UNIVERSITÉ DE LORRAINE

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

Rapport N : Année : 2005

Institut National Polytechnique de Lorraine
Centre de Recherche en Automatique de Nancy

THESE

Présentée à l'Institut National Polytechnique de Lorraine

en vue de l'obtention du titre de

Docteur de l'Institut National Polytechnique de Lorraine

Spécialité: Automatique

par

Karim HAMIDI

Contribution à un modèle d'évaluation quantitative des performances fiabilistes de fonctions électroniques et programmables dédiées à la sécurité

Soutenue le 27 octobre 2005 devant le Jury :

Examinateurs

Prof. Yves DUTUIT	(président)	LAPS (Bordeaux)
Prof. Etienne CRAYE	(rapporteur)	LAGIS (Lille)
Prof. Christophe BERENGUER	(rapporteur)	LM2S (Troyes)
Prof. Jean-François AUBRY	(directeur de thèse)	CRAN (Nancy)
Prof. Claude IUNG		CRAN (Nancy)
Dr. Olaf MALASSE		ENSAM (Metz)

Invité

Dipl. Ing. Martin ROTHFELDER	Siemens CT (Munich)

A mes parents,

A mon frère et

A ma chère et tendre

« Le génie est fait d'un dixième d'inspiration et de neuf dixième de transpiration »

Thomas Edison

REMERCIEMENTS

Je souhaiterais par ce court paragraphe remercier tout ceux qui m'ont éclairé et soutenu lors de cette thèse, de par leurs compétences techniques, leurs expériences et leur soutien moral.

Mes premières pensées vont à Mrs Jean-François Aubry et Olaf Malassé qui ont su encadrer ce travail, malgré la distance, et m'orienter dans mes recherches. Je remercie tout particulièrement le professeur Jean-François Aubry pour ses conseils, son sens critique et son intérêt pour le travail, mais aussi pour sa sympathie et son suivi attentif des travaux. Je lui souhaite un prompt rétablissement, au vu de son état de santé actuel, et apprécie son courage pour le suivi des dernières phases de ce travail. Je tiens aussi à remercier Olaf Malassé qui m'a permis, lors de cette thèse, de m'investir dans les activités de recherche et de formation du centre A3SI de Metz, nouveau pôle, ouvert sur l'industrie, et dédié à la sûreté de fonctionnement et à m'intégrer à l'équipe existante (Gregory Buchheit, Calogero Beltrami).

Je tiens ensuite à remercier la division de Siemens (CT PP2- analyse et simulation des risques) à Munich dans son ensemble qui a financé ces travaux de thèse, a montré son intérêt pour un thème, permettant le développement ultérieur d'outils d'évaluation à portée industrielle. L'environnement industriel et la largeur du spectre des projets en cours (ferroviaire, systèmes automobiles embarqués, industrie, nucléaire) m'a permis de mieux appréhender certains problèmes concrets et de dégager certaines réflexions sur les contraintes architecturales et les limites pratiques de mise en œuvre d'étude et d'évaluation d'architectures électroniques et programmables. Je tiens particulièrement à remercier Martin Rothfelder pour son encadrement sur place, le Prof. Albert Gilg pour son soutien, en qualité de chef de division et les autres membres de l'équipe (Hans, Johannes, Juan, Petra, Manfred,…) et d'autres divisions de Siemens (Philippe, Leopold..) qui ont su m'apporter leurs points de vue et me remonter le moral dans les moments difficiles.

Je tiens, enfin, à remercier les membres de ce jury de thèse qui ont accepté d'être les rapporteurs et les examinateurs de ce travail, et qui ont fait preuve de leurs intérêts au sein de conférences et d'échanges sur les résultats et perspectives de celui-ci. J'apprécie, particulièrement, la venue du Professeur Yves Dutuit, rencontré au congrès Qualita à Bordeaux, dont l'intérêt pour ce thème de recherche et les contributions de ces dernières années sur la fiabilité dynamique sont riches d'enseignement et celle des professeurs Christophe Bérenguer et Etienne Craye qui ont accepté d'être les rapporteurs de ce travail. Je les remercie tous deux pour leur accueil dans leurs laboratoires respectifs et pour leurs travaux de recherche, qui permettent d'apporter des éclairages différents sur cette thèse. Je félicite au passage Mr Craye pour sa récente nomination à un poste de direction de l'EC Lille et le remercie d'avoir su trouver le temps nécessaire à la lecture de cette thèse.

Mes pensées les plus chaleureuses vont à ma famille et à mes amis en France (Pierre, Julien, Antoine, Olivier,…) et en Allemagne (Siham, Yann, Fettie…) dont le soutien moral fut mon principal appui lors de cette thèse. Je pense en particulier à mon amie et à mon frère, à qui je dédie ce travail et qui m'ont toujours soutenu malgré l'éloignement géographique.

Je m'excuse par avance pour ceux qui ne se seraient pas retrouvés dans les précédents paragraphes, et les assure de ma sympathie. J'ajoute simplement un remerciement pour les soutiens techniques de Mrs Hutinet et Thomas (GFI puis Dassault System) en vue de l'utilisation de l'interface graphique de Jagriph (diagramme de fiabilité et graphes de Markov).

En vous souhaitant, à tous, une bonne lecture …

RESUME

La prévention des risques joue un rôle de plus en plus important dans les secteurs industriels, les applications de transport et les installations nucléaires. L'ajout d'une fonction dédiée à la sécurité est un choix visant, par la détection précoce d'une situation accidentogène, à éviter un accident par la mise en œuvre de moyens de réaction permettant le retour du système dans un « état sûr ». L'estimation de ses indicateurs de performances doit donc permettre de juger des avantages en terme de réduction du risque mais aussi des inconvénients en terme des pertes économiques induites par son déclenchement intempestif.

L'utilisation d'architectures électroniques et programmables (EP) s'est accompagnée ces dernières années de contraintes grandissantes (partage de ressource matérielles, informationnelles, modes dégradés de fonctionnement…), qui tendent à influencer fortement la qualité de service de telles fonctions. Il devient donc très difficile d'évaluer correctement les performances (au sens fiabiliste) de telles fonctions. Devant l'influence grandissante des défaillances multiples sur la fiabilité de fonction de sécurité, il nous a semblé important de proposer une méthode permettant, par une meilleure prise en compte de l'aspect temporel, d'estimer de manière plus précise les indicateurs de performances d'une fonction EP dédiée à la sécurité.

Ce travail introduit le concept d'information et interprète un mode de défaillance de la fonction d'étude comme le résultat de l'initiation et de la propagation d'information(s) erronée(s) jusqu'au niveau des actionneurs. Il propose donc de distinguer d'une part les phénomènes d'apparition et de disparition d'erreurs (matérielles et environnementales), et les séquences d'activation de celles-ci (défaillances locales) aboutissant à un mode de défaillance du système d'étude. Cette distinction permet d'expliquer le concept de « défaillances simultanées » et d'aborder de manière différente les problématiques de défaillances de cause commune et d'activation de mode de défaillance latent.

On construit, en partant de la distinction entre existence et activation d'erreur, un modèle dynamique général d'estimation des indicateurs de performances. Il est basé sur un ensemble de modèles stochastiques, permettant de simuler l'évolution temporelle des probabilités d'erreurs matérielles pour chaque ressource matérielle (prise en compte implicite d'exposition mutuelle à des contraintes environnementales), et sur deux listes caractéristiques, donnant les combinaisons d'erreurs activées menant à un mode de défaillance du système d'étude.

La méthode d'identification des listes proposées se base, d'une part, sur une représentation de haut- niveau, de type informationnelle de l'architecture fonctionnelle (incluant les procédures de test et de reconfiguration), et sur une représentation par automate d'état fini du comportement de ses entités constitutives. Elle peut être interprétée comme le résultat d'une propagation de langage (agrégation et réduction) le long de flux élémentaires d'information. Ce processus de construction a fait l'objet d'une automatisation (logiciel AFMCI en Java).

La méthode proposée de construction des processus stochastiques portant sur les états d'erreurs matériels sera, quant à elle, plus classique puisque basée sur des processus markoviens. Nous privilégierons, néanmoins, l'utilisation de chaîne de Markov non homogènes (taux de transitions non constants) pour permettre de prendre en compte l'influence de l'évolution des contraintes environnementales sur les taux de pannes des composants constitutifs de ces ressources matérielles, et expliquerons leur extension dans le cas de ressources préventivement maintenues.

Après avoir montré l'intérêt de cette méthode d'évaluation sur un cas d'étude, nous expliquerons comment utiliser les listes caractéristiques, si le niveau de risque résiduel estimé est trop élevé. Nous proposerons, au vu des listes caractéristiques, la mise en place d'une stratégie pertinente de modification architecturale, permettant de garantir l'atteinte d'un niveau de risque tolérable. Nous aurons ainsi démontré l'intérêt de la méthode de construction des listes caractéristiques, à la fois à des fins d'évaluation de performances, mais aussi leur intérêt dans la mise en place d'une stratégie pertinente d'amélioration des performances d'une fonction EP dédiée à la sécurité.

Tables des matières

CHAPITRE 0 - INTRODUCTION 1

CHAPITRE 1 - PLACE DES FONCTIONS PROGRAMMABLES DEDIEES SECURITE DANS UNE STRATEGIE DE REDUCTION DES RISQUES : ENJEUX ET INDICATEURS DE PERFORMANCES 4

1- Notion de risque, rôle de l'ajout d'une fonction dédiée sécurité et hypothèses d'étude 4
1-1- Mesure et réduction du risque – Contexte d'étude 4
1-2- Fonctions dédiée sécurité – hypothèses de départ 7
1-3- Hypothèses préliminaires 14

2- Allocation de performances fonctionnelles 16
2-1- Détermination du niveau de risque résiduel et hypothèses simplificatrices 16
2-2- Détermination de la perte économique induite et hypothèses simplificatrices 19
2-3- Bilan 20

3- Incomplétude des standards actuels en sécurité fonctionnelle 20

4- Méthode d'évaluations actuelles et limites 23

5- Frontières du système à modéliser 29

6- Concepts de base et proposition (modèle informationnel) 32

7- Conclusion du chapitre 1. 37

CHAPITRE 2 - PROPOSITIONS POUR UN MODELE GENERAL D'EVALUATION DE PERFORMANCES 39

1- Typologie fautes/erreurs 39
1-1- Concept et dénomination des modes de défaillance de ressources matérielles. 40
1-2- Similitudes et différence avec l'approche faute > erreur > défaillance proposée par le LAAS. 43
1-3- Sources de couplages. 44

2- Aspect temporel et définition du concept de défaillances «simultanées» 46
2-1- Utilisation d'un pas temporel T pour prendre en compte les phénomènes d'apparition/ disparition d'erreurs 47
2-2- Avantage d'une étude de type flux d'information 49
2-3- Inconvénients pour les contraintes de temps de test et proposition de résolution 51

3- Structure du modèle général d'évaluation 52
3-1- Vers une structure basée sur une approche multi-échelle de temps 52
3-2- Prise en compte implicite du couplage fautes/environnement 54
3-3- Structure du modèle général d'évaluation des indicateurs de performances. 55
3-4- Evaluation des indicateurs de performances. 57

Conclusion de la partie 2 et introduction des parties 3 et 4 59

CHAPITRE 3 - METHODE SYSTEMATIQUE DE CONSTRUCTION DES LISTES CARACTERISTIQUES 61

1- Propositions et idées fondamentales 62

1-1- Notion de ressource 62
1-2- Notion d'entité sous fonctionnelle 63
1-3- Notion d'utilisation et de consommation de ressources 64
1-4- Démarche générale 67

2- Modèle de haut niveau 68
2-1- Formalisme de haut niveau 68
2-2- Construction du modèle de haut niveau sur exemple 72

3- Modèle de bas-niveau 76
3-1- Automate à états finis et langages marqués 76
3-2- Construction de modèle de bas niveau et automate minimal 78
3-3- Classes d'événements constitutifs et explication 79

4- Processus de construction des listes La et Ld 82
4-1- Propagation de langages – principe et opérations de base 82
4-2- Ordres des opérations d'agrégation de bloc pour la construction de La et Ld. 87

5- Résultats 89

6- Mise en œuvre pratique 90
6-1- Cahier des charges du logiciel 90
6-2- Module de saisie manuelle des graphes et de calcul des langages marqués 91
6-3- Opération de manipulation des listes 91
6-4- Détermination des séquences d'agrégation 92
6-5- Calcul final 93

7- Conclusion du chapitre 3 94

CHAPITRE 4 - CONSTRUCTION DES MODELES D'EVOLUTION DES PROBABILITES D'ERREURS MATERIELLES 96

1- Choix de processus Markoviens et extensions. 96
1-1- Rappel sur les processus markoviens 96
1-2- Utilisation d'un processus markovien pour l'évaluation temporelle des probabilités de fautes d'une ressource matérielle non réparable. 97
1-3- Extension au cas de ressources correctivement maintenues 101
1-4- Extension au cas de procédure de maintenance préventive sur la ressource matérielle 103

2- Détermination des relations entre taux équivalents et stress environnementaux.s 105

3- Augmentation du temps T. Démarches et hypothèses. 106
3-1- Conditions d'élargissement du pas d'itération des modèles et du modèle équivalent. 107
3-2- Modèles équivalents et réduction du temps de calcul des indicateurs de performance. 109

4- Conclusion de la partie 4 110

CHAPITRE 5 - UTILISATION DE LA METHODE SUR CAS D'ETUDE 112

1- Cas d'étude de départ. 112

2- Ajout d'une redondance capteur et d'un contrôle temporel pour la communication 115

3- Ajout d'une redondance actionneur et d'une politique de test périodique 118

4- Cas d'étude réel et discussion 120
4-1- Présentation du cas d'étude réel 121
4-2- Décomposition de la partie 2oo3- 1oo2 124
4-3- Détermination du modèle de haut niveau de ce cas d'étude 124

5- Conclusion du chapitre 5 127

CHAPITRE 6 - VERS UNE ORIENTATION PERTINENTE D'UNE STRATEGIE DE MODIFICATION ARCHITECTURALE 129

1- Vers la mise en place d'une stratégie d'amélioration architecturale pertinente 130
1-1- Hypothèses simplificatrices pour une stratégie de diminution du risque résiduel 130
1-2- Leviers d'amélioration : discussion sur leurs conséquences au niveau local et global. 131
1-3- Contraintes de modifications architecturales et intérêts. 135

2- Stratégie proposée à partir des listes L_d. 136
2-1- Notions de combinaisons d'erreurs minimales. 136
2-2- Notion de combinaisons d'erreurs minimales réduites 137
2-3- Orientation stratégique en deux temps. 138

3- Hiérarchisation des combinaisons minimales et évaluation du gain garanti 138
3-1- Notion de distance à objectif et mise en œuvre de notre stratégie 138
3-2- Critères de hiérarchisation des combinaisons d'erreurs détectable de longueur supérieur à 2 et calcul de g*140
3-3- Critères de hiérarchisation des erreurs de C_d^1 et mise en place d'une stratégie dans le cas 2. 143

4- Bilan sur la stratégie complète. 147

5- Bilan du chapitre 6 148

CHAPITRE 7 - CONCLUSIONS ET PERSPECTIVES 150

Chapitre 0

Introduction

Chapitre 0 - Introduction

L'introduction d'une **fonction dédiée sécurité** dans une installation a pour but de rendre possible la détection d'une situation jugée potentiellement dangereuse et d'**éviter l'occurrence d'un accident**. Le principe d'évitement est initialement basé sur la **détection** d'une situation anormale, précurseur de l'accident, et sur l'activation, dans ce cas de figure, d'un ensemble d'actions. Ces actions, dites de **réaction**, doivent forcer le retour de l'installation dans un état « sûr ». Dans cet état, l'installation doit être capable de poursuivre ses missions de base sans engendrer de conséquence catastrophique pour ses utilisateurs ou son environnement.

Le **spectre d'utilisation** de telles fonctions est donc extrêmement large et va des domaines nucléaire, énergétique, productique, jusqu'aux systèmes de transport aéronautique, automobile ou encore ferroviaire. On peut citer, par exemple, les fonctions d'arrêt d'urgence de réacteurs nucléaires, les fonctions airbag ou de freinage dans les applications de transport.

La démonstration de la **qualité de service** d'une fonction dédiée sécurité est donc primordiale. Elle doit permettre à l'exploitant de garantir que les pertes de production ou de service, induites par l'ajout de cette fonction, restent acceptables et qu'une réduction satisfaisante du **niveau de risque** est atteinte. Cette démonstration est nécessaire à l'acceptation des choix architecturaux et de maintenance mis en œuvre pour assurer cette fonction (phase de **validation architecturale**). Dans cette optique, l'utilisation d'architectures électroniques et programmables (EP) pose le problème de l'estimation quantitative précise **d'indicateurs de performances fonctionnelles**, permettant de juger des conséquences de l'ajout d'une fonction dédiée sécurité.

Ces dernières années, la croissance de la complexité des architectures opérationnelles a créé de nouvelles contraintes dont la prise en compte semble désormais indispensable. Le recours toujours plus fréquent à des procédures de réparation, de diagnostic et de recouvrement fonctionnel distribuées dans l'architecture, rend, en effet, de plus en plus difficile la prise en compte par les méthodes actuelles de l'influence de l'ensemble de ces contraintes sur les performances fonctionnelles, notamment en termes de fiabilité. Nous pensons que certaines approximations faites lors de l'utilisation de méthodes usuelles d'évaluation fiabiliste n'a pas un effet négligeable sur les indicateurs estimés. Elles peuvent être la source d'un biais important, notamment dans le cas d'architectures EP fortement tolérantes aux fautes.

Pour vérifier cette hypothèse, après avoir clairement identifié les indicateurs de performance à estimer, nous proposerons un **modèle original d'évaluation quantitative** de ceux-ci. La structure de ce modèle se basera sur une prise en compte rigoureuse des phénomènes de dépendances pouvant exister dans une architecture EP complexe. Elle s'appuiera sur certains points non complètement pris en compte par les modèles actuels.

L'intérêt et l'originalité de la méthode proposée dans ce travail résident dans le fait qu'elle se base sur une approche complète et dynamique de l'architecture. Elle permet de considérer implicitement les concepts d'initiation et de propagation d'erreurs d'information et les influences des contraintes temporelles liées à l'architecture. Ce modèle s'appuiera sur une méthode de construction systématique, mettant en évidence l'influence des choix matériels, logiciels et de maintenance sur l'aptitude de la fonction dédiée sécurité à satisfaire ses missions. Nous expliquerons ses avantages pour la prise en compte de certains modes de défaillances avant de comparer, sur un cas d'étude, les estimations obtenues par cette méthode à celles données par une méthode d'évaluation « classique ». Nous achèverons notre travail par une discussion sur l'utilisation de certaines listes constitutives du modèle d'évaluation afin de choisir, de manière pertinente, une stratégie de modification architecturale, visant à garantir un niveau de risque résiduel tolérable.

La première partie de ce travail permet de replacer l'étude dans le contexte général de la **maîtrise des risques.** Après avoir rappelé certaines notions fondamentales, nous détaillerons un **nombre limité d'indicateurs de performance fonctionnelle** permettant de justifier la qualité de service d'une fonction EP dédiée sécurité. Nous présenterons ensuite rapidement les méthodes actuelles d'évaluation quantitative et expliquerons pourquoi leur utilisation ne semble aujourd'hui plus suffisante pour une **prise en compte correcte de certains modes de défaillances** (défaillances de cause commune, modes latents de défaillances…). Ce constat nous poussera à introduire la notion d'information et à redéfinir les frontières du système d'étude. Nous baserons ensuite notre réflexion sur le principe d'initiation et de propagation d'erreur(s) le long de flux d'information au sein d'une architecture fonctionnelle reconfigurable.

La seconde partie présentera ensuite la **structure générale** de notre modèle d'évaluation. Cette structure se basera sur une distinction forte entre l'évolution temporelle de l'état d'**intégrité** des ressources matérielles et les mécanisme d'initiation et de propagation d'information(s) **erronée(s)** au sein d'une architecture. Après avoir détaillé les sources de couplage existant dans l'architecture, nous introduirons **un temps caractéristique** nous permettant de clarifier la notion de « défaillances matérielles simultanées » et de justifier la construction de deux listes d'événements, appelées listes caractéristiques, dont nous définirons les propriétés. Nous expliquerons comment, à partir de ces listes, il est possible de proposer un modèle général d'évaluation dynamique en trois niveaux, permettant d'estimer les indicateurs de performances de la fonction étudiée. Nous détaillerons les **avantages d'un tel modèle** et expliquerons quelles sont les étapes nécessaires à sa construction. Ces étapes de construction seront ensuite détaillées dans les parties 3 et 4 de notre travail.

La partie 3 sera dédiée à la construction systématique des **deux listes caractéristiques** exposées dans le modèle général. Ces listes reflèteront les combinaisons complètes d'erreurs activées se traduisant par un mode de défaillance de la fonction de sécurité. Elles seront construites à partir d'une représentation de haut niveau de l'architecture fonctionnelle, de type flux d'information, et d'une représentation plus fine du comportement de chacune de ses entités constitutives, donné par un automate d'états finis. La méthode d'identification de ces listes caractéristiques utilisera des règles d'agrégation et de réduction de langages permettant une prise en compte des contraintes liées au **partage de ressources matérielles**, **informationnelles** et de **précédence** entre sous fonctions du système étudié (fonction de sécurité). Elle sera automatisée grâce à un logiciel développé en Java (AFMCI).

La partie 4 présentera, pour sa part, une méthode de construction systématique des sous modèles permettant d'évaluer l'évolution au cours du temps des probabilités d'erreurs de chaque ressource matérielle. Nous proposerons de nous orienter vers la construction de sous modèles de faible dimension, évoluant parallèlement, basés sur des chaînes de Markov non homogènes. Nous expliquerons ensuite comment sont attribués et actualisés, dans la pratique, leurs taux de transition. Puis, nous nous intéresserons aux hypothèses permettant de remplacer ces modèles de pas d'itération temporel T par des modèles « équivalents » de pas d'itération plus large. Ces transformations permettront, en général, de réduire le temps de calcul du modèle d'évaluation des indicateurs de performances.

Enfin, nous nous intéresserons à la mise en œuvre pratique du modèle exposé, en proposant des cas d'étude. Le premier, très simple, permettra de vérifier la cohérence de la méthode en comparant les résultats numériques obtenus par notre méthode à ceux donnés par une méthode classique (mixte Markov/bloc de fiabilité). Les cas d'étude suivants seront, pour leur part, basés sur l'évaluation d'une architecture plus complexe, dans laquelle nous aurons volontairement introduit certaines contraintes architecturales dont l'influence est actuellement peu considéré dans les méthodes existantes. Nous expliquerons notamment la construction des listes caractéristiques d'un cas d'étude réel, extrait du domaine industriel.

Pour terminer ce travail, nous discuterons de l'apport possible de l'identification des listes caractéristiques pour la mise en œuvre d'une approche d'évaluation/modification architecturale, identique à celle préconisée par l'**IEC61508**. Nous proposerons, dans le cas où une métrique de performance est jugée insuffisante en termes de risque résiduel, de réutiliser les listes construites au chapitre 3 pour déterminer une stratégie de modification architecturale pertinente (en termes de coût de mise en œuvre) qui permette de **garantir**, sous contraintes environnementales fixées, l'atteinte de l'objectif de risque résiduel. Cette méthode présentera un avantage important vis-à-vis des méthodes d'améliorations architecturales actuelles puisqu'elle se basera, de manière complémentaire, sur l'amélioration des capacités de diagnostic de l'architecture de départ et sur l'amélioration de la fiabilité de certaines ressources matérielles jugées « critiques » pour les performances du système d'étude.

-

Chapitre 1

Place des fonctions programmables dédiées sécurité dans une stratégie de réduction des risques : Enjeux et Indicateurs de performances

-

Chapitre 1 - Place des fonctions programmables dédiées sécurité dans une stratégie de réduction des risques : Enjeux et Indicateurs de performances

Juger des conséquences de l'ajout d'une **fonction EP dédiée à la sécurité** dans une installation nécessite un rappel préliminaire sur la place d'un tel ajout dans une politique générale de maîtrise de risque. Ce rappel doit permettre d'expliquer clairement le cadre général de notre travail et les hypothèses permettant d'établir une relation directe entre les exigences portant sur un nombre limité d'indicateurs de performances fonctionnelles et les objectifs à atteindre en termes de risque résiduel et de perte économique induite par l'ajout d'une fonction de sécurité dans une installation. Ces relations expliqueront le besoin d'une évaluation quantitative de ces performances pour valider les choix architecturaux retenus.

Suite à ce rappel, nous examinerons les limites imposées par les méthodes d'évaluation actuelles et montrerons que celles-ci s'avèrent insuffisantes pour modéliser de manière complète l'influence de certaines contraintes architecturales. Nous nous référerons en particulier à la prise en compte de certains modes de défaillances (dits de causes communes et de modes latents). Nous en déduirons une série d'orientations permettant d'aboutir à la mise en place d'un modèle d'évaluation plus précis dans le cas de fonctions EP dédiées à la sécurité, distribuées et tolérantes aux fautes. Ce modèle aura pour particularité de prendre en compte à la fois les propriétés d'évolution temporelle de l'état d'intégrité des ressources matérielles mais aussi les capacités de test et de reconfigurations fonctionnelles de l'architecture.

1- Notion de risque, rôle de l'ajout d'une fonction dédiée sécurité et hypothèses d'étude

1-1- Mesure et réduction du risque – Contexte d'étude

La sécurité des applications industrielles et de transport est devenue, lors de ces dernières décennies, un enjeu significatif. De nombreuses catastrophes industrielles aux conséquences tragiques ont montré l'intérêt d'entreprendre des démarches de réduction du niveau de risque. Les drames de Bhopal, Loughborough et Seveso ont permis une réelle prise de conscience sur le sujet de la part des industriels, des pouvoirs publics mais aussi de la population.

Eviter les accidents est ainsi devenu une préoccupation majeure. Cependant, force est de reconnaître que, dans de nombreux domaines, un niveau de risque nul ne peut jamais être atteint. Il existe toujours des conditions de fonctionnement anormales d'une application pouvant entraîner des dégâts humains, matériels et environnementaux. Ces conditions peuvent résulter à la fois de **causes internes** (c'est-à-dire lié au dysfonctionnement des applications de contrôle de l'installation) et de causes **externes**. Il est donc du devoir de l'industriel :

- d'**estimer** le niveau de risque de son installation,
- de **prendre les mesures** qui s'imposent pour détecter au plus vite les situations accidentogènes,
- de déclencher des **actions** d'évitement de l'accident ou de réduction de son impact.

Ces impératifs ont d'ailleurs été rappelés par les législations européennes qui imposent désormais une démarche de maîtrise et de réduction du risque pour les applications industrielles et de transport. Cette démarche est assujettie à l'obligation de mise en œuvre et de démonstration du niveau de risque résiduel par une **étude de dangers** **[Loi03]**.

Toute stratégie de maîtrise des risques se base donc sur l'identification préliminaire des situations dangereuses auxquelles une installation peut être exposée. Les techniques permettant

cette identification sont nombreuses, elles peuvent s'appuyer sur des analyses exploratoires causales **[Brab02]**, par mots clefs et flux énergétiques **[Klet92]** … Un aperçu de ces méthodes est donné par **[PHA03]**. Nous ne nous intéresserons pas pour notre part à ces méthodes d'identification dans ce rapport. Néanmoins, ces méthodes font appel à une terminologie consacrée dont nous nous devons de rappeler quelques éléments ici.

On parlera de **situation potentiellement dangereuse** pour qualifier toute situation accidentogène suffisante à l'occurrence ultérieure d'un accident.

La notion d'**accident** est relativement usuelle, elle correspond à un événement résultant souvent d'une dérive de l'installation et se traduisant par des dommages, comme par exemple des blessures physiques ou atteintes à la santé des personnes. Ces dommages peuvent être vus comme une conséquence directe ou indirecte de dégâts causés aux biens ou à l'environnement. L'accident est donc le résultat d'un scénario, initié par une situation potentiellement dangereuse, souvent appelée **situation accidentogène**. Ce scénario est conditionné par l'occurrence d'une séquence d'événements qui vont avoir tendance à influencer la dynamique d'apparition de nouveaux dommages au cours du temps. On parlera, en général, **d'événements contributeurs** (**[Leve00]**), pour les événements dont l'occurrence a tendance à augmenter les dommages induits par l'accident, en augmentant la vitesse de propagation du processus de destruction des biens ou des personnes ou leur degré d'exposition. Par opposition, on parlera d'**événements mitigatifs ou réducteurs**, pour les événements dont l'occurrence à tendance à réduire la vitesse du processus de destruction des biens ou des personnes, voir à l'arrêter, et qui aboutissent de ce fait à des dégâts moins importants. Cette vision de mise en œuvre séquencée de différentes « moyens », visant à éviter ou à réduire les dommages causés, est très présente dans le domaine nucléaire, qui propose de modéliser les différents scénarii résultant d'une situation accidentogène par l'usage d'arbre d'événements, en admettant pour chaque moyen de réaction un temps limite de déclenchement **[Fleu02]**.

La **notion de risque** est communément définie comme une « mesure d'un danger associant une mesure de l'occurrence d'un accident et une mesure de ses effets ou conséquences » **[EN50126].**

La mesure des effets d'un accident, en termes de dégâts ou de dommages, est appelée **impact d'un accident** et notée I. Elle peut être définie comme la somme cumulée des dommages induits par la situation accidentogène. L'impact est donc nul, si la situation accidentogène ne se traduit pas par un accident. Dans les autres cas, l'impact sera évalué sur un horizon temporel, commençant à l'instant d'occurrence de la situation accidentogène et se terminant au moment où la somme des dommages aura atteint son maximum. Ce maximum est toujours défini, puisque cette somme est croissante et bornée (l'ensemble des biens et personnes pouvant être exposés à un accident est supposé fini). Cet impact est **mesuré en fonction d'une échelle de référence**, préalablement choisie comme référentiel d'impact et appelée **échelle de gravité**. Il est donc donné suivant une unité sans dimension dont les degrés dépendent des différents dégâts mesurables (nombre de morts, coûts de dégâts matériels, dose chimique rejetées…). On peut trouver de nombreux exemples d'échelles de gravité dans différents secteurs d'activité : ferroviaire (en nombre de victimes), chimique (**[Joly04], [Fite03]** en dose de produits rejetés), aéronautique…Pour prendre en compte la diversité de nature des dommages occasionnés (humain, matériels, environnementaux..), il est généralement proposé un critère d'impact composite, résultant de la pondération d'échelles de **gravité** de nature différente (par exemple, l'impact chimique se référant à la dose et à la nature de rejets, l'impact humain se référant au nombre de morts et victimes…). Le choix d'une échelle de gravité et son acceptation reste **un <u>préalable</u> obligatoire** à toute étude de risque. Nous supposerons donc que cette échelle a été clairement établie.

La mesure de l'occurrence d'un accident sera quant à elle associée à sa probabilité. On peut, en supposant l'impact d'un accident I comme constant quelque soit son instant d'occurrence, choisir comme mesure du risque la valeur

-

$$R = P \times I \qquad \textbf{(Eq. 1)}$$

avec P, sa probabilité d'occurrence moyenne sur la durée de vie de l'installation et I son impact associé.

Dans le cas où la probabilité d'occurrence d'un accident est supposée constante, on peut estimer la probabilité moyenne de l'accident au moyen de sa fréquence d'occurrence f. On se ramène alors à la notion de criticité d'un accident : $C = f.I$ **(Eq. 2)**.

Le niveau de risque d'une installation est défini comme la somme des risques associés aux accidents potentiels auxquels elle peut être soumise. Une stratégie naturelle de réduction des risques d'une installation consiste donc à ramener par l'ajout de mesure d'évitement, de prévention et/ou de mitigation les niveaux de risques prépondérants à des niveaux de risques résiduels tolérables. La somme des risques résiduel doit permettre de se placer en dessous d'une valeur d'acceptation de risque pour l'installation.

Le choix des valeurs de chaque niveau d'acceptation dépendra d'une étude de faisabilité (experts et éventuellement retour d'expérience) permettant de juger de la marge de réduction potentielle de la mesure de risque R associé à chaque accident redouté. Ce choix est donc une hypothèse de travail a priori. Nous ne nous attacherons pas au processus de décision aboutissant à l'allocation de ces niveaux d'acceptation **[CEN01], [Dear00]**.

On rappelle que le **risque résiduel**, noté $\mathbf{R_{res}}$ est défini comme la **somme** des risques relatifs aux accidents pouvant être induits par une situation accidentogène, suite à l'ajout d'une mesure visant à prévenir l'accident ou à réduire son impact. Cet ensemble regroupe l'ensemble des accidents dérivant de la situation accidentogène mais aussi tout **nouvel** accident provoqué de manière direct par les modifications apportées à l'installation.

Cette définition peut être illustré par l'exemple d'un système d'airbag dans une automobile. La fonction première de l'airbag est de réduire les lésions corporelles lors d'un choc frontal. On cherche donc à détecter le choc (situation accidentogène) et à déployer l'airbag en un temps assez court pour réduire les lésions corporelles du conducteur. En notant i l'occurrence d'un choc frontal, on peut définir R_{res}^{i} comme étant la somme des risques attachés aux accidents dérivant des situations suivantes:

- non déploiement de l'airbag lors du choc frontal
- déploiement de l'airbag lors du choc frontal
- déploiement de l'airbag en absence de choc frontal.

Le scénario dérivé du déclenchement intempestif de l'airbag doit donc aussi être prise en compte dans le calcul du risque résiduel R_{res}^{i}, puisqu'il induit la mise en danger du conducteur.

Dans ce rapport, nous considérerons l'ajout, dans une installation existante, d'une fonction dédiée sécurité, visant à la prévention d'un accident dont la probabilité d'occurrence sera supposée estimée et constante (fréquence d'accident f) et dont l'impact sera noté I_{abs}. Nous désirons atteindre un niveau de risque résiduel tolérable.

Cette approche de **prévention** est basée sur la notion de situation accidentogène. Le principe d'une fonction dédiée sécurité est de permettre l'identification de cette situation et de mettre en œuvre en un temps raisonnable la réaction permettant d'éviter l'accident et ainsi de ramener l'installation dans un état de fonctionnement sûr[1]. La condition de déclenchement d'une fonction dédiée sécurité est donc spécifiée de manière **UNIQUE**. Elle correspond à la

[1] Un état de fonctionnement sûr est un état dans lequel l'installation est à même de poursuivre ses missions de base sans conséquence catastrophique pour l'utilisateur ou son environnement.

vérification d'un prédicat logique portant sur un ensemble de paramètres environnementaux observables, à un temps fixé. Cette condition est appelé **condition de demande** de la fonction dédiée sécurité. Cette condition est en général simple. Elle est souvent reliée à des « seuils de bon fonctionnement » ou à des événements extérieurs (sollicitation d'utilisateur). On dira dans toute la suite de notre travail que :

- la **demande est présente au temps t**, si l'état des paramètres environnementaux observables exige le déclenchement des réactions pour assurer l'évitement d'un scénario d'accident.
- la **demande est absente au temps t**, si l'état des paramètres environnementaux observables n'est pas suffisant à prouver l'imminence d'un accident et ne nécessite donc pas le déclenchement des réactions.

Nous allons nous intéresser dans toute la suite de notre travail, à l'étude d'une classe particulière de fonction dédiée sécurité, regroupant l'ensemble des **fonctions électroniques et programmables** (notées EP). Contrairement aux dispositifs dédiées sécurité de type mécanique (soupapes…), dont la fiabilité peut être évaluée par étude d'une architecture simple et dont le déclenchement est conditionné par leur dimensionnement, les fonctions EP peuvent contenir différents processus de contrôle, de diagnostic et de reconfiguration qui rendent leur étude beaucoup plus complexe.

Le déclenchement des moyens de réactions de telles fonctions est, en outre, basé sur la vérification de la conformité de conditions observables à une condition préétablie suivi d'un jugement **[Z61-102]**. Ce jugement peut être basé sur des **processus de diagnostic** et de **reconfiguration interne**.

Cette étude pose le problème d'évaluation des conséquences que peut avoir l'ajout d'une telle fonction sur une installation, en termes de risque mais aussi de perte de sa qualité de service durant sa durée d'exploitation. Elle permet de valider les choix architecturaux (matériels, fonctionnels) et les politiques de maintenance retenus pour qu'un tel système ramène le risque résiduel et la perte économique induite par son déclenchement intempestif, sous un niveau acceptable.

1-2- Fonctions dédiée sécurité – hypothèses de départ

1-2-1- Notions de fonction dédiée sécurité et niveau de service

On veut démontrer que l'ajout d'une fonction EP dédiée sécurité, appelée par la suite fonction de sécurité, se traduit par :

- la garantie d'un niveau de risque résiduel R_{res} acceptable
- une perte d'exploitation raisonnable pour l'installation, c'est-à-dire inférieure à une valeur économique acceptable C_{spec} .

Nous supposerons l'ajout d'une fonction dédiée sécurité **UNIQUE** dans une installation existante. La prévention de l'accident considéré ne résultera donc que de cette fonction et non de ses modes de collaboration avec d'autres fonctions implantées.

L'architecture générale d'une fonction EP dédiée sécurité est constituée:

- d'un ensemble fini de capteurs permettant de mesurer les variables représentatives de l'état de demande et de l'état d'intégrité de ressources matérielles testées,
- d'un ou plusieurs organes logiques programmables permettant le processus de décision (contrôle-commande à appliquer) et

- d'un ensemble d'actionneurs commandés électriquement (valves, moteurs…).

Elle vise à assurer l'identification de la situation potentiellement dangereuse et à exiger dans le cas d'existence de celle-ci (présence de demande), l'activation dans un délai spécifié, de moyens de réaction visant au retour de l'installation dans un état sûr (évitement d'accident).

La fonction étudiée peut être considérée comme une fonction temps réel, au sens proposé par **[Pala98]** car elle doit « satisfaire des contraintes de temps de réponse explicites et bornées, le non-respect de ces bornes entraînant l'apparition de dégradations de performances et de mauvais fonctionnement », voire de danger.

Les interfaces capteurs doivent permettre à chaque période d'acquisition de juger de l'état de demande (présence/absence). Nous pouvons remarquer qu'il est tout à fait envisageable, d'utiliser deux observations provenant d'un même capteur sur une période d'acquisition pour créer une redondance d'information. De même, il est possible dans l'architecture de cette fonction d'utiliser différents organes de décisions distribuées pour synthétiser les ordres de contrôle de l'ensemble des actionneurs.

Nous ne restreindrons donc pas notre étude à une architecture simple capteur(s)-contrôleur unique- actionneur(s), mais nous nous placerons dans le cas général de **centres de décision distribués et communicants**. La période d'acquisition T sera définie comme la période entre deux observations successives de l'état de demande par la fonction de sécurité. En ce sens, cette architecture est un **système à événements discrets** (SED).

L'analogie entre les architectures de ces fonctions électroniques et programmables dédiées sécurité et celles d'autres fonctions de contrôle- commande est évidente. Cependant, les fonctions dédiées sécurité ont pour but premier d'assurer le retour dans un état sûr par une activation d'un certain nombre d'actionneurs, appelés moyens de réaction.

Nous ne nous intéresserons pas ici pour juger de la qualité de service d'une fonction de sécurité **au temps nécessaire permettant le changement d'état des actionneurs** (« phase transitoire »), par exemple au temps de fermeture d'une vanne après la réception de l'ordre de contrôle correspondant. Cette durée sera considérée comme une contrainte de temps de l'architecture, appelée **« temps inertiel** ». Dans le cas d'une activation conjointe de différents actionneurs, le « temps inertiel » de référence sera pris comme le maximum des « temps inertiels » de tous les actionneurs. Cette contrainte temporelle sera **uniquement** prise en compte pour évaluer les conséquences de chaque scénario d'activation ou de non activation des moyens de réactions de la fonction dédiée sécurité. Elle permettra d'estimer le temps total nécessaire entre observation de l'état de demande et l'action éventuellement contrôlée.

Nous supposerons, enfin, que la fonction de sécurité commande **une** unique **action de réaction**, assurée par un ensemble fini d'actionneurs redondants. La fonction sera donc dite de type **SDSA** (single demande single action), par analogie au sigle SISO (Single Input Single Output) utilisé en automatique classique. L'activation de ce moyen de réaction en un temps suffisamment court après le passage dans l'état présence de demande doit permettre l'évitement de l'accident. On pourra considérer le cas d'actionneurs redondants, c'est-à-dire visant par leur action conjointe au bon déroulement d'une action de mise en sécurité de l'installation. La relation entre ces modes d'activation et l'état du service fourni sera conditionnée par les choix du schéma de montage de ces actionneurs. On supposera, donc implicitement, qu'il existe, **pour un état de demande fixé à un instant d'acquisition t**, une **proposition logique unique** portant sur l'état des actionneurs permettant de juger du caractère **correct ou incorrect** de l'action de la fonction de sécurité vis-à-vis de son environnement (au sens large, en y incluant l'installation).

Si on considère, par exemple, une fonction de sécurité simple d'arrêt d'urgence de l'alimentation en comburant d'un réacteur par fermeture de deux vannes séries (notées V1 et V2) sur une conduite C dans le cas du dépassement d'un seuil de température critique. En présence de demande (seuil de température dépassé), on suppose que l'organe logique

doit émettre un ordre de fermeture des deux valves, alors qu'en absence de demande ils doit maintenir l'ouverture des deux valves pour permettre la continuité des missions du système. Cette fermeture doit être effectuée au plus tard à une période Tc après l'occurrence de demande pour éviter l'accident.

On pourra donc résumer dans le tableau 1 suivant les propositions permettant de juger de l'état correct ou incorrect de notre fonction de contrôle-commande dédiée sécurité suivant l'état de demande au temps t en notant :

P1(t) la proposition « valve 1 ouverte au temps t + Tc», où Tc est le temps de réaction critique de la fonction dédiée sécurité, c'est-à-dire le temps au-delà duquel un évitement de l'accident par activation des moyens de réaction n'est plus possible, et

P2(t) la proposition « valve 2 ouverte au temps t+Tc»,

Etat de demande au temps t	Service correct	Service incorrect
Présence de demande	P1(t) faux ou P2(t) faux	P1(t) vrai et P2(t) vrai
Absence de demande	P1(t) vrai et P2(t) vrai	P1(t) faux ou P2(t) faux

Tableau 1 : niveau de service d'une fonction de sécurité sur l'exemple

Dans le cas général de plusieurs actions de réactions (fonction de classe SDMA : single demande multiple action), il est possible que certaines combinaisons d'actions se traduisent par des modes dégradés. Il peut donc exister plusieurs niveaux de service intermédiaires, correspondant à la bonne réalisation des différentes actions. On pourra associer à chaque combinaisons d'actions, notée j, {état de l'action au temps t+Tc} trois métriques :

- une métrique d'impact de l'accident induite par cette combinaison d'action au temps t+Tc sous l'hypothèse d'une demande au temps t (valant 0 en absence d'accident), noté $I_{dem}(j)$,
- une métrique d'impact de l'accident induit par cette combinaison d'action au temps t+Tc sous l'hypothèse d'une absence de demande au temps t (valant 0 en absence d'accident), noté $I_{abs}(j)$ (correspondant aux scenarii de mise en danger du système par déclenchement intempestif des moyens de réaction),
- une métrique de perte économique induite par cette combinaison d'action au temps t+Tc sous l'hypothèse d'une absence de demande au temps t (valant 0 en absence d'accident), noté $C_{abs}(j)$ (correspondant aux pertes économiques de l'installation causées par le déclenchement intempestif des moyens de réaction).

Ces résultats pourront être mis sous la forme d'arbre, comme proposé dans l'**annexe A,** sur un exemple de deux actions, notées A et B. Dans ce cas, on devra plus aux probabilités moyennes d'état de chaque actionneur au temps t+Tc, sous condition de demande fixée au temps t, prises indépendamment, mais bien aux probabilités de combinaisons de ces états, sous condition de demande fixée au temps t. Les relations (Eq.13) et (Eq.16) données dans la suite de ce chapitre seront donc dans ce cas plus général remplacées par les équations (Eq.13a) et (Eq.16a) données dans **l'annexe A**.

1-2-2- Définition du système d'étude, relations avec son environnement et hypothèses simplificatrices

Pour résumer, nous limiterons donc notre étude à celle d'une **fonction PE dédiée sécurité de classe SDSA** ayant pour but d'éviter l'accident redouté par un unique moyen de réaction, faisant intervenir éventuellement le contrôle de plusieurs actionneurs redondants. Ces hypothèses imposent pour la fonction de sécurité étudiée, une **condition de demande unique** (défini par le cahier des charges fonctionnel), **l'existence d'un temps de réaction limite** entre l'observation d'une présence de demande et l'activation effective des actionneurs permettant d'éviter l'accident après l'occurrence de la demande, et l'existence d'une loi logique entre l'**état des actionneurs**

contrôlés et **la réalisation de cette action**. Pour éviter tout malentendu, nous utiliserons dans toute la suite de ce rapport, la terminologie suivante :

Installation : application industrielle ou de transport (appelée « Entity Under Contrôl » dans les normes CEI) dans laquelle nous avons décidé d'implanter la fonction de sécurité dans un but de réduction du niveau de risque industriel attaché à un scénario d'accident.

Système d'étude (= système): ensemble des ressources matérielles et logicielles visant à **assurer correctement** la fonction de sécurité et ce quelque soit l'état de demande.

Environnement : ensemble extérieur au système pouvant influer, directement ou indirectement sur son intégrité et l'état de demande, auquel il est soumis. Il est important de noter que l'installation est incluse dans l'environnement, c'est-à-dire qu'elle peut influencer sur l'évolution de contraintes environnementales sur le système d'étude. On distinguera donc, en général, deux types d'influence portant sur les contraintes environnementales : les événements de causes externes, non commandés par l'installation, et les événements internes à l'installation, constitué de l'ensemble des actions de contrôle de l'installation visant à réagir vis-à-vis d'un ensemble de paramètres observables (lois de contrôle).

Demande : condition permettant de juger du besoin de mise en œuvre de la réaction et portant sur un ensemble de paramètres environnementaux rendus observables par l'implantation d'un nombre fini de capteurs

Réaction : action prévue dans le cahier des charges du système et visant à éviter l'accident redouté.

On peut donc considérer le système étudié, comme un système de contrôle commande échantillonné à durée de cycle d'acquisition fixée, dans lequel un processeur mettra en oeuvre la loi de réaction principale. La fonction de sécurité étudiée peut donc être schématisée par la **figure 3**. Pour **toute période d'acquisition T**, le système va demander une observation de l'état de demande (présence ou absence de demande) par la mise à jour des données délivrées par ses interfaces capteurs. Il va ensuite répondre à cet état par l'activation ou l'inactivation des moyens de réaction. Dans le cas d'un fonctionnement normal du système d'étude et en présence de demande, on aura une réaction de la fonction de sécurité en une durée maximale D. Cette durée est égale à la somme du temps de cycle τ et du temps inertiel Ti, induit par les actionneurs à activer. Cette activation des moyens de réaction va ensuite influencer l'évolution de l'environnement. Ce phénomène d'interaction entre le système d'étude, que l'on peut considérer comme un système à événements discret, et son environnement, dont l'évolution peut être donné par un ensemble fini de lois à évolution différentielle continue, est un système hybride et qui peut être modélisé comme tel. Un exemple est montré dans **[Chab98]**.

En sus de ce phénomène d'interaction directe entre le système et son environnement, que nous pourrions qualifier de **boucle de sécurité**. L'environnement peut aussi perturber le mode de fonctionnement du système en dégradant ou restaurant (dans le cas de réparation) l'intégrité de ses ressources matérielles ou informationnelles. Ces interactions peuvent donc être symbolisées par la **figure 3**, où nous avons fait figurer ces contraintes environnementales.

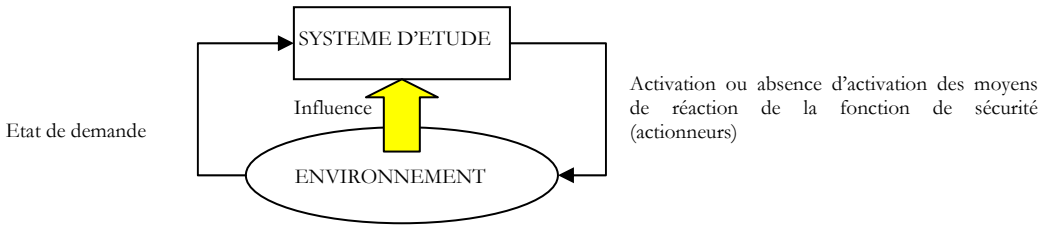


Figure 3 : Modes d'interactions entre fonction de sécurité et environnement

Nous proposerons dans notre étude de découpler ces deux phénomènes d'interaction suivants:

- l'effet perturbateur de l'environnement sur le comportement fonctionnel du système d'un côté (contraintes environnementales)
- et l'évolution de l'état de demande au cours du temps.

Pour ce faire, nous imposerons les hypothèses supplémentaires suivantes :

- L'évolution des contraintes environnementales est supposée **estimable en absence de demande** par simulation des profils de mission de l'installation et retour d'expérience.
- En **présence de demande**, l'absence de réaction de la fonction dédiée sécurité au delà d'un temps critique Tc après son occurrence aboutit de manière certaine à l'accident redouté. Dans le cas de figure inverse, on suppose un retour rapide de l'installation dans un état d'absence de demande, permettant de **négliger l'effet de cette phase transitoire sur les lois d'apparition** de pannes **ultérieures** des composants matériels du système.
- L'occurrence de demande (passage de l'état absence de demande à présence de demande) est un événement **ne dépendant pas de l'évolution normale de l'installation**. Il peut, dans ce cas, être assimilé à une variable aléatoire dont on peut estimer l'évolution probabiliste au cours du temps, **cette évolution sera supposée connue** (soit par jugement d'experts, soit par retour d'expérience).
- Le passage de l'état présence de demande à absence de demande ne peut être atteint que si la fonction de sécurité réagit avant un temps Tc après son occurrence (contrainte de temps maximal d'activation)

En partant de ces hypothèses, on peut symboliser les interactions de notre système et de son environnement sous la forme simplifiée suivante (**figure 4**). L'effet de perturbation de l'environnement sur l'état d'intégrité des ressources du système y est alors estimé en absence de demande grâce à un profil de variation estimé des contraintes environnementales (pire scénario d'utilisation prévu en regard des missions de l'installation). L'événement d'occurrence de demande (passage à un état présence de demande) y est modélisé par un événement de nature aléatoire. Cette simplification revient à dire qu'une absence de réaction de la fonction de sécurité sur un durée Tc suivant l'occurrence d'une demande se traduira toujours par l'accident redouté et aboutira de ce fait à la destruction du système, tandis qu'une réaction durant cette période permettra à la fois d'éviter l'accident et le retour à un profil de mission normal en un temps assez faible, sans que ce retour ait un quelconque effet sur l'état d'intégrité de ses ressources matérielles.

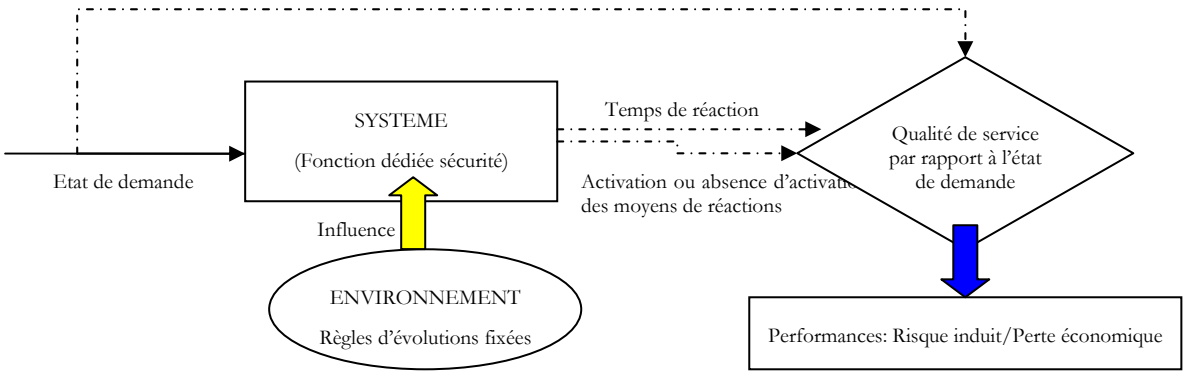


Figure 4 : Modèle simplifié des relations système/environnement.

La probabilité d'occurrence de demande sera en général estimée comme égale à la probabilité d'occurrence de l'accident avant l'ajout de la fonction de sécurité (donnée par retour d'expérience ou simulation). Bien que cette probabilité puisse évoluer au cours du temps, du fait, par exemple, de l'augmentation des probabilités de ces événements initiateurs (dysfonctionnement d'autres fonctions de l'installation…), nous la supposerons dans notre étude comme étant constante (variation temporelle négligeable).

Cette hypothèse supplémentaire nous permettra de démontrer dans la partie suivante l'existence d'une relation directe entre les notions de risques induits et de perte économique provoqués par l'ajout d'une fonction de sécurité et les moyennes temporelles de certains indicateurs, appelés indicateurs de performances. Nous pourrons donc lier la qualité de service de la fonction dédiée sécurité au temps de réaction et à la nature de l'action (activation ou absence d'activation des moyens de réaction) induite par une observation de l'état de demande par le système d'étude. Cette qualité de service sera liée aux notions de risque résiduel et de perte d'exploitation (appelé aussi perte économique induite), introduites dans les paragraphes suivants. Nous proposons donc de déterminer quels indicateurs de performances de la fonction dédiée sécurité peuvent être jugés comme déterminants pour conclure sur sa qualité de service envers l'installation.

La **qualité de service** d'une fonction dédiée sécurité correspondra à la propriété de satisfaire à la fois les besoins exprimés (*capacité d'évitement d'accident*) et implicites (*non perturbation des missions du système en absence de situation dangereuse*) (**ISO8402**). Elle est en général définie par rapport **à sa capacité à fournir un service correct** (aspect fonctionnel). Ce service dépendant à la fois des actions entreprises et du temps de leur mise en œuvre. On peut donc la rattacher à la propriété de **disponibilité** (*capacité d'une entité de fournir un service correct, en un temps raisonnable, quand celui-ci est demandé*) de la fonction de sécurité.

Nous attacherons donc une importance particulière aux deux modes de défaillance de la fonction de sécurité étudiée:

- **l'indisponibilité sur demande**, c'est-à-dire à l'incapacité de fournir un service correct, en un temps raisonnable (*c'est-à-dire inférieur à Tc*) après l'occurrence de demande,
- et le **déclenchement intempestif**, correspondant à la possibilité de déclencher les mesures de réaction de la fonction de sécurité en absence de demande.

On voit qu'il se dégage dans les deux propriétés précédemment évoquées deux notions complémentaires, permettant de juger le caractère correct ou non de la réponse donnée par une fonction de sécurité:

- la **nature de la réponse**, dépendant de l'état d'activation (activation ou non activation) des moyens de réaction de la fonction de sécurité (actionneurs), faisant suite à l'observation de l'état de demande à une période d'acquisition fixée.
- **le temps de cycle D** défini comme le temps maximal séparant l'acquisition de l'état présence de demande et la réaction correspondante, dans le cas d'un fonctionnement normal de la fonction de sécurité.

Pour une fonction de sécurité électronique et programmable, il est possible de supposer le temps nécessaire à l'établissement d'une loi de contrôle au niveau actionneur comme borné. Ce temps dépend essentiellement des caractéristiques du réseau de communication entre entités et des temps de cycle automates fixés lors de la conception. L'ordre de commande transmis à l'actionneur est donc soit transmis sur un intervalle de temps maximal fixé, appelé temps de cycle, soit perdu. Le temps de réponse est donc majoré par la somme de ce **temps de cycle** et du **« temps inertiel »**. Cette somme est **notée Tr**.

Une réduction de la valeur de temps de cycle peut être atteinte grâce à un ordonnancement des tâches du processeur (introduction de priorité entre tâches, procédés d'encapsulation temporelle de tâches **[Jong02]**…), une amélioration des performances de réseaux, en termes de vitesse de transmission et d'accessibilité etc. De nombreuses études ont été faites à ce sujet, en particulier sur des architectures de type réseaux distribués. Aujourd'hui, on peut affirmer que l'utilisation de méthodes d'ordonnancement (RdP, file d'attentes…) sont largement mises en œuvre et que le facteur limitant dans la réduction de la valeur de Tr reste, le plus souvent, l'ordre de grandeur du temps inertiel.

Dans le cas d'un fonctionnement normal, le temps maximal de réaction d'une fonction de sécurité vis-à-vis d'une occurrence de demande, noté Tc, sera donc strictement inférieur à la somme de T, période d'acquisition de l'état de demande, de D, temps de cycle, et de Ti, temps inertiel. Cette condition est nécessaire pour être en mesure de prévenir l'accident, dans le cas d'un fonctionnement normal de la fonction de sécurité. On supposera donc l'inégalité :

$$T + D + T_i < T_c \qquad \textbf{(Eq.6)}$$

Au vu de ces constats, nous définirons deux indicateurs de performances instantanées, permettant de juger de la qualité de service de notre système à l'instant d'acquisition t du système étudié:

- **la probabilité de défaillance sur demande au temps t**, notée PFD(t) (pour probability of failure on demand), correspondant à la probabilité de non réaction de la fonction de sécurité au temps t+Tr si la demande est présente au temps d'acquisition t.
- **la probabilité de déclenchement intempestif au temps t**, notée PFS(t) (pour probability of spurious trip), correspondant à la probabilité d'une activation intempestive des moyens de réactions au temps t + Tr si la demande est absente au temps d'acquisition t.

Les valeurs de ces deux indicateurs (PFD(t),PFS(t)) sont les conséquences des choix architecturaux, tant au niveau fonctionnel que matériel, des choix d'implantation (exposition aux contraintes environnementales des composants) et des choix des politiques de maintenance appliquées aux ressources matérielles. Il est à noter que l'on ne parle pas, dans ce contexte, de probabilité de défaillance en cours de mission mais bien de l'inaptitude de remplir correctement un service pour un état de demande fixé, c'est-à-dire au regard d'une condition observable, permettant de juger de la présence d'une situation accidentogène.

Nous désirons, dans la suite de notre travail, montrer les relations entre ces métriques et les notions de risque résiduel et de perte d'exploitation, mais aussi être en mesure de les évaluer en connaissant les propriétés de fiabilités matérielles des ressources du système d'étude, les politiques de maintenance préconisées et l'architecture opérationnelle (matérielle + fonctionnelle) du système, en incluant ses capacités de test, de recouvrement et de reconfigurations (matérielles et fonctionnelles) au cours du temps. Elles peuvent donc être vues comme une conséquence des choix en termes de matériel et de logiciel, des règles de maintenance (corrective ou préventive) et du contexte d'utilisation imposé aux ressources.

Nous allons dans la sous partie suivante proposer certaines hypothèses permettant de relier les moyennes temporelles de ces probabilités aux notions de risque résiduel et de perte économique. Pour ce faire, on s'intéressera au niveau installation, à deux propriétés essentielles **[Lapr04]**:

- la **sécurité-innocuité**, propriété permettant de justifier l'absence de conséquences catastrophiques sur l'utilisateur ou son environnement, qui sera mesurée grâce à la métrique de risque résiduel induit, notée précédemment $\mathbf{R}_{\mathbf{ind}}^{\mathbf{i}}$ et que nous noterons par abus **R** dans toute la suite du travail.

- la **continuité des missions de l'installation,** propriété permettant de justifier de l'absence d'atteinte au mission de l'installation par l'ajout de la fonction dédiée sécurité, qui sera mesurée en fonction des pertes économique engendrées, notés $\mathbf{C_{ind}}$, lors de la durée d'utilisation de l'ensemble {fonction de sécurité + installation} et mesurée en fonction du nombre d'heure moyen estimé de déclenchement intempestif.

1-3- Hypothèses préliminaires

La définition de la sûreté de fonctionnement, donnée par **[Lapr96]** et précédemment rappelée fait intervenir le concept de **« confiance justifiée »** dans l'estimation des propriétés du système étudié à délivrer un service correct. Cette notion de **confiance**, impose la connaissance correcte et complète des choix architecturaux (matériels, logiciels) et organisationnels, en termes de politique de maintenance (actions et conditions de réalisations), retenue pour le système d'étude.

Elle impose donc que notre architecture soit connue tant au niveau matériel que logiciel et qu'elle soit validée dans son contexte d'utilisation. Il importe donc de vérifier **avant la mise en route de la fonction de sécurité** que

- la cohérence et la conformité des décompositions sous fonctionnelles et leur conformité avec les spécifications de la fonction de base, ses capacités de diagnostic et de tolérance aux fautes, (**hypothèse 1**)
- les lois entrée/sortie de ces fonctions ont été testées après leur implantation logicielle et matérielle (**hypothèse 2)**
- le partage de ressources matérielles entre les sous fonctions du système d'étude et les autres fonctions de contrôle de l'installation a été si possible évité. Il ne peut être autorisé qu'après vérification de l'absence d'une situation de blocage et de conflit sur l'utilisation de la ressource par ces sous fonctions (**hypothèse 3)**
- que les sous entités fonctionnelles de l'architecture du système aient été testées et possèdent un comportement identique à celui figurant dans leur cahier des charges (**hypothèse 4**)

Nous n'entrerons pas dans le détail des méthodes de conception architecturale et de test d'implantation. On peut rappeler, en conception logicielle, l'existence de méthodes de décomposition fonctionnelle structurée (SART, UML **[Tich02]**[**Geri02**]…), de méthodes formelles (langage B et Z) permettant de vérifier par des preuves mathématiques la cohérence lors du développement de module informatique (code), leur utilisation dans la mise en place de politiques de tests efficaces (**[Behn00]**[**Smet99]**). On peut, par exemple, se référer à leur utilisation pour le développement logiciel de la ligne METEOR de la RATP (méthode B) ou pour le développement logiciel de certaines firmes automobiles (UML)… Dans de nombreuses normes (notamment l'**ICE61508**), un accent particulier a été porté sur la phase de développement, notamment par une alternance entre décomposition sous fonctionnelle et vérification ascendante des propriétés et hypothèses choisies.

Ces étapes de conception et de test (rappelées dans **[Holg02]**, **[ECSS-Q-80-03]**…) sont, certes nécessaires, pour vérifier le bon fonctionnement de l'architecture en l'absence d'altérations de ressources matérielles ou informationnelles et valider notre modèle architectural de départ (au niveau fonctionnel et matériel) mais **n'apportent aucune garantie quant au maintien de cette qualité de service lors de la durée de vie du système**. Il est donc important de noter que les hypothèses imposées lors de la conception et avant le démarrage du système **ne sont pas suffisantes** à la détermination de la qualité de service d'une fonction dédiée sécurité.

Ainsi la vérification de la conformité d'une architecture logicielle, définie par **[Haye94]** comme « une spécification abstraite consistant à décrire un module de manière fonctionnelle en

définissant ses sources d'informations, ses interfaces et les interconnections composants-composants mises en œuvre pour supporter ces opérations » est nécessaire pour éviter **les erreurs systématiques de conception**, qui peuvent déboucher sur des comportements dangereux dès la mise en route du système, mais **pas suffisante** pour démontrer sa fiabilité ou le maintien de la qualité de service du système étudié vis-à-vis des altérations de ses ressources matérielles et informationnelles.

Elle n'est donc, généralement, pas suffisante pour démontrer l'évitement de **modes de défaillances systématiques** de notre système au cours de sa durée d'utilisation. Ces défaillances sont définies par l'**IEC61508** comme « la cessation d'aptitude à délivrer un service correct liée de manière certaine à une cause, qui ne peut être éliminée que par modification de la conception, du procédé de fabrication, du mode d'emploi, de la documentation ou d'autres facteurs appropriés ». Les contraintes d'utilisation de ressources matérielles peuvent, en effet, générer ce type de défaillance, et ce malgré l'absence d'erreur de conception.

De même elle n'est pas suffisante pour démontrer qu'une déviation dans les lois d'entrées ou une erreur dans les ressources matérielles assurant la réalisation d'une fonction programmée ne puissent provoquer la mise en danger de l'installation. Le domaine aéronautique (notamment le standard **DO178B**) a mis en exergue cette possibilité et à souligné l'importance d'une étude de ces modes de déviation et de leurs conséquences lors de l'intégration et de la spécification d'architectures logicielles (lois d'interactions entre ressources matérielles et informations).

Nous supposerons donc dans toute la suite de notre étude un **niveau de connaissance** suffisant de notre architecture fonctionnelle et matérielle, et une vérification de ces propriétés de départ. Ces hypothèses préliminaires doivent nous permettre d'être en mesure de décrire l'architecture matérielle et logicielle, suivant un processus séquentiel et reconfigurable d'opérations de traitement d'information pour chacune desquelles un ensemble fini et clairement identifiable (mais non forcément disjoint) de ressources matérielles sont utilisées. Nous imposerons, en outre, une connaissance complète des propriétés de maintenabilité du système et des règles de maintenance (préventive/corrective) de ses ressources matérielles. On rappelle que la maintenabilité est définie comme l'aptitude d'une entité, dans un contexte d'utilisation fixé, à être maintenue ou rétablie dans un état où elle puisse remplir une fonction requise. Nous y ajoutons une connaissance des procédures de test, de recouvrements fonctionnels et des modes de fonctionnement dégradés permettant ces propriétés.

Enfin nous ajouterons à ce modèle de connaissance de l'architecture opérationnelle, deux hypothèses supplémentaires primordiales:

- une propriété de **sécurité-confidentialité**, garantissant l'absence d'altération par modification non autorisée d'information (c'est-à-dire de sources externes au système et non prévu par les procédures de maintenance).
- l'interdiction de remplacement d'un module matériel et/ou logiciel, jugé défaillant ou obsolète, par un module de fonctionnalités non équivalentes ou dont les propriétés de résistance aux stress n'ont pas été évaluées (en particulier pour les composants de remplacement de type COTS (components of the shelf) c'est-à-dire des remplacements standards au moyen de **modules sur étagères**.

La propriété de sécurité-confidentialité est en général garantie par des mesures de protection (interdiction d'accès) vis-à-vis de manipulation de données. Dans la plupart des systèmes industriels (absence de réseau avec l'extérieur, contrôle d'accès et interdiction de modification sur les postes opérateurs), ce problème est relativement bien maîtrisé. La problématique reste encore en suspens pour les réseaux de communication ouverts (reliés à Internet, par exemple). Ce second point est plus complexe, il convient donc de le souligner et d'apporter le cas échéant des assurances en termes de contrôle d'accès et de droit de modification sur une installation.

L'interdiction de remplacement d'un module matériel et/ou logiciel par un module de fonctionnalités non équivalentes ou dont les propriétés de résistance aux stress n'ont pas été évaluées, est imposée par la plupart des standards qui demandent une réévaluation de la fiabilité de l'architecture dans son ensemble, suite à toute modification de ce type.

2- Allocation de performances fonctionnelles

Afin d'expliquer l'intérêt d'évaluer les probabilités de défaillance sous demande et de déclenchement intempestif d'une fonction de sécurité, il est nécessaire de préciser les hypothèses permettant de relier les moyennes temporelles de ces probabilités, appelées **indicateurs de performance**, aux notions de risque résiduel et de perte économique induits par le déclenchement intempestif d'une fonction de sécurité.

2-1- Détermination du niveau de risque résiduel et hypothèses simplificatrices.

Si on considère un instant d'acquisition t_0 de la fonction de sécurité, on peut définir deux situations de départ, selon que la demande est présente au temps t_0 alors qu'elle ne l'était pas à la période d'acquisition T précédente (occurrence de demande sur $]t_0 - T ; t_0]$) ou que la demande est absente au temps t_0.

2-1-1- Cas 1: occurrence de demande sur l'intervalle $]t_0 - T ; t_0]$.

En présence de demande, la fonction de sécurité est supposée exiger une réaction dans un temps suffisamment court pour éviter l'accident. Après l'occurrence d'une demande à t_0, la fonction dédiée sécurité va identifier la présence de demande au temps t_d, vérifiant l'inégalité $t_d - t_0 < T$. Elle va ensuite synthétiser l'ordre de contrôle des actionneurs (ordre de réaction) au temps $t_d + D$ (D est le temps de cycle de la fonction) ce qui va conduire à la réaction de la fonction dédiée sécurité au plus tard au temps $t_d + D + T_i$. On rappelle que

- T_i est le temps inertiel des actionneurs,
- D est le temps de cycle séparant l'observation de l'état de demande et l'ordre de contrôle des actionneurs et
- T est la période d'acquisition de la fonction de sécurité.

En notant $T_R = T + D + T_i$, le temps de réaction maximal de la fonction de sécurité, on doit vérifier (contrainte de conception) l'inégalité $T_R < T_C$.

Dans le cas d'un fonctionnement « normal » de la fonction dédiée sécurité, on aura donc le schéma de réaction modélisé par l'axe temporel suivant :

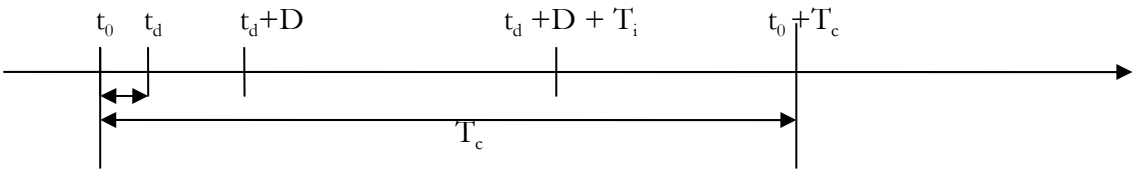


Figure 5 : Temps de réaction et temps critique d'évitement d'un accident

Dans le cas d'une absence de réaction au temps $t_d + D + T_i$, on parlera de **défaillance sur demande** de la fonction dédiée sécurité **au temps** t_0. Sa probabilité sera notée $PFD(t_0)$. La fonction de sécurité étant un système discret de pas d'acquisition T, cette probabilité sera identique quelque

soit le choix de t_0 sur l'intervalle $]t_d - T; t_d]$. En outre t_d est un multiple de T, si on note $t_d = kT$, on aura donc dans ce cas de figure, $PFD(t_0) = PFD(kT)$.

On peut considérer trois cas différents de fonctionnement après l'occurrence d'une demande à t_0 :

- Le premier cas est celui d'**une absence de réaction** de la fonction dédiée sécurité après t_0, ce qui se traduit par l'accident redouté, d'impact noté $\mathbf{I_{dem}}$.
- Le second cas est celui d'**une réaction retardée mais permettant tout de même l'évitement** de l'accident (temps de réaction inférieur à T_c). Ce cas de figure consiste à dire que le système possède des propriétés de tolérance aux fautes, lui ayant permis de réagir de manière appropriée sur l'intervalle $]t_d + D + T_i; t_0 + T_C]$ alors qu'il était défaillant sur demande au temps t_0. Le système a donc été en mesure de détecter au moins une erreur responsable de sa défaillance sur la période $]t_d + D; t_0 + T_C - T_i]$ et de commander la mise en position sûre des actionneurs (réaction de la fonction dédiée sécurité), en un temps assez court pour éviter l'accident. Cette restauration peut être due soit à la réparation d'une ressource matérielle permettant la détection de la défaillance en ligne, soit à une contrainte de périodicité de celui-ci ne lui permettant pas de couvrir son spectre de détection sur un cycle de largeur T et retardant de ce fait son résultat.
- Le troisième cas est celui une réaction retardée au delà du temps $t_0 + T_c$. Dans ce cas, l'accident ne peut être évité. Néanmoins, ce déclenchement retardé peut avoir un effet réducteur sur son impact.

Nous allons volontairement nous placer dans une approche « pire cas » pour laquelle une défaillance sous demande de la fonction de sécurité implique de manière certaine l'accident redouté dont la fonction de sécurité doit assurer la prévention. Cette hypothèse permettra de négliger le second et le troisième cas. Sous ces hypothèses la contribution d'une défaillance sous demande de la fonction de sécurité au risque résiduel est la moyenne temporelle, sur le temps de vie utile du système (T_{life}) de la fonction en escalier, F_1, suivante:

$$F_1(t) = \left\{ \int_{t=kT}^{t=(k+1)T} p(dem,t)dt. \right\} PFD(kT).I_{dem}, \qquad \forall t \in [kT,(k+1)T[$$

avec,

k, entier variant de 0 à $E\left[\frac{T_{life}}{T}\right]$ (où E est la fonction partie entière)

p(dem,t) la probabilité d'occurrence de la demande au temps t

PFD(kT) la probabilité de défaillance sur demande de la fonction dédiée sécurité au temps d'acquisition t=Kt

I_{dem} l'impact de l'accident redouté.

2-1-2- Cas d'absence de demande à t_d

Nous sommes en droit de nous demander si la fonction dédiée sécurité ne peut pas aussi induire des risques en **absence de demande**. Cette possibilité a été passée sous silence dans la norme **IEC61508**. **Cependant, il est clair qu'elle peut exister.** Le déclenchement intempestif des actions de réaction de notre système peut se traduire par:

- **la mise en danger** de l'utilisateur et de son environnement entraînant une augmentation du risque induit par l'ajout de notre système,
- la **génération d'une indisponibilité** de tout ou partie de l'installation, synonyme de dégradation du service rendu par celle-ci.

-

Le problème d'initiation de situation potentiellement dangereuse du fait du déclenchement intempestif (en absence de demande) d'une fonction dédiée sécurité, a été traité très partiellement dans la littérature. Il a été soulevé dans les applications de transport, où un déclenchement fonctionnel intempestif peut entraîner la mise en danger de l'utilisateur et de son environnement. Un exemple simple est le cas d'un airbag dans une automobile. Des cas de déclenchement intempestif ont été recensés dans les années 1980-1990 avec des impacts, en termes humain et matériel qui sont loin d'être négligeables. Ce problème de mise en danger, par le déclenchement intempestif d'une fonction dédiée sécurité, a été soulevé lors de ces dernières années essentiellement au travers des normes machines. La **[98/37/CE]** et la **IEC62061** précisent ainsi « qu'un défaut affectant la logique du circuit de commande ou une défaillance ou une détérioration du circuit de commande ne doit pas créer de situations dangereuses ». Cette clause est souvent **difficile à respecter** et **incomplète** puisqu'elle n'intègre pas implicitement d'autres défauts pouvant avoir pour conséquences un déclenchement intempestif. Il convient donc de prendre en compte ces situations.

Pour ce faire, on va supposer qu'en cas de déclenchement intempestif l'impact de l'accident qui en résulte, s'il existe, est estimable. Il sera noté $\mathbf{I_{abs}}$. Cet impact sera estimé suivant le profil de mission du système. Sous ces hypothèses la contribution d'un déclenchement intempestif de la fonction de sécurité au risque résiduel est la moyenne temporelle, sur le temps de vie utile du système (T_{life}) de la fonction en escalier, F_2, suivante:

$$F_2(t = kT) = (1 - p(dem,t)PFS(kT).I_{dem}$$

avec, PFS(kT) la probabilité de déclenchement intempestif de la fonction dédiée sécurité au temps d'acquisition t=Kt

2-1-3- Bilan

On peut donc, en considérant les deux cas précédemment énoncés, déterminer le niveau de risque résiduel à la suite de l'ajout d'une fonction de sécurité, suivant la formule:

$$R_{res} = \frac{1}{T_{life}} \sum_{k=0..N} \left(\int_{t=kT}^{t=(k+1)T} p(dem,t)dt. \right) PFD(kT).I_{dem} + \frac{1}{T_{life}} \sum_{k=0..N} \left(\int_{t=kT}^{t=(k+1)T} (1 - p(dem,t)).dt. \right) .PFS(kT).I_{abs} \quad \textbf{(Eq. 11)}$$

Cette somme est justifiée par les faits suivants:

- La non occurrence d'une demande sur un intervalle]kT, (k+1) T] est équivalente au maintien d'une absence de demande si on suppose une absence de demande à l'instant kT
- La défaillance sur demande de la fonction de sécurité se traduit toujours par un accident, jugé catastrophique, ne permettant plus le retour dans un état de fonctionnement normal de l'installation (destruction).

En l'absence d'information, on peut supposer constante la probabilité d'occurrence de demande au cours du temps. Cette supposition nous permet de proposer l'estimation:

$$\int_{t=kT}^{t=(k+1)T} (1 - p(dem,t)).dt = f.T \qquad \textbf{(Eq. 12)}$$ avec f, la fréquence d'occurrence de demande.

Si la situation dangereuse détectée par la fonction de sécurité (présence de demande) entraîne de manière certaine l'accident : f sera égale à la fréquence moyenne d'occurrence d'accident avant l'ajout de la fonction dédiée sécurité. On en déduira que

$$R_{res} = \frac{1}{T_{life}} \sum_{k=0..N} fT.PFD(kT).I_{dem} + \frac{1}{T_{life}} \sum_{k=0..N} .(1 - f.T)PFS(kT).I_{abs} = f.PFD_{avg}.I_{dem} + (\frac{T_{life}}{T} - f).PFS_{avg}.I_{abs} \qquad \textbf{(Eq.13)}$$

avec $PFD_{avg} = \frac{T}{T_{life}} \sum_{k=0\cdots N} PFD(kT)$ **(Eq. 14)** $PFS_{avg} = \frac{T}{T_{life}} \sum_{k=0\cdots N} PFD(kT)$ **(Eq. 15)** et N défini dans l'**Eq. 8**.

-

On a donc bien une relation à un degré de liberté entre les deux indicateurs de performances PFD_{avg} et PFS_{avg} et le risque résiduel. Imposer une prescription en termes de risque résiduel acceptable revient donc :

- soit à fixer la valeur maximale de l'indicateur PFD_{avg} si I_{abs} est nul.
- soit à fixer le seuil maximun de l'indicateur PFD_{avg} par rapport au seuil maximum prescrit de PFS_{avg} si I_{abs} est non nul.

2-2- Détermination de la perte économique induite et hypothèses simplificatrices.

Une fois la relation entre le risque résiduel et les métriques PFD_{avg} et PFS_{avg} définies, nous souhaitons estimer la perte économique induite par le déclenchement intempestif. Nous noterons C_{ind}, cette métrique.

On rappelle que la conséquence de l'évitement d'un accident ne peut **jamais** être considérée comme une perte économique et ce même si elle peut s'accompagner d'une perte de productivité. En effet, les conséquences d'un évitement sont toujours négligeables devant celle d'un accident non évité. Nous supposerons, en outre, que la perte économique induite par **l'accident** redouté rentre directement en ligne de compte dans l'évaluation des métriques d'impact I_{pres} .

La réaction intempestive d'une fonction dédiée sécurité peut, pour sa part, induire une indisponibilité de tout ou partie de l'installation. Cette indisponibilité se traduit par un coût sur les missions de l'installation pouvant être associé à une perte économique. Les conséquences du déclenchement intempestif de notre système (fonction de sécurité) sont souvent synonymes de dégradation du service rendu par l'installation, en termes de production ou de mission. Le premier problème est de savoir comment ce déclenchement peut porter à conséquences.

On peut distinguer deux approches pour estimer cette perte:

- une approche linéaire, consistant à associer à une mise en indisponibilité un coût moyen induit (en termes de perte économique par unité de temps T), noté c* (1)
- une approche cumulative, consistant à associer à une mise en indisponibilité du système un coût dépendant du temps de persistance de ce mode de déclenchement, et donc de la politique d'inspection permettant d'en sortir. (2)

Nous choisissons l'hypothèse simplificatrice (1). Nous supposerons donc qu'un déclenchement intempestif peut être un processus réversible. Cette propriété sera supposée comme une conséquence exclusive des propriétés de maintenance corrective et préventive du système et de ses capacités de reconfiguration.

Nous supposerons, en outre, que le maintien injustifié de la réaction d'une fonction de sécurité (c'est-à-dire en absence de demande) **sur un cycle de largeur T** (décision) a le même effet que l'apparition d'un déclenchement intempestif au début de ce cycle. Cette hypothèse d'indépendance entre la perte économique générée sur un cycle d'indisponibilité de largeur T et la durée d'indisponibilité antérieure à celle-ci est **capitale**.

Ces hypothèses nous permettent d'écrire que:

$$C_{ind} = \sum_{k=1..E\left(\frac{T_{life}}{T}\right)} \{[1 - p(demand, kT)].PFS(kT).c^*(kT)\} = (\frac{T_{life}}{T} - f).PFS_{avg}.c^* \qquad \textbf{(Eq. 16)}$$

On peut distinguer deux cas:

- soit I_{abs} est nul, on peut alors affirmer que c* ne l'est pas, ce qui impose une prescription maximale du PFS_{avg} , si le niveau de perte économique maximal du au déclenchement intempestif est prescrit.

- soit c* est nul, dans ce cas I_{abs} ne l'est pas, on conserve dans ce cas un degré de liberté entre les équations 16 et 13. Il faut donc choisir un seuil de prescription pour le PFS_{avg} qui imposera un seuil de prescription pour le PFD_{avg} suivant la connaissance du niveau de risque résiduel acceptable.

L'indicateur de performance PFS_{avg} nous permet donc de juger de la perte économique induite, quand elle existe, par l'introduction de la fonction dédiée sécurité.

2-3- Bilan

Au vu d'un niveau de risque résiduel acceptable et d'un niveau de perte économique maximal admis, il est possible de fixer de manière univoque les seuils de prescriptions des deux indicateurs de performances PFD_{avg} et PFS_{avg} permettant de garantir le respect des objectifs affichés, sous les hypothèses mentionnés et en supposant que le déclenchement intempestif ne se traduise pas par un accident.

Dans le cas où le déclenchement intempestif se traduit par un accident (c*=0, I_{abs} non nul), l'industriel se doit de fixer un rapport entre les deux seuils de prescriptions portant sur les indicateurs de performances. Ce rapport permettra de justifier, ensuite, au regard du niveau d'acceptabilité du risque résiduel, les seuils de prescriptions des deux indicateurs de performances PFD_{avg} et PFS_{avg} de la fonction de sécurité (par le respect de la relation 13). Il revient à fixer une pondération entre la contribution de la défaillance sous demande et du déclenchement intempestif au risque résiduel.

On supposera dans la suite de notre étude, que les prescriptions des deux indicateurs de performances ont été effectuées suivant ces préceptes. Ces prescriptions permettent donc, en accord avec les hypothèses avancées, de garantir un niveau de risque résiduel acceptable de l'installation ET une perte économique raisonnable provoquée par le déclenchement intempestif de la fonction de sécurité. Ces prescriptions resteront valables sous les hypothèses suivantes :

- approximation de la probabilité d'occurrence de la demande sur un intervalle de largeur T par la métrique f. T constante.
- impact d'un scénario d'accident suite à une défaillance sur demande de la fonction dédiée sécurité supposé égale à celui de l'accident redouté et constant (pire cas)
- existence d'une métrique de coût unitaire c* constante, reflétant la perte économique générée par le déclenchement intempestif sur un cycle de la fonction de sécurité.

Il importe de noter que ces prescriptions sont toujours basées sur le scénario du « pire cas » consistant à associer à une défaillance fonctionnelle une conséquence fixée en termes d'impact. Elles permettent donc de garantir que le niveau de risque résiduel se trouve en deça d'une limite de tolérance mais pas toujours d'estimer de manière précise le risque résiduel. Cette estimation peut, en fait, être conduite a posteriori si on connaît la variation au cours du temps des probabilités de défaillance intempestive et sous demande de la fonction de sécurité. Il est, dans ce cas, possible de simuler plus avant les comportements installation-système (fonction de sécurité) en utilisant les techniques de « fiabilité mécaniques » introduite par **[Labe00]**, **[Berg04]**. Cet aspect ne sera pas développé dans notre étude.

3- Incomplétude des standards actuels en sécurité fonctionnelle

L'évaluation quantitative de certains indicateurs de performances (fiabilistes) pour les fonctions de sécurité électroniques et programmables a été proposée par la norme **IEC61508**, dite généraliste qui a ensuite été déclinée dans les domaines de l'industrie de process (**IEC61511**), de l'industrie manufacturière (machine **IEC62061**), nucléaire (**IEC61513**)… Ces standards, appelés

standards de sécurité fonctionnelle, assimilent la qualité de service d'une fonction dédiée sécurité à un niveau d'intégrité, appelé SIL (Safety Integrity Level). L'obtention de ce niveau de SIL est principalement basée sur quatre conditions d'attributions :

- une condition de qualité lors de la conception et la vérification de l'architecture, mettant en avant l'utilisation de méthodes structurées pour démontrer les propriétés sous fonctionnelles de base et leur conformité avec le cahier des charges fonctionnelles.
- une condition de bonne mise en oeuvre des procédures de maintenance lors de la durée d'utilisation de cette architecture.
- une condition liée à la valeur numérique de l'indicateur PFD_{avg}, qui doit suivant la fréquence de sollicitation de la fonction (fréquence d'occurrence moyenne de la situation à détecter) se trouver dans un domaine d'acceptation.
- le respect de certaines spécifications minimales en termes architecturale, permettant d'assurer certaines propriétés de tolérance aux fautes (diagnostic avec prescriptions sur taux de couverture, redondance…).

Le principe de l'IEC61508 est louable dans le sens où cette norme essaie de justifier la mise en place d'une allocation d'objectifs en termes d'indicateur de performance (PFD_{avg}) et qu'elle insiste sur le besoin d'outil de vérifications architecturales et de politiques de test, permettant d'éviter les erreurs systématiques lors de la conception et de l'implantation de l'architecture opérationnelle. Cependant, nous nous permettons d'y relever quelques lacunes majeures:

- Elle ne s'applique qu'à des **fonctions dédiées sécurité pour lesquelles deux uniques niveaux de service (correct/ incorrect) peuvent être définis** en présence de demande. Elle est donc bien adaptée aux fonctions PE de classes SDSA mais s'avère insuffisante pour le traitement de fonctions de sécurité possédant plusieurs réactions mises en œuvre de manière collaborative (démarche proposée dans l'**annexe A**).
- Elle distingue pour l'attribution des objectifs de PFD_{avg} en fonction du niveau de SIL deux niveaux de demande (haute fréquence/basse fréquence) sans montrer le lien entre la notion de risque résiduel et cette fréquence de référence (comme nous l'avons proposé dans l'**équation 13**), alors que cette distinction n'est pas toujours possible dans la pratique **[Sato00]**. Cet aspect sera amené à disparaître progressivement.
- Elle néglige **les effets d'un déclenchement intempestif** sur le niveau de risque résiduel induit en supposant implicitement que ce mode de déclenchement n'est jamais contributeur d'accident. Ce point n'est pas mentionné dans l'**IEC61508** et apparaît uniquement dans l'**IEC61511** (normes process).
- Elle propose un certain nombre de **voies d'améliorations architecturales** (isolement et diversité de voies redondantes, test sur ressources matérielles…), en se bornant à leur intérêt sur des cas d'étude extrêmement simples et ne donne pas de méthode permettant d'intégrer de manière réaliste ces choix au calcul du PFD_{avg} pour des architectures complexes, où le partage d'information ou de ressources rend plus difficile l'établissement de modèles d'évaluation quantitatif (intelligence distribuée, réseaux…)
- Elle donne la fâcheuse impression que l'évitement d'erreur de conception est une condition suffisante pour assurer le caractère négligeable de défaillances dites systématiques, ce qui est **généralement faux**. En effet, une conception valide d'une architecture, n'implique nullement sa robustesse aux fautes éventuelles. Il est de fait possible qu'une architecture puisse être bien conçue, c'est-à-dire qu'elle réponde initialement au cahier des charges fonctionnel fixé, sans pour autant qu'elle présente des performances satisfaisantes. En effet, les choix architecturaux, basés sur la décomposition

de notre spécification fonctionnelle, puisqu'ils s'appuient, généralement, sur des hypothèses de « bon fonctionnement », ne permettent pas de prendre en compte l'ensemble des déviations fonctionnelles locales pouvant avoir lieu dans notre architecture.

Nous pouvons donc affirmer que même si ce groupe de normes a pour avantage de montrer **la nécessité de mise en place d'objectifs quantitatifs** sur des indicateurs de performances fonctionnelles et d'assurer une vérification approfondie de l'architecture fonctionnelle avant sa mise en service, elle n'offre que peu d'éléments permettant de mener à bien cette évaluation dans le cas d'architectures complexes (tolérante aux fautes, réparables…). Elle offre, en effet, une vision trop étriquée du système à modéliser en passant sous silence les vulnérabilités que peuvent faire apparaître les politique de test, de reconfiguration et de réparation dans la bonne mise en œuvre de la fonction. En ce sens, elle ne permet pas de mener des estimations permettant de juger des avantages et inconvénients des choix architecturaux d'une fonction de sécurité. Elle a, en outre, l'inconvénient majeur de ne pas s'intéresser aux conséquences éventuelles du déclenchement d'une fonction dédiée sécurité, négligeant, à la fois, la mise en danger de l'installation qui peut en découler et occultant le compromis nécessaire entre risque résiduel et perte économique induite.

Au regard de ces limitations, un **autre groupe** de normes s'est développé ces dernières années, motivé principalement par les problèmes d'interactions de fonctions de contrôle sur le bon fonctionnement d'une installation. Ces normes, appelés normes **FMDS** (ou RAMS en anglais), mettent en regard l'adéquation des moyens mis en œuvre, c'est-à-dire des choix architecturaux et de maintenance pour remplir une fonction, et des conséquences en termes de qualité de service de l'installation qu'induit l'ajout de cette fonction (en termes de sécurité-innocuité et de propriété de disponibilité). Une représentante de ces normes dans le domaine ferroviaire est l'**IEC62278**.

Cette norme insiste beaucoup plus sur l'aspect de choix et d'allocation d'objectifs fonctionnels. Une partie importante est dédiée aux interactions fonctionnelles de fonctions coopératives (simultanées ou séquencées) et à l'aspect temporel dans le traitement des situations accidentogènes. L'avantage majeur de ces normes est de replacer les choix architecturaux dans une démarche d'atteintes d'objectifs fonctionnels et de considérer les propriétés de fiabilité et de maintenabilité d'une architecture comme déterminantes vis-à-vis de la qualité de service qu'un ensemble fini de fonctions pourra délivrer à l'installation. Ces normes permettent donc la mise en place d'objectifs en termes de performances fiabilistes fonctionnelles et soulignent le besoin de leur démonstration sans présager de quelconque choix architecturaux ou de politique de maintenance, **considérés simplement comme des moyens** pour atteindre ces objectifs finaux. Elles soulignent le besoin d'une démarche de vérification architecturale, sans présenter les méthodes de vérification mentionnées dans l'**IEC61508** et la définissent comme un prérequis de notre étude permettant d'assurer un niveau de connaissance satisfaisant de notre architecture opérationnelle au temps de mise en route de la fonction considérée. Le caractère général de ces normes (s'appliquant autant aux fonctions de contrôle qu'à celle dédiée sécurité) permet d'insister plus sur le besoin d'allocation et de démonstration de indicateurs de performances fonctionnelles en **refusant de proposer des voies d'amélioration architecturale figées**. On va donc considérer l'architecture et les choix de procédures de test et de recouvrements comme des contraintes dont l'influence sur ces indicateurs doit être implicitement prise en compte lors de leur évaluation.

Notre étude s'intéresse aux conséquences de l'ajout d'une fonction de sécurité PE unique. Nous ne nous intéresserons pas aux interactions externes entre notre système et son environnement comme nous l'avons précédemment souligné. Cependant, nous nous placerons dans l'état d'esprit des normes FMDS pour mettre en exergue l'importance des interactions sous fonctionnelles (test, contrôles, décision, transformation…) ainsi que des règles de couplages dans notre système (partage de ressources, contraintes temporelles…) et prendre en compte leur influence dans

l'évaluation des indicateurs de performances retenus (PFD_{avg} ; PFS_{avg}). De ce fait, notre démarche ne nécessitera pas l'acceptation de contraintes architecturales fortes, proposées par l'**IEC61508** mais rarement respectées dans la réalité. Elle placera la justification d'un niveau de risque résiduel et d'une perte économique acceptable au centre de nos préoccupations et devra prendre en compte :

- l'exposition des ressources à des stress environnementaux communs
- le partage de ressources matérielles et informationnelles entre sous fonctions
- l'existence de différentes constantes de temps pour les procédures de test, de recouvrement et de réparation…

Nous allons monter, dans la partie suivante, l'importance que peuvent avoir certains modes de défaillance sur la qualité des estimations d'indicateurs de performances et montrer les difficultés de leur prise en compte au travers des modèles actuels d'évaluation. Ce point permettra de justifier la nécessité d'une méthode d'évaluation plus respectueuse des défaillances multiples, pouvant apparaître dans des fonctions EP tolérantes aux fautes et/ou réparables.

4- Méthode d'évaluations actuelles et limites

On peut principalement distinguer deux classes de méthodes d'estimation des performances fiabilistes en sûreté de fonctionnement: les méthodes qualitatives et les méthodes quantitatives. Dans cette thèse, nous nous intéresserons seulement aux méthodes d'évaluation quantitative. Elles s'appliquent à des systèmes statiques ou dynamiques, au sens de leur fonction de structure[2] **[Kauf75]**. La notion de fiabilité dynamique, développé dans la littérature **[Labe00]**, concerne plus l'interaction entre le système d'étude et son environnement dont l'évolution déterministe dans le temps est donnée par des systèmes d'équations d'états (avec l'aide d'automate hybride déterministe, par exemple). Elle ne sera pas ici abordée puisque l'étude du système se fait en boucle ouverte (cf. **figure 4**).

Nous proposons dans le tableau suivant (**figure 6**) de donner un aperçu des modèles d'évaluation quantitative existants.

Modélisation	Description	Référence	Exemples d'outils
Méthodes statiques			
RBD (diagramme de fiabilité)	Représentation de la structure de système sous forme d'un diagramme inspiré des schémas électriques (impédances, circuit série, parallèle…)	**[Vill87]**	Jagrif-Bloc Diagram, Exida, Riskspectrum, SuperCAB,..
Fonction de structure	Formalisation de la structure par une fonction algébrique construite sur les entiers 0 et 1	**[Kauf75]**	
Réseau de fiabilité	Représentation de la de structure du système cohérent par un multigraphe	**[Kauf75]**	

[2] La fonction de structure décrit la dépendance de l'état de fonctionnement du système vis-à-vis de l'état de tous ses constituants.

FTA	Représentation de la structure du système sous forme d'arbre logique et d'un polynôme booléen	**[Ville87]**	Aralia- SimTree, CapTree, Fault Tree +, CAP-Tree, Risk Spectrum, ..
BDD (diagramme de decision binaire)	Représentation de la structure du système sous forme d'arbre binaire (Th d'expansion de Shanon)	**[Rauz00]**	Aralia
Modèles Markoviens	Formalisation de la structure du système par un graphe associé à un système d'équations différentielles sur les probabilités de chacun des états du système.	**[Gond80]**	Jagrif MOCA, CARMS, SURF2, SuperCab
GSPN (réseau de Petri stochastique)	Matérialisation de la fonction de structure du système par le graphe d'accessibilité du réseau de Petri	**[Noye90]**	JaGrif RP, Design CPN
Méthodes dynamiques			
Processus stochastique déterministe par morceaux	Représentation de l'état du système par utilisation de processus stochastique dont les lois de transitions entre états dépendent de modes. La transition entre ces modes est donnée par un système d'équation différentielle portant sur un ensemble de variables temporelles. **[Ever00]**	**[Dutu02]**	0
GSPN (réseau de Petri stochastique)	Modélisation comportementale du système par réseau de Petri interprété puis évaluation par transformation en réseau stochastique.	**[Scho02], [Triv96], [Bore96]**	Développement de modules sur JaGRIF, SURF2
Logique linéaire	Formalisation en logique linéaire d'un réseau de Petri stochastique	**[Demm02]**	0
BDMP (Boolean Driven Markov Process)	Représentation du système dynamique par un arbre des causes contrôlé par l'évolution de processus Markoviens.	**[Boui01]**	FigSeq
Altarica (automates à contraintes)	Représentation du système par le langage Altarica, basés sur la théorie des automates à contraintes	**[Poin00], [Page04]**	ARBoost (incluant Aralia générateur et évaluation de FTA), Moca-RP (module de génération de GSPN et simulation de Monte Carlo)
Reconfigurable Binary Decision Diagram	Représentation du système dynamique par un diagramme de décision binaire contrôlé par l'évolution de processus Markoviens	**[Duga99]**	Galileo Fault Tree
-	Equations de Chapman-Kolmogorov généralisées	**[Labe00]**	-

-

Au vu de cette brève présentation des méthodes d'évaluation quantitatives existantes, on peut affirmer qu'aujourd'hui seules les méthodes dites dynamiques peuvent permettre de prendre en compte les propriétés de maintenabilité et de reconfiguration d'une architecture tolérante aux fautes et d'évaluer leur impact sur les indicateurs de performances évalués. Cependant, l'utilisation de telles méthodes pose encore quelques problèmes:

- **Explosion de la taille des modèles** rendant difficile la résolution des problèmes (équation de Chapman Kolmogorov) ou impliquant des approximations liées à leur simplification quand elle est possible (par méthode d'agrégation quand certaines propriétés dynamiques ou de convexité peuvent être réunies **[Scho02]**).
- **Difficulté de structuration de la complexité ;** les RdPs qui constituent un bon outil de représentation de l'hybridité, ne se prêtent pas bien à la modélisation hiérarchisée**.**
- **Difficulté de construction des modèles**, la plupart étant des méthodes exploratoires et donc forcément incomplètes, ou basées sur une représentation à RdPs avec contraintes sur les événements de transition. Il n'y a pas de méthode de construction systématique.
- **Difficulté de prise en compte simultanée de l'aspect** déterministe et stochastique du temps (contrainte de test, contrainte de maintenance).
- Difficulté de prise en compte des reconfigurations fonctionnelles autres que tout ou rien et des processus de décision.
- **Difficulté de prise en compte du partage de ressources** (notamment d'information, mais aussi matérielles).
- **Difficulté de modélisation** des contraintes architecturales et de temps de test dues aux procédures de **reconfiguration fonctionnelle**.
- **Difficulté de prendre en compte l'interaction logiciel/matériel**

Les évolutions de ces méthodes au cours des dernières années ont amélioré la lisibilité et réduit la complexité des modèles (notamment par la prise en compte d'interactions entre sous modèles). Cependant cette **évolution ne s'est pas en général faite au bénéfice du pouvoir de représentation de tels modèles**. Certaines contraintes, notamment de temps de test, de partage de ressources (matérielles et informationnelles), de relations d'ordre entre sous fonctions, de persistance de pannes matérielles (par réparation) et de maintenance apparaissent rarement, ce qui pose un problème face à l'évolution actuelle des architectures EP dédiées sécurité. Ces évolutions architecturales, se fondent sur :

- des partages plus importants de ressources matérielles et d'information entre sous fonctions distinctes,
- des implantations plus fréquentes de tests en ligne et de procédures de reconfiguration,
- la manipulation d'informations de nature distincte (données ou diagnostic) entre sous fonctions.

Elles sont, à notre avis, une cause importante de difficulté des méthodes d'évaluations actuelles des indicateurs de performances. Pour s'en convaincre, il suffit d'observer les retours d'expériences de certains grands organismes comme la NASA qui affirme que **plus de 30%** des accidents actuellement recensés pour des fonctions de contrôle commandes tolérantes aux fautes sont dues à des défaillances matérielles multiples. Sachant que la NASA accorde une importance particulière à l'étude de modes communs de défaillances et aux possibilités de propagation d'erreurs au sein des architectures, nous sommes en droit d'estimer cette part comme pouvant être beaucoup plus importante qu'annoncée dans de nombreux secteurs d'activités (les études accidents ne permettent pas toujours de remonter de manière correcte à l'ensemble des causes

ayant provoqué une défaillance fonctionnelle). Nous sommes donc aujourd'hui tentés d'affirmer que les estimations de disponibilité et de défaillances sous demande, données par les modèles actuels qui négligent certaines défaillances multiples, peuvent être fortement biaisées.

Cette tendance semble, d'après nos sources (retour d'expérience Siemens), beaucoup plus marquées dans le cas d'architectures distribuées programmables, possédant un nombre important de procédures de diagnostic, et des niveaux de redondances multiples, et notamment dans l'industrie du process où certaines hypothèses d'indépendances en termes d'informations et de ressources matérielles sont, en général, moins scrupuleusement vérifiées et documentées que dans les domaines aéronautiques et nucléaires où les logiques câblées continuent à prédominer.

Il convient donc de proposer une méthode d'évaluation basée sur le caractère dynamique de l'architecture, c'est-à-dire prenant implicitement en compte l'ordre des sous fonctions, les contraintes de partages de ressources matérielles et informationnelles entre celles-ci et les capacités de test et de modifications fonctionnelles de cette architecture. On va donc se déplacer d'une vision statique ou faiblement reconfigurable de l'architecture du système à une vision dynamique, basée à la fois sur l'évolution temporelle de l'état d'intégrité des ressources matérielles utilisées, mais aussi sur les règles d'utilisation de ces ressources.

Le problème principal induit par l'évolution des architectures actuelles dédiées sécurité est que les procédures de tolérances aux fautes, souvent développées au niveau local, **l'ont toujours été dans l'hypothèse d'un fonctionnement parfait du reste du système** (et donc à l'absence d'utilisation d'autres procédures de recouvrement fonctionnel ou de défaillances éventuelles d'autres sous fonctions). Or ces procédures interagissent avec le reste du système, et il convient donc de prendre en compte **les contraintes et les vulnérabilités** qu'elles peuvent induire sur le système. On peut citer l'utilisation par exemple :

- de ressources matérielles dont le niveau d'intégrité et certaines fonctionnalités **sont testés périodiquement** (capteurs et actionneurs dits « intelligents », valve testée périodiquement en fermeture par un contrôleur…)
- de **redondances d'information complètes** (dans le cas de signaux jugés équivalents) ou **partielles** (dans le cas d'ajout de signature de contrôle générée de type CRC ou Check Sum)
- de **contrôle d'actualisation et de bonne réception** d'information (horloge de contrôle pour protocole maître esclave, temps limite d'acceptation des résultats d'une procédure logicielle…)
- de réseaux permettant la transmission séquencée d'information de **contrôle** mais aussi de **diagnostic** entre modules matériels distants.

Nous sommes tout à fait conscients que l'évolution des architectures PE actuelles vers une répartition des procédures de diagnostic et de décision va logiquement tendre à augmenter dans les architectures complexes le poids de ces contraintes sur les indicateurs de performances de tels systèmes. La complexité accrue du système d'étude peut donc être un vecteur de nouveaux mécanismes de propagation d'erreurs dont l'influence sur les indicateurs de performances pourrait altérer de manière importante l'exactitude des résultats numériques avancés, voire même fausser l'ordre de grandeur de ceux-ci. Il est donc primordial de prendre en compte dans l'évaluation de nos indicateurs de performances non seulement les parties testées mais aussi les processus de décision. En effet, il est illusoire de penser que dans une architecture EP, tout signal de diagnostic se traduise nécessairement par l'activation des moyens de réactions de la fonction de sécurité. De même, il est illusoire de penser que deux voies de redondance partageant certaines ressources matérielles ou informationnelles ne se trouvent pas conjointement affectées par le défaut d'une ressource partagée.

L'importance de ces défaillances a été soulignée dans la littérature au travers des cas particulier de défaillance de causes communes et des problèmes de latence de mode de défaillance. Nous allons donc analyser le traitement actuel de ces classes de défaillances particulières. Nos critiques nous permettront de proposer une définition plus claire des limites de notre système d'étude, et déboucheront sur l'explication des mécanismes d'initiation et de propagation d'information erronée dans l'architecture fonctionnelle de notre système d'étude.

Dans la pratique, l'approche combinatoire pour déterminer la fonction de structure semble donc incomplète (que ce soit un arbre logique, un graphe Markovien ou un diagramme de décision binaire, ce sont toujours des **graphes d'états**). L'apparition d'une défaillance du système étudié est néanmoins toujours liée à l'occurrence d'une séquence ordonnée d'événements. Pour construire notre fonction de structure, nous nous baserons sur des séquences ordonnées d'événements et non sur des combinaisons d'événements. Le modèle de base sera donc de **type graphe d'événements.**

Pour ce faire, nous proposons de modéliser notre système comme une séquence reconfigurable de processus de traitement d'informations. Cette idée de base devrait nous permettre tout d'abord de clarifier les notions d'initiation, de propagation et d'inhibition d'informations erronées dans une architecture EP. Elle pourra ensuite mettre en exergue l'influence des procédures de test et de reconfiguration implantées ainsi que de l'état d'intégrité des ressources matérielles sur le mode de fonctionnement correct/incorrect de la fonction de sécurité pour un état de demande fixé. Nous expliquerons dans quelles mesures une telle idée peut permettre de mieux représenter les problèmes de défaillances simultanées, et en particulier d'apporter une estimation plus précise de l'influence des défaillances de causes communes et des modes latents de défaillances qui semblent aujourd'hui, peu ou incomplètement modélisés. Nous nous accordons par exemple avec **[Char02]** pour ce qui relève des défaillances de causes communes CCF et avec **[Bier03]** pour ce qui est des modes latents de défaillance.

Les définitions de défaillances de causes communes sont nombreuses mais font état de la même évolution. Avec le développement d'architectures réparables et tolérantes aux fautes, on a vu un glissement d'une définition, valable dans les années 1970, et essentiellement orienté matériel qui assimilait principalement une défaillance de cause commune à « l'occurrence simultanée de modes de défaillances dans des voies de redondances matérielles due à une cause partagée et aboutissant à une défaillance fonctionnelle du système (fonction de sécurité) », à la définition proposée par l'IEC61508 : « **défaillance [due à l'occurrence simultanée, ou quasi simultanée, de deux événements (ou plus) causés par l'occurrence d'une cause unique**»[3]. On a donc remplacé la notion de défaillances matérielles (au pluriel) par la définition d'**événements** et on a laissé une interprétation ouverte sur la définition de « quasi simultanéité ». Cette définition permet donc d'accepter que le partage d'une ressource par plusieurs sous fonctions puisse affecter de manière importante les lois fonctionnelles de celle-ci et permettre la propagation d'une ou plusieurs informations erronées dans notre architecture. Cette erreur matérielle induira donc une défaillance de cause commune et ce quelque soit les sous fonctions (appartenance à des voies redondantes ou non) dans lesquelles la ressource matérielle est utilisé. De même, le partage d'information entre différentes voies de traitement peut être considéré potentiellement comme une cause commune de défaillance. Nous proposons donc de prendre implicitement les modes de défaillances de causes communes dans notre modèle en les intégrant au regroupement des deux notions complémentaires:

---

[3] **Cette définition inclut les coupes minimales d'ordre 1** quand celle-ci résulte de l'activation répétée d'une ressource matérielle défectueuse dans l'architecture dédiée sécurité. On rappelle la définition de défaillance de cause commune de l'**IEC61513**: « défaillance résultant d'un ou plusieurs événements induit par une unique cause, provoquant la défaillance simultanée de

- deux ou plusieurs canaux séparés d'un système multi canal ou de plusieurs sous-systèmes, et
- conduisant à la défaillances du système considéré »

-

- La notion d'ICDE[4] introduite dans **ECSS-Q-80-03** et dans le milieu nucléaire définie comme une défaillance de la fonction de sécurité due à la révélation d'au moins deux erreurs matérielles distinctes sur un intervalle de temps défini et pouvant être vues comme le résultat direct d'une cause partagée.
- Les défaillances fonctionnelles, induites par le partage d'une ressource matérielle ou d'une information erronées par plusieurs sous fonctions.

Un autre problème fréquemment débattu est la prise en compte réaliste des modes de défaillances latents.

En présence de certains modes de défaillances matériels, notre fonction de sécurité peut, en effet, présenter un mode de fonctionnement, qualifié de normal. On pourrait définir ce type d'état de fonctionnement particulier comme un « mode dégradé sans altération des performances fonctionnelles ». Ce type de mode est fréquemment rencontré dans les systèmes dits « tolérants aux fautes » et nous ne citerons que l'exemple simple du basculement sur une seule voie d'une d'architecture d'acquisition duale dont l'autre voie est diagnostiquée défaillante et qui subit à ce titre une réparation. Dans ce type d'état (« mode dégradé sans altération des performances fonctionnelles »), il est tout á fait possible qu'un autre événement (par exemple la défaillance d'une ressource matérielle de l'autre voie) entraîne une défaillance fonctionnelle. Si cet événement seul n'est pas suffisant pour provoquer, en l'absence du « mode dégradé » considéré, la défaillance fonctionnelle, alors il est appelé événement **activateur et le mode dégradé est appelé mode latent de défaillance**. Ce phénomène est lié à la révélation d'une erreur du fait de la sollicitation dans un mode particulier de la ressource présentant cet état.

Ces « modes latents de défaillance » ont été souvent relevés dans les études d'accidents de ces dernières décennies. Ils sont (à tort) souvent assimilés à des erreurs d'ordre humain (maintenance, procédure manuelles), qui sont souvent les précurseurs de l'événement activateur. Cependant, ils relèvent complètement des contraintes architecturales, puisqu'un événement activateur sans mode de défaillance latent ne conduit pas à la défaillance du système d'étude. L'étude des modes latents est donc actuellement extrêmement limitée et ce, malgré leur importance relative, comme souligné dans **[Holn01]**. Cet impact des modes de défaillances latents est d'ailleurs souligné dans **[Bier03]** au moyen de quelques exemples.

En outre, de nombreux modes de défaillances fonctionnels résultants de la défaillance combinée de ressources matérielles sont parfois plus assimilables à l'activation de modes latents qu'à des défaillances de cause commune, leur occurrence pouvant être éloignées dans le temps (en contradiction avec la notion de quasi simultanéité). La difficulté d'évaluation de l'influence de ces modes de défaillances a été soulignée dans les enquêtes qui privilégient une approche fonctionnelle exploratoire ascendante et qui peuvent se révéler incapable de distinguer cette différence.

La prise en compte de mode latent impose donc, à notre avis, une définition plus poussée de la notion de temps que celle proposée dans les modèles d'évaluations actuels. En effet, cette notion est basée sur les concepts de persistance d'erreurs sur plusieurs cycles de réaction T de la fonction de sécurité, de leur activation éventuelle et de l'ajout d'un autre événement, dit « révélateur », pouvant être lié soit à une faute matérielle, une action de maintenance, une altération transitoire d'une information, etc.

[4] Extrait des actes du séminaire EuroSafe 2003 - An "ICDE event" is defined as follows: "Impairment of two or more components (with respect to performing a specific function) that exists over a relevant time interval and is the direct result of a shared cause" (lien: http://www.eurosafe-forum.org/forum2003/seminaires/seminaire_1_9.pdf).

Une approche événementielle devrait nous permettre de mieux traiter ces points. Nous nous proposerons donc dans notre étude de nous intéresser à l'évaluation des indicateurs de performance d'une fonction de sécurité en privilégiant une approche événementielle, plus à même de représenter la structure dynamique de la fonction étudiée. Nous devons pour ce faire définir les limites du système étudié et les phénomènes pouvant influencer le niveau de service de la fonction de sécurité.

5- Frontières du système à modéliser

Maintenant que nous avons souligné certaines difficultés des modèles actuels, nous allons définir les limites de notre système d'étude (distinction entre système et environnement) et expliquer les interactions entre ce système et son environnement.

Nous supposons que les choix architecturaux en termes fonctionnels et matériels ainsi que les choix de politique de maintenance sur les ressources matérielles sont connus. Nous supposerons, en outre, que les moyens de tests et les procédures de tolérance aux fautes implantées pour assurer une meilleure qualité de service de la fonction dédiée sécurité, sont décrites de manière complète (spectre de détection, contrainte de temps de test, ressources matérielles mises en oeuvre).

Nous admettrons que cette architecture puisse posséder des « vulnérabilités », évoluant au cours du temps et que ces vulnérabilités sont les conséquences des choix architecturaux (logiciels et matériels) avancés et de l'existence d'erreurs dans le système due aux contraintes environnementales. Elles ne sont donc pas à confondre avec les « erreurs de conception » qui se traduisent par une défaillance fonctionnelle d'une ou plusieurs sous fonctions lors de leur première sollicitation en l'absence d'altération de l'état d'intégrité des ressources matérielles qu'elles mettent en œuvre. Il est donc possible que la conjonction de différentes erreurs puisse induire l'existence d'un mode de défaillance particulier pour une fonction de sécurité. Ces « combinaisons » dépendront de l'état de demande observable, des règles d'utilisation des ressources matérielles dans les fonctions de base, de test et de recouvrement, des règles de reconfiguration architecturale et de l'ordre entre les sous fonctions permettant d'induire l'état d'activation des moyens de réaction de la fonction de sécurité (actionneurs). Dans la suite de notre travail, nous supposerons que :

- le processus de vérification de l'architecture a été correctement mené avant sa mise en service et permet d'assurer un bon fonctionnement de la fonction dédiée sécurité et l'intégrité des ressources matérielles au début d'utilisation du système étudié (cette condition implique PFD(t=0)=PFS(t=0)=0, en notant t=0 l'instant de démarrage de l'installation).
- les règles d'utilisation des ressources sont de nature déterministe et peuvent être seulement influencées par des procédures de contrôle (y compris temporel) et de diagnostic en ligne.
- les règles de maintenance sont soit correctives, soit planifiées (intervalle d'inspection fixé)

Ces hypothèses vont nous permettre d'une part de définir clairement les limites du système à étudier et d'autre part d'introduire la notion d'information qui sera par la suite à la base de la construction de notre modèle d'évaluation générale.

Nous allons en premier lieu introduire un référentiel. Ce référentiel est basé sur une définition précise des limites du système à étudier, c'est-à-dire de notre fonction dédiée sécurité et de ses interactions éventuelles avec son environnement. On retiendra d'abord deux notions principales: celle de boucle de sécurité et celle de fonction de sécurité.

La notion de **boucle de sécurité** est définie comme : « une architecture matérielle et logicielle visant à transformer une information observable, correspondant à la présence ou à l'absence de demande à une période d'acquisition fixée, en une action, en termes d'activation/désactivation de l'ensemble des moyens de réaction prévus dans le cahier des charges de la fonction de sécurité ».

Le système d'étude est constitué

A- au niveau fonctionnel :

- **des fonctions dites de base**, permettant à la fois de remplir les spécifications fonctionnelles de la fonction de sécurité et d'assurer la reconfiguration fonctionnelle ou la mise en mode dégradé de cette fonction (forcer une variable à une valeur de référence, par exemple)
- des fonctions dites de diagnostic, regroupant

  - des tests par comparaison de signal (redondances)

  - des tests de ressources matérielles (en ligne) qui sont des tests périodiques ou commandés, basés sur la sollicitation d'un ensemble de ressources matérielles et l'observation des sorties en résultant (auto- tests, les tests par sollicitation périodiques…)

  - les contrôles temporelles de bonne réception d'informations ou de réalisation de procédure (temps limites des horloges)

B- au niveau matériel

- **toutes les ressources matérielles** pouvant être amenées à être utilisées par une ou plusieurs sous fonctions du niveau fonctionnel (fonctions de diagnostic et de recouvrement incluses).

Pour que cette description soit valable, on définira une **ressource matérielle** comme un ensemble de composants électroniques et programmables, devant remplir une fonctionnalité de base unique (et donc non reconfigurable) durant la durée d'utilisation du système, et dont la réparation ne peut être menée séparément, mais pouvant être utilisée successivement par différentes sous fonctions. On admet qu'une ressource matérielle puisse être à la fois utilisée par plusieurs sous fonctions, qu'elles soient de base ou de diagnostic, mais qu'elle puisse aussi être testée par le système (non indépendance des parties testées et utilisées). Le système d'étude, ainsi défini, subit de la part de son environnement :

- **des événements perturbateurs**, appelés « fautes transitoires », causées par les contraintes environnementales (EMI, T,…) pouvant altérer un signal transmis ou stocké par les fonctions de base et de diagnostic.
- des **contraintes environnementales** influençant l'apparition ou la disparition de pannes sur les composants de ses ressources matérielles.

L'interaction entre l'environnement et le système sera donc restreinte :

- **aux facteurs de contraintes environnementales** (température, pression, EMI…) qui peuvent contribuer aux mécanismes de dégradation des composants matériels ou perturber les signaux transmis ou stockés
- **aux facteurs « humains** » qui peuvent influer sur la qualité et la mise en œuvre d'une action de maintenance portant sur une ressource matérielle (réussie ou non).

On pourra de ce fait définir notre fonction de sécurité comme l'utilisation ordonnée d'un ensemble fini de ressources matérielles suivant des règles imposées par l'architecture

-

fonctionnelle. Ces règles d'utilisation peuvent évoluer au cours du temps suivant l'état des informations échangées dans cette architecture. Elles peuvent donc évoluer suivant les mécanismes de détection et de reconfiguration fonctionnelle et/ou matérielle. Le système d'étude peut donc être assimilée à un système à événement discret qui permet de passer de l'observation d'un ensemble fini de paramètre (état de demande, présence d'erreurs matérielles détectables) à une période d'acquisition fixée e(k)=[kT, (k+1)T[(avec k entier) à une action (activation/désactivation) ou à une absence d'action de la part des moyens de réaction de la fonction dédiée sécurité, au temps $kT+T_R$, avec $T_R = D+T_i$, temps maximal de réaction admissible de la boucle.

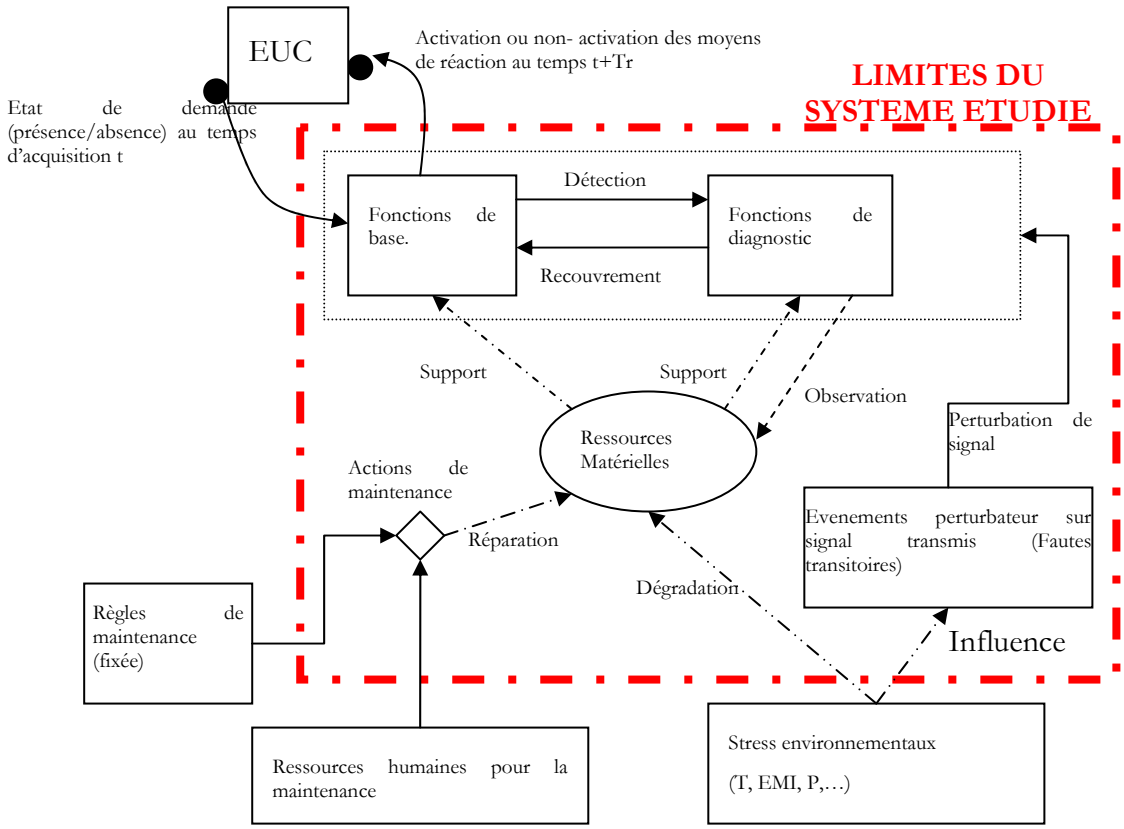


Figure 6 : Limite du système et interactions.

Ce type de description du système (**figure 6**) permet de prendre en compte implicitement :

- les contraintes d'architectures aussi bien matérielles que logicielles (en termes d'applicatif)
  - au niveau fonctionnel, en soulignant l'importance des échanges d'information entre procédure de diagnostic et fonction de base et en soulignant l'influence de l'ordre d'utilisation des ressources matérielles sur la loi de sortie (action) du système.
  - au niveau architecture opérationnelle, en soulignant la possibilité de partage de ressources matérielles et d'informations entre différentes sous fonctions (et donc en prenant en compte implicitement l'influence de partage de ressources matérielles défectueuses).
- les états d'erreurs matérielles existant à l'instant d'utilisation de la ressource matérielle, intégrité ou non intégrité de la ressource matérielle vis-à-vis de la sous fonction pour laquelle elle a été implantée.

- les événements, de causes externes (EMI, T…) perturbateurs sur la transmission et le stockage de signal et son influence sur les lois d'entrée-sortie de notre système (état de demande à t; état des moyens de réaction à t+Tr)
- les contraintes temporelles sur l'apparition et la disparition de « fautes matérielles ».
- les règles de maintenance et les imperfections possibles de ces opérations (soit dans leur mise en œuvre, soit dans l'activation de mode latent malgré leur caractère a priori correct **[Holn01]**).

En distinguant clairement l'aspect fonctionnel et matériel, nous séparons donc implicitement l'état d'erreurs des ressources matérielles et leur activation. Nous intégrons, en outre, les choix d'architectures logicielles, en définissant chaque module logiciel utilisé comme « une spécification abstraite consistant à décrire un ensemble de fonctions en définissant leurs opérations, leurs sources d'informations, leurs interfaces et leurs interactions» **[Haye94]**. Ces spécifications permettent de fixer les règles d'utilisation déterministe d'un ensemble de ressources matérielles.

Nous imposerons l'hypothèse supplémentaire, suivant laquelle, **le code source du logiciel reste inchangé au cours du temps**, et que **les procédures logicielles ont été vérifiées lors de leur implantation** (test exhaustif entrée – sortie sur support matériel). On pourra, en outre, supposer que certaines procédures logicielles sont contraintes en temps, par une procédure de contrôle temporel (horloge). Cette dernière hypothèse est généralement forcée par le temps de cycle général du processeur ou par l'utilisation de bloc d'instruction (OB) imposant un temps limite de déroulement pour un ensemble de procédure contenant la procédure considérée.

Nous supposerons, en outre, que

- **les lois de dégradation d'une ressource matérielle ne dépendent pas de l'état d'intégrité des autres ressources matérielles** appartenant au système, mais seulement des contraintes environnementales auxquelles cette ressource est soumise,
- **les défaillances systématiques** d'une ressource matérielle due à des erreurs de conception, de fabrication ou à un mauvais dimensionnement entraînant sa défaillance immédiate lors de sa sollicitation ne seront pas considérées.

On s'oriente donc vers une définition dynamique de notre système qui s'appuie à la fois sur les notions de flux d'information entre sous fonction et d'évolution temporelle de l'état d'intégrité des ressources matérielles. Dans le paragraphe suivant, nous allons clarifier cette notion d'information. Nous proposerons une description de l'architecture fonctionnelle par flux d'information, permettant à notre avis de mieux appréhender les contraintes de partage de ressources et d'information pour une architecture fixée.

6- Concepts de base et proposition (modèle informationnel)

Une défaillance est définie comme « la cessation pour une entité de son aptitude à assurer une fonction requise **[IEC50191]**», elle est donc toujours définie vis-à-vis d'une fonction dont les spécifications sont fixées dans un contexte de sollicitation précis. Avant toute chose, nous désirons distinguer la défaillance d'une ressource matérielle et la défaillance de notre système d'étude (fonction dédiée sécurité).

La défaillance d'une ressource matérielle est définie comme une cessation pour cet ensemble de composants de son aptitude à assurer la fonction pour laquelle elle a été implantée et ce **indépendamment de la validité des informations d'entrées utilisées** par cette ressource. Elle peut être révélée par la sollicitation de la ressource matérielle dans un mode particulier. Un mode de défaillance sera défini comme l'effet par lequel une défaillance est

observée **[Zwin95]**. Une défaillance matérielle n'est donc pas définie en regard des conséquences qu'elle peut faire porter sur le niveau de service de la fonction dédiée sécurité.

La défaillance de notre système d'étude (fonction de sécurité) est quant à elle définie comme la « cessation pour ce système de son aptitude à assurer une fonction requise ». Nous avons distingué deux modes de défaillance : la défaillance sur demande et le déclenchement intempestif de la fonction dédiée sécurité. Cette défaillance est donc définie au niveau **global**, puisqu'elle se base sur un écart entre les états d'activation des moyens de réaction de la fonction dédiée sécurité et les états attendus, au vu du cahier des charges fonctionnel, en supposant connu et fixé l'état de demande (présence/absence) au début de la période d'acquisition du système. Ce type de défaillance est donc beaucoup plus complexe puisqu'il peut être vu comme le résultat de l'occurrence d'une séquence de défaillances matérielles ou de fautes transitoires (altération de l'état d'un signal transmis ou stocké) suffisantes pour induire une cessation de la fonction requise.

Dire qu'une ressource matérielle est dans un état de mode de défaillance, signifie donc qu'elle présente une erreur, c'est-à-dire un état interne incorrect. En outre, cette erreur sera activée et donnera lieu à une défaillance de la ressource matérielle (suivant un référentiel entrée-sortie local) si la ressource est utilisée suivant une sollicitation révélant ce mode de défaillance. Cette distinction est **capitale** dans la plupart des architectures EP existantes. En effet, elle explique qu'une ressource matérielle utilisée par différentes sous fonctions et dont l'état d'intégrité est jugé constant entre ces utilisations successives puisse entraîner une défaillance de tout ou partie de ces différentes sous fonctions. Ce concept a d'ailleurs été repris dans le cas particulier de mode commun de défaillance du au partage d'une ressource matérielle défectueuse par des voies redondantes, voire même plus récemment dans le cas particulier de l'utilisation répétée d'une porte logique défectueuse pour différentes procédures en cascade menées par un même microprocesseur. **[Purw01]**

On voit donc qu'il existe une distinction importante entre existence d'une erreur matérielle et activation d'une défaillance matérielle. Cette distinction peut permettre de considérer l'architecture fonctionnelle de notre système (fonction de base et fonctions de test) sous la forme d'une séquence de sous fonctions, assurée par un ensemble fini de ressources matérielles, transformant une information observée (état de demande) en une action résultat (activation/désactivation ou absence de réaction de la fonction de sécurité).

Cette distinction permet de nous appuyer sur le concept d'erreur et la notion d'information pour expliciter les mécanismes d'initiation et de propagation d'erreur d'informations dans notre architecture fonctionnelle.

Aujourd'hui la notion d'erreur donnée dans les standards **[IEC61508]** semble trop large pour être réutilisée de manière effective sans autre forme de précision. Nous rappelons néanmoins sa définition générique:

« Ecart ou discordance entre une valeur ou une condition calculée, observée ou mesurée et la valeur ou la condition vraie, prescrite ou théoriquement correcte ».

Cette définition est à notre avis trop floue car elle est soumise à l'interprétation. Ainsi, la définition d'erreur peut dépendre de différents référentiels suivant les acteurs concernés. Un ingénieur informatique définira l'erreur suivant les différences comportementales du module logiciel construit et les spécifications fonctionnelles de celui-ci en supposant les registres d'entrées utilisés comme reflétant bien l'état réel de l'environnement (hypothèse de fonctionnement normatif amont) et dans l'hypothèse forte d'une absence de défauts des ressources matérielles permettant cette opération. Un spécialiste des sciences humaines définira l'erreur suivant le comportement possible de l'acteur observé en supposant les niveaux d'alarmes comme correct et en ne se souciant pas des effets de la procédure de maintenance dans le cas d'un fonctionnement non normatif de notre boucle de sécurité…

-

Le LAAS **[Bore96]** introduit la notion intermédiaire de faute en proposant l'enchaînement suivant : **faute => erreur => défaillance.** Cet enchaînement est supposé cyclique et dépendant du niveau d'observation des phénomènes. Il est supposé que la défaillance d'une entité peut être considérée comme la cause d'une faute et que cette faute peut dans certains cas « activer » une erreur. La faute est définie comme un événement à l'origine d'une défaillance. Un événement de défaillance matérielle est lié à la sollicitation d'une ressource présentant un certain nombre d'altérations au niveau composant, empêchant un comportement fonctionnel spécifié. Si on veut relier la notion d'erreur à la notion de défaillance du système d'étude, il est donc nécessaire d'établir un référentiel commun, qui permet de définir globalement la notion d'erreur et ainsi de définir les phénomènes d'initiation, de propagation et d'inhibition de celles-ci dans notre boucle de sécurité. Il semble, néanmoins, que le LAAS privilégie l'idée d'une décomposition d'une unité fonctionnelle en niveaux, dont chacun constitue à son tour une unité fonctionnelle. Cette décomposition hiérarchique peut poser des problèmes importants de couplage entre niveaux et implique des relations de dépendances fixes entre ces niveaux. Elle semble donc possible pour un flux direct d'information mais difficile à mettre en œuvre dans le cas de structure de tolérance aux fautes basées sur le diagnostic de ressource (par tests périodiques), puisque dans ce cas, certains événements initiateurs, peuvent se trouver directement liés aux transitions, présents dans d'autres niveaux hiérarchiques. La prise en compte de mécanisme de décision, et notamment l'introduction de procédures de diagnostic et de recouvrement (systèmes bouclés) ne permet plus de considérer notre fonction de sécurité comme une chaîne directe demande => action mais comme un ensemble de flux d'information qui peut aussi être influencé par d'autres signaux (contrôle, diagnostic) pouvant être entachés ou non d'erreurs.

Nous introduisons donc les concepts d'information et de propagation d'information afin de privilégier une approche de type graphes d'événements. La notion d'information est définie par **[Perp00]** comme un signal, c'est-à-dire un ensemble de données (..), auquel est adjoint une signification. La signification du signal sera de notre point de vue donnée par l'interprétation qui peut en être faite. On distinguera deux classes d'information :

- les F-informations, qui sont définies par des signaux permettant de juger, de manière correcte dans le cas d'un fonctionnement parfait, de la présence ou de l'absence de demande au début d'un cycle d'acquisition fixée.
- les D-informations (diagnostic), qui sont définies par des signaux permettant de juger, dans le cas d'un fonctionnement correct des autres ressources de l'architecture, de la présence d'une erreur détectable, c'est-à-dire soit d'un écart entre le comportement réel d'une ressource testée et son cahier des charges, soit d'un écart entre deux signaux, jugés équivalents. Par son utilisation ultérieure, ce type d'information doit permettre soit la mise en position sûre du système, par perturbation d'une ou plusieurs F- informations existantes, soit la reconfiguration dans un mode de fonctionnement dégradé.

Nous définirons une **erreur d'information** (information erronée) comme un écart entre une information obtenue à partir d'une observation (interface capteur) et devant refléter soit l'état de demande (présence/absence), soit la présence d'erreur matérielle détectable (information issue d'un test en ligne) et l'information qui devrait être obtenue, sous ces mêmes conditions d'observation (même période d'acquisition) et en l'absence de défaillance sur les fonctions ayant permis de la délivrer.

Cette définition de l'erreur d'information pour les informations de classes F et D est claire et univoque. Elle est liée à une référence fixée, à la fois dans le temps (période d'acquisition) et dans l'espace (point d'acquisition) de manière globale. L'intérêt principal est que cette référence ne dépend aucunement des choix de sous fonctions, excepté les contraintes de test donnant les modes de défaillances détectables. Le concept d'erreur d'information est donc conditionné à une séquence de défaillance des sous fonctions ayant permis sa délivrance.

Nous allons donc modéliser, dans une première approche, l'architecture fonctionnelle de notre fonction de sécurité avec un formalisme à base d'échanges de flux d'information entre sous fonctions. L'architecture fonctionnelle y sera assimilée à un processus de traitement d'information permettant à partir d'un ensemble d'observation de l'état de demande (présence/absence) et des tests en ligne, effectué sur cette période. Le résultat de ce processus est soit sur une action, en termes d'activation ou de désactivation des moyens de réactions, soit une absence d'action (conservation en l'état des actionneurs).On peut y distinguer deux types de flux :

- Les flux principaux, constitués de l'ensemble des flux de F-informations, dont l'orientation va des sous fonctions d'acquisition (interface capteurs) aux sous fonctions d'action (interface actionneurs).
- Les flux « contributeurs », constitués de l'ensemble des flux de D-information, partant de procédure de test matériels, de contrôle temporel de bonne réception ou de comparaison de F-informations. Ces flux aboutissent à des sous fonctions de décision, qui ont pour mission de forcer la mise en mode dégradée ou la reconfiguration fonctionnelle du système.

Dans l'exemple simple suivant (**figure 7**), on suppose qu'il est possible de modéliser par 6 sous fonctions (notées par les blocs f1, f2, f3, f4, f5, f6) une architecture fonctionnelle particulière pour une fonction de sécurité simple. Les sous fonctions f1 et f2 symbolisent, ici, deux sous fonctions d'acquisition de l'état de demande à un instant d'acquisition fixé et de transmission de ce signal, la fonction f3 symbolise une décision de type 1oo2, la fonction f4, une fonction de test d'une ressource en ligne, la fonction f5 le processus de décision suite à ce diagnostic (influence directe sur l'ordre de contrôle de l'actionneur) et la fonction f6 la sous fonction de contrôle et activation/désactivation du moyen de réaction de la fonction de sécurité).

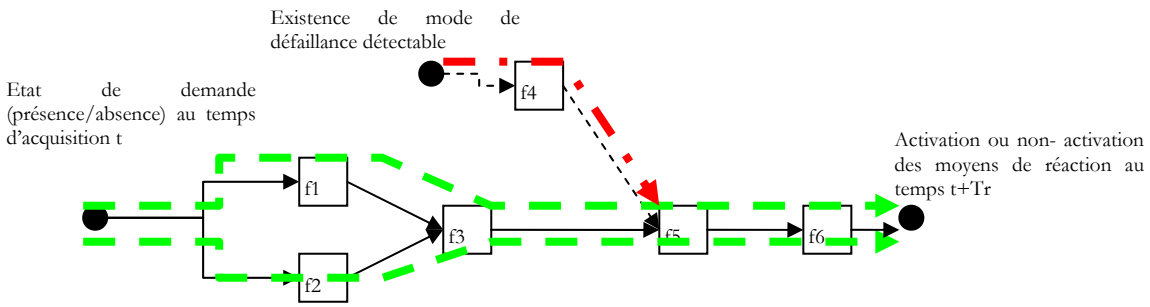


Figure 7: Flux d'information dans une architecture fonctionnelle

On note par une flèche le transfert d'une information d'une sous fonction à la sous fonction lui succédant. Pour simplifier la lecture, on a ici noté en flèches pleines le transfert de F-information et par des flèches pointillés le transfert de D- information. Les deux flux de F-information sont symbolisés par les flèches vertes (pointillés gras) tandis que le flux de D-information est symbolisé par la flèche rouge (flèche intermittente rouges).

Nous baserons notre modèle sur les mécanismes d'initiation, de propagation et d'inhibition d'erreur(s) le long d'un flux de F-information.

On parlera de perturbation d'information au niveau d'une sous fonction si l'information d'entrée et de sortie de cette sous fonction ne sont ni toutes les deux valides, ni toutes les deux erronées. On pourra dans ce cas considérer soit le passage d'une information d'entrée valide à une information de sortie erronée, soit le passage d'une information d'entrée erronée à une information de sortie valide. Une perturbation d'un flux de F-information au niveau d'une sous fonction peut être causée :

- soit par l'existence et l'activation d'un ou plusieurs modes de défaillances de ressources matérielles utilisées par la sous fonction pour assurer les opérations de traitement du signal
- soit par l'utilisation d'une autre information d'entrée entrant en ligne de compte dans un processus de décision (par exemple prise en compte d'une D-information ou d'une F-information)
- soit par l'occurrence d'une faute transitoire, due à une source externe (environnement) et ayant perturbé l'état du signal traité lors de son transport ou de son stockage.
- soit par certaines combinaisons de ces causes individuelles.

Ce phénomène de perturbation d'information considéré se traduit par :

- soit par l'initiation d'une erreur (information erronée) dans le flux d'information considéré,
- soit (exclusif) par l'inhibition d'une erreur (information valide) dans le flux d'information considéré

Ce deuxième point est rarement souligné dans la littérature. Mais il n'est pas absurde d'affirmer que dans certains cas la présence d'une défaillance matérielle en aval du lieu d'initiation d'une erreur d'information puisse empêcher à elle seule sa propagation. On peut citer pour exemple le collage permanent d'un relais en aval du lieu d'initiation d'erreur.

On peut définir le mécanisme de propagation d'une erreur le long d'un flux d'information, comme la réception et la délivrance d'information erronée par une séquence consécutive et ordonnée de sous fonctions. On définira enfin le phénomène de perte d'information par une sous fonction comme l'absence d'un signal de sortie en présence d'une information (de classe F ou D) consommée par celle-ci.

On distinguera donc clairement la notion de **perturbation d'information** de la notion de **perte d'information**. Une perte d'information se traduira par une privation d'information d'entrée de toutes les sous fonctions qui succèdent (liaison fonctionnelle) à celle responsable de la perte, alors qu'il existera souvent la possibilité par une succession de phénomène de perturbation locale de flux de F-informations d'inhiber une erreur d'information précédemment initiée. La perte d'information est donc considérée comme un phénomène irréversible sur une période T et est héritée tout le long d'une séquence de sous fonctions. La perturbation d'une F-information influence l'état d'erreur (vraie ou faux) le long du flux d'information mais ne conduit pas à son interruption. Cette distinction n'est pas, en général, présente dans les approches classiques, basées sur une vision « tout ou rien » du signal transmis.

Cette idée d'un mécanisme d'initiation/propagation d'erreurs a déjà été proposée dans la littérature, notamment par **[Allo01].** Cependant, il est à noter qu'aucune méthode de construction systématique n'a permis à ce jour d'évaluer une architecture complexe suivant ces principes en se basant par exemple sur une méthode de graphes d'événements en la mettant en regard avec l'idée consistant à considérer une défaillance fonctionnelle comme une combinaison d'événements. En outre, la notion d'inhibition accidentelle d'erreur, due à une défaillance matérielle non prévue ou à un comportement humain inopportun, n'a jamais été soulignée dans la littérature et mérite d'être prise en compte.

L'intérêt porté dans la notion d'erreur d'information réside dans le constat simple qu'une F-information de sortie issue de la dernière sous fonction de notre architecture fonctionnelle (fonction f6 pour l'exemple) est équivalente à un mode de défaillance de la fonction dédiée sécurité sur la période d'acquisition considérée, et ce quelque soient les mécanismes d'initiation et de propagation d'erreur qui la sous-tendent. Elle correspond, en effet, à un état d'activation des

moyens de réactions de la fonction de sécurité au temps t+Tr en désaccord avec l'état de demande existant à l'instant t.

Ce type de représentation doit permettre la prise en compte implicite de partage de ressources entre sous fonctions (informationnelles ou matérielles) et de leur conséquence en termes d'activations répétées de modes de défaillances matériels existants. Il est donc tout à fait possible de symboliser la persistance d'une erreur matérielle pour plusieurs sous fonctions, notamment dans le cas d'un partage de ressources pour des sous fonctions distinctes.

Par exemple, dans le cas précédent, on peut imaginer le cas de figure où, sous un état de demande fixé au temps t, un mode de défaillance matérielle unique existe pour une ressource, notée R, et que ce mode de défaillance ne soit pas détectable par la procédure de test. Si ce mode de défaillance est activé, du fait de l'utilisation de cette ressource dans les sous fonctions f1 et f2, on aura une initiation de deux erreurs aux points, notés X et Y sur la **figure 7**. L'erreur se propagera par la suite au niveau de la sortie de la sous fonction f3 (puisque ces deux informations d'entrées, bien qu'erronées, seront jugées équivalentes), puis au travers des fonctions f5 (aucune perturbation du flux de F-information en absence de diagnostic) et f6, pour se traduire finalement par une défaillance fonctionnelle de notre architecture. Ce cas souvent souligné dans la littérature sous la terminologie de mode de défaillance commun dans des voies d'information redondantes sera donc implicitement pris en compte par une approche fonctionnelle de type flux d'information. Il est aussi tout à fait possible d'envisager le partage de ressource matérielle dans des sous fonctions appartenant au même flux d'information.

7- Conclusion du chapitre 1.

Cette partie a permis de rappeler l'intérêt d'une quantification des performances en termes de fiabilité des fonctions dédiées sécurité. Elle a montré les relations existant entre ces performances et les notions de risque résiduel et de coût induit. On a pu ainsi, justifier le recours à deux indicateurs de performances fonctionnelles : le PFS et le PFD, et la mise en place d'objectifs sur leur moyenne temporelle (calculée sur la durée de vie du système). La nécessité d'évaluation de ces indicateurs apparaît donc comme primordiale pour l'utilisateur final qui doit juger de la pertinence des choix architecturaux et organisationnels de sa fonction de sécurité.

La complexité croissante de ces architectures, possédant souvent des propriétés de tolérance aux fautes et des procédures de réparation et de recouvrement, nous a amené à poser le problème de l'exactitude des modèles d'évaluation quantitatifs existants. Du fait de ces propriétés, nous avons jugé nécessaire de clarifier les limites d'études de notre « système » et nous avons montré que certaines problématiques, telles que les défaillances de causes communes ou les modes latents, restaient encore mal intégrés à ces méthodes et ce, malgré l'impact important qu'elles pouvaient avoir sur les performances du système évalué.

Pour résoudre ce problème, nous avons donc opté pour une représentation dynamique de notre système, basée sur les notions de F- et de D-informations. Ce choix a été porté par les limites des approches hiérarchiques, généralement utilisés dans la construction des modèles usuels, la nécessité d'une définition plus claire des couplages existant dans notre système et l'idée d'une meilleure définition de la notion de simultanéité et de temps. Nous avons vu qu'une décomposition sous fonctionnelle de l'architecture du système étudiée (fonction de sécurité) par flux d'information pouvait permettre la prise en compte de contraintes de précédence/postériorité entre sous fonctions, mais aussi de partage d'information (et donc d'erreur sur cette information) et de ressources matérielles. Cette idée ne peut être utilisée que s'il existe un référentiel temporel permettant de juger de l'existence et de la persistance de modes de défaillances matérielles entre leurs utilisations successives (on rappelle que ce procédé est cyclique de période T. Ce constat rejoint le problème de la définition claire du concept de « défaillances matérielles simultanées », introduit dans les années 1970 au début des études de causes

communes de défaillances et de « persistance » de ces fautes matérielles au cours du temps. En outre, il pose le problème des relations entre cet aspect temporel et l'évolution de l'environnement qui tend à augmenter (facteur de stress), ou à réduire (facteur humain pour les actions de réparation), la probabilité d'existence de ces fautes.

En nous attachant à la clarification de ces deux points, nous allons donc proposer dans la partie suivante la structure d'un modèle d'évaluation générale de performances (PFD_{avg}, PFS_{avg}) pouvant être appliqué à l'évaluation de toute fonction dédiée sécurité EP, tolérante aux fautes, de classe SDSC, pour laquelle les hypothèses soulignées en 2-2- et en 5- sont supposées être respectées.

Chapitre 2

Propositions pour un modèle général d'évaluation de performances

Chapitre 2 - Propositions pour un modèle général d'évaluation de performances

Ce chapitre a pour objectif de proposer la structure d'un modèle général d'évaluation des indicateurs de performances (PFD_{avg}, PFS_{avg}) pour une fonction EP dédiée sécurité. Ce modèle doit permettre une prise en compte accrue des phénomènes de défaillances simultanées en accordant une place particulière à la représentation temporelle. Nous aborderons donc de manière approfondie la notion de défaillances matérielles simultanées.

Nous rappellerons dans un premier temps, la typologie des erreurs prises en compte dans notre système d'étude et les sources de couplage portant à la fois sur les conditions d'existence de ces erreurs et sur leurs conséquences en termes de modes de défaillances du système. Nous montrerons ensuite que ces couplages peuvent être pris en compte par l'introduction d'un temps de référence permettant de modéliser notre système d'étude en différents niveaux et d'évaluer quantitativement les indicateurs de performances.

1- Typologie fautes/erreurs.

Les hypothèses préliminaires proposées dans le paragraphe 2-2- du chapitre 1 permettent de supposer l'absence d'erreurs de conception et d'erreurs sur les ressources matérielles implantées au moment de la mise en service d'une fonction dédiée à la sécurité (origine des temps t=0).

Nous nous intéresserons dans notre étude par les fautes pouvant se produire dans notre architecture au cours de sa durée d'utilisation. Puis nous étudierons comment elles peuvent être responsables de mécanismes d'initiation, de propagation ou d'inhibition d'information(s) erronée(s) dans le système étudié (fonctions de base + fonctions de diagnostic). Le système peut être considéré comme un processus périodique de décision, éventuellement reconfigurable, basé sur l'utilisation séquencée de ressources matérielles. A partir de l'état de demande observé à un cycle d'acquisition fixé par les différentes voies d'acquisition (présence ou absence de demande) et du résultat d'opération de diagnostic, le système va **choisir ou non d'activer** les moyens de réactions (actionneurs) prévus pour la mise en sécurité de l'installation après une durée T_r.

Une défaillance est définie par la **IEC61508-4** comme « la cessation de l'aptitude d'une unité fonctionnelle à accomplir une fonction requise ». Cette définition est locale et dépendante des limites de la fonction considérée. En général, on pourra distinguer :

- les défaillances dites **aléatoires** du matériel, causées par les mécanismes d'usure d'un ensemble de composants matériels (module) ;
- les défaillances **systématiques** : « défaillance reliée de façon déterministe à une certaine cause, ne pouvant être éliminée que par une modification de la conception ou du processus de fabrication, des procédures d'exploitation, de la documentation ou d'autres facteurs appropriés ».

On appellera, par la suite, **faute**, toute cause (événement, circonstance) adjugée ou hypothétique de la présence d'une erreur. **[Lapr96].** On va principalement s'intéresser à deux types de phénomènes dans notre système pouvant influer sur les facteurs d'initiation et d'inhibition d'information(s) erronée(s) dans l'architecture fonctionnelle (cf. **définition au chapitre 1**): les **erreurs matérielles** et les **fautes « environnementales »**.

Les **erreurs matérielles** d'un ensemble de composants (électrique, électronique, mécanique, …) dans une ressource matérielle se manifestent par **un mode de défaillance** de celle-ci, traduisant son inaptitude à remplir la fonction pour laquelle elle a été implantée sous un mode fixé de sollicitation. On rappelle que nous avons défini dans le chapitre 1 une ressource matérielle

comme « un **ensemble de composants** électroniques et programmables devant remplir **une fonction de base unique** (et donc non reconfigurable) durant l'utilisation du système, et dont la réparation doit être menée globalement. ».

A chaque ressource matérielle partagée, on peut donc associer plusieurs modes de sollicitation. A chacun de ces modes, la manière dont l'erreur se manifeste est différente. On nomme mode de défaillance chacune d'elles. L'**apparition d'une erreur matérielle d'une ressource** peut toujours être interprétée comme le résultat de mécanismes successifs d'apparition (dégradation) et de disparition (réparation) de pannes sur ses composants. L'existence d'une erreur matérielle est donc liée à l'aspect défectueux d'un ensemble de ses composants (notion de coupe).

Le passage d'un état d'erreur matérielle d'une ressource matérielle à un autre état d'erreur se manifeste par un mode distinct de défaillance. Il peut donc être lié à l'apparition ou à la disparition d'un ensemble fini de pannes des composants constitutifs de cette ressource. On peut néanmoins avoir une **accumulation d'erreurs sur les composants avec conservation de l'état**. On peut citer en exemple les dispositifs de commande de sécurité de catégorie 4, suivant la norme **EN954-1**, qui doivent permettre une continuité de leur fonction dans le cas d'une telle accumulation. Cette accumulation peut influencer la probabilité conditionnelle de passage de cet état d'erreur à un autre état d'erreur matérielle.

Une erreur d'une ressource matérielle (appelée par la suite **erreur matérielle**) a donc une **durée d'existence** liée d'une part aux mécanismes de dégradation auxquels sont soumis ses composants et d'autre part aux propriétés de détection et de réparation portant sur la ressource matérielle considérée. La présence d'une erreur dans une ressource matérielle peut être **temporaire** pour plusieurs raisons : la réparation (disparition d'une ou plusieurs causes), la tolérance aux fautes…

Une « **faute environnementale** » correspond au passage à un niveau de stress environnemental (interférences électromagnétiques, température…) causant l'altération d'un signal. Un exemple de ces fautes est donné dans la littérature par le phénomène dit de « bit flip » qui se traduit par l'inversion d'un signal binaire du fait des rayonnements (essentiellement électromagnétiques) auxquels est soumis son support de transmission ou de stockage. L'existence de cet écart entre le signal transmis et le signal devant être transmis en l'absence de ces fautes, peut être interprétée comme une **erreur, appelée** par abus erreur environnementale. Elle est conditionnée par l'occurrence d'une faute environnementale et la transmission du signal pouvant être affectée par celle-ci (**condition d'activation**). Elle donnera lieu à une défaillance du processus de transmission de l'information se traduisant par une altération du signal transmis. Cette erreur est de type « **commande »** puisque son existence disparaît avec sa cause. L'existence d'une erreur environnementale est donc liée à l'existence d'un niveau de stress environnemental à même de perturber un signal. On peut souligner le caractère transitoire de ces erreurs, qui est lié à l'existence du signal pouvant être affecté. Certains auteurs dont **[Purw01]** parlent d'ailleurs d'erreur transitoire (« transient fault ») pour mieux les opposer aux erreurs persistantes.

1-1- Concept et dénomination des modes de défaillance de ressources matérielles.

On dira qu'une ressource matérielle présente un mode de défaillance quand son état la rend inapte à remplir la totalité des fonctions **pour lesquelles elle a été implantée**. Les défaillances matérielles sont donc définies localement en fonction des lois entrées/sortie de la ressource matérielle considérée et **jamais** en fonction des règles d'interaction de cette ressource avec le reste du système. La défaillance d'une fonction dédiée sécurité dépend pour sa part :

- du contexte d'utilisation, correspondant à l'état de demande (présence/absence) observé ;

- de l'état d'intégrité des ressources utilisées dans l'architecture du système ;
- des règles d'utilisation de ces ressources (requête d'utilisation et ordre d'utilisation).

La révélation (aussi appelée activation) d'une erreur matérielle ne peut avoir lieu que durant son existence. Elle correspond à l'utilisation de cette ressource matérielle **selon le mode de sollicitation** se traduisant par cette défaillance (inaptitude à remplir la fonction qu'elle est supposée supporter).

On peut donc définir le concept de mode de défaillance d'une ressource matérielle, suivant l'état des informations d'entrée à traiter et de sortie à émettre par cette ressource. On propose donc, au plus, trois modes de défaillances, notés S, D et I. On notera que ces initiales n'ont rien à voir avec le mode de défaillance du système d'étude. L'état de la ressource matérielle permettant son fonctionnement normal quel que soit son mode de sollicitation sera noté 0 (absence d'erreur matérielle).

Dans le cas d'une F-information entrante (resp. D-information), on distinguera deux cas, **basés sous l'hypothèse d'absence de défaillance dans le système d'étude :**

- soit l'information d'entrée reflète un état de présence de demande (resp. de présence d'erreur détectable) **[mode de sollicitation 1] ;**
- soit l'information d'entrée reflète un état d'absence de demande (resp. d'absence d'erreur détectable) **[mode de sollicitation 2].**

On observera alors le signal de sortie. **Sous l'hypothèse d'une absence de défaillance dans le système d'étude**, on pourra distinguer les trois cas suivants :

- l'information de sortie reflète un état de présence de demande (resp. de présence d'erreur détectable) **[réponse A] ;**
- l'information de sortie reflète un état d'absence de demande (resp. d'absence d'erreur détectable) **[réponse B] ;**
- aucun signal n'est émis par la ressource **[réponse C].**

Le mode de fonctionnement correct de la ressource est défini par deux lois inductives

« [mode de sollicitation 1] donne [réponse A] » et « [mode de sollicitation 2] donne [réponse B] »

On définira donc les trois modes de défaillances S, D, I **d'une ressource matérielle** en accord avec les règles données **figure 8**. Ces modes sont définis suivant l'écart entre la réponse de sortie et la valeur attendue pour son fonctionnement normal :

[mode S] ⇔ *« [mode de sollicitation 2] donne [réponse A] »*

[mode D] ⇔ *« [mode de sollicitation 1] donne [réponse B] »*

[mode I] ⇔ *« tout mode de sollicitation donne [réponse C] »*

Figure 8 Définition des modes de défaillance d'une ressource matérielle

Pour éviter toute confusion entre les modes, on définira toujours les modes de défaillance d'une ressource matérielle par rapport à sa première utilisation dans l'architecture à t=0, c'est-à-dire en début de vie du système (mode de fonctionnement « normatif »). Si la ressource matérielle n'est

utilisée que dans un mode de fonctionnement dégradé de l'architecture, on considérera sa première utilisation dans ce mode pour définir les modes de défaillances S, D, I.

On appellera erreur de type S (resp. D, I) de la ressource matérielle R, notée d(S,R), l'erreur matérielle qui se manifeste par le mode de défaillance S (resp D, I). On appellera état de type 0 de la ressource matérielle R, noté d(0,R), l'ensemble des états de la ressource matérielle, ne se traduisant par aucun mode de défaillance. Celle-ci est alors intègre (apte à remplir ses fonctionnalités telles que spécifiées).

Le choix des initiales S, D et I effectué précédemment est loin d'être neutre. Il provient de la dénomination souvent employée par les organismes de certification (TuV) pour désigner les modes de défaillance induits par une ressource matérielle utilisée dans une architecture de type mono-flux en l'absence de mécanisme de reconfiguration et sous l'hypothèse d'une utilisation unique de ladite ressource. Cette désignation est conservée car la pratique de test de ressource matérielle modulaire envisagée par ces organismes se fait souvent en absence de connaissance des propriétés de tolérances aux fautes et de reconfiguration de l'architecture dans laquelle elle est implantée.

Dans le cas simple, d'une fonction dédiée sécurité pouvant être interprétée comme une séquence non reconfigurable de sous- fonctions et pour laquelle chacune de ces sous-fonctions est assurée par une ressource matérielle distincte, on peut affirmer qu'en l'absence d'autre erreur environnementale ou d'autre erreur matérielle :

* un mode de défaillance S, comme « safe », sur une <u>unique</u> ressource matérielle, aura pour conséquence un déclenchement intempestif de la fonction de sécurité ;

* un mode de défaillance D, comme « dangereux », sur une <u>unique</u> ressource matérielle, aura pour conséquence une défaillance sur demande de la fonction de sécurité ;

* un mode de défaillance I, comme « interruption », sur une <u>unique</u> ressource matérielle, aura pour conséquence une conservation de l'état des actionneurs de la fonction de sécurité.

Il est à noter que ce lien de causalité n'est vrai que pour des architectures câblées très simples (1oo1 non diagnostiquées et non reconfigurables) et en l'absence de toute autre erreur pouvant affecter le bon fonctionnement du système. Dans le cas général, le lien causal entre ces modes de défaillance locaux et le mode de défaillance du système étudié n'est pas vérifié. Il est ainsi tout à fait possible pour certaines architectures, dites tolérantes aux fautes, qu'un mode de défaillance d'une ressource ne se traduise pas nécessairement par un mode de défaillance de la fonction dédiée sécurité. Il est aussi tout à fait possible qu'un mode de défaillance d'une ressource inhibe un phénomène de propagation d'erreur le long d'un flux de F-informations. Le cas d'école le plus simple est la situation où, en présence de demande, le collage en position « sûre » d'un relais dans une architecture câblée force le déclenchement d'un actionneur quels que soient les modes de défaillance matérielles existant en amont de ce point.

Une défaillance matérielle dans le mode S ou D se traduit soit par le passage d'une information erronée à une information valide soit par celui d'une information valide à une information erronée. On appellera ce passage « perturbation d'information ». Une défaillance matérielle dans le mode I se traduit quant à elle par la perte d'une information. Ces possibilités d'initiation ou d'inhibition d'erreurs sur un flux d'information restent en général non étudiées par les approches classiques. Elles sont, en effet, la caractéristique principale des systèmes non cohérents (« une défaillance survenant sur un composant du système défaillant annule les effets d'une défaillance antérieure et permet ainsi au système de fonctionner à nouveau »). Nous inclurons néanmoins ces possibilités dans notre modélisation car elles s'avèrent indispensables à l'étude de certaines fonctions dédiées à la sécurité pour lesquelles le partage de ressources entre

sous fonctions distinctes et agissant sur un même flux d'information implique des propriétés de non cohérence **[INERIS03]**.

1-2- Similitudes et différence avec l'approche faute > erreur > défaillance proposée par le LAAS.

La plupart des études fiabilistes de fonctions dédiées sécurité s'appuie sur le cycle **faute> erreur>défaillance** introduit par le LAAS **[Bore96]**. Comme nous l'avons vu, cette description reste valable pour les deux types d'erreurs précédemment définis. On associe, pour notre part, le terme **faute**, soit à des pannes de composants d'une ressource matérielle (***fautes matérielles***), soit à des ***fautes environnementales***. **L'existence d'une erreur matérielle** est la conséquence de l'occurrence d'une séquence de pannes matérielles, tandis que celle d'une **erreur environnementale** est directement induite par l'occurrence d'une faute environnementale (correspondant au dépassement d'un seuil de contraintes environnementales). L'activation des **erreurs matérielles** (apparition de défaillances locales) est conditionnée par la sollicitation d'une ressource matérielle, suivant un mode révélant la défaillance (défaillance matérielle ou altération d'un signal). Le cycle faute>erreur>défaillance reste donc respecté.

Cependant, nous allons nous démarquer d'une approche par **décomposition hiérarchique de l'architecture fonctionnelle, basée sur des comportements de type « tout ou rien ».** Cette décomposition fait correspondre une défaillance dans un niveau de représentation bas à une faute dans un niveau de représentation plus élevé (en termes hiérarchique). Ce choix est justifié par le fait que les **architectures de contrôle-commande tolérantes** aux fautes sont souvent des **systèmes non cohérents** (au sens fonction de structure fiabiliste), ainsi il est nécessaire pour évaluer correctement leur fiabilité de prendre en compte les mécanismes d'initiation et d'inhibition d'informations erronées le long de flux d'information.

Ces architectures, du fait de leurs bouclages internes (au sens de boucle en automatique) se prêtent, en effet, mal à une décomposition fonctionnelle hiérarchique. L'interaction de différents flux d'information (F- et D-information) et le partage possible de ressources, qu'elles soient matérielles pour supporter des sous fonctions distinctes, ou informationnelles entre sous fonctions, rendent difficile cette décomposition.

On va donc introduire le concept d'information (définition au paragraphe p.37 chapitre 1) pour prendre en compte l'initiation, la propagation et l'inhibition d'information erronée le long de flux d'information, voire même la perte d'information le long de ces flux. Nous pouvons, en effet, remarquer que l'activation d'erreurs dépend fortement des choix d'architectures avancés (règles et séquences d'utilisation de ressources matérielles) et de l'état des signaux transmis. Les mécanismes d'inhibition d'information erronée le long d'un flux d'information peuvent être dus soit à des processus de reconfiguration et de recouvrement, soit à l'existence d'autres erreurs matérielles ou environnementales, ayant tendance à perturber une information erronée.

C'est pourquoi nous préconisons une approche, basée sur une distinction forte entre **existence d'erreurs**, activation d'erreurs et conséquences de ces **séquences d'activation** (le terme séquence est ici primordial) en termes de lois entrées (état de demande) / sortie (état d'activation des moyens de réaction) du système étudié. Le résultat de ces différentes perturbations locales d'informations dans l'architecture (activation d'erreurs) se traduit soit par la présence soit par l'absence d'un mode de défaillance fonctionnelle de notre système d'étude. Cette défaillance du système d'étude pouvant être interprétée comme le résultat direct d'une propagation d'information erronée jusqu'aux interfaces actionneurs (transformation d'une information de contrôle en action).

Cette approche est compatible avec les propositions du LAAS, en termes de terminologie, mais pas en termes de modèle, puisque la décomposition hiérarchique fonctionnelle y est

remplacée par un **modèle basé sur la notion de « flux d'information »** pour lequel le signal peut être éventuellement perdu mais aussi altéré et réutilisé tel quel, lors d'opérations de traitements successifs. Cette idée de propagation doit s'appuyer sur une définition claire d'un temps de référence permettant de juger de l'existence ou de la non-existence d'erreur, affectant les missions des ressources utilisées (en termes de transformation de signal ou de transport).

Pour choisir ce temps de référence et justifier son usage, il est nécessaire de comprendre quels phénomènes de dépendance peuvent altérer la fiabilité du système. Nous nous proposons, dans le paragraphe suivant, de présenter les sources de couplages environnementales et architecturales, que nous modéliserons par la suite.

1-3- Sources de couplages.

On peut distinguer deux classes de couplages dans notre système d'étude qui doivent être prises en compte pour permettre une évaluation correcte des indicateurs de performances du système: les **couplages environnementaux** et les **couplages architecturaux**.

Les premiers se justifient par la sensibilité des ressources matérielles implantées et des signaux échangés vis-à-vis de l'environnement. Celui-ci peut influencer **l'apparition** et la **disparition** d'erreurs distinctes (matérielles et environnementales).

Les seconds sont liés aux **contraintes architecturales** et aux dépendances temporelles et causales qui peuvent en découler. Ces couplages doivent être pris en compte dans la description des phénomènes d'initiation et de propagation d'information(s) erronée(s).

Nous nous restreindrons, dans toute notre étude, à la prise en compte de ces deux problèmes. Les phénomènes d'accélération de la dégradation d'une ressource matérielle, induits par la panne d'autres ressources matérielles augmentant les contraintes environnementales auxquelles elle est soumise, et les défaillances dues à une mauvaise maîtrise des procédés de fabrication et de montage des ressources matérielles ne seront pas prise en compte dans cette étude.

1-3-1- Couplages environnementaux

Les couplages qui portent sur l'apparition ou la disparition de plusieurs fautes (classes précédemment définies), seront qualifiés d'**environnementaux, du fait de l'existence** d'un lien causal entre ces apparitions/disparitions et les contraintes environnementales. Les couplages sur l'apparition de plusieurs fautes matérielles dépendent essentiellement :

- de contraintes environnementales participant à la dégradation progressive (usure) ou immédiate (choc) des composants constitutifs des ressources matérielles (température, pression,…), pour une classification de ces facteurs on pourra se référer à **[Maur00]**.
- de contraintes d'architectures matérielles, liées à la disposition des composants dans la ressource et à l'exposition de la ressource aux sources de contraintes environnementales extérieures (concept de zone d'exposition **[Popo01]**)

Les **phénomènes de fautes environnementales** dépendent essentiellement de contraintes environnementales de types rayonnement **[Chen00]**, dont la forte variation peut influencer l'état d'un signal sur un support matériel devant assurer son transport ou son stockage. Ces probabilités de fautes sont donc liées à l'état pris, sur un cycle de réaction, de certains stress environnementaux et à l'exposition du support matériel à ces stress (lien causal). Elles dépendent de la variation de ces stress au cours du temps. On supposera que ces probabilités peuvent varier au cours du temps suivant le niveau de stress environnemental auquel est soumis le système à chaque cycle. Une faute environnementale sera donc supposée maintenue sur chaque cycle T du système étudié.

Les **phénomènes de disparition d'erreurs matérielles** sont liés aux opérations de maintenance matérielle. Il est possible qu'une action de maintenance puisse forcer le passage d'un état d'erreur matérielle à un autre état d'erreur différent de l'état 0 (cas de maintenance non parfaite). Il est possible que des actions de maintenance préventive sur des ressources matérielles distinctes, impliquent une dépendance temporelle entre ces phénomènes. Cette dépendance doit être prise en compte au moyen d'horloges locales.

1-3-2- Couplages architecturaux

Les couplages dits **« architecturaux »** sont liés aux contraintes de l'architecture fonctionnelle. Les choix d'architecture fonctionnelle induisent des règles d'utilisation successive de ressources matérielles et d'échange de signaux entre ces ressources. Elles permettent d'expliquer les mécanismes d'initiation, de propagation et de partage d'informations erronées dans l'architecture opérationnelle. Les modes de défaillances du système d'étude peuvent donc être vus comme le résultat d'activation d'erreurs successives le long de flux d'information (erreurs matérielles et erreurs environnementales).

L'existence d'un mode de défaillance du système d'étude est donc fortement dépendant des règles d'utilisation des ressources matérielles, c'est-à-dire de l'ordre de ces utilisations et du partage éventuelle de ressources matérielles par des sous fonctions distinctes de notre système d'étude. Ces couplages peuvent être de trois types : le couplage par **partage de ressource matérielle**, le couplage par **partage d'information** et le couplage par **transmission d'information**.

Le couplage par **partage de ressource matérielle** est dû à l'utilisation par différentes sous fonctions (répétées ou distinctes) du système d'étude d'une ressource matérielle présentant une erreur persistant sur une période temporelle séparant ses utilisations successives (S,D ou I, suivant la terminologie précédente). On pourra avoir une défaillance répétée de la ressource matérielle, suivant le même mode de sollicitation. L'utilisation d'un bus de communication (couche physique) pour transmettre deux informations distinctes au contrôleur en est un exemple simple. Nous constatons qu'une rupture de bus entraînera une défaillance de ces deux sous fonctions si la réparation du bus ne peut être effectuée entre les deux opérations de transfert successives.

Le couplage par **partage d'information** est dû à l'utilisation par différentes sous fonctions d'une même information d'entrée (définie de manière unique dans le temps et l'espace). L'information étant unique, une information erronée en entrée d'une des sous fonctions implique nécessairement une information erronée sur l'entrée des autres sous fonctions (et réciproquement). Ce phénomène de dépendance par des sources d'informations partagées a bien été souligné par **[Char02]** dans son étude des modes communs de défaillances, qui expose le problème du partage d'une information d'entrée unique par deux voies de traitement redondantes.

Le couplage par **transmission d'information** est dû au fait que l'architecture fonctionnelle du système est organisée autour de flux d'information entre sous fonctions. Cette structure ordonnée entre sous fonctions contribue à la propagation d'une information erronée le long d'un flux d'information ou entraîner la perte de certaines informations (non récupérable). Le lieu d'apparition d'une information erronée est donc non neutre vis-à-vis de ses possibilités de propagation ou d'inhibition ultérieure.

1-3-3- Bilan

La distinction entre ces deux types de couplage (environnementaux/architecturaux) passe par une définition plus précise du facteur temporel. En effet, en détaillant les modes de couplages

architecturaux et environnementaux, nous pouvons remarquer que les termes d'existence (et notamment de conservation) d'erreurs et d'occurrence de fautes matérielles ont été distingués La notion d'activation de ces erreurs et la prise en compte de leurs effets dépendent de règles d'ordonnancement portant sur l'utilisation des ressources, au sens large c'est à dire des ressources informationnelles et matérielles, par une architecture fonctionnelle.

Un prise en compte plus rigoureuse de l'aspect temporel est d'autant plus nécessaire qu'elle permettrait de préciser le concept de « défaillances quasi-simultanées » qui apparaît dans la définition des défaillances de causes communes proposée dans certains standards (**IEC61508**, **ECSS-Q-80-03)**.. Ce concept, bien qu'utilisé dans les normes, n'y est jamais défini de manière précise.

Nous proposons donc d'introduire et de justifier la définition d'une fenêtre temporelle d'observation, permettant de définir précisément le concept d'événements et d'erreurs simultanées. Ceci nous permet de distinguer :

- les dépendances entre les phénomènes d'apparitions et disparitions d'erreurs,
- les possibilités en termes d'activations de ces erreurs, liées aux contraintes architecturales et à leur co-existence,
- et les conséquences qu'elles induisent sur le bon fonctionnement du système d'étude.

Dans la littérature, le thème de contraintes et de dépendances temporelles est régulièrement occulté dans les modèles d'évaluation de performances fiabilistes de fonction de sécurité. Certains modèles **[Duga00]**, **[Boui01]**, **[Triv96]** ont cherché à atteindre une prise en compte plus précise de mécanismes de reconfigurations fonctionnelles au cours du temps. Cependant, ils ont rarement mis en avant le problème de mode de sollicitation des ressources matérielles. Les modèles se sont souvent contentés de mécanismes de reconfiguration simples laissant de côté la prise en compte de reconfiguration fonctionnelle dépendant de l'état de validité de certaines informations manipulées.

Ce thème semble cependant être la pierre angulaire de l'étude de tout système reconfigurable et réparable. Nous avons donc choisi puisqu'il semble être au centre d'un certain nombre de phénomènes permettant l'initiation et la propagation d'erreurs d'information(s) dans notre architecture fonctionnelle, de lui accorder un traitement particulièrement attentif. Aussi, nous tenterons de définir clairement la notion de « défaillances simultanées » pour une architecture EP.

Nous verrons que cette définition permet de proposer un modèle d'évaluation des performances de notre système d'étude, en intégrant les phénomènes de couplage précédemment soulignés et en permettant une prise en compte, par voie de conséquence, plus réaliste de l'influence de modes de « défaillances simultanées » dans le processus d'évaluation de nos indicateurs de performance.

2- Aspect temporel et définition du concept de défaillances «simultanées»

La notion de temps a été peu développée dans la littérature s'intéressant aux performances fiabilistes des fonctions de sécurité. Le temps est un paramètre borné d'étude du système. On peut rappeler que l'évaluation des indicateurs de performances, basée sur un calcul de type moyenne temporelle, se fait sur un horizon temporel fini, noté $[0 ;T_{life}]$ avec 0, le temps origine de mise en service du système et T_{life} sa durée d'utilisation. Nous avons souligné dans la partie 1, que nos indicateurs de performances instantanées (PFD(t),PFS(t)) ne pouvaient être définis que comme des fonctions discrètes du temps suivant la période T d'acquisition du système étudié.

On peut donc définir une première échelle de temps, dit **globale**, de nature échantillonnée et évoluant de 0 à T_{life} **suivant le pas T**. Par rapport à cette échelle globale, il est possible d'introduire un certain nombre d'horloges locales de périodes fixées pour permettre de modéliser les instants d'occurrence des tâches de maintenance préventive planifiées des ressources. On conservera, dans la suite de notre travail, la notation générique TI et TR proposée par l'ICE61508 comme définition de la période d'inspection préventive et du temps d'inspection et de réparation hors ligne d'une ressource matérielle maintenue préventivement. On pourra éventuellement considérer le cas d'une fonction TI, évoluant après chaque nouvelle inspection préventive, permettant de prendre en compte toute politique de maintenance préventive planifiée non périodique. Il est ainsi possible sur une durée d'utilisation de 10 ans du système, de planifier la maintenance d'une ressource matérielle au bout de 5 ans, 7 ans et 8 ans, ce qui nous permettra de définir une fonction TI prenant les trois valeurs : 5 ans, 2 ans et 1 an.

Il peut être intéressant de se demander si le pas de référence T de cet espace est bien adapté à la modélisation des phénomènes d'apparition et de disparition des fautes précédemment évoquées (fautes environnementales, existence de modes de défaillances matérielles) et quelle plus value il peut apporter vis-à-vis de la modélisation de phénomènes d'initiation et de propagation d'erreurs d'information le long des flux informationnels de l'architecture fonctionnelle du système. Nous expliquerons dans quelle mesure ce choix nous semble judicieux et quelle mesure supplémentaire s'avère nécessaire pour la prise en compte des contraintes de temps de test.

2-1- Utilisation d'un pas temporel T pour prendre en compte les phénomènes d'apparition/ disparition d'erreurs

Nous supposons comme propriété préliminaire, que sur une période T fixée, correspondant au temps de cycle maximal acquisition de l'état de demande, l'état d'intégrité des ressources matérielles est un **invariant**. L'ensemble des composants matériels constitutifs de chaque ressource matérielle conserve le même état d'intégrité (normal ou erroné) durant une période de largeur T. L'état normal sera noté par la suite 0, tandis que les états d'erreurs seront notés S,D,I suivant le mode de défaillance par lequel ils se manifestent lors de leur utilisation (suivant les notations données en 1-1). Cette propriété **peut être justifiée** sous trois hypothèses nécessaires:

- une hypothèse portant sur le **temps de réparation des ressources matérielles,** imposant que le temps moyen de réparation de chaque ressource après détection d'un état de défaillance de celle-ci, soit largement supérieur à la période T. On doit donc vérifier l'égalité,

$$Min_{i \in E}(MTTR_i) >> T \qquad \textbf{(condition 1)},$$

où $MTTR_i$ est le temps moyen de réparation de la ressource i et E est l'ensemble des ressources matérielles réparables

- une hypothèse portant sur les procédures de réparation des ressources matérielles: « si un ensemble de composants inclus dans une ressource matérielle est jugé défaillant (détecté comme tel), on suppose que le processus de réparation consécutif à cette détection rendra impossible l'utilisation de tout sous ensemble de ces composants par l'architecture fonctionnelle durant toute la durée de la réparation » **(condition 2)**
- une hypothèse portant sur la remise en ligne d'une ressource réparée : « si une ressource matérielle est remise en service durant une période de traitement T et postérieurement à l'instant d'acquisition de l'état de demande, elle ne pourra être considérée comme utilisable par l'architecture fonctionnelle qu'au cycle d'acquisition suivant» **(condition 3)**

La **condition (1)** est en général vérifiée pour des fonctions dédiées sécurité car le temps de cycle est supposé suffisamment court pour éviter une situation critique (accident). Pour s'en

convaincre, il suffit d'observer que le temps de réparation d'une ressource matérielle est toujours d'un ordre de grandeur supérieur à la minute, voir á l'heure, tandis que le temps de cycle T d'un automate industriel est en général d'ordre inférieur à la seconde.

La **condition (2)** peut être validée si on suppose que toute réparation de ressources matérielles s'effectue « hors ligne », c'est-à-dire que l'ensemble de ses composants est déconnecté du reste de l'architecture durant la phase de réparation (toujours vrai pour les ressources matérielles).

La **condition (3)** est quant à elle validée dans le cas d'architectures de types programmables, puisque la présence des périphériques est vérifiée en ligne avant chaque cycle de calcul. Dans les autres cas, on peut supposer que le comportement du système sera estimé à un cycle près, ce qui au vu des ordres de grandeurs des taux de défaillances des ressources matérielles (événements rare) ne portera pas préjudice à la qualité de l'évaluation finale des indicateurs de performance du système.

On peut donc accepter la proposition suivante : *« Pour toute période T, l'état d'intégrité des composants (normal/ défectueux) constitutifs de chaque ressource matérielle sera supposée comme invariant »* **(P1)**

La conséquence directe de **(P1)**, est que **chaque ressource matérielle ne peut présenter au plus qu'un unique mode de défaillance sur un cycle de réaction T (P2)**. L'occurrence de ce mode de défaillance étant liée à l'existence d'une erreur et au mode de sollicitation de cette ressource. Au vu de cette définition, on dira que la ressource matérielle est défaillante à t pour $t \in [kT,(k+1)T[$ avec k entier, si et seulement si les deux conditions suivantes sont réunies :

- il **existe** une erreur matérielle sur le cycle de réaction $[kT,(k+1)T[$
- la ressource est sollicitée au temps t suivant un mode (de sollicitation) révélant cette erreur.

On distingue ainsi clairement la notion d'existence d'erreur matérielle et la notion d'occurrence d'une défaillance matérielle correspondant à l'activation de cette erreur par l'utilisation de la ressource matérielle erronée suivant un mode de sollicitation précis.

Cela va nous permettre de définir clairement le concept de « défaillance matérielle simultanée » (ou quasi-simultanée) introduit dans certaines définitions et de clarifier la problématique des défaillances de cause commune, en accord avec les hypothèses préliminaires restrictives données au chapitre 1 (notamment en 5-), et des modes latents de défaillance. On dira que **deux défaillances matérielles sont « simultanées »** quand elles se produisent sur le même cycle de réaction **$e(k)= [kT,(k+1)T[$** (avec k entier). Deux événements de défaillances matérielles « simultanés » peuvent donc être dus:

- soit à la coexistence d'erreurs sur e(k) dans au moins deux ressources distinctes et de l'activation de ces erreurs matérielles, résultat de leur utilisation suivant le mode de sollicitation les activant.
- soit à l'existence sur e(k) d'une erreur matérielle révélée plusieurs fois, du fait de l'utilisation répétée de cette ressource matérielle suivant le mode de sollicitation l'activant.

Un exemple simple de ces deux cas de figure peut être donné par un système d'acquisition bi-canal relié par un réseau filaire à un voteur 1oo2 (par exemple un comparateur analogique) et ayant pour but de prévenir l'accès non autorisé d'une personne sur une zone exposée. La coexistence d'une erreur matérielle sur chacun des capteurs aura pour conséquence, une non-détection de la situation dangereuse (croisement des barrières immatérielles par la personne) puis une défaillance de la fonction de sécurité, dans le cas où le reste des ressources matérielles sont

intègres. Une erreur matérielle du réseau (coupure) aura les mêmes conséquences puisque le voteur 1oo2 ne sera pas en mesure de recevoir l'information « présence de demande » (fil simple sans contrôle de présence de périphériques).

La notion de **défaillance matérielle simultanée** n'est donc pas dissociable de la définition de la période de référence T, choisie pour le système étudié. Ce temps est donc à la fois la période d'observation de l'état de demande, permettant de juger du besoin de l'activation des moyens de réaction de la fonction de sécurité, et la période de référence permettant de juger des propriétés de « simultanéité » de deux événements (ici de défaillances).

Par abus, on étendra cette notion de simultanéité aux erreurs décrites dans la partie 1-1. On dira que deux erreurs sont simultanées si elles **co-existent** sur un intervalle e(k).

Le concept de « faute environnementale » a été préalablement lié à un niveau de stress. Ces fautes se manifestent lors de la transmission d'un signal, c'est-à-dire sur une fenêtre temporelle de largeur inférieure à T. Elles se traduisent par une perturbation de l'information associée à un signal (cette erreur dépendant de l'émission effective du signal). On peut alors parler de défaillance dans le processus de transfert de cette information. On dira que cet événement de défaillance est « simultané » à tout autre événement de défaillance de même type ou de défaillance matérielle se produisant sur la même période e(k).

D'après les définitions précédentes, on définira un **mode de défaillance de cause commune** de la fonction de sécurité, comme un mode de défaillance causé par au moins deux événements de défaillances simultanées, dus à l'activation d'un ensemble d'erreur non vide (erreur matérielle et/ou environnementale), dont l'apparition peut être vue comme le résultat directe d'une cause partagée (contraintes environnementales et/ou partage de ressource). Le cas d'un mode de défaillance de cause commune, induit par le partage d'une unique ressource erronée (matérielle ou informationnelle) est appelé mode commun de défaillance de la fonction de sécurité (MCF, définition extraite **[Maur00]** identique à celle de l'**IEC62278**)

On définira une **défaillance de mode latent** du système comme une défaillance du système, résultant, soit de l'apparition ou la disparition d'une défaillance locale (matérielle ou perturbation d'information), soit d'un changement des règles d'utilisation des ressources matérielles révélant une erreur non activée jusque là. On parle en général d'activation de mode latent. On peut noter qu'une défaillance de mode latent est le résultat de la coexistence et de l'activation de plusieurs erreurs (matérielles et/ou environnementales) dans le système. Cette classe de défaillances appartient donc, au même titre que les défaillances de causes communes, à la classe des défaillances (matérielles et environnementales) simultanées. Ces défaillances sont définies comme le résultat d'activations « simultanées » (sur une même période T) d'une ou plusieurs erreurs dont l'existence est avérée sur cet intervalle de référence.

Nous proposons donc de prendre en compte l'influence de ces deux types de défaillances particulières (défaillance de cause commune, activation de défaillance de mode latent) comme une sous classe de défaillances « simultanées ». La prise en compte de leur influence sur les indicateurs de performances de la fonction de sécurité sera incluse dans la prise en compte des couplages environnementaux (permettant de justifier l'apparition d'erreurs dues au même stress) et des couplages architecturaux (permettant de prendre en compte implicitement les séquences d'activation d'erreurs menant à un mode de défaillance du système).

2-2- Avantage d'une étude de type flux d'information

Pour déterminer les séquences d'erreurs activées menant à un mode de défaillance, nous interpréterons notre architecture fonctionnelle, comme une séquence reconfigurable de sous fonctions. Les possibilités de reconfigurations fonctionnelles sont nécessairement induites par

l'état des informations échangées. Nous rappelons (cf. chapitre 1) que nous inclurons dans l'architecture fonctionnelle du système d'étude l'ensemble des procédures de test, de recouvrement et de reconfiguration de la fonction de sécurité.

Une modélisation fonctionnelle de notre architecture peut ainsi permettre en prenant en compte successivement les perturbations et pertes d'information dans chaque entité sous fonctionnelle successive, de déterminer de manière complète l'ensemble des combinaisons d'erreurs « simultanées » activées sur une période de référence menant à ce mode de défaillance. Nous distinguerons les perturbations d'information et la perte d'information le long d'un flux.

Le phénomène de perturbation d'information est un phénomène local (sur un flux d'information). Il résulte de l'activation d'une erreur (matérielle ou informationnelle) et se traduit soit par la transformation d'une information valide en une information erronée, soit par la transformation d'une information erronée en une information valide. L'occurrence de plusieurs perturbations d'information le long d'un flux peut donc changer plusieurs fois l'état de validité de l'information. On a donc une propriété de réversibilité.

Le phénomène de perte d'information est un phénomène local, résultant de l'activation d'une erreur matérielle (mode de défaillance I). Il se traduit par une perte de l'information le long d'un flux d'information, qui empêche son utilisation ultérieure.

On distingue deux modes de défaillance distincts pour le système d'étude :

- défaillance de la fonction dédiée sécurité sous l'hypothèse d'une présence demande au moment d'acquisition du cycle e(k) considéré
- déclenchement intempestif de la fonction dédiée sécurité en l'absence de demande au moment d'acquisition du cycle e(k) considéré

On appellera liste caractéristique, l'ensemble des séquences d'activation d'erreurs possible sur un cycle e(k) (cycle de largeur T) et qui sont nécessaires et suffisantes à l'existence d'un mode de défaillance du système d'étude fixée. On notera Ld la liste caractéristique relative au mode de défaillance sur demande de la fonction de sécurité, et La la liste caractéristique relative au déclenchement intempestif de la fonction de sécurité en absence de demande.

Il s'agit de **séquences d'activation d'erreurs**, l'ordre d'activation est contraint par les choix architecturaux du système. On prendra en compte toutes les séquences. Le problème de disjonction d'états n'existe pas puisque l'état d'une ressource matérielle ne dépend pas de celui des autres ressources, d'après les hypothèses faites en 5- du chapitre 1. L'existence d'une séquence d'activation d'erreurs dans une des listes caractéristiques exclut celle de toute sous-séquence extraite de longueur inférieure comme séquence à part entière de cette liste. Nous pouvons justifier de l'existence de telles listes, car toute architecture fonctionnelle peut être décomposée en un nombre fini de séquences de sous fonctions. C'est sur ce principe que se base la décomposition sous forme de BDD proposée par **[Rauz99]**.

Nous définirons, dans le chapitre 3, une méthode systématique menant à cette identification. En outre, nous pouvons déjà justifier que les combinaisons construites par cette méthode seront constituées:

- d'erreurs matérielles portant sur des ressources distinctes, du fait de l'impossibilité de coexistence sur une même période e(k) de deux modes de défaillances distincts pour une même ressource matérielle (proposition **(P2)** – cf. chapitre 2, paragraphe 2-1).
- et d'événements d'« erreurs environnementales » distincts.

2-3- Inconvénients pour les contraintes de temps de test et proposition de résolution

Pour le moment, nous avons considéré, dans notre système d'étude, comme seules sources de perturbation et de perte d'information possibles, les erreurs dites matérielles et les erreurs dites « environnementales ». Nous avons ensuite supposé que l'architecture fonctionnelle, du fait de l'utilisation ordonnée de ressources matérielles et de transmission de signaux entre ressources, allait permettre d'activer certaines combinaisons d'erreurs existant sur une période de référence T et que ces séquences d'activation pouvaient se traduire par une défaillance ou un fonctionnement correct de la fonction de sécurité. Cependant, en réduisant l'horizon de détermination de ces combinaisons à un cycle de réaction T, nous n'avons pas pris en compte les contraintes dites de **temps de test**, intrinsèques au système.

Nous avions précédemment (partie 1) introduit la notion de D-information. Ces informations doivent permettre de diagnostiquer la présence d'erreurs détectables (test en ligne), l'incohérence de certaines informations (redondance) ou la non réception d'une information (horloge de contrôle). Nous allons dans cette partie nous intéresser aux D-informations issues de test en ligne.

On peut aisément prendre en compte les capacités de détection d'erreurs et les processus de décision en découlant dans la représentation de notre architecture fonctionnelle. Le spectre de détectabilité du test et le partage de ressources entre parties testantes et testée peut donc être intégré aux méthodes d'identification de nos listes caractéristiques Ld et La.

Le problème principal réside en fait sur le nombre de cycles nécessaires pour mener à bien un test « en ligne » et de fait le décalage possible, dans le cas d'un fonctionnement « normal », entre l'instant d'apparition d'une erreur matérielle détectable et l'instant d'émission d'un bit de diagnostic par le test sensé la détecter.

Nous appellerons, dans la suite de ce paragraphe, « test en ligne », tout test basé sur une sollicitation en entrée d'un ensemble de ressources matérielles, sur une observation du signal de sortie (signature) en résultant et sur une prise de décision suivant cette signature (principe d'injection/observation). Ces opérations ne doivent pas interagir sur les flux de F-informations échangés dans le système. Les autres tests, basés sur le principe de redondance d'information (par exemple sur deux canaux d'acquisition ou deux voies de traitement parallèles) n'entreront pas dans cette catégorie, puisque leur intervention est conditionnée par une information, actualisée à chaque période T. Il est ainsi tout à fait possible qu'une erreur matérielle « détectable », apparue au cycle e(k), ne soit détectée, par un test en ligne de périodicité supérieur à T, qu'au cycle e(k+p) (avec p>1).

Ce problème peut être constaté par exemple pour le test périodique d'un actionneur par sollicitation de la part d'un microcontrôleur dont la période pourrait être de 6h (avec 6h>>T). Cela implique de rechercher un compromis entre taux de couverture et périodicité moyenne du test (en nombre de cycle).

Pour prendre en compte cette distorsion entre la période T et la périodicité de certains « tests en ligne », nous proposons d'introduire l'événement (non observable) « échec de test », qui correspondra à la non détection sur le cycle e (k) considéré d'une erreur détectable, malgré un fonctionnement correct du test (absence d'erreurs matérielles sur les ressources le supportant) et la présence d'une erreur matérielle détectable par le test.

Cet événement ne dépend ni de l'état d'intégrité des ressources matérielles mises en jeu, ni de l'existence de stress environnementaux. Sa probabilité ne dépendra que du temps, et sera conditionnée par les contraintes de durée et de périodicité du test.

La prise en compte des événements « échecs de test » complète utilement la typologie des erreurs proposée dans la partie 1. Elle permet de considérer cet événement comme induisant la perte d'une D-information (absence de retour d'information de diagnostic).

Elle est tout à fait intégrable à une modélisation de l'architecture fonctionnelle de type flux d'information en l'interprétant comme une perte d'information. Il est de ce fait possible de construire les listes caractéristiques Ld et La dont les combinaisons pourront être constituées:

- d'**erreurs matérielles portant sur des ressources distinctes**,
- **d'événements d'« erreurs environnementales » distincts**.
- et **d'événements « échec de test » distincts**.

Nous ajouterons uniquement comme hypothèse que les événements « échec de test » appartenant à une même combinaison sont supposés indépendants (au sens probabiliste). Cette hypothèse ne sera pas justifiée dans notre étude et impose une indépendance entre instants de déclenchement des différents tests en ligne. Nous supposerons donc, dans notre étude, cette indépendance.

3- Structure du modèle général d'évaluation

3-1- Vers une structure basée sur une approche multi-échelle de temps

La construction des listes caractéristiques doit pouvoir être réalisée en modélisant les possibilités de perturbations de l'information dans chacune de ces entités par un sous modèle permettant la prise en compte de l'existence d'état d'erreurs matérielles et environnementales sur un cycle de réaction T. L'exposé d'une méthode de construction de ces listes, permettant la prise en compte des problématiques de partage de ressources par différentes entités sous fonctionnelles et de la transmission d'information, est une condition nécessaire à l'établissement du modèle général d'évaluation, prenant en compte les différentes sources de couplages architecturales. Elle sera donc détaillée dans le chapitre 3 de notre travail (mise en place d'une méthode de construction systématique).

Ces listes ne dépendent que des règles d'utilisation des ressources matérielles et informationnelles dans une architecture fonctionnelle représentant les flux de F et de D-informations. Elles sont uniquement **dépendantes des choix d'implantation fonctionnelle de départ** et des propriétés de test et de reconfiguration implantées. Leur détermination peut donc être conduite sous l'hypothèse d'une connaissance complète de l'architecture fonctionnelle, incluant les processus de :

- traitement d'information, (transformation, transmission, stockage)
- de décision (choix d'informations de sortie au vue d'informations d'entrées multiples)
- de test par redondance d'information et de test en ligne
- de contrôle de bonne réception d'une information (chien de garde, horloge d'actualisation de variable)
- de reconfigurations fonctionnelles et de modes dégradés.

Nous supposerons dans toute la suite de ce chapitre, ces listes comme construites. Elles pourront être vues comme des **invariants du temps**, et prendrons en compte les procédures de reconfigurations fonctionnelles. Le calcul de ces combinaisons pourra donc être effectué en une seule fois en identifiant les conséquences d'erreurs possibles (matérielles, environnementales ou

d'événement « échec de test »). Nous conserverons les seules combinaisons de ces événements d'activation d'erreurs induisant l'un des deux modes de défaillance du système d'étude.

Si on suppose que dans une liste chaque combinaison est distincte, on pourra donc en accord avec la définition de ces listes, dites « caractéristiques », écrire:

$$PFD(t=kT) = p(Ld, e(k)) \qquad \textbf{(Equation 21)} \quad \text{et}$$

$$PFS\ (t=kT) = p(La,e(k)) \qquad \textbf{(Equation 22)}$$

avec

p(Ld, e(k)) la probabilité d'existence des combinaisons d'erreurs Ld sur la période d'acquisition e(k)

p(La, e(k)) la probabilité d'existence des combinaisons d'erreurs La sur la période d'acquisition e(k)

Nous désirons nous appuyer sur ces deux équations (21 et 22) pour construire notre modèle d'évaluation. Il est, en effet, simple de remarquer que si nous sommes en mesure de calculer ces deux variables PFD(t=kT) et PFS(t=kT) pour chaque cycle, il sera possible d'évaluer dynamiquement nos indicateurs de performances en utilisant les équations 14 et 15 données au chapitre 1 (p.19).

Il nous reste donc à évaluer la probabilité de chaque combinaison constitutive de ces listes pour toute période d'acquisition k fixée. Le problème provient du fait que ces séquences d'événements, bien que distinctes, peuvent contenir des événements correspondant à l'activation d'une même erreur ou d'erreurs induits par des contraintes environnementales subits par différents matériels. Ce problème est bien connu dans les approches de type arbres de fautes (FT), pour lesquelles l'indépendance des événements de base doit être assurée.

On peut, nénamoins, calculer pour toute période e(k)=[kT, (k+1)T[d'acquisition fixée, la probabilité d'occurrence de ces listes comme la somme des probabilités des combinaisons la constituant, en s'appuyant sur les hypothèses du chapitre 1. Le fait pour une ressource matérielle d'être dans un état d'erreur est, en effet, indépendant de la proposition d'existence d'un autre état d'erreur pour une autre ressource sur la même fenêtre de référence e(k) d'après les hypothèses d'étude, bien qu'elle puisse dépendre du l'effet cumulé de facteurs environnementaux sur les périodes précédentes. En outre, l'activation de ces erreurs est un événement certain et déterministe car contraint par les choix architecturaux.

En effet, les probabilités d'erreurs matérielles et environnementales sont liées aux facteurs de stress environnementaux. L'idée proposée est donc de s'abstenir de chercher une règle de corrélation fixe entre chaque probabilité d'erreur, mais de remonter au niveau des causes d'apparition et de disparition de celles-ci. Nous allons, de ce fait, contourner la difficulté précédemment évoquée en utilisant la dimension temporelle du problème. Ainsi on remarque que les mécanismes d'apparition et de disparition des fautes dépendent essentiellement de facteurs environnementaux dont l'évolution temporelle peut être supposée estimable. Il est possible, en modélisant l'évolution des probabilités d'erreurs, par des processus stochastiques parallèles de pas discret T, dont les règles d'évolution dépendent directement de l'évolution temporelle des stress environnementaux, de modéliser l'évolution de ces probabilités d'erreurs en prenant implicitement en compte les couplages environnementaux. Les probabilités de ces combinaisons correspondent donc au produit des probabilités d'erreurs la constituant sur e(k), puisque ces événements sont jugés indépendants (au sens probabiliste).

3-2- Prise en compte implicite du couplage fautes/environnement

On pourra s'intéresser aux trois classes d'événements précédemment évoquées : les événements d' « **échec de test** », les fautes **environnementales** et enfin les fautes **matérielles (pannes)**.

La probabilité d'occurrence d'un événement « **échec de test** » sur la période e(k) peut être définie suivant les contraintes de périodicité (redémarrage tous les N cycles) et de durée (en nombre de cycles p) du test. La probabilité de cet événement au cours du temps sera supposée estimée (hypothèse d'un fonctionnement normal du test) et ne dépendra que de la valeur de ces deux grandeurs N et p et de la structure du test (nombre de détection possible d'une faute matérielle par le test sur un cycle et répartition des instants de détection). Elle sera donc indépendante des événements de fautes environnementales et matérielles qui dépendront des contraintes environnementales auquel le système est soumis. Nous la supposerons donc estimable.

La probabilité d'existence d'une erreur environnementale, dépend essentiellement de critères de stress dits « instantanés ». On associera la cause d'apparition d'une telle erreur à une probabilité de « choc environnemental », causé par une variation brusque des interférences électromagnétiques au moment de la transmission ou sur la période de stockage de l'information considérée. Sa persistance sera prise égale à T. Ce choix est lié au fait qu'une erreur environnementale ne peut affecter qu'une information et que sa condition d'activation est donc extrêmement courte puisqu'elle est liée à un transfert ou à un stockage de celle-ci. On supposera que l'activation de ce type d'erreur dépend uniquement de l'émission effective de l'information pouvant être affectée par celle-ci. L'apparition d'un tel événement sera considérée **comme indépendante des phénomènes d'apparition et de disparition des erreurs matérielles puisque l'existence d'une** erreur environnementale ne sera pas liée à l'historique en termes de contraintes environnementales sur les périodes précédentes. La probabilité d'existence d'une erreur environnementale est donc directement liée au niveau de stress environnemental sur la période T. Sa durée de persistance sera prise au plus égale à cette valeur de référence comme le propose les études par simulation par perturbations externes **[Chen00]**.

Dans notre approche, la probabilité d'existence de ces erreurs environnementale sera donc prise comme directement dépendante d'un niveau de stress environnemental. On pourra donc assimiler l'existence de deux erreurs environnementales lors de leur utilisation soit à l'effet de facteurs environnementaux indépendants au sens probabiliste, soit à un événement externe (choc environnemental) dont la probabilité au cours du temps sera supposée estimables et pourra dépendre de la fenêtre de référence e(k) considérée.

La probabilité d'existence d'une **erreur matérielle** sur la période e(k) sera quant à elle directement reliée à la distribution des probabilités d'existence de mode de défaillance matérielle de cette ressource R au début de la période e(k) suivant les modes {0,S,D,I}. L'évolution de ces probabilités sera donnée par des modèles stochastiques associés à chaque ressource matérielle. Ces modèles, de type discret et de pas d'itération T, donneront l'évolution des probabilités d'existence des états d'erreurs {0,S,D,I} pour chaque ressource matérielle. Leurs taux de transition correspondent à l'occurrence de séquences de pannes des constituants de ces ressources ou à la réalisation d'une opération de maintenance. Ces événements dépendant des stress environnementaux, les taux de transitions (règles d'évolution de nos modèles stochastiques) seront donc assimilés à des variables du temps (échelle discrète de pas T) et dépendront du niveau de contraintes environnementales auxquelles est soumise la ressource au début de chaque période T. On pourra de ce fait simuler l'influence des contraintes environnementales sur les lois d'évolution de chaque probabilité d'erreurs matérielles, sans négliger la possibilité d'une exposition des différentes ressources à des stress partagés, en proposant une évolution parallèle des différents modèles. Nous prenons de ce fait, implicitement dans l'estimation des probabilités

d'erreurs matérielles en compte l'exposition de différentes ressources matérielles à des contraintes environnementales (stress et réparation) communes (contraintes environnementales). Nous expliquerons dans la partie 4 comment obtenir le modèle permettant de simuler l'évolution des probabilités d'erreurs de chaque ressource matérielle. Notre choix se portera sur une utilisation de chaînes de Markov non homogènes dans le cas de ressource matérielle maintenue non préventivement et sur une extension de cette classe de modèle pour la prise en compte de l'influence des actions de maintenance préventive.

Ces choix de modélisation permettent de justifier que la probabilité d'activation d'une combinaison d'erreurs matérielles sur un cycle fixé e(k) (de largeur T) puisse être calculée comme égale aux produits des probabilités d'existence de chacune de ces erreurs matérielles, dont l'évolution est donnée par ces modèles. Ceci permet de justifier la proposition P3 faite ultérieurement dans l'exposé de l'algorithme de calcul des indicateurs de performances (paragraphe 3-4).

3-3- Structure du modèle général d'évaluation des indicateurs de performances.

On propose de ce fait une structure en trois niveaux pour le modèle d'évaluation des indicateurs de performances de la fonction de sécurité étudiée (**figure 9**).

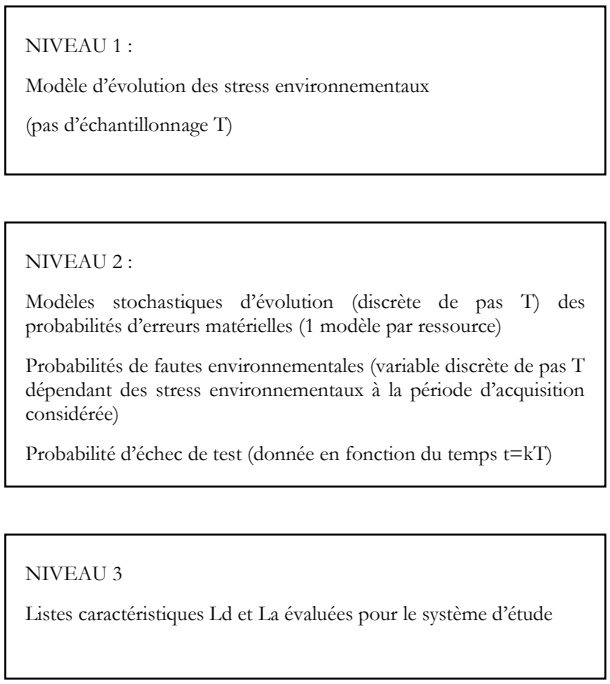


Figure 9: Structure du modèle général d'évaluation des indicateurs de performances

Le **premier niveau de représentation** fournit un modèle d'évolution temporelle des contraintes environnementales auquel est soumis l'architecture. Il donne en un nombre de points, jugés comme représentatifs des zones d'implantations des ressources matérielles, l'évolution des contraintes environnementales pouvant influer sur les règles de dégradation (température, pression, radiations…) et de réparation des composants au cours du temps. Ces modèles d'évolution sont discrets et de pas d'échantillonnage T. En général, ces modèles sont dérivés d'une estimation d'évolution de l'environnement souvent effectuée en fixant un certain nombre d'hypothèses sur l'évolution de l'environnement en absence d'occurrence de demande. Nous n'entrerons pas ici dans le détail du maillage de l'espace d'implantation **[Popo01]**,**[Auch04]** qui

permet d'approximer l'évolution temporelle de certains facteurs d'influence (température, pression) sur l'espace d'implantation des composants. Des profils estimés d'évolution de paramètres environnementaux en cours de mission (souvent divisée en phase) peuvent être trouvés dans un large spectre de domaines que ce soit aéronautique **[FIDES]**, automobile (notamment pour les domaines de vibrations mécaniques), verrier (profil Température/pression)… On pourra trouver dans **[Maur00]** une classification claire des contraintes environnementales influant sur les lois de dégradation des composants. Nous supposerons **donc** ce modèle d'**évolution des contraintes environnementales comme fixé** et si possible, validé par retour d'expérience soit sur l'installation existante (hypothèse d'absence de demande), soit sur des installations pour lesquelles des comportements analogues (sources de chaleurs, sources de radiation…) ont déjà été étudiés. Notre étude supposera donc ces profils discrets comme suffisant pour juger de manière réaliste des influences des contraintes environnementales sur les lois de dégradation des composants. On limite généralement ces profils aux paramètres environnementaux dont la variation est jugée suffisante pour faire varier de manière conséquente le taux de pannes des composants implantés ou pour contribuer à l'apparition de fautes environnementales, les autres paramètres étant souvent jugés constants.

La prise en compte de contraintes humaines, portant sur les opérations de maintenance, est actuellement fortement limitée par le manque de modélisation stochastique des comportements humains. On supposera donc que les probabilités d'erreurs dans les opérations de maintenance sont données par une estimation basée sur une approche empirique (retour d'expérience) et/ou qualitative (pondération par les facteurs d'influence **[Rasm97]**, **[Swai83]**; **[Holl98]**). On supposera le temps d'indisponibilité pour une tâche de maintenance comme faible et directement pris en compte par l'estimation du temps moyen de réparation.

Ce premier niveau, appelé modèle d'évolution des contraintes **environnementales**, permet de définir pour chaque temps t=kT, avec k entier et t variant de 0 à T_{life}, une donnée des facteurs de stress environnementaux. Ces profils étant supposés comme indépendants des réactions de la fonction dédiée sécurité (cf hypothèses chapitre 1-3), ils pourront être :

- soit simulés dynamiquement (en entrant les jeux d'équations différentielles d'évolution ce qui consomme en général un temps de calcul important),
- soit donnés par des profils d'évolution polynomiale par morceaux d'ordre faible, qui réduisent le nombre de paramètres à conserver), si l'écart introduit est jugé comme négligeable en utilisant des règles de périodicité.

Un exemple simple est donné dans le logiciel **FIDES** par la simulation de la température sur un profil de vol (profil linéaire par morceau).

Le **second niveau** de représentation contient trois classes d'éléments :

- Un ensemble de processus stochastiques, de type chaînes de Markov non homogènes (cf. chapitre 4 pour détails), permettant de modéliser l'évolution des probabilités d'erreur de chaque ressource matérielle. La particularité de ces modèles réside dans le fait que les probabilités conditionnelles de passage d'un état d'erreur à un autre sont directement reliées, par des règles de couplage causales, aux stress environnementaux auxquels est soumise la ressource durant la période de référence e(k)= [k.T ; (k+1)T[avec k=E(t/T). Ces niveaux de stress sont supposés constants sur cette période de référence. Ils permettent donc l'évaluation des probabilités d'erreur matérielle par simulation dynamique, en connaissant les règles de couplages et l'évolution des stress environnementaux au cours du temps (sous un pas d'échantillonnage minimal T). L'estimation de l'évolution temporelle de chacun de ces processus stochastiques peut

donc être effectuée en parallèle après actualisation des probabilités conditionnelles de transitions sur la période correspondante.

- Un ensemble d'attributs, correspondant aux probabilités de fautes environnementales, dont les valeurs sont directement liées à l'évolution temporelle de stress environnementaux conditionnant leur existence (facteurs d'influence)
- Un ensemble d'attributs, correspondant aux probabilités des événements de classe « échec structurel de test » dont la valeur est donnée par une fonction ne dépendant que du temps (cf paragraphe 3-2).

Le **troisième niveau** de représentation est donné par les deux listes caractéristiques, Ld et La, définies de la manière suivante :

- Ld est la liste complète des combinaisons d'erreurs distinctes (matérielle, environnementale et/ou événement « échec de test ») et simultanées pouvant être activées sur un cycle de réaction T et induisant de manière nécessaire et suffisante une absence de réaction de la fonction de sécurité sous l'hypothèse d'une présence de demande au moment d'acquisition.
- La est la liste complète des combinaisons d'erreurs distinctes (matérielle, environnementale et/ou événement « échec de test ») simultanées **pouvant** être activées sur un cycle de réaction T et induisant de manière nécessaire et suffisante un déclenchement intempestif des actions de réaction de la fonction dédiée sécurité (réaction en absence de demande).

Une représentation correcte de ces trois niveaux va nous permettre d'aboutir à l'évaluation des indicateurs de performance, PFD_{avg} et PFS_{avg}, du système étudié par une simulation dynamique décrite dans la partie suivante. Cette simulation est relativement simple et peut être automatisée sans difficulté majeure.

3-4- Evaluation des indicateurs de performances.

L'évaluation de nos indicateurs de performances est basée sur l'algorithme suivant. Elle commence au temps t=0, pour lequel deux variables S1 et S2 sont initialisées à la valeur 0 et une variable compteur k, initialisée à la valeur k=0. Elle est menée pour k variant de la valeur 0 à la valeur E(Tlife/T), noté N avec E la fonction partie entière et T_{life} le temps d'utilisation du système étudié. Le temps t=0 est l'instant de mise en service de notre système.

On pose t=kT

Initiation de l'algorithme :

Pour k=0,

A- Les stress environnementaux sont évalués à l'instant de référence t=0 grâce au profil d'évolution des stress environnementaux (premier niveau).

B- Les valeurs de ces paramètres de contraintes environnementales permettent ensuite dans le second niveau:

- l'actualisation des probabilités conditionnelles des processus de Markov associés à l'évolution de l'état d'erreur des ressources matérielles (dans chaque processus),
- l'évaluation des probabilités de fautes environnementales,
- l'évaluation des probabilités « échec de test » au temps t = k.T.

C- On effectue, ensuite, dans le second niveau, l'itération des processus markovien (à un pas temporel T) ce qui nous permet d'actualiser les probabilités d'erreurs pour chaque ressource matérielle.

D- On utilise, ensuite, les probabilités d'erreurs évaluées dans les étapes B et C, pour calculer les probabilités pd et pa d'occurrences des listes Ld et La (troisième niveau) au temps t=kT.

La probabilité d'occurrence de chacune de ces listes est donnée par la somme des probabilités de chaque combinaison d'erreurs « simultanément » activées qu'elle contient. Ces dernières probabilités sont données par le produit des probabilités d'erreurs, actualisées lors des étapes B (erreur environnementale et événements « échec de test ») et C (pour les erreurs matérielles). **(P3)**

E- On effectue l'addition (intégration par morceaux):

$$S1 := pd .T + S1 \qquad \textbf{(Equation 23)}$$

$$S2 := pa. T + S2 \qquad \textbf{(Equation 24)}$$

$$k := k+1 \qquad \textbf{(Equation 25)}$$

<u>Continuation de l'algorithme</u>

Evaluation de k, si k<N

A- Les stress environnementaux sont évalués à l'instant t= k.T grâce au profil d'évolution des stress environnementaux (premier niveau).

B- Les valeurs de ces paramètres de contraintes environnementales permettent ensuite, en connaissant k, dans le second niveau:

- l'actualisation des probabilités conditionnelles des processus de Markov associés à l'évolution des probabilités d'erreurs des ressources matérielles (dans chaque processus),
- l'évaluation des probabilités de fautes environnementale,
- l'évaluation des probabilités « échec de test » au temps t = k.T.

C- On effectue, ensuite, dans le second niveau, l'itération des processus markovien (pour un pas temporel T, correspondant à un pas d'itération) ce qui nous permet d'actualiser les probabilités des erreurs pour chaque ressource matérielle.

D- Puis on utilise les probabilités d'erreurs et de fautes évaluées dans les étapes B et C, pour calculer les probabilités pd et pa d'occurrences des listes Ld et La (troisième niveau) au temps t=kT.

E- Enfin, on effectue l'addition (intégration par morceaux):

$$S1 := pd .T + S1$$

$$S2 := pa. T + S2$$

$$k := k+1$$

<u>Arrêt de l'algorithme</u> Déclenche pour k=N (sortie de boucle)

On arrête la boucle précédente et on affiche les indicateurs de performances :

$$PFD_{avg} := S1/ T_{life} \qquad \textbf{(Equation 26)}$$

$$PFS_{avg} := S2/ T_{life} \qquad \textbf{(Equation 27)}$$

qui correspondent à l'évaluation de nos indicateurs de performances.

L'algorithme proposé fonctionne et va nous permettre d'estimer les valeurs correctes de nos indicateurs de performances, sans négliger les influences conjointes dues aux couplages environnementaux et architecturaux, sous la condition que les listes La et Ld soient bien construites (complètes et prenant en compte les partages de ressources dans l'architecture) et que les processus stochastiques permettant l'évaluation des probabilités d'erreurs de chaque ressource matérielle soient corrects, c'est-à-dire que les règles de couplage entre stress environnementaux et probabilités conditionnelles de transitions dans nos processus markoviens discrets soient validés.

En outre, l'algorithme possède certaines propriétés de robustesse, ainsi le saut d'un indice k n'entraînera qu'une faible distorsion des résultats des estimations d'indicateurs de performance, si on suppose une évolution lente des indicateurs instantanées pa et pd, et un nombre de pas d'itération important.

Son inconvénient majeur réside dans le temps de calcul, du fait de l'ordre de grandeur en général conséquent de notre valeur N= E(Tlife/T). Il est, cependant, à noter que certaines hypothèses sur le profil d'évolution des stress environnementaux, détaillées au chapitre 4, peuvent permettre de remplacer les processus stochastique de pas d'itération T par des processus de pas d'itération temporel supérieur, sans trop altérer la qualité des probabilités d'erreurs matérielles estimées. L'utilisation de ces processus « équivalents » permet de remplacer T par ce nouveau pas d'itération dans l'algorithme de calcul des indicateurs de performances et de réduire d'autant le temps de calcul nécessaire à leur estimation.

Ce type de modèle présente cependant un nombre d'avantages importants vis-à-vis des méthodes actuelles dans l'étude d'architectures réparables et reconfigurables:

- Il distingue clairement l'influence des contraintes architecturales, données par les listes La et Ld, et celle des contraintes environnementales, prises en compte de manière dynamique par les interactions des niveaux de représentation 1 et 2.
- Il préconise la prise en compte implicite et dynamique des problèmes de couplages environnementaux, en évaluant pour chaque période T, les processus stochastiques modélisant l'évolution des probabilités d'erreurs pour chaque ressource matérielle.
- Il considère du fait de la construction des listes La et Ld les problèmes de partage de ressources (matérielles et informationnelles) dans l'architecture, de contraintes de test et de reconfiguration.
- Il permet une distinction forte entre existence et activation d'erreur, permettant de mieux appréhender l'influence des modes de défaillances simultanées (et donc des modes de défaillances de causes communes, de modes communs et de modes latents).
- Il prend en compte les propriétés de tolérance aux fautes, de reconfiguration et de réparation intrinsèques au système d'étude.

Conclusion de la partie 2 et introduction des parties 3 et 4

Dans ce chapitre, nous avons proposé en nous appuyant sur les concepts de couplage et de simultanéité, et en introduisant un temps caractéristique, appelé période d'observation et prise égale au temps d'acquisition T de notre boucle de réaction, un modèle d'évaluation général des performances (PFD et PFS). Celui-ci peut s'appliquer aux fonctions de sécurité électronique et programmable. Les problèmes de causes communes de défaillances et de modes latents, ont été traités comme sous classe des défaillances fonctionnelles, dues à l'occurrence simultanée de défaillances. Le modèle s'est appuyé sur une division du temps de vie du système suivant deux échelles :

- une échelle de temps global, discrète de pas d'échantillonnage T qui permet la prise en compte de l'évolution des stress environnementaux et humains, l'évolution des modes de défaillances matérielles ainsi que la simulation de l'évolution des valeurs instantanées du PFD et du PFS.
- une échelle de temps, dite logique, liée aux choix d'architectures fonctionnelles (procédé de traitement de l'information), dans une fenêtre temporelle T. Celle-ci doit permettre la construction de deux listes Ld et La et ainsi de prendre en compte les couplages architecturaux, dus à la transmission d'information, au partage de ressources et au partage d'informations.

Il peut permettre aisément l'évaluation des indicateurs de performances sous l'hypothèse :

- d'une description réaliste des profils de stress environnementaux (niveau 1),
- d'une modélisation correcte des processus stochastiques du niveau 2,
- d'une exactitude dans les règles de corrélation entre lois d'évolutions des modèles stochastiques de niveau 2 et stress environnementaux,
- d'une détermination complète et précise des listes Ld et La.

La confiance dans les résultats avancés dépend néanmoins fortement de la qualité de construction des modèles stochastiques du niveau 2 et de la construction complète des listes La et Ld. C'est pourquoi nous proposerons dans les chapitres 3 et 4 de ce travail, les outils et méthodes nécessaires à leur construction.

De notre point de vue, la construction des listes La et Ld est l'étape de loin la plus critique, puisqu'elle doit permettre de prendre en compte les contraintes architecturales, en termes de dépendance temporelle entre sous fonctions et de partages de ressources dans notre architecture. Il reste, en particulier, à démontrer qu'il est à la fois possible de construire les listes Ld et La évoquées, en respect des contraintes architecturales, tout en assurant leurs propriétés en termes de complétude. Une méthode de construction systématique de ces listes sera donc décrite dans le chapitre 3. Elle débouchera sur l'implantation d'un prototype informatique (outil d'évaluation automatique des listes) permettant de faciliter sa mise en œuvre.

La construction des processus stochastiques de niveau 2, permettra quant à elle de souligner l'intérêt d'utilisation de chaînes de Markov non homogènes, les règles d'évolution de leurs attributs mais aussi la possibilité de la prise en compte des opérations de maintenance préventive planifiée. Elle débouchera sur une réflexion sur l'élargissement du pas de calcul T pour leur évaluation (étape nécessaire à la réduction du temps de calcul du modèle d'évaluation général). Ces points seront donc abordés dans la partie 4.

Chapitre 3

Méthode systématique de construction des listes caractéristiques

Chapitre 3 - Méthode systématique de construction des listes caractéristiques

Nous venons de montrer l'importance de la construction complète de deux listes, dites caractéristiques, permettant de modéliser les combinaisons d'événements simultanés nécessaires et suffisantes menant à un mode de défaillance de la fonction dédiée sécurité. Ces deux listes, notées La et Ld, sont à la base de notre modèle d'évaluation dynamique des indicateurs de performances (PFD_{avg} et PFS_{avg}) du système étudié Leur construction doit donc être complète pour permettre un niveau de confiance suffisant dans le modèle d'évaluation et dans ses résultats.

Cette exigence impose la mise en place d'une méthode d'identification de ces listes, à partir de modèles pouvant être vérifiés, et suivant des règles de construction systématique et clairement justifiés. Cette construction devra prendre en compte deux aspects majeurs. En premier lieu, elle devra accorder une importance particulière aux phénomènes de perturbation d'information(s) le long d'un ou plusieurs flux d'information, qui sous-tendent les mécanismes d'initiation et de propagation d'information erronée, et au phénomène de perte de signal (appelé interruption de flux d'information). En second lieu, la construction devra permettre la prise en compte implicite de l'effet des partages de ressources (informationnelle et matérielle) dans l'architecture fonctionnelle étudiée.

Nous débuterons ce chapitre en précisant l'analogie entre les concepts de base que nous nous proposons d'utiliser et les notions, introduites en logique linéaire. Cette analogie nous permettra d'expliquer clairement notre démarche de fond, basée sur une représentation de haut niveau de l'architecture fonctionnelle, et nous permettra d'expliquer l'utilisation de langages finis (au sens de la théorie des automates d'états) pour construire de manière systématique les deux listes recherchées.

La deuxième partie sera ensuite dédiée à la définition claire des éléments constitutifs du formalisme de haut niveau et exposera une méthodologie de construction d'un modèle de haut niveau associé à l'architecture fonctionnelle étudiée.

Nous aborderons la troisième partie par un bref rappel sur la théorie des automates d'états finis. Puis nous justifierons le fait que le comportement de chaque entité sous fonctionnelle de notre modèle de haut niveau peut être décrit par un automate d'états finis. Nous imposerons une définition de classe d'appartenance pour chaque événement et un certain nombre d'attributs pour chacune d'elle. Un événement sera, par la suite, défini uniquement par la donnée de sa classe et des valeurs de ses attributs.

La quatrième partie nous permettra, en partant de ces deux niveaux de représentation, de construire nos listes finales. Nous définirons le processus de propagation et de réduction de langage le long d'un flux d'information qui constitueront les opérations élémentaires permettant la construction complète des listes caractéristiques recherchées. Nous expliquerons ensuite comment les langages sont propagés et réduits, au regard de la structure du modèle de haut niveau. Cette propagation aboutira à la construction finale des listes recherchées La et Ld dont les propriétés seront justifiées (structure, complétude, prise en compte des contraintes architecturales)

Nous présenterons, dans la cinquième partie, un outil de construction automatisé de ces listes, implanté lors de la thèse, appelé AFMCI. Cet outil basé sur les règles de propagation et de réduction des langages permet de mener à bien la construction systématiques des listes La et Ld, par une saisie manuelle des modèles de haut et bas niveau, au moyen d'une interface graphique. Il facilite, en outre, la réutilisation des listes générées et leur exportation vers des outils de calculs

existants, pour une mise en œuvre efficace de la méthode d'évaluation des indicateurs de performances (chapitre 2).

1- Propositions et idées fondamentales

Nous avons précédemment souligné que la construction des listes Ld et La s'appuie sur une vision fonctionnelle de notre architecture, basée sur le concept d'information introduit au chapitre 1 et sur les mécanismes d'initiation et de propagation d'information erronée dans cette architecture.

La mise en œuvre de cette idée de base s'appuie sur un nombre limité de concepts que nous allons expliquer dans les sous parties suivantes :

- La définition claire du concept de ressource, dans un sens élargi (et pas simplement de ressource matérielle).
- La définition du concept d'utilisation et de consommation de ressources.
- La notion d'entité sous fonctionnelle.
- La notion de flux d'information et d' « héritage » le long d'un flux d'information.

Ces notions de base sont nécessaires à la justification de notre méthode systématique de construction des listes et sous-tendent les concepts de propagation et de réduction de langage le long de flux d'information, utilisés par la suite pour construire nos listes La et Ld.

1-1- Notion de ressource

La notion générale de ressources a déjà longuement été définie dans **[Gira97]**, nous conserverons donc la définition générique suivante :

Ressource = « chose qui peut être produite, utilisée et consommée»

On va restreindre cette définition à notre problématique (description de l'architecture matérielle et logicielle du système étudié). Nous ne nous intéresserons pas aux ressources dites humaines. Elles entrent en jeu au niveau des processus de réparation matérielle et qui seront, de ce fait, prises en compte dans le chapitre 4 de notre travail, dédié à la construction de modèles temporels d'évolution des probabilités d'erreurs matérielles.

On peut distinguer dans notre étude deux types de ressources : les ressources matérielles et les ressources informationnelles. Nous rappelons la notion de **ressource matérielle**, en accord avec la définition donnée dans le chapitre 1, comme :

« un ensemble de composants mécaniques, électroniques et/ou programmables, pouvant remplir une fonctionnalité unique (et donc non reconfigurable) durant la durée de vie du système étudié, dont la réparation est menée globalement et pour lequel les procédures de réparation (préventive et ou corrective) impliquent une indisponibilité de l'ensemble de ses composants durant la période de réparation »

Le verbe pouvoir, indique bien que la fonctionnalité, n'est pas toujours requise sur une période e(k) fixée. Le terme « système étudié » permet de prendre en considération l'utilisation de cette ressource pour quelques fins que ce soit et inclut de ce fait le spectre de sous fonctions de test, de contrôle, de décision ou de recouvrement, d'après les limites du système imposées dans le chapitre 1. La notion d'indisponibilité durant la période de réparation (si ressource réparable) est réaliste, puisque la réparation d'une ressource matérielle induit l'impossibilité d'utilisation de cette ressource par le système durant la période de réparation.

Nous définirons la notion de **ressource informationnelle**, comme « un ensemble structuré de données (éléments bruts de nature mesurable) auquel est adjoint une signification et ayant pour but de permettre de manière directe ou indirecte la réalisation de la fonction de sécurité ». En se basant sur cette définition, on peut introduire deux types de ressources informationnelles.

Les **ressources informationnelles de classe 1**, définies précédemment sous les notions de F-information et de D-information, permettent d'établir au regard d'observations effectuées sur l'environnement et sur un ensemble de ressources matérielles testées « en ligne », une loi comportementale au niveau des actionneurs de la fonction dédiée sécurité.

Les **ressources informationnelles de classe 2** permettent de fixer les règles d'utilisation de ressources matérielles. Elles sont supposées n'être sujettes ni aux modifications (hypothèse de sécurité-confidentialité) ni aux altérations au cours du temps (hypothèses données à la fin de la partie 1-2). Cette deuxième classe de ressources regroupe l'ensemble des instructions données en code source, les fonctions d'adressage et les règles de communication entre ressources matérielles (protocoles).

D'après nos hypothèses du chapitre 1 (sécurité-confidentialité, non altération du code source pour les parties logicielles), on supposera les ressources informationnelles de classe 2 comme invariantes au cours du temps. Nous prendrons donc uniquement en compte, dans ce chapitre, les ressources matérielles et les ressources informationnelles de classe 1 (F- et D-informations), appelées par la suite **informations**. Ce terme désigne l'ensemble des F- et D-informations échangées dans l'architecture étudiée. Les ressources informationnelles de classe 2, définies comme données invariantes au cours du temps, permettent uniquement de fixer les règles d'utilisation des ressources matérielles dans l'architecture fonctionnelle considérée. Ces règles seront implicitement prises en compte dans la définition des entités sous fonctionnelles.

1-2- Notion d'entité sous fonctionnelle

La définition de la notion **d'entité sous fonctionnelle** est nécessaire pour permettre une subdivision efficace de l'architecture fonctionnelle. On appellera **règle d'utilisation de ressources matérielles**, toute séquence (suite ordonnée) fixée d'opérations de traitement de signal, chacune de ces opérations de base étant supposée comme attribuée à **une ressource matérielle fixée**.

Une **entité sous fonctionnelle** est définie comme un procédé de traitement de signal. Ce procédé utilise un ensemble fini de signaux d'entrées et doit délivrer un signal de sortie unique. Il est basé sur l'application d'une règle d'utilisation de ressources matérielles sur l'ensemble des signaux d'entrées. Cette règle est choisie sur un ensemble fini de règles des ressources matérielles, **uniquement au vu de l'état des signaux d'entrée à traiter,** en accord avec **[Hoc87]**. La propriété de reconfiguration d'une entité sous fonctionnelle est donc uniquement dépendante des règles d'utilisation possible de ressources matérielles pour un processus de traitement du signal et de l'état d'un ensemble fini (pouvant être vide) des signaux d'entrée. On parlera ainsi d'entité **sous fonctionnelle non reconfigurable** lorsque la règle d'utilisation de ressources matérielles permettant le processus de traitement du signal est fixée et non dépendante de l'état des signaux d'entrée.

Maintenant que nous avons précisé les termes ressource et entité sous fonctionnelle, nous allons essayer de dégager dans quelle mesure, on peut affirmer d'une ressource qu'elle est « consommée, utilisée ou produite ». Cette définition va nous permettre de dégager certains concepts intéressants entre le mode d'utilisation et les conséquences de cette utilisation pour les ressources matérielles et informationnelles retenues, puis de mettre en évidence un certain nombre de similitude avec les fondements de la logique linéaire.

1-3- Notion d'utilisation et de consommation de ressources

Nous ne souhaitons pas ici rentrer dans le détail sur les fondements de la logique linéaire. On pourra trouver dans les travaux **[Reto02], [Bräu96]**, une introduction satisfaisante sur les relations historiques d'une telle logique vis-à-vis de la logique classique. On peut rappeler que dans la logique linéaire, la validité d'une proposition (valide/erronée) peut varier au cours du temps. On peut, en outre, attacher à une proposition des conditions d'extinction liée, par exemple, à la délivrance d'une nouvelle proposition.

Le nombre et l'ordre d'apparition des propositions peuvent donc influer sur les conclusions fournies. Il est donc possible de décrire un problème temporel sous la forme de séquents, définissant les règles de modification et de consommation des propositions de base. En d'autres termes, c'est la structure même du contexte qui détermine en grande partie la logique. L'adoption de ce type de logique s'est accompagnée d'une redéfinition des connecteurs logiques de base et de certaines de leurs propriétés **[Gira87]**. On peut illustrer l'utilisation de la logique linéaire par le court exemple suivant:

Soient A, B, C et D les propositions suivantes :

A : Pierre a 5 francs

B : il faut 5 francs pour acheter un pain au chocolat

C : Pierre a un pain au chocolat

D : Pierre a effectué 5 minutes en vélo (trajet boulangerie-maison)

Si on introduit les opérateurs ⊗ assimilables à une conjonction (ET logique), –o l'opérateur inductif et + l'opérateur puis, on pourra formaliser le problème suivant la proposition (A⊗B) –o C et N.C –o D (avec N entier en logique linéaire). La deuxième proposition revient à dire que quelque soit le nombre de pains au chocolat achetés lors d'un passage à la boulangerie, le petit Pierre devra faire 5 min en vélo : On pourra imaginer les deux scénarii suivants, sachant que Pierre est très gourmand (c'est-à-dire qu'il va convertir son argent en pain au chocolat) mais paresseux (il n'aime pas le vélo).

Scénario 1 : Pierre reçoit cinq francs (à une date t1) puis encore cinq francs (à une date t2 ultérieure).

Scénario 2 : Pierre reçoit directement dix francs (à une date t1).

La logique linéaire permet de distinguer les deux scénarii (en supposant que le prix du pain au chocolat est un invariant du temps):

Scénario 1

!B

A puis A

A⊗B puis A⊗B (A⊗B) -o C

C puis C

D puis D

Le petit Pierre aura donc mangé deux pains au chocolat et fait 10 minutes de vélo (2 fois 5 minutes)

Scénario 2

!B

2A

2A⊗2B (A⊗B) -o C

2 C

D

Le petit Pierre aura donc mangé deux pains au chocolat et fait 5 minutes en vélos (1 fois 5 minutes)

Figure 10 : Exemple de raisonnement par séquents (logique linéaire)

On voit, dès lors l'intérêt pratique d'une approche de logique linéaire qui permet de prendre en compte l'ordre d'utilisation des propositions pour déterminer les conclusions des deux scénarii par approche descendante (séquents linéaires). Dans le cas d'une approche par logique classique, on aurait simplement formalisé le problème par la proposition (A ET B) => (C et D) ce qui aurait, certes, permis la résolution du problème, mais sous une forme moins intuitive. La logique linéaire, contrairement à la logique classique, ne se base donc pas sur des propositions dont la validité et surtout l'existence pourraient être considérées comme invariant du temps. Elle considère au contraire des relations d'ordre fortes entre les différentes propositions et une notion de disparition/apparition de proposition qui sous-tend l'idée de disponibilité.

Un parallèle peut être donc fait entre l'utilisation des ressources précédemment évoquées (matérielle, informationnelle) dans une vision de flux d'information et les concepts de précédence et de consommation introduits dans la logique linéaire. Dans notre problème, nous sommes intéressés par l'ensemble des combinaisons d'erreurs permettant, du fait de leur activation, d'arriver à la conclusion « mode de défaillance de la fonction de sécurité ».

Comme nous l'avons souligné dans la partie 1, cette conclusion dépend de l'état d'erreur des ressources matérielles sur la période d'acquisition e(k) considérée mais aussi des règles d'utilisation de celles-ci, et des perturbations qu'elles induisent sur les informations transmises dans l'architecture étudiée. La première idée consiste à interpréter l'état des ressources, état du signal rattaché à une information ou état de fautes, comme des propositions. On peut distinguer les propositions suivant leur caractère périssable et non périssable sur une période de référence T.

Les propositions portant sur l'état d'une information sont, par définition, périssables puisque l'information peut être consommée, c'est à dire utilisée puis détruite par une sous entité fonctionnelle, pour permettre la création d'une autre information, qui peut ou non conserver la même classe (F-, D-). Les propositions portant sur l'état erroné des informations qui transitent dans notre architecture opérationnelle sont périssables, puisque liées à l'existence de ces informations.

Les propositions portant sur l'état d'erreur matérielle sont elles non périssables sur un intervalle de référence T, c'est-à-dire qu'elles conservent leur propriété de validité (vrai ou fausse) sur cette période. En effet, les ressources matérielles qui servent à la transformation d'informations au cours du temps, sont supposés conserver le même état d'erreur sur chaque cycle de largeur T. On utilisera la notion d'exemplaire, introduite en logique linéaire, pour symboliser l'utilisation répétée d'une ressource matérielle:

« une abstraction permettant d'attribuer à différents objets une propriété équivalente ». ***[Gira97]***.

Dans le cas particulier, d'une utilisation répétée d'une ressource matérielle sur un même cycle par différentes sous fonctions, on considérera donc que le mode de défaillance présenté de cette ressource sera identique sur notre cycle d'observation T. Les propositions portant sur le caractère erroné de la ressource matérielle considérée seront donc des invariants du temps sur cet intervalle de référence.

Pour déterminer les listes La et Ld recherchées, une première idée consiste à modéliser notre architecture en une séquence d'opérations de traitement d'information. Nous pourrons ainsi identifier les combinaisons d'erreurs activées permettant d'atteindre une action en contradiction avec l'état de demande supposée.

Pour ce faire, on pourrait décrire pour chaque traitement d'information les combinaisons de défaillances locales (matérielles ou dues à des fautes environnementales) pouvant modifier ses lois entrées/sortie, c'est-à-dire entraîner la perturbation ou la perte de l'information traitée. On pourrait ainsi se ramener à une description du processus de transformation d'information sous la forme de séquents, comme proposé dans **[Demm02]**. L'intérêt d'une telle approche est qu'elle permet la prise en compte de deux phénomènes le long d'un flux d'information.

La perturbation d'information par une sous entité fonctionnelle induit l'utilisation de l'information perturbée en aval du point de perturbation, cette information pouvant éventuellement être partagée par deux sous entités fonctionnelles distinctes. On observera alors le résultat **de perturbations cumulées le long d'un flux d'information**, la perturbation d'une information pouvant être réversible le long d'un flux d'information.

La perte d'information par une entité sous fonctionnelle (perte de signal) constitue, a contrario, toujours un phénomène irréversible puisqu'il implique l'absence d'information en entrées des entités sous fonctionnelles suivantes le long du flux d'information. On parle alors d'héritage puisqu'une perte d'information induit l'absence de traitement d'information par une ou plusieurs séquences d'entités sous fonctionnelles.

Cependant, la construction d'un modèle s'appuyant sur des séquents linéaires est difficile, pour des systèmes fonctionnellement reconfigurables. Les règles comportementales de chaque entité sous fonctionnelle peuvent, en effet, varier non seulement en fonction de la présence d'erreurs sur les ressources matérielles mises en jeu mais aussi en fonction d'interprétation portant sur les informations d'entrées manipulées par ces entités sous fonctionnelles, indépendamment de leur état de validité (valide/erronée).

Nous ne proposerons donc pas, ici, de construire les listes La et Ld en nous appuyant sur une méthode d'exploration ascendante des séquents (logique linéaire), puisque celle-ci risque de poser le problème de conflit entre branches, dues aux contraintes de partage de ressources. Nous proposerons ainsi une méthode de construction des listes caractéristiques permettant la prise en compte implicite du concept de propagation d'informations et de la notion de conflits entre état d'erreur des ressources (« présence de propositions explicites et directement contradictoires » **[Deha00]**). Deux propositions seront donc en relation de conflits si « *il y a dans leurs consommés respectifs un exemplaire commun d'une ressource, que nous appellerons ressource critique pour la relation de conflit* ». Cette ressource pourra être soit matérielle, soit informationnelle.

Notre identification se base, par préférence, sur une décomposition fonctionnelle de l'architecture suivant des flux d'information. Le concept d'information est donc un niveau d'abstraction permettant la prise en compte des contraintes de précédence entre entités sous fonctionnelles induite par les choix architecturaux (logiciels et matériels). Elle nous permet de prendre en compte le partage d'information (partage de flux) et le partage de ressources matérielles (exemplaires) par différentes entités sous fonctionnelles. On peut rappeler que l'architecture fonctionnelle représentée inclut à la fois les fonctions de base, permettant la réalisation correcte de la fonction de sécurité en absence d'erreurs, mais aussi toutes les fonctions visant au contrôle de bon fonctionnement, à la détection d'erreurs et à la reconfiguration fonctionnelle de ces fonctions de base (système tolérant aux fautes).

Cette méthode s'appuiera sur un modèle de délivrance/consommation d'information entre entités sous fonctionnelles, permettant d'identifier les phénomènes d'initiation, de propagation et d'inhibition d'erreurs d'information et de perte d'information le long de ces flux et de juger de leur influence sur le comportement de la fonction de sécurité sur une fenêtre d'observation T fixée.

Elle permettra donc de déterminer, en accord avec l'architecture fonctionnelle et matérielle et l'état de demande au moment de l'acquisition, l'ensemble des **chemins possibles**, en termes de **combinaisons** d'erreurs simultanées activées, aboutissant à la propagation d'une information erronée jusqu'aux interfaces actionneurs. Cette propagation aboutit à un mode de défaillance de la fonction de sécurité.

La **défaillance sous demande** de la fonction de sécurité est équivalente à la propagation d'une information erronée ou à l'absence d'information au niveau de l'interface actionneurs alors que la présence de demande est avérée à l'instant d'acquisition. La liste

complète des combinaisons d'erreurs activées nécessaire et suffisante à ce phénomène est notée Ld.

Le **déclenchement intempestif** de la fonction de sécurité est équivalent à la propagation d'une information erronée au niveau de l'interface actionneurs alors que l'absence de demande est avérée à l'instant d'acquisition. La liste complète des combinaisons d'erreurs activées nécessaire et suffisante à ce phénomène est notée La.

1-4- Démarche générale

Nous proposons pour la construction des listes La et Ld une méthode systématique basée sur deux niveaux de représentation.

Le 1er niveau de représentation est une décomposition fonctionnelle de notre architecture. Il permet, en partant de la connaissance de l'architecture fonctionnelle de notre système d'étude, de modéliser l'architecture par un modèle de type flux d'information entre entités sous fonctionnelles. Ce modèle a pour entrée l'ensemble des paramètres observés, c'est-à-dire l'état de demande et l'état de faute des ressources matérielles testées en ligne à la période d'acquisition considérée comme fixée t= k.T, et pour sortie l'action effectuée au temps t+Tr (Tr étant le temps de réaction maximum autorisé du système et correspondant au temps de cycle auquel est ajouté le temps inertiel maximal des actionneurs).

Ce premier niveau, appelé modèle de haut niveau, permet de modéliser les contraintes de précédence entre entités sous fonctionnelles et le partage éventuel d'information entre différentes entités sous fonctionnelles. Il se base sur une représentation sous forme de schéma bloc, où chaque bloc symbolise une entité sous fonctionnelle. On introduira différentes classes d'entités permettant une représentation complète de l'architecture dans la partie 2. Une méthode de construction structurée de ce modèle de haut niveau, à partir de la connaissance du système, y sera, en outre, donnée et illustrée par un court exemple.

Le second niveau de représentation permet de modéliser le comportement fonctionnel et dysfonctionnel de chaque entité sous fonctionnelle (bloc du modèle de haut niveau) en lui associant un automate d'état fini. Nous définirons, dans la partie 3, certains ensembles d'états marqués, permettant de conclure sur l'existence et à l'état du signal délivré par l'entité. Nous classerons les événements de transitions de ces automates, en associant à chacune de ces classes d'événements constitutifs certains attributs, permettant de les décrire de manière générique et de définir certaines règles de manipulation de langages.

En nous appuyant sur ces deux niveaux de représentation et sur un certain nombre de règles élémentaires de manipulation de langages, nous proposerons, dans la partie 3, une méthode de construction itérative (par propagation de langage le long de la structure graphique définie par le modèle de haut niveau) permettant la construction complète de nos listes La et Ld (la propriété de complétude sera justifiée). Nous expliquerons comment cette méthode permet la construction complète de nos listes sans négliger la prise en compte implicite :

- des propriétés de précédence entre entités sous fonctionnelles
- des contraintes de partage d'information
- des contraintes liées au partage de ressources matérielles.

Les avantages de cette méthode par rapport aux méthodes de synchronisation classique de langage seront exposés. Nous expliquerons enfin de manière succincte, dans quelles mesures une automatisation de ce procédé de construction de listes est nécessaire. Nous expliquerons enfin comment ce besoin a été rempli par la mise en place d'un logiciel spécifique dont le développement en Java a été mené dans l'entreprise Siemens.

2- Modèle de haut niveau

2-1- Formalisme de haut niveau

La mise en place d'un formalisme de haut niveau, pour décrire l'architecture opérationnelle de notre fonction de sécurité doit remplir les objectifs suivants :

- Il doit permettre de décrire génériquement l'ensemble des flux de F- et de D-informations possibles dans l'architecture, le mécanisme de dégradation ou de restauration de leurs attributs (présence ou non d'erreur) et les mécanismes de décisions et de reconfiguration du système.
- Il doit être lisible et s'abstenir dans un premier temps de tout raffinement du mode de description des entités
- Il doit permettre la compréhension par différentes équipes du fonctionnement général de l'architecture
- Il doit s'appuyer sur un formalisme rigoureux (classes d'entités) et sur une méthode de construction systématique et vérifiable par un état des connaissances des procédures de traitement et de diagnostic de notre architecture.
- Il doit permettre d'utiliser dans un second temps la description à bas niveau de granularité de ces entités et permettre de manière univoque la construction de nos listes á partir de ces données et de règles de compositions clairement établies.

L'idée de modélisation d'une fonction par des flux de données (Data Flow) et par des flux de contrôle n'est pas nouvelle. Elle a été développée, en particulier, dans les techniques de génie logiciel **[Jaul90]** comme la modélisation SART introduite par Paul Ward et Stephen Mellor en 1984-85 **[Mell85]** puis étendue en 1986-87 par Pirbay-Hatley **[Pirb87]**. L'idée fondamentale reste celle de développer un modèle d'architecture fonctionnelle en se basant sur 4 concepts de bases :

- l'existence de flux d'information
- l'existence de flux d'événements de contrôle
- des procédures de contrôle
- des entités de stockage d'information.

Nous proposons de modéliser l'architecture fonctionnelle de notre système sous la forme de flux d'informations (F- ou D-information) entre des entités sous fonctionnelles (définition donnée en 1-2). Chaque flux d'informations, symbolisé par une flèche représente le transfert d'une information de sortie d'une entité sous fonctionnelle « origine » qui permet son émission à une ou plusieurs sous entités sous fonctionnelle « but » qui vont utiliser cette information comme entrée. Ce transfert s'effectue par une relation d'équivalence permettant d'affirmer que l'information délivrée par une entité sous fonctionnelle d'origine sera identique à celle utilisée par l'entité sous fonctionnelle « but ». Chaque entité sous fonctionnelle étant représentée par un bloc. De ce fait le schéma suivant, symbolise le transfert d'une information de sortie du bloc i (entité sous fonctionnelle i) vers une entité sous fonctionnelle j, où celle-ci servira de signal d'entrée.

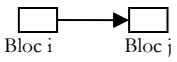


Figure 11 : Flux d'information entre blocs i

On appellera **flux d'information élémentaire** le transfert d'une information du bloc i au bloc j lui succédant directement. On supposera que le bloc j n'est en mesure d'effectuer un transfert

d'information (délivrance de signal de sortie) que si un signal d'entrée a été reçu et traité par celui-ci durant le cycle de traitement T considéré. On distinguera **trois types de flux élémentaire**:

- Les **flux élémentaires de F-information**, correspondant au transfert d'une F-information entre deux sous- entités fonctionnelles successives.
- Les **flux élémentaires de D-information**, correspondant au transfert d'une D-information entre deux entités sous fonctionnelles successives.
- Les **flux élémentaires de contrôle**, correspondent :

A- en l'absence d'émission d'une information par une entité, à la transmission à une procédure de contrôle (horloge) d'un événement « absence d'information », symbolisée par une flèche entre le bloc contrôlé et l'entité sous fonctionnelle de contrôle permettant de détecter cette « défaillance d'émission » (noté par la suite CT).

B- pour une ressource matérielle testée en ligne, à l'existence d'un état ou d'une combinaison d'erreurs matérielles détectable par ce type de test.

Ils seront symbolisés de la manière suivante :

⟶	Flux élémentaire de F-information
- - - - ▶	Flux élémentaire de D-information
-·-·-▶	Flux élémentaire de contrôle

Figure 12: Types de flux élémentaires et symboles

On appellera **flux d'informations**, toute séquence d'entités sous fonctionnelles **pouvant** être reliées par des flux d'information élémentaire (chemin). Nous acceptons donc qu'un flux de contrôle et qu'un flux d'information puissent provenir de la même entité sous fonctionnelle (l'existence de l'un exclura alors l'existence de l'autre). On aura alors le schéma (1) de la **Figure 13**. On acceptera aussi qu'un flux élémentaire d'information soit partagé entre différentes entités destinatrices (ici m et n). On parle alors de parallélisme sous fonctionnel. On symbolisera dans ce cas ce partage par un schéma de type (2a) dans le cas d'une F-information partagée et de type (2b) dans le cas d'une D-information partagée.

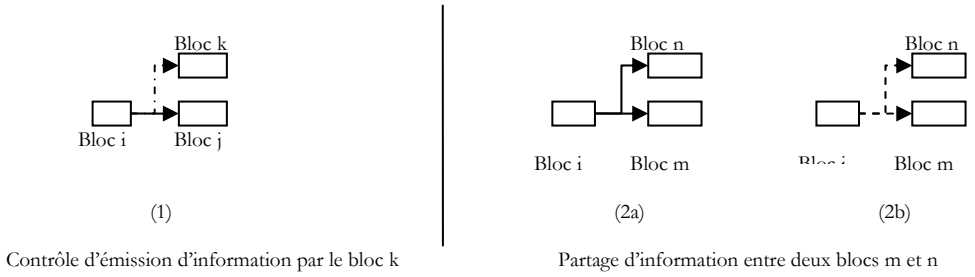


Figure 13: Contrôle de flux (1) et partage d'information (2a)-(2b)

On distinguera cinq classes de sous entités fonctionnelles, ou blocs, permettant de modéliser l'architecture du système étudié. Chaque entité sous fonctionnelle est symbolisée dans le modèle de haut niveau par un bloc contenant les initiales correspondantes à ces classes d'appartenance : **TF, SB, IP, CT, ST**. On ajoute à ces entités sous fonctionnelles, deux classes de blocs spécifiques, appelés bloc sources, et notées SRC1 et SRC2.

Les blocs SRC1 symbolisent les interfaces capteurs, permettant l'observation de l'état de demande (présence/absence de demande). Ils délivrent à chaque période d'acquisition une information, correspondant à l'état de demande observable par notre système. Cette F-information est par nature non erronée et est à la base du processus de décision de notre système

d'étude. Dans la suite (partie 3), on notera presD, l'événement présence de demande à l'instant d'acquisition et absD l'événement absence de demande à l'instant d'acquisition.

Les blocs SRC2 délivrent à chaque période d'acquisition une information, dont l'occurrence est liée sur la période T considérée à l'existence d'une combinaison détectable d'états d'erreurs matérielles par un test en ligne (D-information). Ils symbolisent l'observation par une procédure de test d'un sous-ensemble de ressources matérielles testées. Cette information est par nature non erronée et est à la base du processus d'auto test de ressources matérielles (elle sera toujours reliée à une entité sous fonctionnelle de test en ligne ST). On aura, dans la pratique, autant de blocs SRC2 que de tests de ressources matérielles en ligne indépendants (ces tests seront supposés périodiques). L'attribut de ce bloc est la combinaison des erreurs matérielles **détectables** (et non forcément détectées) par le test considéré, notée L1(SRC2). Ces entités SRC1 et SRC 2 sont modélisées de la manière suivante, dans le modèle de haut niveau proposé pour l'architecture fonctionnelle du système d'étude:

Figure 14: Type et schéma de connexion de blocs

Les entités sous fonctionnelles de classe **TF**, sont définies comme des entités de transformation du signal. Elles permettent la transformation d'une information d'entrée unique en une information de sortie unique par un changement de la nature du signal, tout en conservant leur classe (F- ou D-information). Elles peuvent contribuer à l'appauvrissement des données utilisables (seuillage, amplification, transport de données…) mais ont pour rôle principal de conserver leur signification pour tout processus de décision ultérieure. Elles **peuvent éventuellement** changer la nature du signal (électrique, mécanique…). Un exemple simple de ce cas de figure peut être donné par une sous fonction de contrôle transformant un ordre d'entrée de nature électrique (commutation d'un relais d'alimentation) en une action mécanique (fermeture de vanne). Tout flux de F-information se termine généralement par un bloc TF, symbolisant la transformation finale au niveau des actionneurs d'un ordre de contrôle en action. Ces blocs seront appelés blocs terminaux. On aura donc les possibilités de connexions suivantes:

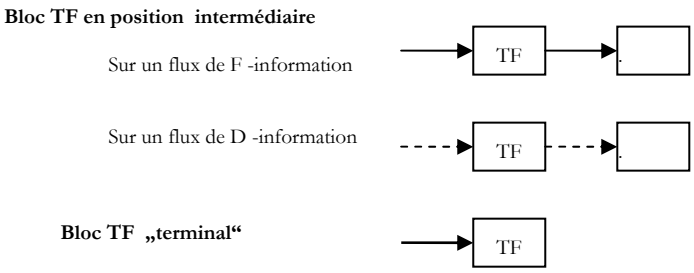


Figure 15: Type et schéma de connexion de blocs TF

Les entités sous fonctionnelles de classe SB sont définies comme des entités de stockage du signal. Elles permettent la mémorisation sur un support d'un signal d'entrée et sa restitution ultérieure. Ces entités peuvent permettre de représenter les procédés de mémorisation nécessaire à la synchronisation des entrées d'un processus de décision. Chaque entité de cette classe possède une information d'entrée et de sortie unique, et conservant la classe d'information (F- ou D-) entre information consommée et délivrée. Le signal de sortie de ce type de bloc sera toujours réutilisé par une autre entité sous fonctionnelle. On aura donc les possibilités de connexion suivantes:

Figure 15: Schéma de connexion de blocs

Les entités sous fonctionnelles de classe IP sont définies comme des entités de décision. Elles permettent de délivrer une information de sortie au regard de deux informations d'entrées. On pourrait donner, comme exemple, les processus de comparaison et de décision sur informations jugées redondantes, les processus d'acceptation de valeur en regard d'une information de diagnostic, … Cette classe d'entités est symbolisée par un bloc possédant **deux entrées** (flux d'information élémentaires entrant) et **une sortie unique** (flux d'information sortant). Dans la pratique, on peut remarquer que toute fonction de logique séquentielle portant sur ensemble fini de n F-informations et de p D-informations et ne s'appuyant pas sur des processus de mémorisation intermédiaire peut être divisée en une structure de bloc IP et SB interconnectés. Cette hypothèse est d'ailleurs corroborée par Rauzy pour justifier de la décomposition d'une structure fonctionnelle combinatoire en BDD **[Rauz99]**. Elle est vérifiée **si on suppose que le processus de décision sur la période T est déterministe et que l'état de demande observable** par les différents capteurs lors de cette période est **invariant**. On aura donc les possibilités de connexions suivantes:

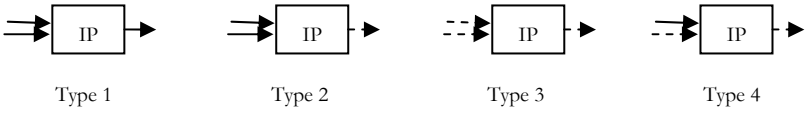


Figure 16: Types et schéma de connexion de blocs « décision »

- le montage de type 1 peut être utilisé pour symboliser le processus de décision, en regard des F-informations provenant de deux voies de traitement redondantes,
- le montage de type 2 peut être utilisé pour symboliser le processus de synthèse de diagnostic à partir de la comparaison de F-informations (par exemple entre voies redondantes)
- le montage de type 3 peut être utilisé pour symboliser le processus de fusion de diagnostic
- le montage de type 4 peut être utilisé pour symboliser le processus de décision sur une F-information au vu d'un diagnostic

Les **entités sous fonctionnelles de classe CT** (contrôle de flux) correspondent aux opérations de vérification de la réception d'une information en un temps raisonnable, permettant leur utilisation ultérieure. Ces entités sous fonctionnelles sont nécessaires pour symboliser les procédures de contrôle d'actualisation de données dans une architecture. Elles sont importantes puisqu'elles permettent de représenter la détection d'une absence de transmission d'information et l'émission d'une information de diagnostic utilisée par la suite à des fins de reconfiguration fonctionnelle, mise en mode dégradé de l'architecture… . Cette classe d'entités sous fonctionnelles permet de prendre en compte les procédés d'actualisation de données dans notre architecture (Watch Dog, bascule d'actualisation…). Cette entité possède pour entrée un flux élémentaire de contrôle et pour sortie un flux de D-information. On aura donc les possibilités de connexion suivantes:

Figure 17: Schéma de connexion de blocs CT (contrôle de flux)

Les **entités sous fonctionnelles de classe ST** (test en ligne) correspondent aux opérations de test en ligne, supposées périodiques, de ressources matérielles. Ces entités possèdent pour entrée un ensemble fini de modes de défaillances matérielles, sollicités lors du test, et jugés détectables par celui-ci. Le flux entrant peut donc être assimilé à une D-information provenant d'une source secondaire (voire paragraphe suivant). La sortie est une D-information (signal de diagnostic) pouvant être utilisée suivant les choix architecturaux retenus. Elles possèdent pour entrée un flux élémentaire de D-information, provenant d'une entité source de type SRC2 et comme sortie un flux de D-information. Leur schéma de connexion est donc le suivant :

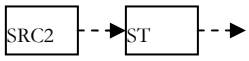


Figure 18: Schéma de connexion de blocs ST (test en ligne)

A partir de la définition de ces classes d'entités, il est possible de représenter l'architecture fonctionnelle du système étudié (fonction de sécurité), de manière complète, en termes de flux d'informations et d'événements. Cependant, cette construction du modèle dit de haut niveau requiert une bonne connaissance du système et doit s'appuyer sur une méthode précise pour éviter tout oubli. Nous proposerons une méthode de construction claire de ce modèle que nous illustrerons par un exemple concret sur une architecture relativement simple.

2-2- Construction du modèle de haut niveau sur exemple

Toute architecture fonctionnelle d'un système dédié sécurité est modélisable en utilisant les blocs élémentaires précédemment définis. En effet, elle peut être décomposée en entités sous fonctionnelle de transformation, de stockage, de test et de décision sous deux conditions

- une non évolution des paramètres observés sur un cycle T (hypothèse (**P2**) du chapitre 2)
- l'existence d'une relation logique entre l'état de demande observé et l'action à mener.

La première condition est, généralement justifiée, par le fait que le temps séparant l'acquisition des paramètres observés par les interfaces capteurs au début d'une même période e(k) est suffisamment court (largement inférieur à T), pour qu'en présence de redondance sur l'observation de l'état de demande (redondance informationnelle), cet état n'évolue pas entre les différents instants d'acquisition effectués au même cycle e(k)=[kT, (k+1).T[.

La seconde condition est remplie par la donnée des spécifications fonctionnelles de la fonction de sécurité.

Nous proposons d'expliquer, sur un cas d'étude simple, la construction progressive du modèle de haut niveau.

2-2-1- Présentation de l'exemple simple

L'exemple choisi provient d'un schéma d'arrêt d'urgence d'un réacteur thermique, correspondant à l'interruption de son alimentation en comburant. La fonction de sécurité doit entraîner l'arrêt du réacteur quand une température seuil, jugée comme dangereuse, notée T_{lim}, est dépassée. Cette température est mesurée au moyen de deux capteurs d'acquisition, notés respectivement C1 et C2, reliés par un bus de communication suivant le protocole maître esclave à un contrôleur (PLC, automate programmable). On suppose un protocole de transmission simple sans ajout de séquence de test (CRC ou Checksum) permettant de vérifier la non altération des signaux transmis lors de leur transport. Ces informations sont acquises à chaque T et stockées dans les registres d'entrée du contrôleur. En cas de non réception de ces informations au bout d'un temps limite de transmission, on attribue au registre d'entrée correspondant une valeur par défaut, correspondant à la valeur attendue en cas de détection effective du dépassement du seuil de température.

Chacune de ces informations est ensuite comparée au seuil de dangerosité T_{lim}. Dans le cas, où au moins une de ces valeurs dépasse cette limite, le contrôleur émet un ordre exigeant l'arrêt du réacteur. Cet ordre est transmis, via le même bus de communication, à deux circuits de coupures. La fermeture de ces relais a pour conséquence l'alimentation de deux moteurs commandant la fermeture de deux vannes V1 et V2 montées en série sur le circuit d'alimentation en comburant du réacteur. La fermeture d'une de ces vannes permet la mise en position sûre du réacteur. Le manque de comburant consécutif à cette fermeture tend à arrêter le fonctionnement du réacteur. On ajoute à cet ensemble de sous fonctions une procédure de test périodique d'une vanne (effectuée toute les 6h, par exemple) par sollicitation en fermeture. Dans le cas d'un échec de cette fonction de test (non fermeture de la vanne sous sollicitation), on prévoit une procédure exigeant la fermeture systématique des deux vannes. La demande de fermeture de deux vannes dans le cas d'une défaillance détectée de l'une d'entre elle est un procédé classique pour des architectures orientées sécurité qui vise à faciliter l'inspection ultérieure par une mise en position sûre de l'installation. Cet exemple est décrit par le montage matériel suivant (**figure 19**).

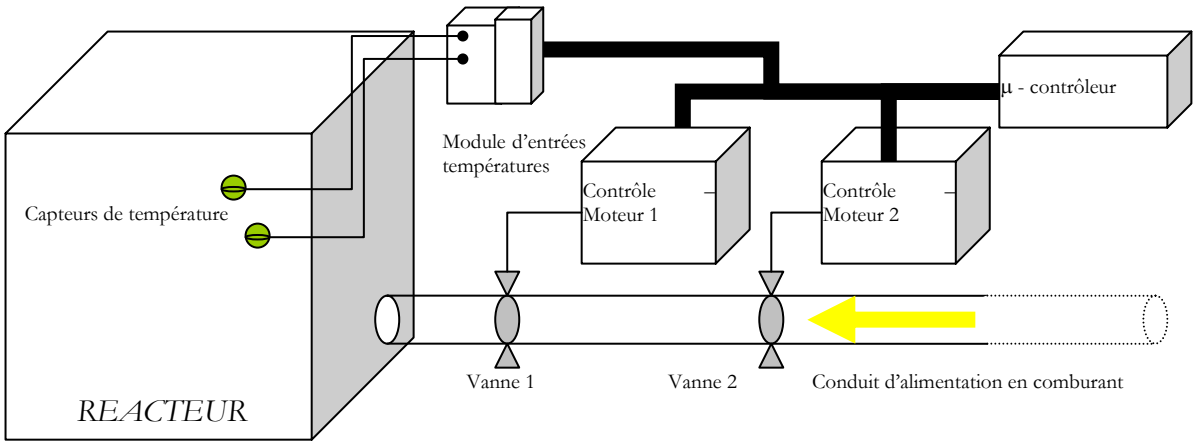


Figure 19 : Schéma de principe - Exemple simple – arrêt d'urgence de réacteur

2-2-2- Etapes de construction de notre modèle de haut niveau

La première étape de construction prend en compte les sous entités fonctionnelles de base. On représente donc les flux de F-information en partant du bloc source SRC1 et en arrivant aux blocs terminaux de classe TF. Il s'agit d'une description de l'architecture sans prise en compte des tests en ligne, ni des procédures de contrôle de bonne réception d'information (contrôle

d'actualisation de donnée). Sous cette hypothèse (notée **H4**), on obtient le schéma simplifié suivant. Chaque entité sous fonctionnelle est associée, dans la **figure 20**, à un numéro dont la description est donnée à titre indicatif en dessous du modèle:

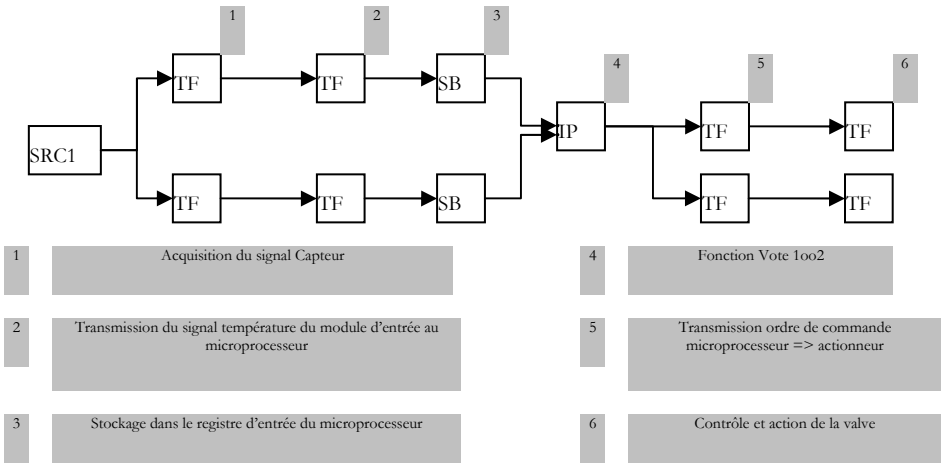


Figure 20: 1ère étape de construction sur exemple

On va ensuite vérifier que chaque flux de F-information décrit est complet, c'est-à-dire qu'aucune autre procédure n'utilisant de F-information n'a été omise, sous l'hypothèse simplificatrice **H4** et qu'il existe :

- Pour toute sous entité fonctionnelle décrite un flux de F-information (chemin de flux élémentaire) permettant le passage de cette entité à au moins une des entités TF terminales.
- Pour toute sous entité fonctionnelle décrite un flux de F-information permettant le passage d'un bloc SRC1à cette entité.

Une fois ces vérifications faites, nous pouvons passer à la seconde étape de construction.

La **seconde étape** de construction prend, cette fois, en compte la liste complète des procédures de test et de contrôle d'actualisation de données dans l'architecture. On ajoute en fait sur **la figure 20** les blocs sources SRC2, symbolisant l'existence de combinaisons de fautes matérielles testées au moyen d'un test en ligne, et on les connecte à une entité de classe CT, et aux procédures de contrôle. On vérifie ensuite qu'aucun test en ligne, ni procédure de contrôle d'actualisation (en général liées à des horloges internes de contrôleur ou à des horloges de bonne réception d'information transmises par bus), n'ont été oubliées (obtention de la **figure 21)**.

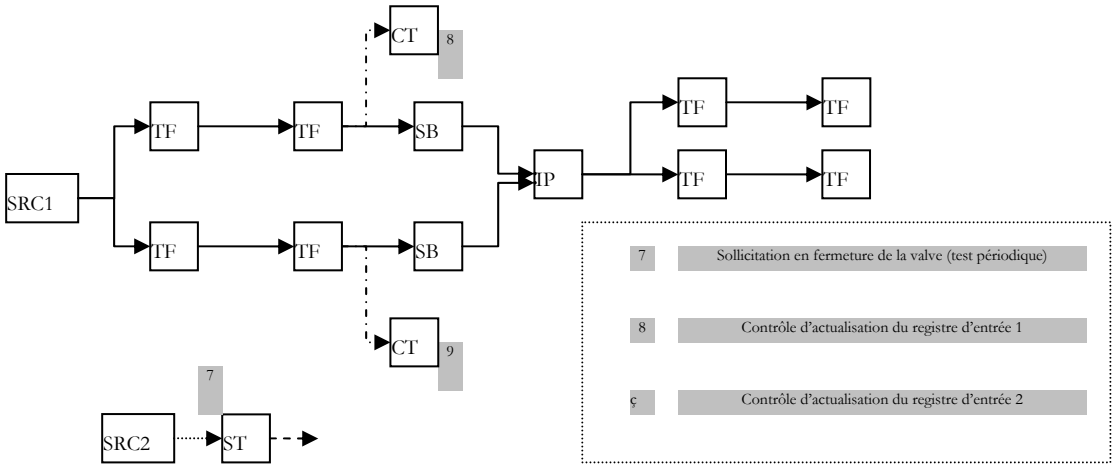


Figure 21 : 2ème étape de construction sur

Une fois cette étape achevée, on va compléter le modèle de haut niveau (**figure 21**) **en décrivant les flux de D-informations issues des entités de classe CT et ST**. Cette description se fait par l'ajout de blocs de classe TF (transformation du signal), SB (stockage) et éventuellement IP de type 3 (fusion de diagnostic).

On va ajouter ensuite sur le flux de F-information les blocs IP correspondant à la modification de la structure de base (donnée par l'étape 1) et symbolisant les propriétés de recouvrement et de modes dégradés. On obtiendra donc le modèle de haut niveau suivant (**figure 22**).

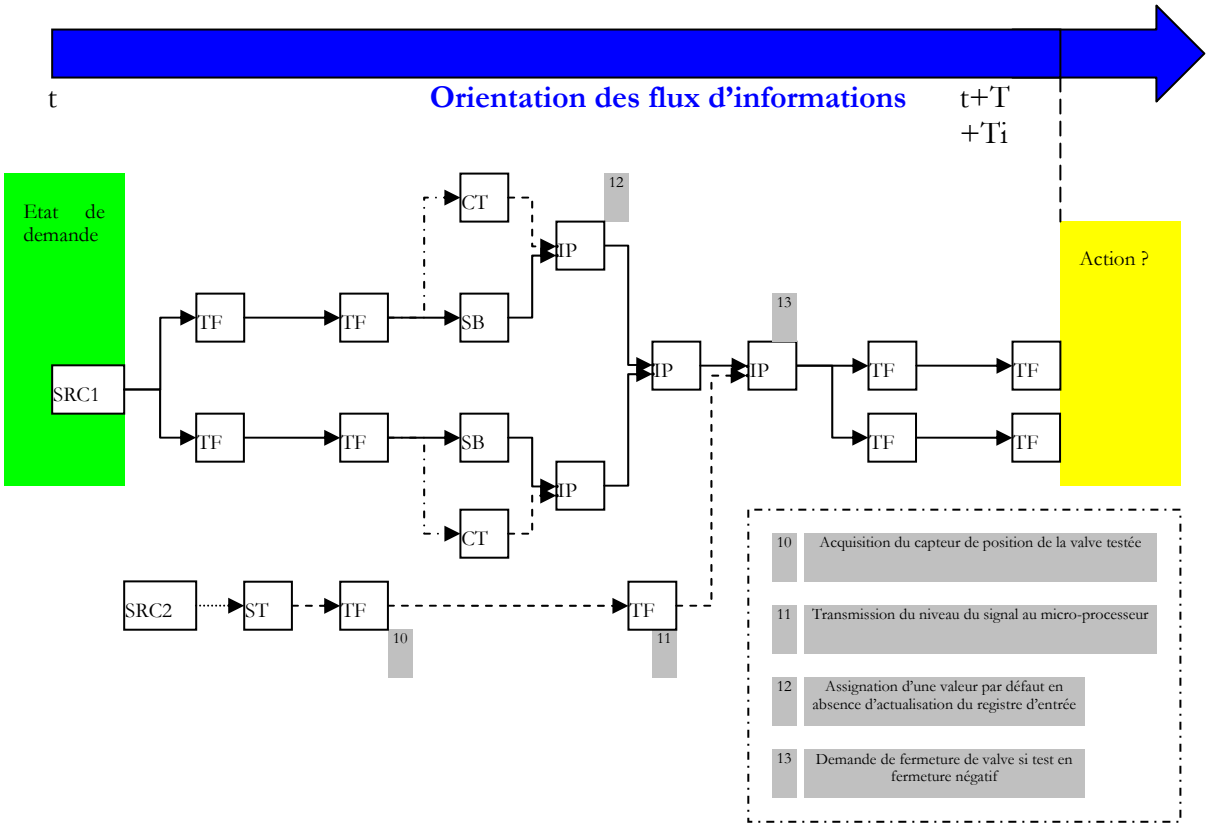


Figure 22 : Modèle de haut- niveau complet pour exemple

Ce modèle sera jugé complet si l'ensemble des propriétés de reconfiguration fonctionnelle et de modes dégradés y figure. Cette complétude devra être vérifiée par l'identification des mesures mises en œuvre en cas de diagnostic et de leurs conséquences locales en termes de reconfigurations. Nous avons ainsi obtenu en trois étapes distinctes de construction, en partant d'une architecture connue, le modèle de haut niveau associée à celle-ci (premier niveau de représentation).

La qualité de ce modèle est fortement dépendante d'une vue exhaustive des procédures de test, de contrôle et de reconfiguration du système. Son caractère progressif facilite cependant sa construction pour un ingénieur ayant une bonne connaissance du système et de ses capacités de reconfiguration. En outre, sa lecture est relativement aisée. On peut prévoir éventuellement la mise en place de librairies de séquences de blocs pour permettre un gain de temps dans l'élaboration du modèle de haut niveau (notamment pour la prise en compte de procédures de décision de type vote majoritaires 2oo3 de plus en plus utilisées dans les architectures dédiées sécurité dans le monde industriel).

La construction « systématique » du modèle de haut-niveau dépend de la vérification d'une identification complète des flux d'information dans l'architecture. La validation du modèle de haut-niveau peut être effectuée par l'identification complète des informations échangées par l'architecture dédiée sécurité, en incluant les procédures de test, de contrôle temporel et de recouvrement.

3- Modèle de bas-niveau

Le 2nde niveau de représentation est un niveau de granularité plus faible. Il doit permettre de décrire le comportement fonctionnel et dysfonctionnel de chaque entité sous fonctionnelle. Nous avons décidé d'utiliser pour ce faire un automate à état fini pour la représentation comportementale de chaque sous entité fonctionnelle. Après un court rappel sur les automates d'états finis et uneexplication sur leur construction, nous introduirons pour chaque événement modélisé une classe d'appartenance et des attributs spécifiques permettant de décrire de manière synthétique ces automates d'états finis et d'utiliser les langages générés pour la construction de nos listes finales La et Ld.

3-1- Automate à états finis et langages marqués

Pour chaque entité, nous nous proposons d'associer un graphe d'événements de dimension finie. Cette description est cohérente avec la définition de notre entité sous fonctionnelle, définie comme un procédé de traitement de signal, basé sur l'utilisation séquencée de ressources matérielles. Le fait que ces règles soient conditionnées par l'état du signal d'entrée nous permet la prise en compte des événements de réception d'un signal d'entrée comme événements initiateurs, c'est-à-dire les événements permettant de quitter l'état « au repos » de l'entité sous fonctionnelle considérée.

Avant d'expliquer la méthode de construction des automates d'états finis décrivant le comportement de chaque sous entité fonctionnelle, un rappel s'impose sur les terminologies de base et les abus de langages que nous nous autoriserons dans notre étude. Du fait que nous ne nous intéresserons qu'à un nombre très faible de notions portant sur ces automates, nous ne détaillerons pas un certain nombre de concepts utilisés dans le domaine de la supervision des SED (systèmes à événements discrets). On pourra se référer pour avoir un panorama plus global à l'ouvrage de référence **[Cass99]**.

On appellera automate d'état fini un quintuplet $A = (X, Y, f, x_0, Xf)$ dans lequel :

- X est un ensemble fini d'états ;

- Y est un ensemble fini d'événements,
- f: X × Y → X est la fonction de transition,
- x_0 est l'état initial
- Xf ⊂ X est l'ensemble des états finaux (ensemble fini)

Généralement, on représente un automate d'états finis par un graphe orienté étiqueté. Un automate d'états est un moyen de représentation de la notion de langage. Un langage est un ensemble de mot. Chaque mot est une suite de lettres. Chaque lettre appartient à un alphabet. Dans la théorie des SED, l'alphabet est l'ensemble des événements possibles, les mots deviennent des séquences d'événements. On utilisera, par la suite, les notions de base suivantes :

Mot = « séquence ordonnée et finie d'événements », notée en général par l'abréviation générique m

Langage fini = « ensemble fini de mots, équivalent à une liste de longueur finie de mots »

On rappelle que l'opérateur qui permet de construire les mots à partir des événements est l'opérateur de concaténation, noté . (et par abus sous-entendu), dont l'élément neutre est noté ε.

On rappelle, enfin, que le langage marqué associé à un automate d'état fini déterministe, noté L_m (A) est donné par l'ensemble des mots associés au **chemin** permettant le passage de l'état initial x_0 à un état de l'ensemble marqué Xf. Il est donc constitué de l'ensemble des chemins d'accessibilité de x_0 aux éléments de Xf.

L'ensemble des états marqués d'un automate d'état fini est, en général, utilisé pour donner les séquences d'événements menant à un ensemble d'états dont les propriétés sont jugées intéressantes. Il est tout à fait possible dans le cas où plusieurs classes d'états accessibles nous intéressent d'utiliser une partition de l'espace des états marqués X_f en un nombre fini (p≥1) d'ensemble d'états marqués $X_{f,1}$; $X_{f,2}$;… $X_{f,p}$ vérifiant :

$$\forall (i;j) \in \{1;2;...;p\}^2, \qquad (i \neq j) \Rightarrow X_{f,i} \cap X_{f,j} = 0 \quad et \quad \bigcup_{i=1..p} X_{f,i} = X_f$$

Si une telle partition est donnée, on notera $\mathbf{L_{m,i}}$ comme l'ensemble des mots (correspondant aux séquences d'événement associés) permettant le passage de l'état initial x_0 à un état de l'ensemble marqué $X_{f,i}$. On peut remarquer sans aucune difficulté que, dans ce cas, l'existence d'un même chemin appartenant à deux de ces langages marqués n'est pas possible dans la pratique, puisque ce chemin ne peut exister que si l'automate est non déterministe (**Proposition 4)**

Pour **en revenir à notre problème de base**, nous nous proposons de décrire le comportement de chaque entité sous fonctionnelle par un automate d'états finis déterministe. On notera A_i= (X(i), T(i), f(i), x_0(i), Xf(i)), l'automate d'état fini associé à la sous entité fonctionnelle i. Cet automate d'état fini aura pour état initial x_0(i).

Le graphe d'événements associé à chaque entité sous fonctionnelle est construit suivant trois étapes. En partant de l'état initial x_0(i), on va d'abord décrire l'ensemble des règles d'utilisation de ressource suivant l'état du signal d'entrée.

En partant du profil de fonctionnement « normal » (absence de fautes), on va construire un automate d'état fini dont les nœuds correspondront à l'utilisation d'une ressource matérielle pour une opération de traitement d'information. Les événements de transitions sont alors des événements liés au transfert d'un signal entre ressources matérielles. On ajoutera deux états finaux permettant de symboliser l'émission effective d'un signal de sortie. Ces états finaux, notés $x_{f,1}$(i) et $x_{f,2}$(i), correspondront respectivement à l'état de repos de l'entité sous fonctionnelle suite

à l'émission d'une information de sortie correspondant aux réponses A (présence de demande) et B (absence de demande) fournies en absence de défaillance (cf. définition **chapitre 2, 1-2**).

Pour chaque état de ce graphe, correspondant à l'utilisation d'une ressource matérielle, on va introduire dans le graphe une transition associée à l'événement défaillance de cette ressource. Cette transition mène à un autre état existant (erreur de traitement) ou à ajouter, dont les sorties devront être définies.

Pour prendre en compte la perte éventuelle du signal, on ajoutera un état « puits » à l'automate d'état, noté $x_{f,3}(i)$, dont l'accès sera lié soit à des défaillances matérielles de mode I (cf. terminologie chapitre 2,2-2), soit à un événement « échec de test » (dans le cas spécifique d'une sous entité fonctionnelle de classe ST).

On obtiendra ainsi un graphe orienté **acyclique** $G(i)=\{X(i),T(i),f(i)\}$ permettant de symboliser les opérations de traitement et les possibilités de perturbation et de perte du signal. Si on note $X_f(i) = \{ x_{f,1}(i), x_{f,2}(i), x_{f,3}(i)\}$, on peut aisément affirmer que ce graphe est à la fois accessible et co-accessible, c'est-à-dire qu'il existe pour tout état x de ce graphe au moins un chemin d'accessibilité permettant de passer de $x_0(i)$ à x et qu'il existe au moins un chemin permettant d'aller de x à un des éléments de $X_f(i)$.

En outre, puisque $X_{f,1}(i) = \{ x_{f,1}(i)\}$, $X_{f,2}(i) =\{ x_{f,2}(i)\}$, $X_{f,3}(i) = \{ x_{f,3}(i)\}$ constituent une partition de $X_f(i)$ et qu'il n'existe aucune séquence d'événement permettant le passage de l'un à l'autre de ces ensembles, on pourra affirmer par contraposé de (**Proposition 4)** que

$$\forall(u,v)\in\{1;2;3\}^2, L_{m,u}(A(i))\cap L_{m,v}(A(i))=0$$

On notera dans toute la suite de notre travail : $L_a(i) = L_{m,1}(A(i))$; $L_b(i) = L_{m,2}(A(i))$; $L_c(i) = L_{m,3}(A(i))$. On peut aisément vérifier que ces trois langages forment une partition de $L_m(A(i))$.

On sait que les événements de transfert de signaux entre deux états de l'automate sont des événements certains en absence d'erreurs. C'est pourquoi on les remplace par l'événement neutre ε. La connaissance de ces événements n'est pas nécessaire pour la détermination des listes Ld et La recherchées. Celle-ci est cependant importante pour ne pas oublier de mots dans le calcul des langages marqués de l'automate. On les conservera donc dans ce calcul.

Une fois l'automate d'état fini construit, il est très rapide de déterminer les langages $L_a(i)$; $L_b(i)$; $L_c(i)$ par une procédure itérative récursive permettant de remonter des états marqués à l'état initial. La structure d'arbre de ces automates d'états finis simplifie leur parcours et ne présente aucune difficulté particulière.

3-2- Construction de modèle de bas niveau et automate minimal

Le procédé de construction de l'automate d'état fini d'une entité sous fonctionnelle peut amener à un nombre important de nœuds intermédiaires, si on considère l'automate d'état fini complet. On dira qu'un état u est équivalent à un état v, si les événements d'entrée permettant d'y entrer et les événements de sorties permettant de le quitter sont identiques (mêmes événements d'entrée et de sortie), et si chacun de ces événements de sortie identiques mène à un état commun. On pourra, dans ce cas, fusionner ces deux états. On arrive de ce fait à un automate d'état fini équivalent dont la dimension est inférieure. Pour toute entité sous fonctionnelle, on ne considérera donc que l'automate d'état fini équivalent de dimension minimale[5].

Pour toute entité sous fonctionnelle i, on pourra donc effectuer le calcul de $L_a(i)$; $L_b(i)$; $L_c(i)$ à partir de cet automate minimal. Les mots constituants ces langages marqués contiendront

[5] En général, on cherchera toujours à construire l'automate de dimension minimale. La construction progressive de notre automate fini ne permet pas d'assurer que l'automate obtenu le soit, on peut donc le vérifier et le cas échéant réduire l'automate. (mais ce n'est pas impératif car théoriquement, ces automates doivent marquer le même langage)

donc six classes d'événements, que nous allons détailler et pour lesquelles une notation générique sera proposée.

3-3- Classes d'événements constitutifs et explication

Pour chaque automate d'états finis, décrivant le comportement d'une entité sous fonctionnelle (bloc), on distinguera, par la suite, six classes d'événements distinctes. Ces classes vont nous permettre par la suite de construire itérativement nos listes Ld et La à partir de la structure du modèle de haut niveau, tout en prenant en compte des problèmes de conflits entre ressources, existants dans l'architecture étudiée. Ces classes permettront, en outre, une saisie rapide des événements constitutifs de chaque automate fini par la donnée de la classe d'appartenance de ces événements constitutifs et des attributs leur étant rattachés.

Les 6 classes d'événements sont les événements initiateurs, finaux, d'activation d'erreur matérielle (transient event), d'activation d'erreur environnementale (transient fault), d'« échec de test » (test fault) et les autres événements correspondants à la transmission de signaux internes à l'entité sous fonctionnelle (other). Ces classes seront notées par la suite, Init, Fin, d, bf, tf, other.

3-3-1- Evénements initiateurs, intermédiaires et événements finaux.

Les événements initiateurs correspondent à la réception par l'entité sous fonctionnelle d'un signal d'entrée. Ce signal provient de l'ensemble des entités sous fonctionnelles directement en amont de celle considérée. On distinguera quatre cas :

Si l'entité est de classe TF ou SB, les événements initiateurs correspondront à la réception d'une information d'entrée par cette entité.

Pour une F-information entrante, l'événement initiateur sera dit de type True, si l'information reçue est identique à celle qui serait reçue en présence de demande et en absence de défaillance du système. Pour une D-information entrante, l'événement initiateur sera dit de type True, si l'information reçue est identique à celle qui serait reçue en présence d'erreur détectable et en absence d'autres défaillances dans le système.

Pour une F-information entrante, l'événement initiateur sera dit de type False, si l'information reçue est identique à celle qui serait reçue en absence de demande et en absence de défaillance du système.

Si l'entité est de classe ST, on aura un seul événement initiateur correspondant à l'existence d'une combinaison d'erreurs détectables par l'entité de test, on le notera Init(True).

Si l'entité est de classe CT, on aura un seul événement initiateur correspondant à la détection de la non réception d'une information par une entité CT, on le notera Init(True).

Si l'entité est de classe IP, on numérotera ses deux entités A et B, et on les distinguera suivant la figure suivante (convention de raccordement graphique) et ce quelque soit la nature des flux entre ces blocs (F- ou D-) et la classe des blocs amonts (A et B) connectés au bloc IP considéré (**figure 23**).

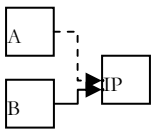


Figure 23 : Schéma de connexion du bloc IP

On aura au plus quatre événements initiateurs (11/12/21/22) correspondant à l'état de l'ensemble des deux informations reçues par la sous entité fonctionnelle.

- l'événement Init(11) correspondra à la réception de deux événements Init(True) provenant de chaque bloc amont (A et B) du bloc IP.
- l'événement Init(12) correspondra à la réception d'un événement Init(True) provenant du bloc A et d'un événement Init(False) provenant du bloc B.
- l'événement Init(21) correspondra à la réception d'un événement Init(False) provenant du bloc A et d'un événement Init(True) provenant du bloc B.
- l'événement Init(22) correspondra à la réception de deux événements Init(False) par chaque bloc amont (A et B) du bloc IP.

Les événements **finaux** correspondent à l'émission par l'entité sous fonctionnelle d'une information de sortie. On distinguera l'événement Fin(True) qui conduit à un état de $\mathbf{X_{f,1}}$ et l'événement Fin(False) qui conduit à un état de $\mathbf{X_{f,2}}$.

Les événements de la classe **«intermédiaires» (noté « Others »),** correspondent à la transmission de signaux dans la sous entité fonctionnelle. Ils incluent tous les signaux transmis autres que ceux correspondants aux événements initiaux et finaux. Ils seront notés par un caractère simple (a,b,c,d…). Leur importance réside dans la détermination de l'ensemble des chemins d'accessibilité aux états marqués, ils contribueront donc à la construction des listes finales La et Ld même s'ils ne figurent pas dans celles-ci.

3-3-2- Evénements activation d'erreur matérielle, d(X,Y)

Les événements **« activation d'une erreur matérielle »** correspondent à la possibilité d'activation d'une erreur matérielle. Ces événements sont induits **à l'existence d'une erreur matérielle** lors de l'utilisation de la ressource associée. Ils seront notés sous la forme générique d(X,y). La première lettre d, signifie l'appartenance de l'événement à cette classe. Le premier attribut X correspond au nom de la ressource matérielle erronée (souvent donné sous la forme xi avec i le numéro de la ressource matérielle >0). Le second attribut Y correspond à l'état d'erreur de la ressource matérielle, il appartient donc au quadruplet {0,S,D,I}. L'activation de l'erreur se traduit par une défaillance matérielle de mode X de la ressource utilisée. Le mode 0 étant le mode de fonctionnement normal. L'utilisation d'une ressource matérielle Xi dont l'état est normal est donc notée d(Xi,0).[6]

Par extension, on fera aussi entrer dans cette classe les événements présence et absence de demande, notés respectivement **d(xo,S)** et **d(xo,D)**. En effet, ces événements correspondent à l'émission d'un signal (activation) suite à une observation. Ils ne peuvent donc pas être vus comme la conséquence du comportement d'une entité sous fonctionnelle. Ce choix est justifié par le traitement apporté aux conflits de ressources matérielles. On peut remarquer que l'état d'erreur d'une ressource et l'état de demande observé sont considérés comme un invariant du temps sur une période T.

3-3-3- Evénements activation d'erreur environnementales bf(i) et événement « échec de test » tf(i)

Les événements **« activation d'une erreur environnementale »** correspondent à la possibilité d'activation d'une erreur environnementale dont l'existence est liée à une faute environnementale (cf. typologie chapitre 2). Ils seront notés sous la forme générique bf(ei), où ei est la faute environnementale et bf signifie l'appartenance à cette classe d'événement. Le nom « bf » est

6 Rappel : les événements d(xi,0) : utilisation de la ressource xi dans un état correct ne doivent pas être oubliés, notamment pour une bonne utilisation de l'opérateur croisement et réduction.

dérivé des erreurs dites de bit-flip dans la littérature anglo-saxonne dont l'étude en termes de conséquence est souvent effectuée par injection de fautes sur des circuits intégrés ou des blocs mémoires **[Chen00]**.

Les événements **« échec de test»** correspondent enfin à la non détection par un test en ligne sur une période suffisamment courte d'une combinaison d'erreurs détectables et même en l'absence d'erreurs sur les ressources de cette procédure de test.. Cet échec induit la non émission d'un signal diagnostic lors du cycle e(k) où l'existence de cette combinaison d'erreur est avérée. Ces événements seront notés **tf(i)** avec i le numéro du test mis en échec. Ils seront uniquement utilisés dans les automates finis de classe ST, les autres entités sous fonctionnelles, n'étant pas sujettes à ce type de défaillance.

3-3-4- Exemple d'automate fini pour un bloc TF

Pour illustrer la définition de ces classes, on propose ci-dessous un exemple d'automate d'état fini associé à un bloc TF, noté i, d'acquisition de signal (sur un flux de F-information). Le comportement de cette entité est donné par le graphe suivant (**figure 24**)

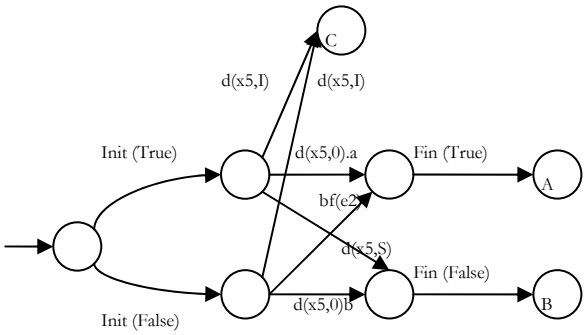


Figure 24 : Exemple d'automate d'état fini pour un bloc TF

On peut remarquer directement par la donnée du graphe que $X_{f,1}(i)=\{A\}$, $X_{f,2}(i)=\{B\}$, $X_{f,3}(i)=\{C\}$. On en déduit les langages marqués suivants :

$$L_a(i) = \{Init(True).d(x5,0).a.Fin(True)\ ;\ Init(False).d(x5,S).Fin(True)\}$$

$$L_b(i) = \{\ Init(True).bf(e2).Fin(False)\ ;\ Init(False).d(x5,0).b.Fin(False)\}$$

$$L_c(i) = \{Init(True).d(x5,I);\ Init(False).d(x5,I)\}$$

La première séquence (appelé mot) de La(i), Init(True).d(x5,0).a.Fin(True) n'est constituée d'aucun événement de classe d, bf ou tf. On peut donc affirmer qu'il s'agit de la séquence d'événements rencontrée pour un fonctionnement normal de l'entité sous fonctionnelle, lorsque l'information d'entrée du bloc TF est non erronée et sous l'hypothèse d'une présence de demande au moment de l'acquisition.

Le second mot de La(i), Init(False).d(x5,S).Fin(True), est constitué d'un événement de classe « activation d'erreur matérielle ». On peut donc affirmer qu'il appartient à un mode de défaillance de l'entité sous fonctionnelle du fait de la contradiction des attributs entre l'événement de classe Init et l'événement de classe Fin constituant cette séquence. Ce mode de défaillance est la conséquence directe de la défaillance de mode S de la ressource x5 lors de son utilisation. Il peut se traduire lorsque l'information d'entrée du bloc TF est non erronée et sous l'hypothèse d'une absence de demande au moment de l'acquisition, à l'émission d'une information de sortie erronée de ce bloc TF (demande de mise en position sûre par le bloc TF). On parle alors d'initiation d'erreur par le bloc TF. Il peut aussi se traduire lorsque l'information d'entrée du bloc TF est erronée et sous l'hypothèse d'une absence de demande au moment de l'acquisition, par

l'émission d'une information de sortie non erronée par ce bloc TF. On parle dans ce second cas, d'inhibition d'erreur le long d'un flux d'information. Ce cas de figure est caractéristique des systèmes non cohérents.

Après cette courte illustration nous allons souligner l'intérêt possible de l'introduction de telles propriétés (classes d'appartenance et attributs), nous expliquerons par la suite les opérations nous permettant à partir de la connaissance du modèle de haut niveau et d'une telle description comportementale (automate d'état fini à propriétés) comment il est possible de construire de manière systématique et complète les listes La et Ld recherchées.

3-3-5- Intérêt de l'introduction de ces classes d'événements

Nous proposons d'utiliser le modèle de haut niveau pour construire de manière itérative et complète les listes La et Ld qui correspondent au résultat de la propagation d'une information erronée jusqu'aux interfaces actionneurs (modélisées par les entités sous fonctionnelles TF terminales).

Contrairement aux approches par synchronisation **[Cass99]** qui risquent de mener à une explosion du nombre d'états de l'automate d'état fini en résultant, nous proposons d'utiliser les propriétés (classes et attributs) des événements de chaque bloc et les propriétés de connexion de ces blocs pour propager des langages le long de cette structure et les réduire successivement en prenant en compte les conflits apparaissant entre les erreurs activées.

Nous expliquerons dans la partie suivante les notions de propagation et de réductions de langages, nécessaires à la construction de nos listes finales, avant d'expliquer comment nous avons automatisé ce processus de construction.

4- Processus de construction des listes La et Ld

Dans cette partie, on prend comme point de départ pour l'élaboration de nos listes La et Ld, la construction préalable du modèle de haut niveau correspondant à l'architecture fonctionnelle du système étudié et celle des automates d'états finis associés à chacune de ses entités sous fonctionnelles (bloc).

4-1- Propagation de langages – principe et opérations de base

Comme nous l'avons souligné, une défaillance du système d'étude peut être considérée comme le résultat des mécanismes d'initiation, de propagation et d'inhibition d'informations erronées le long de flux d'information. Il semble donc légitime pour construire les listes Ld et La d'utiliser une approche, permettant de générer certains langages au niveau d'un bloc et de les propager vers les blocs avals. Cette procédure répétée de proche en proche en partant des blocs sources et en allant jusqu'aux blocs terminaux va nous permettre de déterminer de manière complète les chemins (séquences d'événements) induisant un mode de défaillance du système d'étude. On l'appellera **propagation de langage**, puisque les séquences d'événements sont « propagées » de bloc en bloc suivant les flux élémentaires d'information du modèle de haut niveau et des règles liées aux classes de blocs interconnectés.

Pour chaque entité sous fonctionnelle i, on va associer trois listes, notées L1(i), L2(i) et L3(i). Ces listes peuvent contenir des mots, constitués des événements précédemment décrits dans les automates d'états finis. On suppose qu'au début du processus de propagation, ces listes sont toutes vides, exceptées :

- les listes L1 des blocs de classe SRC1 qui contiennent le mot {d(xo,S)} correspondant à la présence d'une demande au début du cycle d'acquisition.

- les listes L2 des blocs de classe SRC1 qui contiennent le mot {d(xo,D)} correspondant à une absence d'une demande au début du cycle d'acquisition.
- les listes L1 des blocs SRC2 qui correspondent aux séquences (mots) d'erreurs matérielles détectables par la procédure de test en ligne (blocs ST) à laquelle chacun de ces blocs SRC2 est relié.

On dira qu'un bloc est « agrégé » lorsque celui-ci contient au moins une liste L1, L2, L3 non vide et qu'il possède au moins un bloc lui étant connecté en aval (suivant un flux élémentaire d'information) pour lequel toutes ces listes sont vides. Au début du processus de propagation des langages, seuls les blocs de type SRC sont donc « agrégés ».

Les listes L1, L2, L3 seront interprétés comme les séquences d'événements de classe d, bf et tf (incluant les événements présence et absence de demande d'après 2-5-2), permettant d'affirmer respectivement pour l'entité sous fonctionnelle considérée :

- que l'information de sortie émise est identique,

- soit à celle qui le serait en absence de défaillance ou d'événement échec de test dans le système en présence de demande dans le cas où cette information est une F-information,

- soit à celle qui le serait en présence d'erreurs détectables par la procédure de test et en absence d'autre défaillance du système dans le cas d'une D-information,

- que l'information de sortie émise est identique à celle qui le serait en absence de demande et en absence de défaillance ou d'événement échec de test dans le système,
- qu'aucune information de sortie ne sera émise par cette entité sous fonctionnelle.

La procédure de propagation de langage est basée sur l'héritage de la propriété bloc agrégé d'un bloc amont à un bloc aval. Nous allons distinguer deux cas de figure.

Dans le premier cas (cas série), un « bloc agrégé » est connecté en série à un bloc « non agrégé », ne possédant pas d'autre flux d'information entrant. Le premier bloc (agrégé) peut être de classe IP, SB, SRC, CT, TF, ST et le bloc aval peut être de classe SB, ST, CT, TF. On notera A et B ces deux blocs, suivant la **figure 25.**

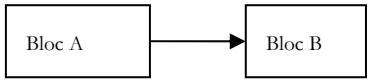


Figure 25 : Cas série

Dans le second cas (cas croisement), un « bloc agrégé » est connecté en série à un bloc « non agrégé », possédant un autre flux d'information entrant. Le premier bloc (agrégé) peut être de classe IP, SB, SRC, CT, TF, ST et le bloc aval peut être uniquement de classe IP. On notera A et B les deux blocs reliés au bloc « non agrégé » et C le bloc IP « non agrégé », suivant la figure suivante :

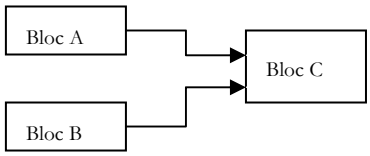


Figure 26 : Cas croisement

4-1-1- Premier cas : agrégation de bloc série (figure 25)

Dans le premier cas (cas série), la détermination de L1, L2 et L3 du bloc B, notée L1(B), L2(B),L3(C) est possible dès que la détermination de L1,L2 et L3 de son bloc amont (bloc A), notées L1(A), L2(A),L3(A) a été effectuée. On rappelle que le bloc B n'est pas de classe IP.

Pour ce faire, on calcule d'abord les langages marqués $L_a(i)$, $L_b(i)$ et $L_c(i)$ du bloc B. Ces langages marqués contiennent des événements pouvant appartenir aux 6 classes proposées dans la partie 2-5, ils sont mémorisés dans les listes **$L'_1(i)$, $L'_2(i)$** et **$L'_3(i)$**, définis comme vecteur de vecteur pour conserver la structure liste/mot/événement. Chaque mot de ces langages commence par un événement de classe Init(True) ou Init(False). On va distinguer alors deux cas de figure :

- soit le bloc B est de classe ST, TF ou SB, dans ce cas, on pose $L_{1,init}$ (B,true) =L1(A) et $L_{1,init}$ (B,false) =L2(A)
- soit le bloc B est de classe CT, dans ce cas, on pose $L_{1,init}$ (B,true) =L3(A)

On détermine ensuite les langages **$L''_1(i)$, $L''_2(i)$** et **$L''_3(i)$.** Pour tout $j\in\{1,2,3\}$, le langage $L''_j(i)$ est constitué de l'ensemble des mots distincts, obtenus par concaténation à gauche des mots de $L'_j(i)$ commençant par l'événement Init(true) avec chaque mot de $L_{1,init}$ (B,true) et de l'ensemble des mots distincts obtenus par concaténation à gauche des mots de $L'_1(i)$ commençant par l'événement Init(false) avec chaque mot de $L_{1,init}$ (B,false).

Pour plus de clarté, nous allons illustrer cette opération par un exemple portant sur la détermination de $L''_1(i)$ en supposant, que le bloc i est de classe TF, que les listes L1(A), L2(A) et L3(A) sont connues et que l'automate d'état fini décrivant le comportement i est donné par l'exemple de la figure 24. On suppose que:

$$L1(A) = \{d(xo,S),\ d(xo,S)d(x5,D)bf(e5),\ d(xo,D)d(x5,S)\},\quad L2(A) = \{d(xo,D),\ d(xo,S)d(x5,D)\}$$

$$et\ L3(A)=\{\ d(xo,S)d\ (x5,I),\ d(xo,D)d\ (x5,I)\}$$

D'après les langages marqués de l'automate de la figure 24, et après simplification (cf. **étape (*)** ci-dessous), on a aussi

$$L_1(i) = \{Init(True)\ d(x5,0)Fin(True)\ ;\ Init(False)d(x5,S)Fin(True)\}$$

$$L_2(i) = \{\ Init(True)bf(e2)Fin(False)\ ;\ Init(False)\ d(x5,0)Fin(False)\}$$

$$L_3(i) = \{Init(True)d(x5,I);\ Init(False)d(x5,I)\}$$

On en déduit que

$$L''_1(i) =\{\ d(xo,S)Init(True)\ d(x5,0)Fin(True),\ d(xo,S)d(x5,D)bf(e5)Init(True)\ d(x5,0)Fin(True),\ d(xo,D)d(x5,S)Init(True)\ d(x5,0)Fin(True),\ d(xo,D)Init(False)d(x5,S)Fin(True),\ d(xo,S)d(x5,D)Init(False)d(x5,S)Fin(True)\}$$

$$L''_2(i) =\{\ d(xo,S)\ Init(True)bf(e2)Fin(False)\ ;\ d(xo,S)d(x5,D)bf(e5)Init(True)bf(e2)Fin(False)\ ;\ d(xo,D)d(x5,S)Init(True)bf(e2)Fin(False)\ ;\ d(xo,D)Init(False)\ d(x5,0)\ Fin(False)\ ;\ d(xo,S)d(x5,D)\ Init(False)\ d(x5,0)Fin(False)\}$$

$$L''_3(i) =\{\ d(xo,S)Init(True)d(x5,I);\ d(xo,S)d(x5,D)bf(e5)Init(True)d(x5,I);\ d(xo,D)d(x5,S)Init(True)d(x5,I);\ d(xo,D)Init(False)d(x5,I)\ ;\ d(xo,S)d(x5,D)Init(False)d(x5,I)\}$$

En absence d'une information à l'entrée, on pourra affirmer pour une entité de classe SB, TF ou ST, il n'existera pas d'information en sortie. On introduira donc trois listes $L'''_1(i)$,$L'''_2(i)$ et $L'''_3(i)$, vérifiant:

- $L'''_1(i)= L''_1(i)$ et $L'''_2(i)= L''_2(i)$ quelque soit la classe de l'entité i
- $L'''_3(i)= L''_3(i)$ si l'entité est de classe CT et $L'''_3(i)= L''_3(i)\cup L3(A)$ sinon **(Eq.26)**

Cette transformation permet de prendre en compte l'héritage d'une perte d'information le long d'un flux d'information. Les listes $L'''_1(i)$,$L'''_2(i)$ et $L'''_3(i)$ contiennent donc bien l'ensemble des séquences d'événements (mots) recherché.

Néanmoins, toutes ces séquences ne sont pas possibles, puisqu'elles permettent de parcourir l'ensemble des combinaisons d'erreurs matérielles sans tenir compte de l'impossibilité pour une ressource matérielle lors d'un cycle T de présenter deux modes de défaillance différents. En outre, elles contiennent un nombre important d'événements non utiles à la détermination des listes La et Ld.

On va donc calculer chacune des listes $L_j(i)$ avec $j \in \{1,2,3\}$ en simplifiant chacun des langages $L'''_j(i)$, de la manière suivante (**étape (*)**):

- les événements de classe Init, Final et « intermédiaire » de chaque mot m de $L'''_j(i)$ seront remplacés par l'élément neutre pour la concaténation ε, leur trace ne sera donc pas conservée
- les événements apparaissant plusieurs fois dans un mot m de $L'''_j(i)$ seront remplacés par l'élément neutre pour la concaténation des mots ε, leur trace ne sera donc pas conservée.
- Enfin, les mots contenant deux événements « erreurs matérielles activées » conflictuels, c'est-à-dire de la forme d(Xi,Yi) et d(Xj,Yj), vérifiant les conditions Xi=Xj ET Yi=Yj , seront exclus du langage, puisque possédant une probabilité nulle d'après l'hypothèse P2 (chapitre 2).

L'ensemble de ces trois opérations est mené par un opérateur de réduction des listes, noté RED. Les deux premières opérations s'apparentent à la notion de projection de langages **[CasLaf97]**

On obtient donc, pour notre exemple,

$$L_1(B)= L_1(i) =RED\ L'''_1(i) = \{ d(xo,S)\ d(x5,0),\ \cancel{d(xo,S)d(x5,D)bf(e5)\ d(x5,0)},\ \cancel{d(xo,D)d(x5,S)\ d(x5,0)},\ d(xo,D)d(x5,S),\ \cancel{d(xo,S)d(x5,D)d(x5,S)}\}$$

$$L_2(B)= L_2(i) =RED\ L'''_2(i) =\{ d(xo,S)\ bf(e2)\ ;\ d(xo,S)d(x5,D)bf(e5)bf(e2)\ ;\ d(xo,D)d(x5,S)bf(e2)Fin(False)\ ;\ d(xo,D)\ d(x5,0);\ \cancel{d(xo,S)d(x5,D)\ d(x5,0)}\}$$

$$L_3(B)= L_3(i) =RED\ L'''_3(i) =\{ =\{ d(xo,S)d(x5,I);\ \cancel{d(xo,S)d(x5,D)bf(e5)d(x5,I)};\ \cancel{d(xo,D)d(x5,S)d(x5,I)};\ d(xo,D)d(x5,I)\ ;\ \cancel{d(xo,S)d(x5,D)d(x5,I)},\ d(xo,S)d\ (x5,I),\ d(xo,D)d\ (x5,I)\}$$

Si on suppose les listes L1, L2 et L3 du bloc A comme complètes, on peut aisément démontrer la complétude des listes L1, L2 et L3 du bloc B si on suppose que l'automate fini associé au bloc B est complet. En outre, les séquences constitutives de ces listes ne contiennent pas d'événements de défaillance conflictuels. On peut donc affirmer que :

- Lj(B) identifient complètement les différentes combinaisons d'erreurs, auxquelles sont associées les états possibles de l'information de sortie
- ces mots contiennent uniquement des événements distincts et non conflictuels.

Maintenant que le cas d'agrégation de bloc série a été effectué, nous allons nous intéresser au cas où un bloc « agrégé» est connecté en série à un bloc « non agrégé », possédant un autre flux d'information entrant.

4-1-2- Deuxième cas : agrégation de bloc IP (cas croisement- figure 26)

Ce cas de figure se rencontre uniquement quand le bloc C (cf figure) est un bloc IP. On distingue deux cas, soit le bloc B est agrégé et dans ce cas l'agrégation du bloc IP est possible. Soit le bloc B est « non agrégé », on doit attendre dans ce cas son agrégation avant de pouvoir agréger le bloc IP.

On se placera donc dans l'hypothèse où les deux blocs A et B sont « agrégés ». On suppose donc connues les listes L1(A), L2(A), L3(A), L1(B), L2(B) et L3(B). Avant de décrire les étapes de constructions de L1(C), L2(C) et L3(C), on rappelle l'opérateur de produit sur les langages noté ⊗. Pour tout langage U et V, l'opérateur ⊗ permet de construire le langage constitué de tous les mots obtenus par concaténation à droite d'un mot de U par un mot de V. Cette opération est facilement automatisable en utilisant deux boucles logicielles :

On va calculer :

$$L11(C) = RED(L1(A) \otimes L1(B)) ;$$

$$L12(C) = RED(L1(A) \otimes L2(B)) ;$$

$$L21(C) = RED(L2(A) \otimes L1(B)) ;$$

$$L22(C) = RED(L2(A) \otimes L1(B)) ;$$

Ces combinaisons d'événements correspondent à l'ensemble des combinaisons d'événements de classes « erreurs matérielles », « erreurs environnementales » et « échec de test » non conflictuels (du fait de l'utilisation de l'opérateur RED) induisant les événements initiaux du bloc IP considéré.

Pour déterminer L1(C),L2(C) et L3(C), on calcule d'abord les langages marqués $L_a(i)$, $L_b(i)$ et $L_c(i)$ du bloc C. Ces langages marqués contiennent des événements pouvant appartenir aux 6 classes proposées dans la partie 2-5, ils sont mémorisés dans les listes $\mathbf{L'_1(i)}$, $\mathbf{L'_2(i)}$ et $\mathbf{L'_3(i)}$.Chaque mot de ces langages commence par un événement de classe Init : Init(11), Init(12), Init(21) ou Init(22).

On détermine ensuite les langages $\mathbf{L''_1(i)}$, $\mathbf{L''_2(i)}$ et $\mathbf{L''_3(i)}$. Pour tout $j \in \{1,2,3\}$, le langage $L''_j(i)$ est constitué de la réunion de :

- l'ensemble des mots distincts, obtenus par concaténation à gauche des mots de $L'_j(i)$ commencant par l'événement Init(11) avec chaque mot de LC(11)
- l'ensemble des mots distincts, obtenus par concaténation à gauche des mots de $L'_j(i)$ commencant par l'événement Init(12) avec chaque mot de LC(12)
- l'ensemble des mots distincts, obtenus par concaténation à gauche des mots de $L'_j(i)$ commencant par l'événement Init(21) avec chaque mot de LC(21)
- l'ensemble des mots distincts, obtenus par concaténation à gauche des mots de $L'_j(i)$ commencant par l'événement Init(22) avec chaque mot de LC(22)

On détermine ensuite $L'''_1(i)= L''_1(i)$, $L'''_2(i)= L''_2(i)$ et $L'''_3(i)= L''_3(i) \cup L3(A) \cup L3(B)$ **(équation 29)**

On obtient finalement $\mathbf{L_j(C)= RED(L'''_j(i))}$, avec $j \in \{1,2,3\}$.

On peut illustrer ces opérations par un court exemple. On suppose calculés les langages suivants (cas de redondance informationnelle utilisant les mêmes ressources matérielles)

$$L1(A) = \{d(x_0,S)\, d(x_5,0)\, ,\, d(x_0,D)d(x_5,S)\}, \quad L2(A) = \{d(x_0,D)\, d(x_5,0)\, ,\, d(x_0,S)d(x_5,D)\}$$

et L3(A)={ d(xo,S)d (x5,I), d(xo,D)d (x5,I)}

L1(B) = {d(xo,S) d(x5,0) , d(xo,D)d(x5,S)}, L2(B) = {d(xo,D) d(x5,0) , d(xo,S)d(x5,D)}

et L3(B)={ d(xo,S)d (x5,I), d(xo,D)d (x5,I)}

et

$L'_1(i)$ = *{Init(11) d(x4,0) aFin(True) ; Init(12) d(x4,0) aFin(True) ; Init(21) d(x4,0) aFin(True) ; Init(22)d(x4,S)Fin(True)}*

$L'_2(i)$ = *{ Init(11) d(x4,D) Fin(False) ; Init(12) d(x4,D) Fin(False) ; Init(21) d(x4,D) Fin(False) ; Init(22)b d(x4,0) Fin(False)}*

$L'_3(i) = \varnothing$

On aura donc, L11(C) = *{d(xo,S) d(x5,0) , d(xo,D)d(x5,S)}*

L12(C) = $\varnothing$

L21(C) = $\varnothing$

L22(C) = *{d(xo,D) d(x5,0) , d(xo,S)d(x5,D)}*

Puis *L"1(i) = { d(xo,S)d(x5,0)Init(11)d(x4,0)aFin(True) ; d(xo,D)d(x5,S)Init(11)d(x4,0)aFin(True) ; d(xo,D)d(x5,0) Init(22)d(x4,S)Fin(True); d(xo,S)d(x5,D) Init(22)d(x4,S)Fin(True)}*

$L''_2(i)$ = *{ d(xo,S) d(x5,0) Init(11) d(x4,D) Fin(False) , d(xo,D)d(x5,S)Init(11) d(x4,D) Fin(False) ; d(xo,D) d(x5,0) Init(22)b d(x4,0) Fin(False) ; d(xo,S)d(x5,D) Init(22)b d(x4,0) Fin(False)}*

$L''_3(i) = \varnothing$

Puis $L'''_3(i)$ *={ d(xo,S)d (x5,I), d(xo,D)d (x5,I)}*

Enfin

$L_1(C)$= *{ d(xo,S) d(x5,0) d(x4,0); d(xo,D) d(x5,S) d(x4,0); d(xo,D) d(x5,0) d(x4,S); d(xo,S)d(x5,D) d(x4,S)}*

$L_2(C)$ = *{ d(xo,S) d(x5,0) d(x4,D), d(xo,D)d(x5,S) d(x4,D); d(xo,D) d(x5,0) d(x4,0); d(xo,S)d(x5,D) d(x4,0) }*

$L_3(C)$ = *{ d(xo,S)d (x5,I), d(xo,D)d (x5,I)}*

Si on suppose les listes L1, L2 et L3 du bloc A et B comme complètes, on peut aisément démontrer la complétude des listes L1, L2 et L3 du bloc C comme l'ensemble des chemins possibles, vérifiant les propriétés énoncés en 3-1. On peut donc affirmer que :

- Lj(C) identifient complètement les différentes combinaisons d'erreurs, auxquelles sont associées les états possibles de l'information de sortie, en accord avec la définition générale de L1, L2 et L3
- Ces mots contiennent uniquement des événements distincts et non conflictuels.

4-2- <u>Ordres des opérations d'agrégation de bloc pour la construction de La et Ld.</u>

Nous avons expliqué dans la partie précédente comment permettre à un bloc d'être agrégé quand son (cas TF,ST,SB, CT) ou ses deux (cas IP) blocs amonts étaient agrégés. Puisque les listes L1, L2 et L3 sont connues pour les blocs sources, il est donc possible d'agréger les blocs suivant la structure de haut niveau jusqu'à évaluer les listes L1,L2et L3 des blocs terminaux et de démontrer par récurrence le caractère complet de ces listes de combinaisons d'événements de classes « erreurs activée » (matérielles et environnementales) et « échec de test ».

Pour l'illustrer, on va reprendre le modèle de haut niveau proposé en 2-2-2. On va numéroter chacune de ces entités constitutives suivant la figure suivante (le choix des numéros n'a aucune incidence).

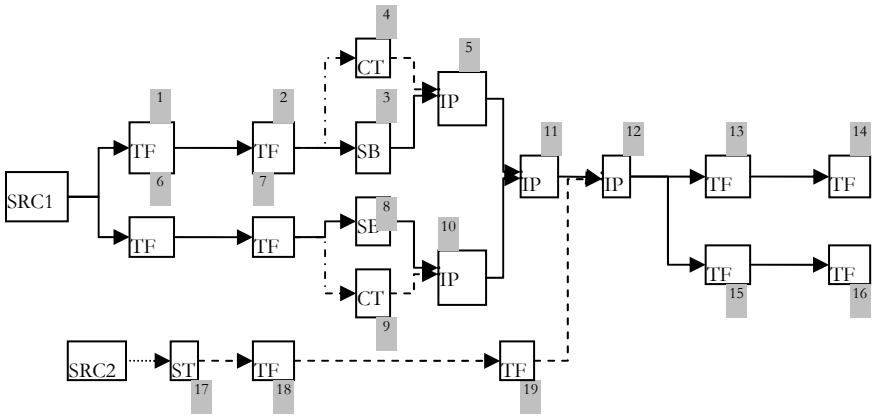


Figure 27 : Modèle de haut- niveau -exemple

Une solution possible pour cet exemple est donc la séquence d'agrégation des blocs suivant l'ordre :

{*1-2-4-3-5-6--7-9-8-10-17-18-19-11-12-13-15-14-16*}

Les blocs 1,2,4,3,5,6,8,7,16,17,18,12,13,14,15 étant des blocs séries, on utilisera la procédure décrite dans 3-1-1, en utilisant les couples (A,B) respectifs suivants : {SRC1,1} {1,2} {2,4} {2,3} {SRC1,6} {6,7} {7,9} {7,8} {SRC2,17} {17,18} {18,19} {12,13} {12,15} {13,14} {15,16}

Les blocs 5,9,10,11 étant des blocs IP, on utilisera pour leur agrégation la procédure décrite dans 3-1-2, en utilisant les triplets {A,B,C} respectifs suivants : {4,3,5} {8,7,10} {5,10,10} {11,19,12}.

On aura ainsi construit les listes complètes L1(i), L2(i) et L3(i) pour les blocs i=14 et i=16 qui sont les blocs terminaux. On notera, par la suite,

- Ld(i) l'ensemble des mots de L2(i) et L3(i) qui commencent par le caractère d(xo,S) (c'est-à-dire présence de demande à l'instant d'acquisition)
- La(i) l'ensemble des mots de L1(i) qui commencent par le caractère d(xo,D) (c'est-à-dire absence de demande à l'instant d'acquisition)

La liste Ld(14) (resp Ld(16)) contient l'ensemble des combinaisons d'événements non conflictuels menant à une non activation de la vanne 1 (resp. la vanne 2) après l'occurrence d'une demande. On a donc :

$$Ld = RED(Ld(14) \otimes Ld(16)).$$

Dans le cas d'un unique actionneur, on aura Ld par simple construction de la liste Ld(i) de l'entité TF terminale.

La liste La(14) (resp La(16)) contient l'ensemble des combinaisons d'événements non conflictuels menant à une activation de la vanne 1 (resp. la vanne 2) en l'absence d'une demande. On a donc :

$$La = La(14) \cup La(16).$$

On a ainsi complètement identifié les deux listes recherchées La et Ld, tout en prenant en compte :

- l'ordre des différentes sous entités fonctionnelles
- la partage de ressources informationnelles (du fait du partage par exemple de l'utilisation des listes L1(2),L2(2) et L3(2) lors de l'agrégation des blocs 3 et 4)
- le partage de ressource matérielle, en interdisant à une combinaison d'événements (opérateur RED) de contenir deux événements d'erreurs matériels conflictuels, en accord avec l'hypothèse d'invariance de l'état des ressources matérielles sur un cycle T.

On peut affirmer que ces listes sont complètes, ne contiennent que des événements d'activation d'erreurs distincts et non conflictuelles.

5- Résultats

En appliquant la méthode précédente, on va construire complètement les listes Ld et La recherchées en prenant en compte

- l'héritage d'informations erronées le long de flux d'information, par le mécanisme de propagation des langages L1,L2, L3
- la possibilité de perte d'information le long d'un flux, modélisée par les **équations 28 et 29** dans les opérations d'agrégation élémentaires décrites en 3-1-1 et 3-1-2.
- le partage d'information, par l'utilisation possible de langages L1, L2, L3 communs par plusieurs blocs,
- les contraintes, en termes d'erreurs, liées à l'utilisation répétée d'une ressource identique, prise en compte grâce à l'utilisation de l'opérateur RED après chaque agrégation.

Les listes finales La et Ld sont complètes puisque les listes L1, L2 et L3 des blocs sources le sont et que nous appliquons une méthode d'agrégation systématique sur l'ensemble des flux de F- et de D-informations, en utilisant les langages marqués des automates d'états finis de chaque entité sous fonctionnelle. Cette propriété peut être démontrée sans difficulté par redondance sur chaque opération élémentaire d'agrégation le long d'un flux et du fait de la complétude de l'opérateur $\otimes$.

En outre, chaque mot de ces listes contient

- un événement correspondant à l'état de demande (supposé vrai et pouvant être remplacé par ε dans Ld et La),
- une combinaison finie d'erreurs matérielles non conflictuelles portant sur des ressources matérielles distinctes, du type d(xi,Y) avec $Y \in \{S,D,I\}$ une combinaison d'utilisation de ressources matérielles saines du type d(xi,0)
- une combinaison d'erreurs environnementales distinctes (pouvant éventuellement dépendre de la même faute environnementale),
- une combinaison d'événements échec de test.

Dans l'hypothèse d'erreurs matérielles rares, on peut supposer, pour réduire le temps de calcul, que le produit des probabilités de combinaisons d'événements d'utilisation de ressources saines est équivalent à 1 (l'erreur induite sur le calcul étant très faible). On remplacera donc ces événements par le caractère ε dans Ld et La.

Les mots de Ld et La ne seront donc plus constitués que d'événements d'activation d'erreurs matérielles portant sur des ressources distinctes, d'erreurs environnementales distinctes et/ou d'événement(s) « échec de test ». Ils permettront de donner l'ensemble complet des combinaisons possibles d'événements simultanées menant à un mode de défaillance du système. Ils pourront donc être directement utilisés dans le modèle général (chapitre 2) pour estimer les indicateurs de performances de la fonction de sécurité étudiée.

Le caractère « systématique » de la construction de ces listes est lié à l'identification de l'ensemble complet des séquences d'activations d'erreurs menant à un mode de défaillance de la fonction de sécurité. Les séquences d'activation d'erreurs réduites sont suffisantes à un mode de défaillance (premier impliquant), et on peut affirmer que si un mode de défaillance de la fonction de sécurité existe, une de ces séquences d'erreurs aura été activée. En outre, ce processus de construction est purement déterministe puisque contrôlé par les flux d'information échangés ou non entre entités sous-fonctionnelles et par la description locale d'automate à état fini de nature déterministe.

La notion d'ordre disparaît si on admet l'hypothèse fondamentale donnée au chapitre 2 (invariance de l'état des ressources matérielles sur une période T). Cependant, la construction de ces séquences, qui peuvent être vues sous ces hypothèses comme des combinaisons minimales, s'appuie sur un processus propagatif et non hiérarchique statique. Elle prend donc implicitement en compte les contraintes temporelles, en termes l'ordre, des entités sous-fonctionnelles dans le traitement d'information. La différence principale avec une recherche de traces par Altarica vient du fait que nous n'utilisons pas ici la synchronisation de graphes d'événement et que les conflits et répétition d'événements sont gérés (réduction des langages) à chaque étape du processus de propagation des langages.

On peut, enfin, remarquer que la longueur de ces listes et de leurs constituants a un faible impact sur le temps d'évaluation. La probabilité d'occurrence de chacune de ces listes est, en effet, donnée par la somme des probabilités de chacun de ses mots. Ces probabilités sont égales aux produits des probabilités d'existence des erreurs activées, des probabilités des « événements d'échec de test » et de celles des erreurs environnementales activées apparaissant dans chaque mot. Cette opération peut donc être facilement automatisée en se basant sur la structure liste/mots/caractère (par exemple par l'utilisation de vecteurs de vecteurs en Java) des listes La et Ld, si on peut simuler l'évolution temporelle des probabilités d'erreurs matérielles et environnementales (chapitre 4).

6- Mise en œuvre pratique

Nous avons précédemment parlé de l'utilisation ultérieure des listes La et Ld pour le calcul des indicateurs de performances de la fonction de sécurité étudiée. Il est néanmoins clair que la construction de ces listes, malgré la simplicité de la méthode d'agrégation de blocs, est très difficile à mener manuellement. Nous avons donc proposé la construction d'un logiciel de calcul de ces listes appelé AFMCI sous Java. Le choix de ce langage a été motivé par l'existence d'interface graphique, de type schéma bloc du logiciel ©Jagrif (v.3-30) qui peut être mise à profit pour déterminer à partir d'une saisie manuelle du modèle de haut niveau les séquences d'agrégations successives de ces blocs constitutifs.

6-1- Cahier des charges du logiciel

Le logiciel AFMCI a pour but de permettre la saisie manuelle du modèle de haut niveau, des graphes et des propriétés (classe du bloc, classe d'événements, attributs et marquage) associés à chaque entité sous fonctionnelle. Une fois cette saisie effectuée, il permet de calculer automatiquement les listes La et Ld et de les mémoriser en conservant la structure

liste/mot/caractères permettant leur utilisation ultérieure pour une estimation numérique (méthode du chapitre 2) des indicateurs de performances de la fonction de sécurité.

Il est donc constitué de quatre modules :

- un module de saisie manuelle des automates d'états finis et une méthode de calcul de leurs langages marqués (notés précédemment La, Lb et Lc)
- un module de calcul permettant la définition des opérations de bases (RED, ⊗) et de mener à bien les étapes d'agrégation élémentaires 3-1-1 et 3-1-2 en reconnaissant la nature du dernier bloc (IP ou autres).
- un module de calcul permettant de mener à bien la méthode, connaissant les séquences successives d'agrégation de blocs de type série ou croisement.
- un module d'interface graphique permettant la saisie manuelle du modèle de haut niveau sous l'interface graphique de ©Jagrif (Stochastic Bloc), de changer les propriétés (graphe) de chaque bloc et de calculer automatiquement les séquences d'agrégation.

6-2- Module de saisie manuelle des graphes et de calcul des langages marqués

Le module de saisie manuelle permet d'entrer l'automate d'état fini de chaque bloc. L'interface graphique se présente sous la forme d'une fenêtre interactive permettant d'entrer le nombre d'états de l'automate fini, de préciser l'état de marquage de l'état considéré et de définir pour chacun de ces états les transitions, en précisant leur classe et leurs attributs. On associe à cet automate d'état fini une structure d'arbre sous Java (Build Tree) qui va nous permettre par la suite, par parcours des chemins d'accessibilité, de calculer itérativement les langages marqués associé à l'automate fini.

On distinguera deux méthodes :

- la méthode AddBlock qui permet de créer un nouveau graphe associé à une entité sous fonctionnelle
- la méthode Calculate(L) qui permet le calcul des langages marqués La, Lb et Lc de l'automate d'état fini.

Nous noterons que les listes utilisées lors des opérations d'agrégation (étape de 3-1-1 et 3-1-2) seront mémorisées sous la forme de vecteur de vecteur. Un langage sera défini comme un vecteur, contenant des vecteurs (mots) dont les constituants seront des caractères liés à leur propriété de classe d'appartenance et à leurs attributs. Cette structure imbriquée permet ensuite une mise en œuvre aisée des opérations «RED » et « croisement ». Elle permet, en outre, une exportation des listes d'un bloc à un autre par passage par un fichier tampon .xml et l'utilisation de la librairie jdom de Java permettant l'importation et l'exportation de ce type d'objets.

6-3- Opération de manipulation des listes

On introduit, outre la méthode de calcul des langages finis, les deux opérateurs RED et ⊗ (notés respectivement Red1Op et Red2Op dans le code source).

L'opérateur RED va permettre le parcours de chaque mot d'un langage (boucle d'exploration des vecteurs d'ordre 1 (mots)), la suppression des caractères répétitifs (boucle sur les composants du mot (vecteur d'ordre 2)) et l'exclusion des mots contenant des caractères conflictuels (états d'erreurs matériels activés contradictoires). Il va mémoriser le résultat de cette opération sous le même emplacement que la liste de départ.

L'opérateur ⊗ va permettre l'opération de multiplication des langages. La difficulté dans sa mise en œuvre réside dans la concaténation de vecteurs d'ordre 2 induisant la création d'un nouveau vecteur d'ordre 2. Elle oblige, pour des raisons de structure manipulée, l'introduction d'un objet tampon permettant de mener à bien cette opération (boucle).

Une fois ces procédures implantées, on prévoit une procédure (FrontEnd) qui permet la mise en œuvre des opérations élémentaires :

- d'agrégation de N blocs séries (avec N entier)
- d'agrégation d'un bloc IP par la donnée de son nom (numéro) et de celui de ces deux prédécesseurs (dans l'ordre A-B donné en 3-1-2) (cas N=3).

Cette procédure est automatique, elle permet si la longueur de séquences des blocs est égale à 3 de mettre en oeuvre automatiquement suivant la classe du dernier bloc (IP ou autre) soit (N-1) opérations élémentaires série (3-1-1) ou 1 opération de croisement (3-1-2). Le premier bloc de la séquence est supposé agrégé tandis que les autres ne le sont pas. Le résultat de cette procédure est le calcul de L1, L2 et L3 pour tous les blocs de la séquence.

Il est donc possible en connaissant les différentes séquences d'agrégations des blocs de calculer L1, L2 et L3 du dernier bloc et d'en déduire si ce bloc est un bloc terminal La et Ld.

6-4- Détermination des séquences d'agrégation

Après avoir implanté ces méthodes, il est possible de calculer automatiquement La et Ld par une saisie manuelle des séquences d'agrégation de blocs. On lance, pour chaque séquence d'agrégation de blocs, la procédure de calcul des listes L1, L2, L3 des blocs la constituant, jusqu'à les avoir calculées pour chaque bloc TF terminal. Pour le cas d'étude du chapitre 5, nous avons utilisé une saisie manuelle de ces séquences successives.

Nous proposons d'utiliser l'interface graphique de©Jagrif (Stochastic Bloc) pour construire automatiquement ces séquences. L'interface graphique de cette outil nous permet d'entrer manuellement le modèle de haut niveau en utilisant deux types d'objets : les objets blocs et les objets liens (permettant de figurer les flux élémentaires d'informations). Dans le cas de flux multiples partant d'un bloc, l'outil génère automatiquement un objet appelé connecteur LC. Dans le cas d'un bloc IP, les deux entrées sont liées par un connecteur de type RC. Pour chaque nouveau bloc crée, on démarre la procédure AddBlock qui permet de donner l'automate d'état fini associé et la classe du bloc. Pour l'exemple précédent, on obtient donc le modèle graphique suivant (**figure 27**).

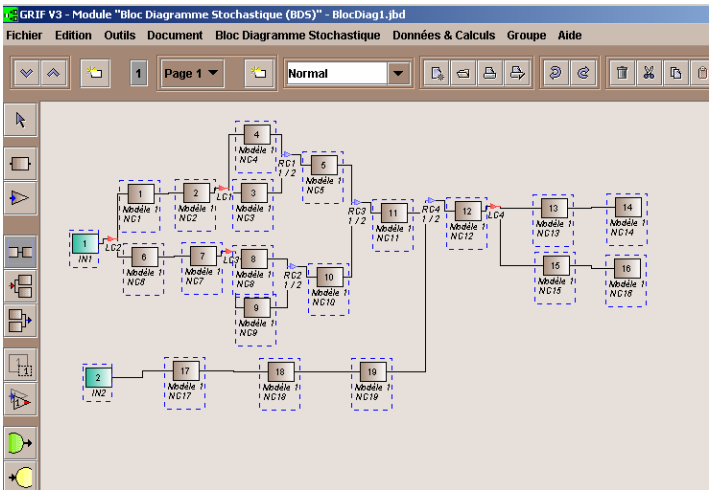


Figure 28 : Saisie graphique du modèle de haut niveau de l'exemple précédent sur interface©Jagrif

L'algorithme de calcul automatique des séquences est basé sur la procédure GetPrevious (resp GetNext) de l'outil ©Jagrif qui permet de déterminer pour chaque bloc ou entité LC ou RC son ou ses blocs amonts (resp avals).

On va donc parcourir la structure en partant des blocs terminaux (pas de sortie donc vérifiant GetNext= vide). On détermine en partant de ces blocs la séquence de blocs permettant d'atteindre le premier bloc qui soit de type IP ou de type LC. On obtient donc les séquences :{13,14}, {15,16}, rangées dans une liste de séquences (vide au démarrage de la procédure), notée U.

On a ainsi U={{13,14}, {15,16}}

On part ensuite des premiers blocs de ces séquences,

- on recommence la même opération si ce bloc n'est pas de classe IP
- on détermine la séquence {A,B,C} (deux blocs amont et bloc IP) dans le cas inverse.

On ajoute les résultats en tête de la liste U obtenue. On obtient ainsi : U= {{11,19,12}, {13,14}, {15,16}}

On recommence la procédure jusqu'au moment où les séquences générées ont toutes pour premier bloc un des blocs sources INIT (arrêt du processus itératif).

On obtient finalement :

U={{init1,6}{init1,1}{6,7}{1,2}{init2,17}{7 ;8}{7 ;9}{2 ;3}{2 ;4}{17 ;18}{8 ;9 ;10}{4 ;3 ;5}{1 8 ;19}{5 ;10 ;11}{11 ;19 ;12}{13 ;14}{15 ;16}}

Cette liste finale U de séquence est la liste des opérations d'agrégation élémentaires à mener. On calculera donc les listes finales L1, L2, L3 des blocs terminaux (14 et 16) en appliquant á cette séquence de séquence la procédure de calcul Front End.

6-5- Calcul final

Le calcul final donne les listes L1, L2 et L3 des blocs terminaux agrégés. Pour des raisons d'encombrement de la mémoire par mise en mémoire des résultats, on peut effacer les listes L1, L2, L3 des blocs ayant été agrégés et non utilisées dans les calculs ultérieurs. Ces listes finales obtenues permettent de déterminer les listes La et Ld ayant les propriétés énoncées dans le paragraphe 4.

Ces listes sont mémorisées sous la forme d'un fichier .xml du type :

<MOT

<caractère

<classe

<attribut

Dans le cas d'un langage Ld={d(x1,S),d(x2,D)bf(5), d(x3,D), ft(5)}, on aura donc le contenu du fichier suivant (**figure 29**):

```
- <block type="FINAL">
- <Ld>
- <wort>
  <char type="TRANSIENT" X="x1" y="S" />
  </wort>
  <wort>
<char type="TRANSIENT" X="x2" y="D" />
<char type="bf" i="5"/>
</wort>
  <wort>
<char type="TRANSIENT" X="x3" y="D" />
</wort>
  <wort>
<char type="test fault" i="5"/>
</wort>
  </Ld>
  <La />
……
  </La>
```

Figure 29 : Exemple de fichier .xml obtenu

Cette structure du fichier résultat (format .xml) permet d'intégrer facilement ces listes au modèle numérique d'évaluation des indicateurs de performances de notre fonction de sécurité (chapitre 2), si on associe à chaque événement sa probabilité évoluant au cours du temps. Elle permet, en effet, d'appeler un à un les mots du langage (possibilité de sommation) puis un à un les événements constitutifs de ces mots (possibilités de multiplication de leur probabilité). On peut donc calculer très rapidement les variables pd et pa données au chapitre 2, si on connaît un modèle permettant de simuler l'évolution individuelle des probabilités d'erreurs.

Une méthode de construction de ces modèles sera donnée au chapitre 4 pour les erreurs matérielles. Pour les événements « échec de test », on supposera soit une attribution déterministe {0 ;1} de la probabilité de cet événement du à la périodicité du test en ligne (cas où la période de test est très supérieure à T), soit une probabilité évoluant au cours du temps, dans le cas d'un test par balayage de périodicité courte. Nous ne nous attacherons pas à cette estimation qui se base, en général, sur le retour d'expérience sur une période longue en présence d'erreur détectable (dispersion des instants de test). Nous supposerons donc dans toute la suite de notre travail, l'évolution temporelle des probabilités « échec de test » comme estimées et l'évolution des probabilités de fautes transitoires estimables à partir des stress environnementaux (supposée déterministe). On note qu'en général les modèles actuels d'estimation probabiliste de ce type de fautes restent encore approchés et qu'on leur attribue donc souvent une probabilité d'occurrence supposée constante sur des phases temporelles déterminées (suivant le profil de mission de l'installation).

7- Conclusion du chapitre 3

Nous avons montré dans ce chapitre que les listes caractéristiques La et Ld pouvaient être construites de manière systématique à partir d'une modélisation de haut niveau, de type flux d'information, de l'architecture fonctionnelle et d'une modélisation par automate d'état fini des comportements des entités sous fonctionnelles de cette architecture.

La proposition de règles de propagation et de réduction de langages le long de cette structure (processus propagatif) nous permet d'assurer leur complétude et la prise en compte des contraintes dues au partage de ressources (matérielles et informationnelles) et des contraintes de précédence entre entités sous fonctionnelles. En outre, elle permet d'assurer le caractère possible (probabilité non nulle) de chaque combinaison. L'avantage de cette méthode est de réduire la taille des listes obtenues en permettant l'exclusion lors du processus d'agrégation de séquence correspondant à l'activation d'erreurs matérielles dont la co-existence va à l'encontre des hypothèses d'invariance proposées au chapitre 2 (hypothèse P2). Elle permet en outre de prendre en compte les problèmes de partage de ressources entre voies parallèles par l'utilisation successive des opérateurs de croisement et de réduction.

Bien que la longueur des listes obtenues soit souvent importante et ne permette pas, de ce fait, une interprétation manuelle des résultats, l'automatisation de leur processus de construction et leur mémorisation permet leurs utilisations ultérieures à des fins de calcul, notamment pour l'évaluation de performances proposée au chapitre 2 et pour l'identification de combinaisons minimales (cf. chapitre 6).

L'utilisation conjointe de ces listes pour la construction d'arbres dynamiques (©AltariCa/©SimTree, ©Fault Tree +) dont les feuilles seraient les événements constitutifs des mots de La et Ld (erreurs matérielles, transitoires, échec de test) est possible. L'évolution des probabilités pd et pa serait directement liée aux modèles stochastiques associés à chaque ressource matérielle et donnant l'évolution de ses états d'erreurs au cours du temps et l'évolution temporelle des probabilités de fautes transitoires et d'événements échec de test. Ce type de mise en œuvre du modèle général donné dans le chapitre 2 sera celui retenu dans l'évaluation de notre cas d'étude (chapitre 5).

Mais, avant de discuter de cette mise en œuvre pour un cas d'étude précis, nous devons proposer les modèles stochastiques permettant de simuler l'évolution des probabilités d'erreurs matérielles au cours du temps. Cette construction sera donc exposée dans le chapitre suivant (chapitre 4), où nous justifierons de l'intérêt de l'utilisation de processus markoviens non homogènes.

Chapitre 4

Construction des modèles d'évolution des probabilités d'erreurs matérielles

Chapitre 4 - Construction des modèles d'évolution des probabilités d'erreurs matérielles

Dans la partie précédente, nous avons décrit une méthode permettant d'aboutir aux listes caractéristiques Ld et La. Nous avons ainsi vu comment identifier les combinaisons d'erreurs activées distinctes nécessaires et suffisantes à l'existence d'un mode de défaillance sur un cycle de réaction du système d'étude. Nous allons, dans cette partie, nous intéresser à la construction de processus stochastiques permettant de décrire l'évolution temporelle des probabilités d'existence de ces erreurs matérielles. Nous associerons à chaque ressource matérielle un processus stochastique. Ses règles d'évolution dépendront de l'évolution temporelle des contraintes environnementales et des procédures de maintenance auxquelles la ressource est soumise lors de la durée d'utilisation de notre système. Nous allons privilégier une modélisation de ces processus basée sur l'utilisation de chaînes de Markov non homogènes et sur leurs extensions pour la prise en compte de procédures de maintenance préventives planifiées.

Nous montrerons dans la première partie l'intérêt de tels processus et expliquerons la forme générale des modèles construits. Nous étudierons d'abord le cas de ressources non réparables avant d'expliquer les modifications de modèles dans le cas d'une ressource matérielle maintenue de manière corrective et préventive. Nous expliquerons la notion de taux de transitions équivalents entre les modes de défaillance d'une ressource matérielle et les relations entre ces taux et les taux de pannes des composants constitutifs de la ressource considérée.

La seconde partie sera ensuite dédiée à la détermination des taux de transitions de chacun de ces processus. Nous identifierons les contraintes auxquelles sont soumis les composants et relierons les taux de pannes des composants à l'évolution temporelle des stress environnementaux. Nous proposerons de distinguer l'influence instantanée des contraintes environnementales des phénomènes d'usure affectant ces composants.

Une fois ces processus stochastiques définis pour chaque ressource matérielle (pas d'itération T), nous discuterons des hypothèses nécessaires permettant de justifier l'utilisation de processus équivalents, dont le pas d'itération T* serait supérieur à T. Leur utilisation directe dans le modèle général d'évaluation quantitative, donné au chapitre 2, permettra en remplaçant T par T* dans l'algorithme de réduire le temps de calcul des indicateurs de performance recherchés.

1- Choix de processus Markoviens et extensions.

1-1- Rappel sur les processus markoviens

L'utilisation des processus de Markov pour décrire l'évolution temporelle des modes de défaillances des ressources matérielles est relativement usuelle dans le monde industriel **[Rouv00]**. Ils permettent d'estimer l'évolution au cours du temps des probabilités, pour un ensemble fini de ressources, d'occuper chacun de ses états (processus stochastique). Ils impliquent de ce fait un espace d'états (ou espace des possibles) et un référentiel temporel : discret ou continu.

Nous désirons expliquer l'avantage de l'utilisation de chaînes de Markov non homogènes pour décrire l'évolution des probabilités d'erreur de chaque ressource matérielle. Pour ce faire, nous désirons d'abord rappeler quelques notions de base sur les processus markoviens. Un processus sera dit de Markov (ou markovien) si la probabilité d'être dans un état X à l'instant t, ne dépend que de l'état réel du système à l'instant le précédant directement et non des changements d'états antérieurs à celui-ci **[Schn85]**. Un processus markovien d'ordre consiste donc à décrire le système étudié par un ensemble d'états tels que

- à chaque étape, le système est décrit par un des états du processus,

- la probabilité que le système arrive à l'état à l'étape ne dépende que des états du système lors des étapes précédentes.

On peut donc parler de processus sans mémoire, puisque les probabilités conditionnelles de passage ne dépendent pas de l'évolution antérieure du processus. Il est néanmoins tout à fait possible de prendre en compte la dégradation progressive du système modélisé par l'ajout d'états supplémentaires.

La seule donnée de l'ensemble des probabilités des états d'une chaîne markovienne à l'instant t et des probabilités conditionnelles de passage d'un état à un autre au même instant, permet donc de définir l'ensemble des probabilités des états de cette chaîne à l'instant suivant t+ dt.

Pour toute chaîne markovienne, on notera P(t) le vecteur colonne des probabilités des états du processus. Ce vecteur a une dimension identique au nombre d'états modélisés. On le notera N. On notera, en outre, A(t) la matrice de transition donnée par les probabilités conditionnelles de passage entre états. La matrice A(t), appelée matrice de transition, contient à l'emplacement (i,j) (i-ème ligne, j-ième colonne), la probabilité conditionnelle de passage sur la période [t,t+Δt[entre l'état j et l'état i sachant que nous nous trouvons dans l'état i au temps t.

On peut distinguer les chaînes markoviennes, suivant la nature discrète ou continue du temps. On parlera de chaînes de Markov pour un référentiel temporel discret. Dans le cas d'un pas temporel τ, le processus Markovien pourra s'écrire sous la forme générique :

$$P(t+\tau) = A(t).\ P(t) \qquad \text{avec } t=k\tau \text{ (k entier)} \qquad \textbf{(Equation 30)}.$$

Les termes de la matrice A(t) sont bien les probabilités conditionnelles de passage entre états sur l'intervalle [t,t+T[. On peut retrouver le cas continu en faisant tendre τ vers 0. On retrouve donc une équation différentielle du premier ordre en remarquant que $P'(t) = \lim_{dt \to 0} \frac{P(t+dt) - P(t)}{dt}$, cette équation permet d'écrire que $P'(t) = [A(t) - I].P(t) = \tilde{A}(t).P(t)$. **(Equation 32).**

Une chaîne de Markov est dite homogène si sa matrice de transition associée A(t) est constante au cours du temps. Dans le cas inverse le processus markovien sera dit non homogène. Pour plus de détail sur les processus markoviens, le lecteur pourra se reporter aux références **[Coco97]**, rappelant les règles de base et de stationnarité de ces modèles et **[Mart04]** qui montrent les limites des modèles homogènes dans le cas de systèmes évolutifs de nature réversible.

Nous serons intéressés par la possibilité de modélisation de l'évolution temporelle des probabilités d'erreurs matérielles **d'une ressource matérielle** non réparable, par **une chaîne de Markov** de pas d'itération τ=T et de **nature non homogène**.

Cette modélisation est possible, puisque nous avons précédemment supposé qu'une ressource matérielle conservait le même état sur une période de référence T (cf. chapitre 2). La probabilité de ces états est, par conséquent, constante sur ces périodes. Nous étendrons ensuite ce modèle de base pour permettre la prise en compte d'opérations de maintenance (correctives ou préventives) sur la ressource matérielle considérée.

1-2- Utilisation d'un processus markovien pour l'évaluation temporelle des probabilités de fautes d'une ressource matérielle non réparable.

L'utilisation d'un processus markovien pour modéliser l'évolution temporelle de l'état de dégradation **d'une ressource matérielle** est relativement naturelle.

La dégradation d'un ensemble fini de composants au cours du temps pour une ressource matérielle non réparable est un phénomène irréversible, dépendant à la fois de l'état actuel de la

ressource au temps t et des contraintes environnementales au même instant pouvant influer sur l'apparition d'autres dégradations. On peut ainsi construire un processus markovien représentant l'évolution des probabilités d'état de la ressource matérielle**, supposée pour le moment non réparable**. On associera à chaque combinaison de ses composants défectueux (les autres composants étant supposés sains) un état.

On supposera ensuite que les probabilités conditionnelles de passage entre ces états pour un temps de référence t=kT dépendent de l'état des stress environnementaux à cet instant de référence. Ces transitions correspondent à l'apparition de défauts sur les composants de la ressource. Cette chaîne de Markov possède en général une dimension importante puisqu'on considère toutes les combinaisons d'éléments défectueux. Elle est de nature **non homogène,** puisque la probabilité de pannes de composants dépend, de manière déterministe, des stress environnementaux auxquels les composants de la ressource matérielle ont été exposés et notamment du temps **[Hass95]**.

Une fois ce modèle construit, on va affecter chaque état à une classe d'appartenance. Ces classes correspondront aux modes de défaillances S, D, I et au mode de fonctionnement normal 0 de la ressource matérielle modélisée. On appellera par la suite macro-état X (avec $X \in \{0, S, D, I\}$) l'ensemble fini d'états appartenant à la classe X.

Nous désirons remplacer le modèle précédent par une chaîne de Markov agrégée, permettant de décrire directement l'évolution des probabilités pour la ressource de se trouver dans un des macro-états précédemment définis. Actuellement, la plupart des méthodes d'agrégation d'états proposés dans la littérature (**[Seri96]**, **[Scho02]**) le sont pour des graphes de Markov homogènes. Elles peuvent utiliser soit des propriétés structurelles de ces graphes (connexité, diagonalisation matricielle…), soit s'appuyer sur une analyse numérique permettant de déterminer l'évolution de la distribution des états dans un macro-état fixé.

Notre principal problème est que nous ne pouvons pas a priori déterminer de manière précise les répartitions au temps t des états dans un macro-état fixé X. Nous pouvons cependant affirmer que

- pour tout état de 0, il existe au moins un chemin permettant d'accéder à un autre macro-état (phénomène naturel de dégradation) ;
- aucune dégradation ne permet le retour d'un macro-état S, D ou I au macro-état 0 (hypothèse de non réparation) ;
- le macro-état I est absorbant, puisqu'une perte de signal dans la ressource matérielle est un phénomène irréversible pour une ressource non réparable, toute apparition de défauts ultérieure sur les composants sains restants (au sens de non défectueux) sera donc sans effet ;
- s'il existe une dégradation permettant le passage d'un état a de X à un état b de Y (avec X et Y deux macro-états distincts), aucune séquence de dégradation de composants ne permet le passage de cet état b à l'état a (absence de boucle due à l'irréversibilité du processus de dégradation).

Sous ces hypothèses, l'évolution de l'état d'erreur d'une ressource matérielle peut être donnée par une chaîne de Markov non homogène de la forme (**figure 30**).

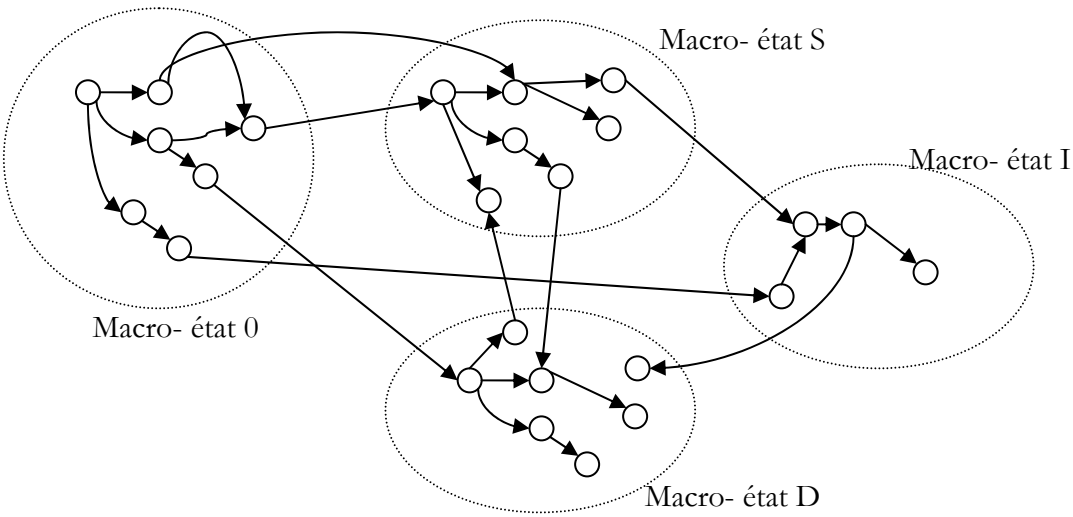


Figure 30 : Chaîne de Markov non agrégée.

Dans ce modèle, chaque transition correspond à l'apparition d'une panne de composant, durant la période de référence T. Le problème majeur d'un modèle de ce type est le nombre d'états et donc de transitions à modifier au cours du temps. Pour simplifier, on cherche à établir une chaîne de Markov non homogène de pas temporel T dans laquelle ne figure plus que les macro-états (**figure 31**).

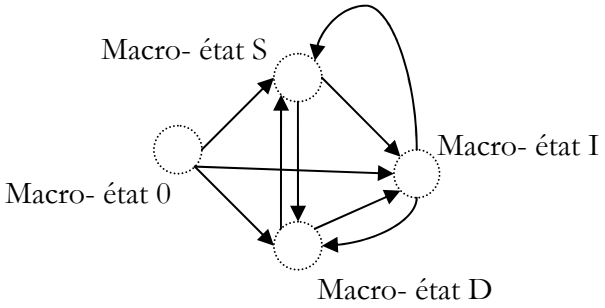


Figure 31 : Chaîne de Markov recherchée.

Pour construire, ce processus markovien, nous allons donc proposer, dans le cas d'une ressource matérielle non réparable, les hypothèses suivantes :

- Il n'existe aucune transition permettant le passage d'un des macro états de l'ensemble {S, D} à un macro état distinct du même ensemble (**hypothèse 1**) ;
- Il n'existe aucune transition permettant le passage du macro état I à un autre macro état (**hypothèse 2**).

L'hypothèse 1 est justifiée si on suppose que la ressource matérielle est un système cohérent (dans le cas inverse, on la subdivisera en plusieurs ressources cohérentes prises en compte dans la construction des listes du chapitre 3). En effet, le passage pour une ressource matérielle d'un mode de défaillance S à un mode de défaillance D impliquerait que, sous un mode de sollicitation 1 (cf. notation chapitre 2, 1-1), la ressource défaillante puisse passer à un mode de fonctionnement correct. De même, le passage pour une ressource matérielle d'un mode de défaillance S à un mode de défaillance D impliquerait que, sous un mode de sollicitation 2 (cf. notation chapitre 2, 1-1), la ressource défaillante puisse passer à un mode de fonctionnement correct. Ces deux possibilités sont proscrites pour une ressource matérielle cohérente.

L'hypothèse 2 est justifiée puisque le mode de défaillance I peut être vu comme le résultat de la perte d'un signal dans l'architecture matérielle et que cette propriété est héritée le long du processus de traitement de signal en aval de ce point de coupure.

On considère donc des **ressources matérielles cohérentes** et on conservera uniquement

- les transitions de l'état 0 vers un macro état S, D ou I et
- les transitions des macro états {S, D} au macro-état I.

On aura donc déterminé sous ces hypothèses la chaîne Markovienne de la forme (**figure 32**)

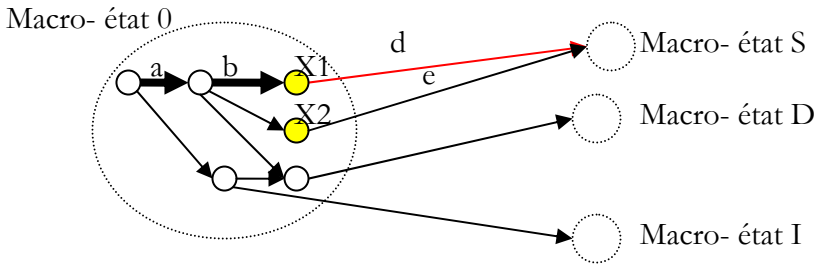


Figure 32 : Mise en évidence de séquences de pannes.

On note $E_{0\to i}$ l'ensemble des séquences de pannes (chemins) permettant de passer de l'état initial de la ressource matérielle (correspondant à l'état ne possédant aucun arc entrant) au macro état i, avec $i \in \{S, D, I\}$. Il est ensuite possible de déterminer pour chaque élément de cet ensemble, les sous-séquences de longueur maximale permettant de rester dans le macro-état 0. Ces sous-séquences amènent à un ensemble $x_{0\to i}$ d'états.

Dans la **figure 32**, la séquence de pannes abd constitue, par exemple, une séquence de $E_{0\to S}$. La séquence ab est sa sous-séquence de longueur maximale permettant de rester dans le macro état 0. L'ensemble $x_{0\to S}$ de la figure 32 est donné par les deux états notés X1 et X2.

Il est ensuite possible de déterminer un taux de transition équivalent entre le macro-état 0 et le macro-état S au temps t, noté p(S/0,t). On aura :

$$p(S/0,t) = p(\mathrm{X1}/0,t).d(t) + p(\mathrm{X2}/0,t).e(t) \qquad \textbf{(Equation 33)}$$

avec t=kT ;

p(Xi/0,t), la probabilité d'être dans l'état Xi à t ;

Cette formule peut être généralisée sans difficulté, pour un nombre de chemins différent et en considérant les macro-état D et I en sus du macro état S.

p(i/0,t) sera alors appelé taux de transition équivalent et sera noté $\Gamma_{i/0}(t)$, on supposera une relation algébrique directe (fonction linéaire), donnée par une relation analogue à la précédente.

On peut donc construire le modèle de Markov non homogène simplifié suivant pour une ressource non réparable (**figure 33**), où chaque taux de transition équivalent $\Gamma_{i/0}(t)$ avec $i \in \{S, D, I\}$ est donné par une fonction algébrique en fonction d'un nombre fini de probabilités d'apparition de pannes de composants sur la période [t,t+T[.

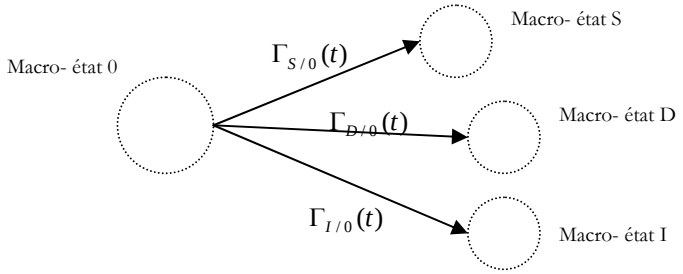


Figure 33 : Résultat de la procédure d'agrégation.

Nous allons désormais justifier l'extension de ce modèle au cas de ressources matérielles correctivement maintenues.

1-3- Extension au cas de ressources correctivement maintenues.

Le cas d'une ressource correctivement maintenue va maintenant être abordé. La réparation est conditionnée par une procédure de détection d'erreurs en ligne. On suppose, en général, que l'action de réparation est exigée dès lors que la détection de l'erreur matérielle est effectuée. Cette information de diagnostic est souvent de type tout ou rien. L'existence d'un retour d'information au niveau opérateur, suite à une détection par la procédure de test, est dépendante de l'état des ressources permettant cette remontée d'informations. On peut identifier une liste de ressources dont le fonctionnement doit être normal pour assurer la bonne réalisation du test et l'absence de perte d'information. On néglige, en général, le retard de retour d'informations par rapport au MTTR (durée moyenne d'inspection + réparation de la ressource) et on suppose qu'il est sûr (contrôle de la réception périodique d'un message de diagnostic positif ou négatif).

En partant de ces hypothèses, on peut identifier pour chaque macro état i avec $i \in \{S, D, I\}$ deux sous-ensembles d'états constitués mode(i,det) et mode(i,u).

Le premier, noté « mode (i, det) », est constitué des combinaisons de pannes détectables alors que le second, noté « mode (i, u) », est constitué des combinaisons de pannes non détectables. En utilisant les mêmes arguments que dans la partie 1-2-, on peut, en absence de détection et de réparation, proposer une chaîne de Markov non homogène similaire à la **figure 34**.

Les coefficients $\Gamma_{I/0}(t)$ $(i \in \{S, D, I\})$ de la **figure 33** sont donc remplacés dans la **figure 34** par deux coefficients de la forme $\Gamma_{I,d/0}(t)$ avec d={det, u}, correspondant respectivement à la possibilité d'entrée dans un état d'erreur détectable ou non détectable par le test. L'ajout d'un coefficient $\alpha_{i,u\to\det}(t)$ dans la **figure 34**, signifie la possibilité de passage entre ces deux modes. Il permet d'expliquer que l'apparition d'une panne supplémentaire sur un composant puisse rendre détectable un état d'erreur matérielle de la ressource antérieurement non détectable. On peut facilement vérifier par construction que $\Gamma_{I/0}(t) = \Gamma_{I,\det/0}(t) + \Gamma_{I,u/0}(t)$ pour tout $i \in \{S, D, I\}$ (**Equation 34**).

Le taux de couverture du test, noté DC_i, défini par le rapport $\frac{\Gamma_{I,\det/0}(t)}{\Gamma_{I/0}(t)}$ dans **[IEC61508]** et **[Gobl98]** n'est pas toujours constant, du fait de l'hétérogénéité, en terme de robustesse, des composants constitutifs de la ressource. La détermination des taux, notés $\alpha_{i,u\to\det}(t)$ dans la figure, sera évaluée au moyen des combinaisons de pannes suffisantes au passage d'un mode de défaillance indétectable à un mode de défaillance détectable.

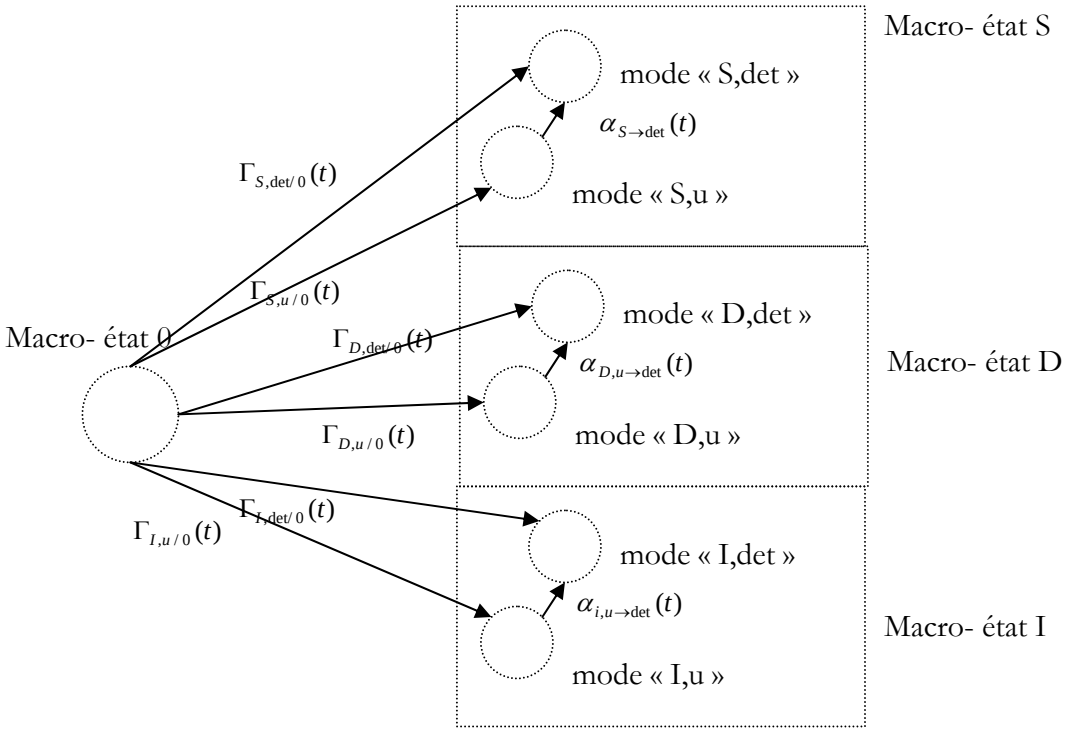


Figure 34 : Prise en compte des modes détectables.

Une fois ce modèle admis, on va désormais compléter la chaîne de Markov pour prendre en compte le procédé de diagnostic et de réparation. Pour ce faire, on va distinguer la notion d'erreur matérielle détectable et d'erreur matérielle détectée. Le passage de « détectable » à « détectée » se fait en deux étapes : la mise en œuvre de la procédure de test (action de périodicité m.T avec m entier supérieur à 1) et la réussite du test de suivi de la transmission de l'information au superviseur. La seconde étape est une étape de sollicitation – transmission de signal simple, elle peut donc être symbolisée par l'intervention successive d'un ensemble de ressources matérielles. Il est donc possible de déterminer les combinaisons d'erreurs de ces ressources menant à une défaillance de cette opération. La probabilité de cette combinaison est supposée constante sur la durée T.

On suppose que le MTTR est un nombre entier h de la période T. On pourra compléter le processus markovien précédent sous la forme suivante pour prendre en compte les opérations de diagnostic et de réparation corrective. L'hypothèse d'une faible variation de la probabilité de l'état grisé permet dans la figure 35 d'assimiler l'opération de réparation à un taux de transition constant de la forme 1/h, puisque h est un entier supérieur à 1.

Ce modèle (**figure 35**) est plus complexe que le modèle standard proposé par **[Sato03]** qui ne prend pas en compte l'échec possible de la procédure de détection. Il permet, en outre, l'évolution temporelle des taux $\Gamma_{I,\det/0}(t)$ et $\Gamma_{S,\det/0}(t)$, ce qui n'est pas possible dans les chaînes de Markov homogènes.

Il permet de traiter le cas d'une ressource correctivement maintenue. Cette construction s'est basée sur l'hypothèse de faibles variations des probabilités des états grisés, notés A et B, pendant la période de réparation, de durée estimée égale au MTTR (temps moyen de réparation). Ces hypothèses relativement réalistes pour des périodes de tests prises courtes du fait de la faible valeur des transitions équivalents, ne sont plus valables pour des périodicités longues (cas général pour les procédures de maintenance préventive). Il convient donc de proposer un autre modèle dans ce cas de figure.

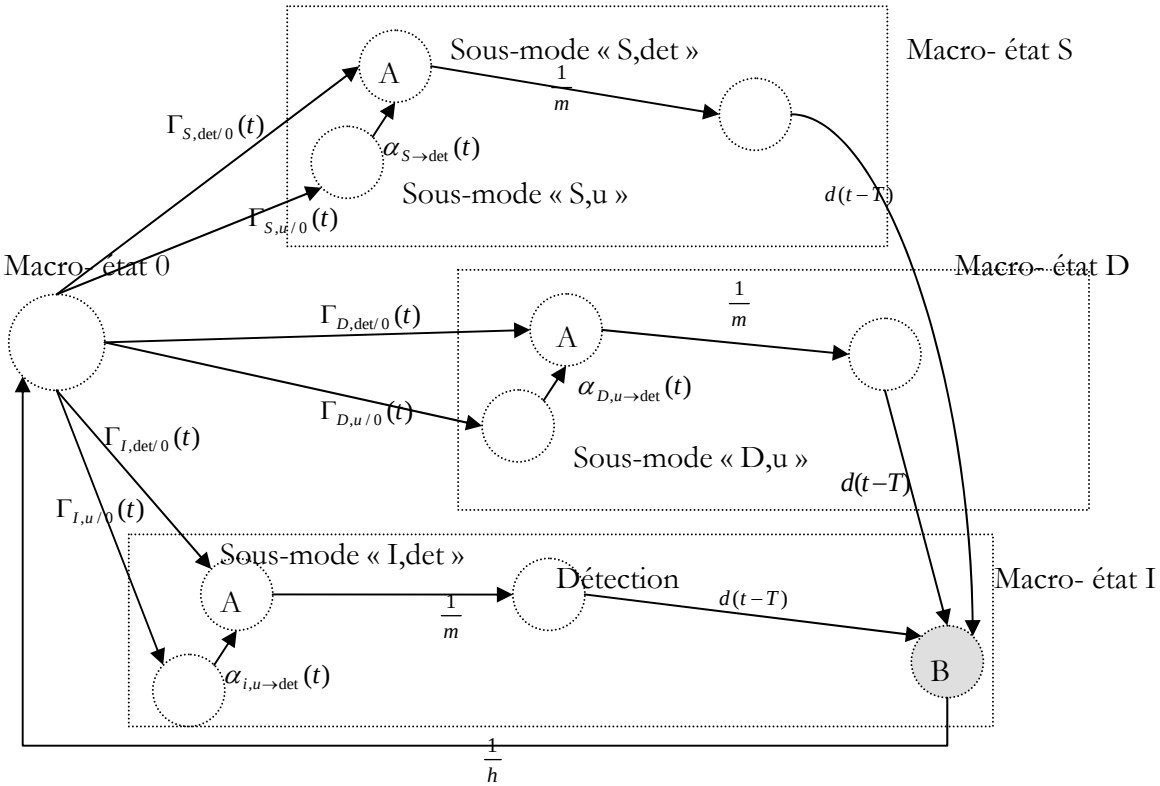


Figure 35 : Chaîne de Markov associée à une ressource maintenue

Il est nécessaire de proposer une extension de modèle pour prendre en compte l'ajout d'une procédure de maintenance préventive planifiée (de périodicité TI et de durée TR) pour une ressource matérielle. On peut optimiser la fiabilité d'un système en personnalisant les périodes d'inspection pour chaque ressource matérielle **[Zhao03]**. On supposera désormais pour une ressource **correctivement et préventivement** maintenue l'inégalité

$$T_{test} << MTTR << TI.$$

Notre proposition va se baser sur l'introduction d'une horloge locale permettant de prendre en compte les opérations planifiées de maintenance préventive.

1-4- Extension au cas de procédure de maintenance préventive sur la ressource matérielle

On suppose une opération de maintenance préventive de périodicité TI pour la ressource matérielle étudiée, de durée fixée TR et dont le succès est donné par une matrice R_L reliant la distribution des états d'erreurs avant la K-ième action de maintenance préventive à celle des états d'erreurs après cette action de maintenance préventive. Cette matrice R est carrée de même dimension que le nombre de lignes de P et vérifie la propriété selon laquelle la somme de ses termes suivant une même ligne (resp. une même colonne) est égale à 1.

On supposera que TI et TR sont des multiples entiers de T (dans le cas inverse on les approximera par le multiple de T leur étant le plus proche). On supposera que TI est fonction du nombre d'opérations de maintenance effectuées, on aura en toute rigueur défini TI(L) comme la période séparant la L-ième opération de maintenance préventive et la L+1-ième opération de maintenance préventive. TR sera supposée néanmoins fixée par soucis de lisibilité. On pourrait aussi la faire varier en fonction de L, en arguant une augmentation de la rapidité des opérateurs suivant l'âge de l'installation ou au contraire d'une difficulté croissante de maintenance. Cette

distinction entre phase de maintenance et phase opérative a bien été montrée dans **[Buko00]** et a été aussi souligné dans les modèles de Markov multi-phases **[Dutu02]**.

On peut distinguer deux modes de fonctionnement pour L variant de 0 au nombre total d'opérations de maintenance préventive.

Le premier mode de fonctionnement (mode 1) correspond à l'évolution normale de la ressource en dehors des phases de maintenance préventive, c'est-à-dire sur les périodes du type [τ(L) ; τ(L) +TI(L)[avec

$$\tau(L) = \sum_{i=1..L}(TI(i)+TR) = L.TR + \sum_{i=1..L} TI(i)$$ **(Equation 34)** (on note par abus $\sum_{i=1..0} TI(i) = 0$).

Sur ces périodes, l'état d'erreur de la ressource matérielle suit l'**équation 30** avec une matrice de transition A déduite de la figure 33, si la ressource est non correctivement maintenu et de la figure 30 sinon.

Le second mode de fonctionnement (mode 2) correspond à l'évolution de la ressource lors des phases de maintenance préventive, c'est-à-dire sur les intervalles de type [τ(L) +TI(L), τ(L+1)[si L ne correspond pas à la dernière action de maintenance et les intervalles du type [τ(L) ; T_{life}] sinon. Sur ces périodes, l'état d'erreur est bloqué sur le mode de défaillance I (déconnexion de la ressource). Le vecteur P(t) est donc un vecteur de probabilités contenant uniquement des 0 sauf pour le macro état d'erreur I, correspondant à une probabilité égale à 1. On notera P_I ce vecteur. On peut donc représenter l'évolution de notre processus stochastique par l'automate temporisé de la **figure 36**.

Cette extension garde les propriétés d'un processus stochastique. En effet, la somme des termes du vecteur P(t) est toujours égale à 1. Elle modélise de façon réaliste les deux modes de fonctionnement de la ressource durant et en dehors des phases de maintenance préventive. Ce type de modèle est facilement programmable du fait de son déterminisme et de la connaissance préalable de A(t).

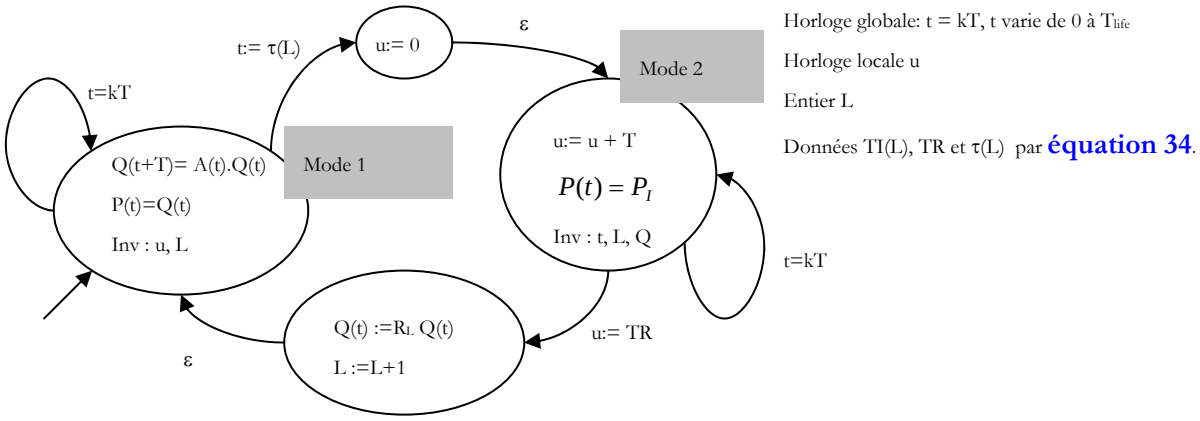


Figure 36 : Extension du modèle associé à une ressource préventivement maintenue.

Nous sommes donc capables de construire pour toute ressource matérielle un processus stochastique discret (de pas d'échantillonnage T) permettant de donner l'évolution des probabilités d'erreurs matérielles. Nous allons donc désormais discuter de la détermination des valeurs de nos taux de transitions équivalents notés précédemment par les variables $\Gamma_{.../0}(t)$ en fonction de l'évolution des stress environnementaux. Pour ce faire, il suffit d'expliquer le lien de causalité entre l'évolution des probabilités d'apparitions de pannes des composants constitutifs de la ressource étudiée et l'évolution temporelle des stress environnementaux, au vu des bases de

données existantes. Nous utiliserons ensuite l'**équation 33** pour déterminer les taux de transitions équivalents.

2- Détermination des relations entre taux équivalents et stress environnementaux

D'après les formules précédentes **(Equation 33)**, chaque taux de transition de type $\Gamma_{.../0}(t)$ correspond à une formule linéaire liant les différents taux d'apparition de pannes. Ces taux d'apparition peuvent être supposés constants sur les fenêtres d'observation [kT, (k+1)T[. A chaque période, on prendra une valeur moyenne des stress environnementaux.

La définition de taux d'apparition de pannes variant à chaque période T permet de prendre implicitement en compte l'existence de stress environnementaux et de leurs impacts sur ces taux de pannes. Actuellement, la littérature portant sur la fiabilité des composants privilégie une approche contrainte/résistance **[Snoo05]**. Elle considère que le taux de défaillance d'un composant (taux de pannes) résulte à la fois du niveau d'usure du composant et de l'exposition aux stress environnementaux.

On peut donc proposer une première équation portant sur le taux d'apparition de panne d'un composant, noté λ_c. On définit le taux d'apparition de panne d'un composant sur la période [kT,(k+1)T[, noté $\lambda_c(k)$ comme étant égale, sous l'hypothèse d'une sollicitation effective du composant sur cette période, comme la probabilité d'apparition d'une panne sur l'intervalle [kT, (k+1)T[.[7] La distinction principale avec le taux de défaillance de ressource matériel tel qu'usuellement défini provient du fait que nous permettons à la ressource de ne pas être utilisées dans certaines configurations fonctionnelles du système d'étude, une erreur matérielle ne sera donc pas nécessairement activée lors de son cycle d'apparition

Pour k fixé, on supposera alors l'existence d'une fonction Ω_c portant sur les stress environnementaux (Température, Pression, Niveau de rayonnement) auxquels le composant est soumis, permettant de proposer la relation générale suivante :

$$\lambda_c(k+1) = \alpha_c(k).\lambda_c(k) + \Omega_c(stess,k) \quad \textbf{(Equation 35)}$$

La fonction Ω_c(stress,k) établit un lien de causalité directe entre le niveau de stress auquel est soumis le composant au temps t=kT et son taux d'apparition de pannes. On rappelle que, par définition, les composants de nos ressources matérielles sont soumis à des facteurs de stress environnementaux dont le nombre est, en général, restreint. On peut classer ces facteurs de stress par catégories **[Maur00]** résumées dans le tableau suivant (**tableau 3**)

Mécanique et Thermique	Température, Poussière, Sollicitation (fréquence/amplitude), Vibration, Humidité
Electrique et Corrosif	Rayonnement Electromagnétique, Radiation, Mode de sollicitation (fréquence), Concentration (chimique)
Chimique	Corrosion (acide/oxydation)

Tableau 3 : Classification des stress environnementaux.

La **fonction α(k)** est liée au mode de sollicitation de la ressource. Elle permet la prise en compte d'une usure progressive du composant. **[Kaff01]**

La fonction **Ω_c(stress,k)** sera quant à elle évaluée hors du profil d'usure. Certaines des actuelles bases de données de taux de défaillances des composants comme **[FIDES]**, **[SN29500]**, **[MIL-HBK],** donnent ces fonctions sous forme d'un produit de facteurs correctifs liés aux différentes contraintes environnementales :

[7] Le terme panne est bien ici employé suivant la terminologie de l'**IEC61508.**

$$\Omega_c(stress,k) = F_1(T).F_2(P)....\lambda_{ref}. \quad \textbf{(Equation 36)}$$

où λ_{ref} est une constante et les valeurs des fonctions F_i portent sur chaque contrainte environnementale (elles sont données par des abaques correspondant à des tests de composants).

Les **bases de données** permettant le calcul du facteur correctif sont souvent issues d'un retour d'expérience important ou de tests. On pourrait discuter de l'incertitude de ces données **[Akme01]**, mais cette étude ne rentrera pas dans le cadre de ce travail.

On supposera qu'en général, les composants électriques et électroniques ne sont pas soumis à usure (α=0). Cependant nous devons considérer des composants électromécaniques au niveau des capteurs et des actionneurs (moteur, valve…). On tombe donc sous une hypothèse α non nulle, pour prendre en compte les phénomène d'usure. Il est donc utile de modéliser le phénomène d'usure.

On aura ainsi relié les taux de transitions de nos processus stochastiques à l'évolution temporelle des stress environnementaux auxquels nos ressources sont soumises. L'influence de ces contraintes sera liée à une approche zonale permettant de déterminer leur évolution, sous l'hypothèse d'un profil de mission déterministe de l'installation (absence de situation d'accident) et par maillage du lieu d'implantation des composants. Pour chaque ressource matérielle, il est possible de déterminer suivant un profil de mission, la variation des facteurs de stress auxquels l'ensemble de nos composants est soumis. Pour ce faire, on donne en général un point géographique, lié au lieu d'implantation de la ressource. On peut déterminer en ce point une estimation de l'évolution des facteurs de stress retenus en fonction du temps. Cette évolution sera donnée par découpage de notre environnement en zones d'influence (procédé de maillage) et par superposition des sources de fluctuations des stress en fonction du temps (perturbations supposées déterministes). Cette approche de découpage est très utilisée dans l'aéronautique **[Popo01]** mais peut facilement être étendue au monde industriel en identifiant les sources de perturbations de stress dans notre installation (source de chaleur, enceinte de pression,…).

En utilisant les **équations 33 et 35**, on peut relier directement les valeurs des matrices A(t) au stress environnementaux au temps t considéré (on rappelle que t=kT, temps discret) et aux valeurs de A(t-T). Par hypothèse, on a A(0-T)=0 (condition initiale) puisqu'aucun effet d'usure n'est constaté lors de la première mise en service du composant.

Pour chaque période T, on pourra évaluer les contraintes environnementales auxquelles les composants sont soumis, en déduire les taux de transitions de nos processus stochastiques puis évaluer leur évolution sur cette fenêtre. Ceci nous permettra de déterminer de manière discrète (pour chaque période T) et de manière dynamique l'évolution des probabilités d'erreurs matérielles. Nous prendrons donc implicitement en compte les couplages entre taux d'apparition de pannes dues aux contraintes environnementales auxquelles sont soumis des composants distincts (et donc les ressources auxquelles ils appartiennent).

3- Augmentation du temps T. Démarches et hypothèses.

Les processus stochastiques, précédemment construits, permettant l'évaluation dynamique des probabilités d'erreurs matérielles au cours du temps, possèdent un pas d'itération T petit. Ceci pose le problème de la rapidité du calcul du modèle général, proposé au chapitre 2. Il peut donc être intéressant de se demander sous quelles conditions ces processus stochastiques pourraient être remplacés par des processus stochastiques « équivalents » de pas d'itération temporel T*=LT (avec L entier positif).

Nous proposons donc de remplacer chaque modèle d'évolution des probabilités d'erreur des ressources matérielles, précédemment construit, par **un processus équivalent** de pas d'itération $T_i^*=k_i^*.T$ (avec k_i^* entier supérieur à 1). Le choix de T_i^* sera justifié. Nous

montrerons ensuite qu'un processus équivalent pourra être construit pour tout pas d'itération $T''_i \leq T_i^*$.

En définissant T*, comme la valeur minimale des T_i^*, nous serons en mesure de construire, pour chaque ressource matérielle, un modèle « équivalent » de pas d'itération temporel T* permettant de modéliser l'évolution des probabilités d'erreurs. On pourra donc connaître la probabilité des erreurs matérielles à chaque T*. En supposant que les probabilités d'erreurs environnementales varient peu sur ces intervalles, il sera possible d'utiliser les modèles « équivalents » pour simuler l'évolution des probabilités d'erreurs des ressources matérielles. Il sera donc possible d'estimer pa et pd (cf. chapitre 2) à chaque T*. On pourra ainsi réduire le temps de calcul de l'algorithme d'estimation des indicateurs de performance proposé au chapitre 2, en remplaçant T par T*.

3-1- Conditions d'élargissement du pas d'itération des modèles et du modèle équivalent.

Nous allons volontairement nous placer dans le cas le plus général d'une ressource matérielle correctivement et préventivement maintenue. Cette ressource sera notée i. Certaines hypothèses notées par le sigle (P) ne seront à vérifier uniquement dans le cas d'une ressource matérielle préventivement maintenue, tandis que celles notées par le sigle (C) seront à vérifier pour une ressource matérielle correctivement maintenue. Les autres hypothèses, notées (G), s'appliquent à toutes les ressources maintenues ou non maintenues.

Nous nous intéressons au cas général d'une ressource matérielle dont l'évolution probabiliste des états d'erreurs est donnée par l'automate **figure 36**. On désire remplacer ce processus stochastique par un processus stochastique équivalent de largeur $T_i^*=k_i^*.\ T$. On cherche donc à l'approximer par le modèle de comportement « équivalent » suivant **figure 37**.

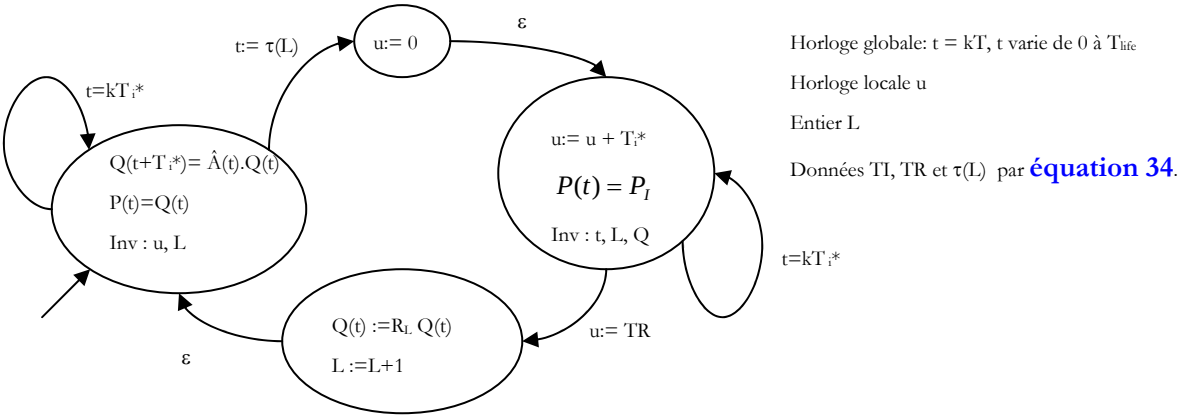


Figure 37 : Modèle équivalent recherché.

On doit donc vérifier :

- que TI(L) est un multiple de T_i^* (pour assurer la transition 0=>1) ; **(P1)**
- que TR est un multiple de T_i^* ; **(P2)**
- et que le processus $P(t+T_i^*)= \hat{A}(t).P(t)$ est équivalent au processus $P(t+T)= A(t).P(t)$. **(P3)**

Vu les ordres de grandeur de TI et TR permettant d'affirmer que TR<<TI, on peut réduire les points (P1) et (P2), à l'hypothèse $\exists L \in \mathrm{N}, L \geq 1$ vérifiant $L.T_i^* \approx TR$. On peut donc affirmer que $Ti^* \leq TR$.

Le traitement du point **(P3)** est, en revanche, beaucoup plus intéressant. Nous supposons connues les contraintes environnementales au temps t, et nous souhaitons déterminer une matrice « équivalente » de transitions Â(t) permettant de vérifier le point (P3).

Si on observe les taux de transitions de la matrice A(t), donnée par le graphe de la partie 1-3, on remarque trois classes de transitions :

1. les transitions correspondant à la panne d'une séquence de composants, notées avec les symboles Γ.. et dépendant à ce titre uniquement du niveau d'usure de ces composants (et donc des fonctions alpha) et des contraintes environnementales auxquelles ils sont soumis au temps t=k.T.
2. les transitions correspondant à des contraintes de temps, notées 1/m et 1/h (pour le temps de test et le MTTR)
3. les transitions liées à la probabilité d'erreurs matérielles au cycle précédent, notées d(t-T).

On supposera que les contraintes environnementales doivent évoluer assez lentement sur les intervalles de largeur T_i* et que le facteur d'usure α des composants (cf chapitre 3-2) y est nul. On peut donc considérer les taux de transitions Γ.. du processus Markovien de départ, comme constants sur cet intervalle. $\Gamma_{i/0}$ est, en effet, une fonction linéaire d'un nombre fini de taux de défaillances λ_c définis par des fonctions récursives du type

$$\lambda_c(k+1) = \alpha_c(k).\lambda_c(k) + \Omega_c(stess,k) \ .$$

On choisira l'intervalle τ_i*= N_i T (avec m entier) maximum vérifiant ces hypothèses en fonction des profils de contraintes retenus et d'une étude de sensibilité des taux de pannes de la ressource matérielle contenus dans $\Gamma_{...}$. On rappelle qu'on a supposé dans la partie 1-2 que h<m. On comparera ensuite la valeur N_i avec les valeurs m et h, c'est-à-dire avec les rapports MTTR/T et Ttest/T. Deux cas de figure sont alors possibles :

* soit N_i <h<m, on prendre alors τ_i**= N_i T (cas 1)

* soit h< N_i , on prendra alors τ_i**= h T (cas 2)

On note dans la suite M_i le nombre entier prenant soit la valeur N_i dans le cas 1 ou h dans le cas 2., Si on suppose la variation du macro-état 0 faible sur τ_i**, c'est-à-dire que les produits M. $\Gamma_{...}$ sont négligeables devant 1, pour tout entier j appartenant à [1,M] et pour tout K multiple de M, en notant t=M_iT, on aura donc l'égalité :

$$P(t+jT) = A(t).P(t+(j-1)T) \quad \textbf{(Equation 37)}$$

On en déduira, par récurrence en dehors des phases de maintenance préventive, le modèle « équivalent » suivant (donnant l'équation d'état Q(t+T_i*)= Â(t).Q(t)), possédant un pas temporel discret de longueur T_i*=M_i.T (**figure 38**).

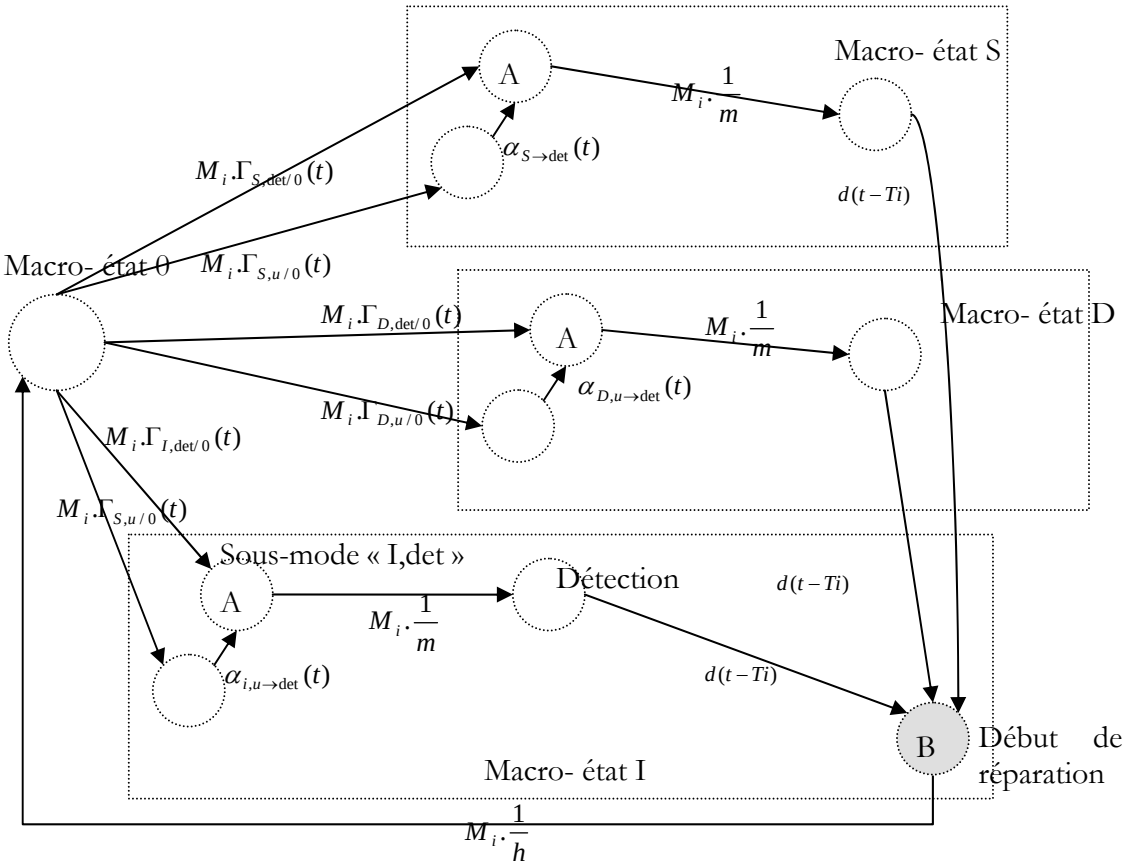


Figure 38: Chaîne de Markov équivalente en dehors des périodes de maintenance

On a ainsi déterminé la matrice Â(t) qui manquait à l'établissement de notre modèle équivalent, définie comme un processus stochastique de pas temporel discret $t=M_i.T$. Cette détermination permet de construire le processus stochastique équivalent global, en accord avec la **figure 38**.

3-2- <u>Modèles équivalents et réduction du temps de calcul des indicateurs de performance.</u>

Dans la sous-partie précédente, nous avons vu sous quelles hypothèses il était possible de remplacer les processus stochastiques construits dans la partie 1 de ce chapitre, de pas discret temporel T, par des processus stochastiques « équivalents », c'est à dire conservant un profil réaliste de l'évolution des états d'erreurs matérielles au cours du temps, de pas discret temporel T_i*.

Nous avons proposé cette démarche dans le cas d'une seule ressource matérielle i. Elle peut être appliquée à l'ensemble des processus stochastiques attachés à chaque ressource matérielle. En outre, on remarque que la démarche reste valide pour tout entier $k<M_i$ non nul et que la matrice Â équivalente est alors déterminée en remplaçant M_i par k dans ces taux de transitions.

On a aussi expliqué comment les entiers M_i peuvent être déterminés pour chaque ressource suivant les hypothèses suivantes :

* stationnarité des contraintes environnementales sur les intervalles de largeur M_i T ; (G)

* fonction d'usure α constante de ses composants sur cette période ; (G)

* majoration de M_i T par la valeur du temps d'inspection et de réparation préventive de la ressource matérielle TR (sous l'hypothèse supplémentaire TR<<TI) ;
(P)

* majoration de M_i T par le MTTR et la période de test en ligne de la ressource ; (C)

* inégalité $\Gamma_{U/0}(t)$ M_i T <<1 , pour tout $u \in \{S, D, I\}$ (Hypothèse de pannes « rares »).
(G)

On peut ainsi déterminer pour chaque ressource i, le nombre entier M_i supposé maximum. Ce maximum n'est pas forcément le maximum réel car il est lié au choix de la largeur maximale d'une fenêtre temporelle où les taux de pannes des composants des ressources sont supposés constants. On est, en outre, capable pour tout entier non nul k inférieur ou égal à M_i de construire le processus stochastique « équivalent », de pas temporel discret de valeur kT, donnant une estimation réaliste de l'évolution des probabilités d'erreurs matérielles de cette ressource. Pour ce faire, il suffit de remplacer la valeur M_i par k dans la **figure 38**.

Si on note E l'ensemble des ressources matérielles modélisées (c'est-à-dire pour lesquelles au moins une défaillance appartient à l'alphabet des langages La et Ld), on peut donc définir la valeur :

$$M = Min_{i \in E}(M_i). \quad \textbf{(Equation 38)}$$

M étant un minorant des M_i déterminé pour l'ensemble des modèles attachés aux ressources matérielles. Il est possible de construire pour chaque ressource le modèle équivalent d'évolution de l'état d'erreurs matérielles en prenant la valeur T_i*=M.T=T*.

On peut, en utilisant ces modèles, évaluer l'évolution des probabilités d'erreurs matérielles de notre système avec un pas temporel de largeur MT. Il est donc possible de remplacer le modèle d'évaluation des indicateurs de performances par un modèle équivalent, où les modèles du niveau 2 seraient remplacés par ces modèles équivalents. Il est ensuite possible d'appliquer l'algorithme de calcul proposé dans le paragraphe 3-4 du chapitre 2 en remplaçant T par MT. Cette approximation reste valable si T* reste faible devant Tlife, car elle revient à remplacer une intégrale d'une fonction en escalier de pas T, par l'intégrale d'une fonction en escalier de pas T*, obtenue par échantillonnage et blocage d'ordre 0 de celle-ci tous les MT. Cette approximation est justifiée par l'hypothèse de « pannes rares » précédemment donnée. Cette simplification du modèle nous permet de réduire le temps de calcul de nos indicateurs de performances dans un rapport M.

4- Conclusion de la partie 4

Nous avons vu, dans cette partie, comment les processus stochastiques permettant de décrire l'évolution de la probabilité d'erreur des ressources matérielles étaient construits dans le cas général de ressources matérielles maintenues correctivement et préventivement (partie 1). Nous avons, en outre, expliqué les relations entre les matrices de transitions de ces processus stochastiques et l'évolution temporelle des stress environnementaux, dans le cas d'un profil déterministe d'évolution de ces contraintes (partie 2). Nous nous sommes enfin intéressés à la détermination de processus stochastiques dont l'évolution est jugée équivalente suivant une période d'observation plus large (MT).

Cette extension nous permet de modifier le modèle général d'évaluation (chapitre 2) en remplaçant la période de calcul T par une période de calcul MT, tout en conservant les mêmes listes La et Ld et sans sacrifier la qualité des estimations obtenues (hypothèse MT<<T_{life}). Elle nous permet donc d'effectuer ces estimations en un temps de calcul plus raisonnable et

d'appliquer cette méthode à des cas d'études de dimension réaliste. L'augmentation du pas de simulation est justifiée par trois points principaux :

- fluctuations faibles des contraintes environnementales
- conservation des hypothèses de base sur les limites des taux de transitions des modèles markoviens [0;1]
- événements « rares » pour les mécanismes de dégradation (remplacement de l'exponentiel par son développement limité d'ordre 1) ou contraintes temporelles (MTTR, TI, T_{test}) multiples du nouveau temps de référence.

Nous nous proposons dans la partie suivante de mener ces calculs sur un cas d'étude particulier et de comparer les estimations d'indicateurs de performances obtenues par notre méthode avec celles obtenues par une méthode usuelle du monde industriel (RBD/graphes Markov). Nous montrerons ensuite les intérêts et la possibilité d'utilisation de la méthode de construction des listes (chapitre 3) pour une architecture plus complexe et conclurons sur les limites et les avantages de mise en œuvre de la méthode proposée.

Chapitre 5

Utilisation de la méthode sur cas d'étude

Chapitre 5 - Utilisation de la méthode sur cas d'étude

Ce chapitre a pour but d'illustrer notre méthode d'évaluation. Pour ce faire, nous partirons d'un cas d'étude très simple, basé sur une chaîne de réaction directe capteur- contrôleur- actionneur. Nous construirons ses listes caractéristiques Ld et La et nous discuterons des différences entre ces listes et les combinaisons de défaillances considérées dans une approche de type diagramme de fiabilité. Cette comparaison nous permettra de commenter les différences numériques obtenues sur l'évaluation des indicateurs de performance de cette architecture.

Nous rendrons ensuite cette architecture de base plus complexe par l'ajout d'une redondance capteur et d'un protocole de communication (maître-esclave avec temps limite de réception d'informations). Nous discuterons des changements apportés par la construction de nos listes caractéristiques et montrerons au vu de certaines listes intermédiaires L1, L2 et L3, que notre méthode permet bien la prise en compte des problèmes de partage de ressources matérielles et informationnelles dans l'architecture. Nous en déduirons certaines propriétés en termes de longueur des listes finales obtenues et expliquerons leur manipulation ultérieure à des fins d'évaluation de nos indicateurs de performances.

Nous complexifierons ensuite davantage l'architecture en prenant en compte la possibilité d'actionneurs redondants et de tests périodiques commandés sur l'un d'entre eux (vanne périodiquement sollicitée). Nous en déduirons, en accord avec les listes caractéristiques de ce nouveau système, l'intérêt et les limites de tels tests et discuterons de l'impact de la périodicité du test sur la variation des performances de notre système d'étude.

Afin de montrer que cette méthode est généralisable pour des architectures plus complexes, nous construirons le modèle de haut niveau d'un système réel basé sur l'utilisation d'un vote majoritaire et d'une logique 1oo2. Cet exemple est tiré de la littérature et permet de démontrer que notre méthode est généralisable pour un large spectre de fonctions EP dédiées sécurité. Sans retranscrire l'ensemble des listes caractéristiques obtenues, nous discuterons de leur longueur et de leur utilisation ultérieure à des fins d'évaluation numérique. Nous en déduirons que l'évaluation dynamique proposée au chapitre 2 n'est possible que si les listes obtenues sont interfacées avec un module de calcul. Nous expliquerons comment cet interfaçage peut être effectué avec Aralia©, dans le cas de stress environnementaux constants et comment généraliser cette approche au cas plus général de stress environnementaux non constants.

1- Cas d'étude de départ.

Le cas d'étude de départ est extrêmement simple. Il se base sur une fonction de détection d'un seuil de température dangereux pour un réacteur. En cas de détection de cette situation dangereuse, l'alimentation du réacteur en comburant doit être fermée par l'intermédiaire d'une vanne de sécurité mise en série avec la vanne de contrôle normal du flux d'alimentation. On peut donc représenter cette architecture par le simple schéma de principe suivant (**figure 39**).

Le capteur de température est relié à un module d'entrée numérique décentralisé monté sur un rack (Siemens **ET200M** pour exemple d'un tel composant). Ce module est relié à un microprocesseur, via une interface de communication et un bus. Il doit assurer la comparaison de la valeur observée avec un seuil défini, correspondant au seuil de dangerosité, en termes de température. Cet organe de contrôle est ensuite relié via le bus à un module de sortie analogique. Ce dernier permet la fermeture ou l'ouverture d'un relais, commandant le sens d'alimentation du moteur de la vanne de sécurité et pouvant exiger soit sa fermeture, soit son ouverture. En présence d'une situation dangereuse, le cahier des charges de la fonction de sécurité impose la fermeture de la vanne de sécurité. En absence d'une situation dangereuse, la vanne de sécurité

doit rester ouverte (l'ordre d'ouverture reste actif). On remarquera que les modules d'entrée numérique et de sortie analogique sont tous les deux alimentés par une source de tension commune, puisqu'il s'agit de la source d'alimentation du rack décentralisé. On supposera pour le moment que le protocole de communication entre le PLC et ses périphériques (modules d'entrée et de sortie) est extrêmement simple, puisque le PLC va émettre à chaque période T une requête d'acquisition au module d'entrée et va émettre, suite au processus de décision (comparaison de la valeur de température acquise avec la valeur seuil), un ordre de contrôle envoyé au circuit de contrôle de la vanne. Nous ne nous intéresserons donc pas pour le moment à la possibilité d'implantation d'une horloge assurant l'actualisation de la donnée reçue, et supposerons que l'information transmise n'est pas redondée (absence de télégrammes de codage de type CRC, de sommes de contrôle (Check-Sum) ou d'émissions répétées d'informations). En outre, nous supposerons connue la probabilité de perte d'informations dans le réseau de communication à chaque émission d'information par un périphérique (notée p) et nous supposerons qu'aucune erreur d'adressage n'est possible vis-à-vis des périphériques.

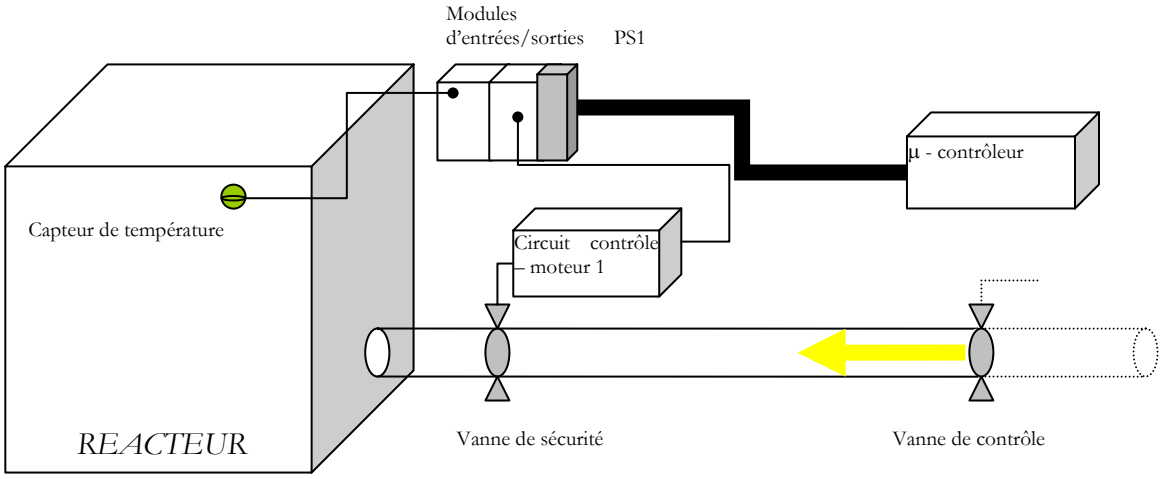


Figure 39 : Schéma de principe de l'architecture de base

Dans ce cas, extrêmement simple, l'architecture fonctionnelle peut être représentée par le modèle de haut niveau suivant (**figure 40**) :

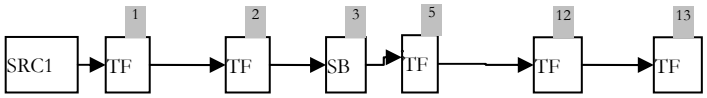


Figure 40 : Graphe de fiabilité (RBD)

Les automates d'états finis correspondant à chaque bloc ainsi que les données numériques de taux de défaillances sont donnés dans l'**annexe B11**.

On supposera pour effectuer une comparaison effective entre la méthode proposée et la méthode usuelle (diagramme de fiabilité/graphes de Markov homogènes) que les stress environnementaux sont constants et que nous ne prenons pas en compte les critères d'usure des ressources matérielles. L'approche par diagramme de fiabilité pour l'estimation de PFD_{avg}, est basée sur le schéma fonctionnel suivant (**figure 41**) où chaque bloc est donné par un graphe de Markov homogène donné dans l'**annexe B12**.

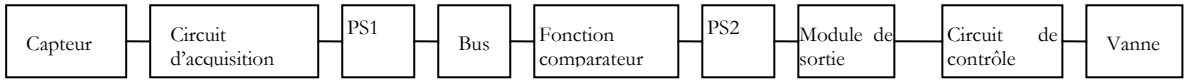


Figure 41 : Diagramme de fiabilité.

Pour le moment, nous supposons une absence de processus de maintenance préventive. Pour prendre en compte ces processus, il suffira de remplacer le graphe de Markov des ressources matérielles par l'automate d'état du **chapitre 4, 1-4, figure 36.**

Le PFD instantané pour une période d'acquisition t=kT fixée sera donc évalué comme la réunion (assemblage en série dans le formalisme RBD) des propositions suivantes :

E1 : x1 (capteur) est dans le mode de défaillance I au temps t ;

E3 : x3 (circuit d'acquisition) est dans le mode I ou (exclusif) dans le mode D (somme) au temps t ;

E7 : x7 (alimentation du rack – PS1) est dans le mode I au temps t ;

E5 : x5 (bus) est dans le mode I au temps t ;

E10 : x10 (comparateur) est dans le mode D au temps t ;

E8 : x8 (alimentation du PLC- PS2) est dans le mode I au temps t ;

E15 : x15 (module de sortie analogique) est dans le mode I ou (exclusif) dans le mode D (somme) au temps t ;

E16 : x16 (circuit de contrôle) est dans le mode I au temps t;

E17 : x17 (moteur +vanne) est dans le mode D ou le mode I au temps t.

Cette probabilité est calculée par une approximation de Poincarré d'ordre 2.

On peut donc facilement en déduire la moyenne de cette probabilité en connaissant les taux de défaillances et de réparation de ses composants, en utilisant par exemple l'outil Bloc de fiabilité de ©Jagrif (pour désignation et taux de défaillances des ressources matérielles se reporter à **Annexe B13**). Grâce à cet outil qui permet d'associer à chaque bloc un graphe de Markov homogène, en accord avec les données numériques données en annexes, on aura :

$$PFD_{avg}=3{,}163\ E\text{-}2.$$

Dans le cas d'une construction systématique de nos listes caractéristiques (chapitre 3), suivant les automates finis donnés en **annexe B11,** on obtiendra les listes données en annexe **B14** par l'utilisation de notre outil d'identification (AFMCI). Connaissant l'évolution des modèles markoviens associés à chaque ressource matérielle, on peut évaluer de la même manière la probabilité de l'ensemble de ces listes. On obtiendra alors, pour les mêmes valeurs de taux de défaillance, la valeur :

$$PFD_{avg}=2{,}783\ E\text{-}2.$$

Ce calcul est effectué, en générant à partir des listes de l'annexe B14, sous **©Jaralia** un arbre logique dont les feuilles, correspondant aux erreurs activées. Ces feuilles ont pour attribut une fonction temporelle (valeur numérique) dont la valeur correspond à la probabilité de cette erreur. Cette probabilité est calculée par itération de la chaîne markovienne de la ressource matérielle correspondante (cf. chapitre 4 pour construction).

On peut remarquer que les séquences sont deux à deux distinctes et qu'aucun mot d'une liste caractéristique ne peut être, en même temps, sous-séquence d'un autre mot de cette liste. Pour des raisons de lisibilité, nous avons supprimé certains caractères de type d(xi,0) dans les listes finales données en annexe, leurs probabilités étant très proche de 1. Ceci impliquerait une faible erreur de calcul si nous travaillons avec les séquences données en annexe (de l'ordre d'un facteur multiplicatif d'environ 10^{E}-5). Cependant le calcul a été effectué avec les séquences complètes directement exportées du logiciel AFMCI à Jaralia.

La différence entre ces deux valeurs de PFD_{avg}, de l'ordre de 15% pour cet exemple, peut être expliquée du fait que l'architecture choisie est non cohérente (en termes de structure fiabiliste) et que la combinaison simultanée de différents modes de défaillances matérielles peut mener à l'inhibition d'une erreur, situation non prévue par la méthode classique. Le cas d'un blocage en position fermée de la vanne contribue par exemple à la réduction de la valeur trouvée (justifiant à lui seul une différence d'environ 8%). On peut cependant remarquer que dans le cas d'un flux unique d'informations (absence de processus de décision), la probabilité moyenne évaluée par la méthode RBD/Markov majorera toujours celle proposée par notre méthode puisque chaque combinaison de Ld contient au moins un des modes de défaillances de ressources matérielles (D ou I) retenus dans l'analyse par l'approche RBD/Markov.

On voit donc que même pour un exemple très simple, notre méthode apporte des résultats différents liés au caractère non cohérent de l'architecture considérée (existence de mode S et D sur les ressources entrant en jeu). Ce premier résultat est très important et montre l'intérêt d'utiliser une telle méthode pour des architectures dont les ressources matérielles possèdent des modes de défaillances autres que ceux se traduisant par la perte d'une information le long d'un flux d'informations unique (i.e. de modes de défaillances différents des modes de défaillances de type I). Elle permet de prendre en compte les propriétés de non cohérence de l'architecture et la possibilité, du fait de l'apparition d'un mode de défaillance local, d'inhiber la propagation d'une erreur d'informations le long d'un flux.

Nous allons désormais discuter de l'intérêt de notre méthode pour prendre en compte de manière réaliste le partage de ressources matérielles et informationnelles dans des voies d'acquisition redondantes et son utilité pour modéliser les contrôles temporels au niveau d'échanges d'informations (via un bus de communication). Nous proposerons pour ce faire, sur la base de notre architecture de départ, une architecture plus complexe et expliquerons l'intérêt de notre modélisation dans la prise en compte de partage de ressources entre sous fonctions.

2- Ajout d'une redondance capteur et d'un contrôle temporel pour la communication

Nous nous intéresserons dans ce paragraphe à la prise en compte par notre méthode de l'ajout de redondances et de contrôles temporels de communication. L'utilisation de redondances informationnelles dans une architecture est très fréquente. Elle se traduit, en général, par le doublement d'un processus de traitements d'informations, c'est-à-dire d'une séquence de sous fonctions partageant les mêmes informations d'entrées et devant fournir une information de sortie équivalente qui servira par la suite à un processus de décision (vote). Le diagnostic qui s'en suit peut aboutir soit sur la mise en mode dégradé du système, c'est-à-dire la mise à une valeur par défaut d'une information, soit par sa reconfiguration fonctionnelle, c'est-à-dire sur un processus de décision basé sur un sous ensemble d'informations disponibles.

Les politiques de reconfigurations fonctionnelles, comme, par exemple, le passage d'un mode majoritaire 2oo3 à un mode de fonctionnement dégradé 1oo2, ou de mise en mode dégradé, comme la mise à une valeur par défaut d'un signal, sont nombreuses et peuvent être rencontrées sous différentes formes suivant les choix d'architectures fonctionnelles avancées pour une fonction de sécurité particulière. Nous allons dans cette partie, nous intéresser à l'ajout

d'une voie de redondance pour les voies d'acquisition de l'état de demande de notre cas d'étude. Pour ce faire, nous allons considérer le schéma de principe suivant, basé sur le cas d'étude du paragraphe 1 (**figure 42**).

On ajoute à l'architecture précédente une nouvelle voie d'acquisition (voie d'acquisition duale) au moyen d'un autre capteur de température. Ce capteur est lié **au même** module d'entrée analogique (voies parallèles possédant des éléments communs d'amplification et de codage CNA). Ce module est intégré dans le même rack que le précédent cas d'étude. Les informations acquises par les deux capteurs sont donc fournies de manière séquentielle au PLC pour chaque période d'acquisition (deux registres d'entrées). On suppose, en outre, l'implantation d'une horloge de contrôle (watch-dog) vérifiant la bonne actualisation des registres d'entrées. En cas d'absence de réception d'une de ces informations après un temps limite (temps maximum admis de transmission du signal), on assignera une valeur de température supérieure à la valeur de seuil de dangerosité. Le microcontrôleur utilise ensuite ces deux informations en les comparant au seuil de température (condition de demande). Il délivre la valeur 1 si au moins une de ces informations indique un dépassement du seuil de température du réacteur et 0 sinon. Ces signaux analogiques sont ensuite utilisés par une porte logique OU (vote 1oo2), le signal de sortie 1 exigeant un signal de contrôle en fermeture de la vanne.

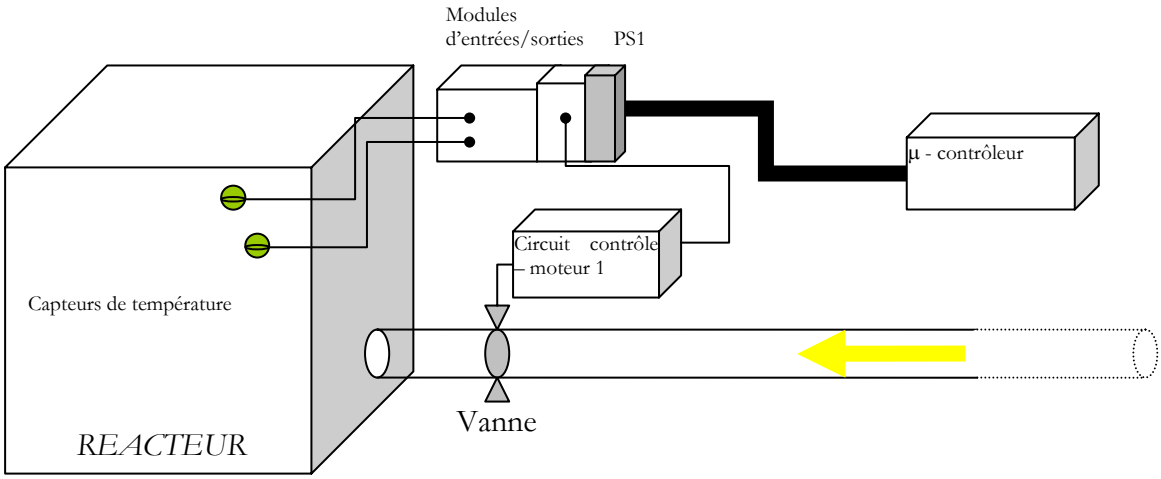


Figure 42 : Schéma de principe de l'architecture considérée.

Ce système d'étude peut être modélisé par le modèle de haut niveau suivant (**figure 43**):

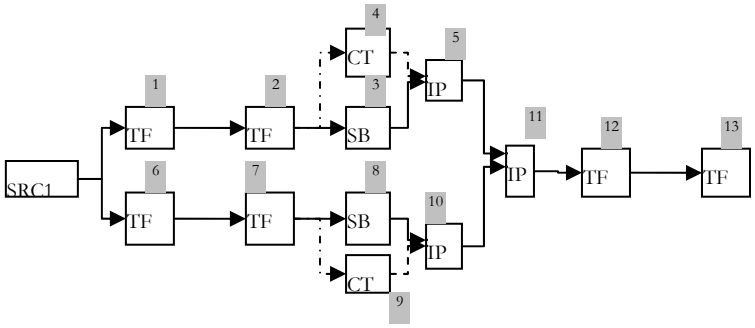


Figure 43 : Modèle de haut niveau.

Dans le cas d'une construction systématique de nos listes caractéristiques (chapitre 3), suivant les automates finis donnés en **annexe B21,** on obtiendra facilement les listes caractéristiques complètes par l'utilisation de notre outil d'identification (AFMCI).

Pour monter l'intérêt de notre méthode pour la prise en compte de la redondance et de l'horloge de contrôle, on va s'intéresser aux listes intermédiaires L1, L2 et L3 identifiées après l'agrégation du bloc 11. Ces listes sont données en **annexe B23**.

On peut comparer ces listes aux calculs effectués en utilisant une méthode diagramme de fiabilité/ graphe de Makov homogène classique. Si on néglige le couplage entre les modes de défaillances des deux capteurs (dus aux stress environnementaux), on pourra, suivant cette méthode, décrire l'architecture grâce au modèle suivant (**figure 44**) :

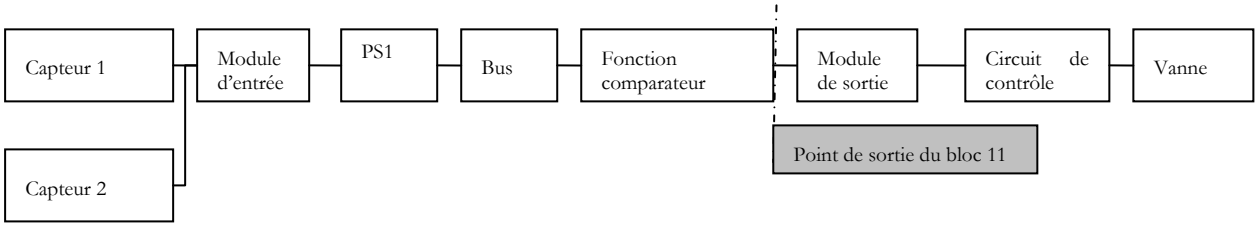


Figure 44 : Diagramme de fiabilité.

En considérant la partie en amont de la sortie du bloc 11, on pourra calculer la probabilité d'une information erronée en sortie de ce bloc au cycle d'acquisition t=kT, pour deux capteurs équivalents en termes de taux de défaillance mais non sensible à des stress environnementaux partagés.

Dans le cas d'une absence de signal par un des capteurs (mode de défaillance I), on peut constater en identifiant nos listes caractéristiques grâce à l'outil AFMCI, que l'absence d'un signal d'acquisition sera souvent détectée par l'horloge de contrôle (inhibition d'information erronée) et aboutira à des combinaisons différentes de modes de défaillances, données dans l'**annexe B22**. On peut remarquer au vu de ces listes deux aspects.

Le premier est que l'utilisation des règles de réduction et d'agrégation des langages se traduit bien par la présence d'erreurs matérielles non conflictuelles. On prend donc bien en compte le **partage des ressources** (modules d'entrées, PS1, Bus…) dans le calcul des listes, ce qui n'est pas toujours le cas si on applique la méthode diagramme de fiabilité dans des architectures avec partage de ressources. Les événements: «défaillance en mode I du bus », « défaillance en mode I de PS1 »… qui apparaissent au niveau des blocs séries du diagramme de fiabilité sont donc bien des conditions suffisantes à l'apparition d'un mode de défaillance sous demande du système.

Le second aspect est qu'**une information non erronée** correspondant **à une absence de demande** n'est **pas vue comme une absence d'informations par le PLC**. Cet aspect explique bien le fait que l'absence d'une information fournie par les capteurs se traduit très souvent par la mise en position sûre du système, ce qui n'apparaît pas dans les méthodes actuelles d'évaluation. Cette procédure de détection supplémentaire est souvent omise dans les méthodes d'évaluation de type RBD/Markov, ce qui peut expliquer, dans certains cas (non partage de ressources matérielles hors des voies redondantes) une surestimation de la valeur moyenne du PFD.

La mise en œuvre pratique du calcul des indicateurs de performances (PFD_{avg} et PFS_{avg}) est possible. Cependant, on peut remarquer que même si les contraintes de partages de ressources matérielles entre sous fonctions tendent à diminuer la taille de ces listes, celles-ci peuvent devenir importantes pour un nombre plus important de ressources matérielles entrant en jeu dans une architecture complexe. On propose donc de convertir directement chaque liste caractéristique obtenue, en une opération de sommation sur l'ensemble des mots, du produit de l'ensemble des événements, correspondant chacun à une erreur (matérielles ou environnementales) activée. L'évolution de ces variables temporelles (probabilités d'erreurs) pourra ensuite être estimée :

- soit dans le cas de stress environnementaux constants, en utilisant des outils numériques comme l'outil **©Aralia ou ©Fault Tree+** qui permet de donner le profil d'évolution de chacune de ces variables suivant le graphe de Markov retenu,
- soit, dans le cas de stress variants avec le temps, en utilisant les algorithmes généraux de couplage entre processus stochastiques discrets (cf. chapitre 4), ce qui nécessite l'implantation d'un outil informatique spécifique (modèles dynamiques).

On pourra ainsi calculer les PFD et PFS pour chaque cycle T* et en déduire les valeurs de nos indicateurs de performances. Pour nos cas d'étude, du fait de la longueur raisonnable des listes caractéristiques, ces calculs ont été effectués par un couplage manuel avec l'outil **©Moca-RP**, en supposant les stress environnementaux et donc les taux de défaillances de ressources comme constants au cours du temps. Un calcul pour des taux de défaillance variant au cours du temps, comme proposé dans les modèles stochastiques du chapitre 4 est aussi possible, mais il impose une amélioration de l'outil informatique existant. Il n'a pas été effectué mais sa réalisation future pourrait se baser sur les outils de simulations itératives de chaînes de Markov tel que **©CARMS** (Excel), en couplant les taux des matrices de transitions des processus Markoviens par une interface en **©Matlab**, donnant l'évolution de ces taux et les profils de stress environnementaux.

3- Ajout d'une redondance actionneur et d'une politique de test périodique

L'ajout d'une politique de test périodique dans une architecture pour assurer des propriétés de tolérance ou de prévention d'erreurs matérielles dans une architecture est un procédé usuel. Nous allons donc proposer une extension de l'architecture précédente en implantant non pas une mais deux vannes en série permettant la fermeture de la conduite d'alimentation de comburant (cf. **figure45**). On propose d'implanter un test périodique sur la vanne dont la fiabilité moyenne est jugée la plus faible (vanne 1). Ce test consiste à solliciter cette vanne périodiquement par l'émission d'un ordre de contrôle par le PLC et d'observer ensuite la position de la vanne par un capteur de position, implanté au niveau de ses battants, qui va permettre de juger de la bonne réalisation de la fonction à une durée fixée après la sollicitation. Cette durée permet de prendre en compte le temps normalement nécessaire à la transmission de l'ordre de contrôle et au temps maximal nécessaire à la fermeture de la vanne en absence de défaillance. En cas de défaillance détectée en fermeture de la vanne 1, le PLC forcera systématiquement la fermeture de la vanne 2. On a donc le schéma de principe suivant :

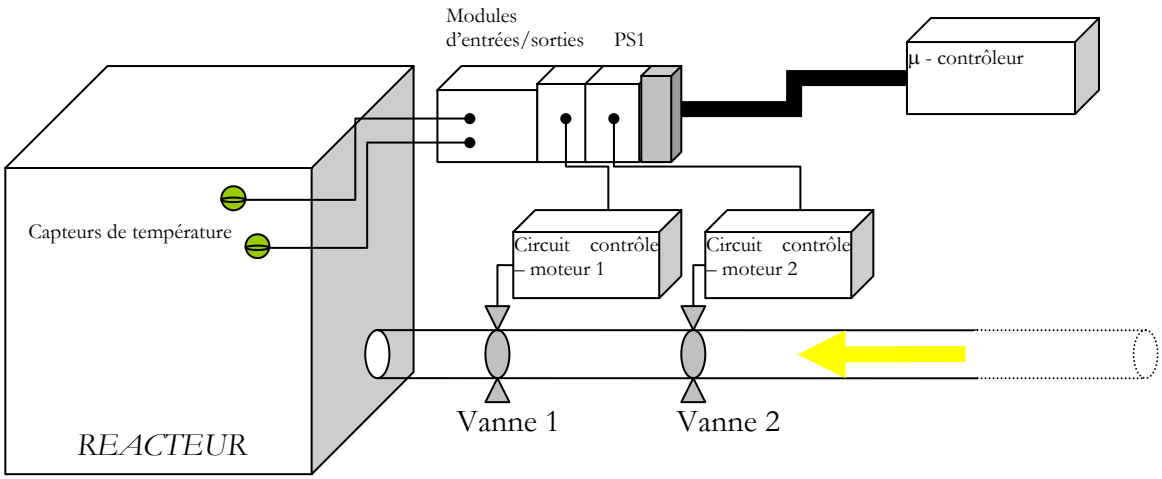


Figure 45 : Schéma de principe de l'architecture considérée.

Ce système d'étude peut être modélisé par le modèle de haut niveau suivant (**figure 46**):

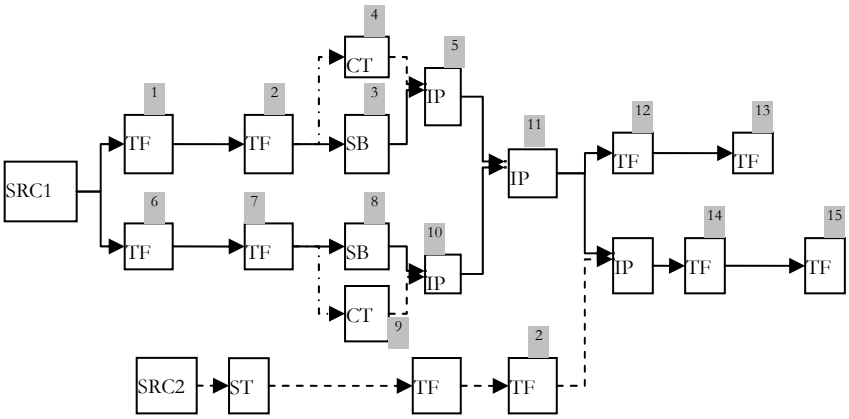


Figure 46 : Modèle de haut niveau.

On propose la construction de l'automate d'états finis associé au bloc ST du modèle de haut niveau précédent.

La difficulté consiste d'abord à déterminer les combinaisons d'erreurs matérielles détectables par l'auto test, c'est-à-dire la liste L1 du bloc SRC2. Cette liste peut être facilement déterminée. La sollicitation de la vanne se fait par le montage série des blocs 12 et 13 du modèle précédent. On peut donc en utilisant la méthode d'identification des listes du chapitre 4, déterminer cette liste à partir du modèle tronqué de haut niveau suivant.

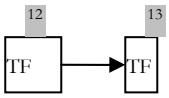


Figure 46 : Modèle de haut- niveau tronqué

La liste L1 de SRC2 est donc donnée par la liste L1 calculée, à partir du modèle tronqué précédent (constitué uniquement des blocs 12 et 13 en série), pour le bloc 13, en supposant la réalisation de l'événement Init(True) du bloc 12. On aura donc, pour le bloc SRC2,

$$L1=\{d(x5,0)d(x7;0)\ d(x16;0)\ bf(e8)d(x17,S)\ ;\ d(x5,0)d(x7;0)\ d(x16;0)\ d(x17,0)\}. \quad (\textbf{Equation 41})$$

L'automate d'états finis du bloc ST est donné par la figure simple (***figure47***) :

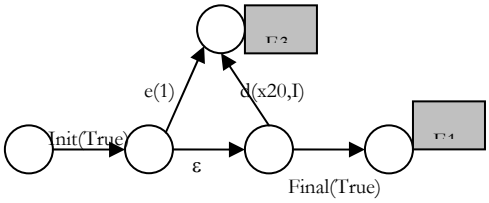


Figure 46 : Automate d'états finis du bloc ST

où e(1) est l'événement « échec de test », correspondant à la non réalisation du test sur la période T (puisque T_{test}>T) ; d(x20,I) correspond à l'activation d'un mode de défaillance I du capteur de position et les événements ε représentent l'échange d'informations à l'intérieur de la fonction de test.

Grâce à ces éléments, on peut identifier les listes La et Ld à partir de notre outil AFMCI et donc évaluer numériquement nos indicateurs de performances, suivant la méthode proposée au chapitre 2. L'influence de l'ajout du test périodique est double :

- elle augmente, en général, la fiabilité moyenne de l'ensemble testé « circuit de contrôle-vanne » en imposant une réparation de certaines combinaisons de modes d'erreurs matérielles.
- elle est neutre en cas de défaillance de certaines ressources matérielles, notamment le bus de communication ou l'alimentation du PLC.

On peut néanmoins remarquer que l'événement « échec de test » combiné à l'événement « vanne bloqué en position ouverte » peut retarder l'exigence de mise en position sûre du système. On peut observer une augmentation significative du PFD_{avg}, si la période de test périodique de la vanne est élevée, puisque cet élément est souvent le plus sensible, dans un cas d'architecture réaliste. On préférera donc plutôt des tests assez fréquents par sollicitation des actionneurs, mais d'amplitude faible (test sur déplacement faible du battant de la vanne), qui éviteront le déclenchement « intempestif » trop fréquent du système (faible perturbation du flux de comburant) et auront pour avantage de peser de manière moins importante sur la valeur du PFS_{avg} du système d'étude.

Si on considère la taille des listes caractéristiques obtenues, on remarque que l'augmentation du nombre de ressources matérielles du système se traduit par une augmentation du nombre de combinaisons de défaillances locales simultanées menant à un mode de défaillances du système, c'est-à-dire des listes La et Ld (on obtient ici des listes respectivement de 4800 et 4200 mots). Bien que cette augmentation soit fortement réduite dans le cas de partages importants de ressources matérielles entre sous fonctions, du fait de nos opérations de réduction et de croisement des listes lors du processus de propagation, elle n'en demeure pas moins un problème important pour coupler ces listes à un outil de calcul numérique.

Il est donc possible de faire des hypothèses suivant l'ordre de grandeur des probabilités maximales d'erreurs sur le caractère négligeable de certaines combinaisons et de réduire la taille des listes manipulées tout en introduisant au niveau des calculs une erreur. Nous ne discuterons pas, dans notre travail, des critères de troncature de ces listes de combinaisons, puisque nous avons avant tout recherché la complétude des listes obtenues. On peut cependant remarquer que les combinaisons d'erreurs activées faisant intervenir plusieurs ressources dont le niveau de fiabilité est important (donné à la fois par la valeur du taux de défaillance mais aussi les politiques de maintenance corrective et préventive) peuvent être négligées devant les combinaisons plus courtes contenant une sous combinaison d'erreurs appartenant déjà à cette liste (au sens d'inclusion stricte). La définition de ces combinaisons, appelées combinaisons minimales, sera reprise dans le chapitre 6 à des fins de diagnostic (ensemble $\boldsymbol{F_{d,mat}}$). Elles peuvent être construites par un traitement informatique (algorithme décrit en 2-2 du chapitre 6), qui peut être implanté par deux boucles logicielles imbriquées, une sur l'ensemble des mots et une sur l'ensemble des événements les constituants.

4- Cas d'étude réel et discussion

Pour clore cette partie, nous allons nous intéresser à un cas d'étude réel, issu d'une étude industrielle. Nous voulons montrer que malgré la croissance de la taille des listes caractéristiques, il est possible de construire un modèle de haut niveau permettant de construire de telles listes pour une fonction de sécurité. Nous rappellerons que la seule hypothèse imposée est qu'il existe

un prédicat logique entre l'état de demande à l'instant d'acquisition (indépendamment de ceux qui l'ont précédé) et l'activation de moyens visant à éviter l'accident.

Nous allons nous intéresser au cas d'un contrôle de niveau dans un réservoir chimique. Nous avons choisi ce cas d'étude réel pour trois causes principales :

- il fait intervenir différentes politiques de tests et une possibilité de reconfiguration fonctionnelle basée sur le passage d'une politique de vote majoritaire (2oo3) à un vote de type 1oo2 dans le cas d'une détection de défaillance de certaines ressources de ses voies d'acquisition redondantes ;
- il utilise un bus de communication, contrôlé par une horloge de contrôle (watch-dog) ;
- il considère le partage de ressources matérielles entre voies redondantes mais aussi au niveau des opérations de base menées par le PLC ;
- il considère le cas d'actionneurs testés en sollicitation, évoqués dans la partie 3.

Le premier point est intéressant car il permet de prendre en compte implicitement par notre méthode de construction de listes la variation de la vulnérabilité du système pour deux configurations fonctionnelles locales distinctes (2oo3 et 1oo2). En outre, il considère un choix de conception orienté vers une logique majoritaire 2oo3, qui peut difficilement être pris en compte par les méthodes RBD ou FTA classiques.

Le second point est intéressant car il distingue la réception d'une information correspondant à une absence de demande en absence de défaillance et la non réception d'une information.

Le troisième point l'est également car il permet d'expliquer la présence d'événements d'erreurs matérielles communs activés, à la fois dans des voies de traitement parallèle, mais aussi dans des processus de décision (IP) ou de transformation (TF) successifs.

L'intérêt du quatrième point a été souligné dans la partie 3, mais on constate que de plus en plus d'industries ont recours à ce type de test pour permettre la maintenance corrective d'un actionneur dans un schéma d'actionneurs redondants et une exigence de mise en position sûre du système.

Nous voulons dans cette partie montrer que la construction du modèle de haut niveau est possible pour ce cas d'étude. La saisie manuelle des différents automates d'états finis de chaque entité permet de mener à bien le calcul des listes Ld et La. Nous nous référerons si besoin aux automates d'états finis des cas d'étude précédents pour construire, par analogie, les automates d'états finis de nos entités. Pour plus de clarté, nous noterons sur le modèle de haut niveau la référence des blocs précédents permettant, par analogie, cette construction.

4-1- Présentation du cas d'étude réel

Nous considérons, dans ce cas d'étude, une fonction de sécurité ayant pour but d'éviter le débordement d'un réservoir de liquide, noté R. Cet exemple est tiré d'un cas réel pris dans une industrie chimique de produits dangereux. Le débordement du réservoir génère en général des conséquences catastrophiques pour les personnes sur site et les équipements (dégagements gazeux corrosifs).

Nous désirons donc maintenir le niveau du réservoir sous une cote d'alerte notée H. En cas de dépassement de la hauteur limite, on suppose qu'on se dirige de manière quasi-certaine vers un débordement du réservoir. Ce réservoir est alimenté par différentes conduites d'alimentation et son niveau varie au cours du temps.

Une **vanne de contrôle** (notée C) à la base du réservoir assure l'alimentation d'un circuit hydraulique de sortie. Cette vanne fonctionne en régime « tout ou rien » et nous savons que quand celle-ci est ouverte, le débit de sortie du réservoir est supérieur à son débit d'entrée.

L'industriel préconise donc lors de la construction du réservoir une conduite de vidange. Cette conduite doit permettre de vider le réservoir quand son niveau est supposé trop haut (i.e. supérieur ou égal à la hauteur limite H). Il propose par conséquent, l'installation d'une conduite en dessous de la cote H. Cette conduite, appelée circuit de vidange, permet si elle est ouverte de descendre en dessous de la cote d'alerte H (retour en position sûre). Cette conduite est fermée par deux vannes V1 et V2 montées en parallèle dans le cas d'un fonctionnement normal.

La fonction de sécurité implantée permet le contrôle en ouverture et en fermeture de ces deux vannes V1 et V2. Elle est basée sur la situation dangereuse suivante :

« Le niveau de liquide dépasse la cote d'alerte H et la vanne de contrôle C est fermée. »

La hauteur du liquide est contrôlée par trois capteurs S1, S2 et S3 placés à 120° (réservoir cylindrique) à la hauteur H. Ces trois capteurs sont de types analogiques et permettent de détecter le dépassement ou le non dépassement de la cote H. Ils sont reliés à des modules d'entrées décentralisés suivant le schéma de principe donné **figure 47**. Ces modules transmettent cette information au microcontrôleur qui décide suivant une logique de vote majoritaire (2oo3) du dépassement ou du non dépassement de la cote H. Chaque circuit d'acquisition analogique est testé périodiquement par un processeur commun à chaque module d'entrée. En cas de détection d'une défaillance de ce circuit, un bit de diagnostic est envoyé au PLC. Celui-ci ne prend alors pas en compte l'information du capteur lié à ce circuit d'acquisition et lui attribue la valeur correspondant au dépassement de la cote d'alerte.

Dans le cas d'une défaillance détectée sur un circuit d'acquisition, on propose de passer du mode 2oo3 au mode 1oo2 sur les deux voies d'acquisition jugées non défaillantes.

Dans le cas d'au moins deux défaillances détectées sur les trois circuits d'acquisition, on exige une action équivalente à celle du dépassement de la hauteur H.

On prend donc en compte deux modes dégradés possibles sur la détection du dépassement ou du non dépassement de la cote d'alerte H.

L'ouverture de la vanne de contrôle C (et donc sa fermeture) est contrôlée par deux capteurs de flux, notés F1 et F2 placés en aval de la vanne C. On suppose que la distance entre ces capteurs et la vanne de contrôle est assez faible pour assurer un délai extrêmement faible (inférieur à T) entre l'ouverture de la vanne de contrôle et son observabilité par ces capteurs. Ces capteurs sont reliés à des modules d'entrées décentralisés. Aucun test périodique n'est prévu sur les circuits d'acquisition correspondants. Les modules transmettent ces informations au microcontrôleur qui juge suivant une logique 1oo2 (assurée technologiquement par une porte NAND) de l'état de fermeture de la vanne de contrôle C. Les montages entre ressources matérielles sont symbolisés **figure 47**.

Le bus de communication fonctionne sur un mode maître-esclave avec horloge de contrôle. Il demande la transmission des informations provenant des différents capteurs à chaque période d'acquisition. Au-delà d'un temps limite, fixé par une horloge de contrôle, il impose à l'information manquante une valeur par défaut correspondant :

- soit à une hauteur de liquide supérieure à la cote limite, si cette information provient des capteurs S1, S2 ou S3 ;
- soit à un état observé de fermeture de la vanne, si cette information provient des capteurs F1 ou F2.

En partant des deux processus de décision précédents permettant de juger de la validité des deux conditions « le niveau de liquide dépasse la cote d'alerte H » et « la vanne de contrôle C est fermée », le microcontrôleur va émettre :

- soit pour une validité des deux conditions, un ordre d'ouverture des deux vannes V1 et V2 ;
- soit pour l'invalidité d'une des deux conditions, un ordre de fermeture des deux vannes V1 et V2.

Cet ordre va être transmis via le bus de communication au niveau actionneur par l'intermédiaire de deux modules de sortie. La sortie analogique de chacun de ces modules est reliée à un circuit de contrôle. Chaque circuit de contrôle (distinct) est constitué d'un montage relais qui permet d'imposer le sens du courant dans un moteur. Chacun de ces moteurs est relié à une vanne (V1 respectivement V2). Ce montage permet de commander en ouverture et en fermeture les vannes V1 et V2 (cf **figure 47**).

Du fait de la fiabilité limitée de la vanne V1, on a décidé d'ajouter une fonction de contrôle en ligne du bon fonctionnement de celle-ci. Cette fonction est basée sur la sollicitation périodique de cette vanne à l'ouverture (même procédé que dans le paragraphe 3). On contrôle l'état d'ouverture de la vanne par un capteur de flux en aval de celle-ci, noté P1. Si l'ouverture n'est pas détectée par retour de cette information au contrôleur (via le bus de communication), on impose un ordre de fermeture de la vanne V2 quel que soit l'état des informations issues des capteurs S1, S2, S3, F1 et F2 et une réparation de la vanne V1.

On peut donc résumer notre cas d'étude par le schéma de principe suivant (**figure** 47).

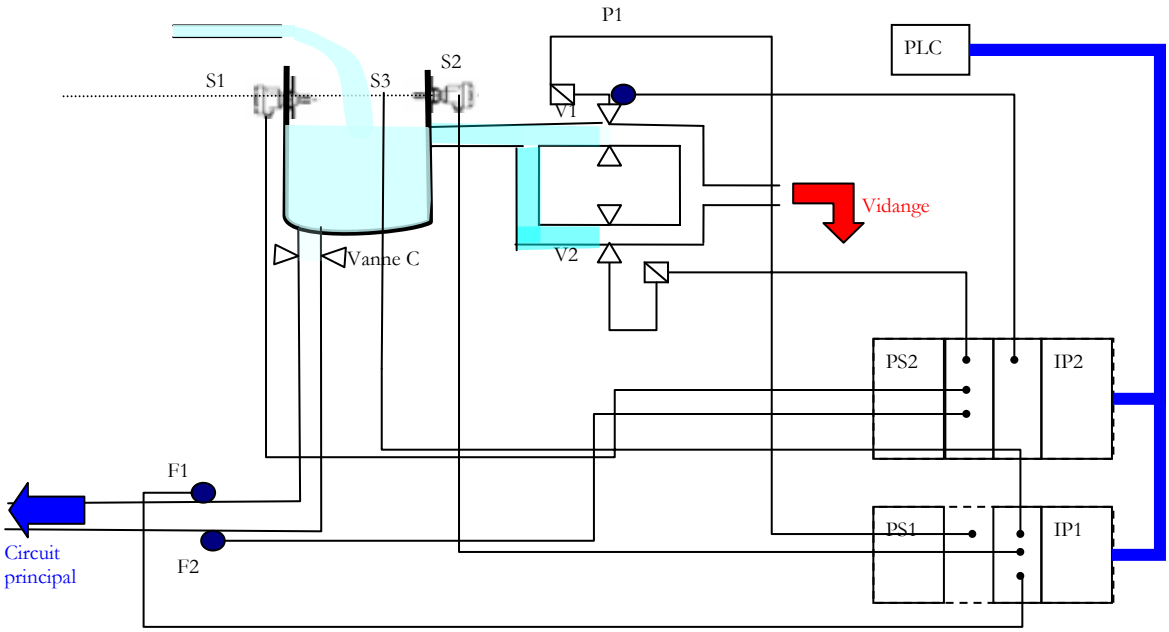


Figure 47 : Schéma de principe, contrôle de niveau de

Le choix de ce cas d'étude a plusieurs intérêts. Il se base d'abord sur une volonté de réduire le déclenchement intempestif, nous pouvons voir que cette architecture n'a pas été proposée au hasard:

- on a utilisé à dessein une logique majoritaire avec un mode de recouvrement sécuritaire (1oo2) pour augmenter la disponibilité de l'installation, sachant notamment que les capteurs de niveau choisis sont moins coûteux mais aussi moins fiables que les capteurs de flux.
- certains efforts ont été fait pour éviter certains modes communs de défaillances (les deux modules de sortie, ont par exemple été montés sur des modules entrée-sortie décentralisés ne possédant pas les mêmes sources d'alimentation).
- l'ajout d'un test par sollicitation périodique de la vanne a été choisi comme la solution la plus économique, même si elle implique des contraintes de partages de ressources du fait de la distance existant entre ressource matérielle testée (vanne) et lieu d'observation des résultats (capteurs de positions) et de décision (PLC).

Nous allons désormais expliquer comment la reconfiguration fonctionnelle 2oo3 – 1oo2 peut être modélisée au niveau du modèle de haut niveau avant de proposer ce modèle pour l'architecture complète.

4-2- Décomposition de la partie 2oo3- 1oo2

Pour comprendre le passage d'une logique 2oo3 aux deux modes dégradés proposés 1oo2 et mise à défaut, un court rappel de logique élémentaire est nécessaire.

On considère trois propositions A, B et C sur lesquelles se base un processus de décision. A chacune de ces trois propositions, on associe une variable booléenne, notée respectivement a, b et c, qui prend la valeur 1 lorsque la proposition correspondante est valide.

Un processus de vote majoritaire sur les trois variables est donc équivalent à la proposition :

« (a et b) ou (b et c) ou (c et a) »

Si on assigne à la variable c, par exemple, la valeur 1, la proposition deviendra « a ou c », ce qui nous ramène à un vote de type 1oo2 sur les deux voies non mises à une valeur par défaut (ici a et c). Ce résultat reste valide par permutation circulaire sur le triplet (a,b,c).

Si on assigne au couple de variables (a,b) la valeur (1,1), on aura une sortie du processus de décision égale à 1 (exigence de mise en position sûre du système). Ce résultat reste vrai avec les couples (b,c) ou (a,c).

En modélisant notre architecture sous forme de flux d'informations, cette fonction 2oo3 et ses modes de reconfiguration qui pourraient paraître, au premier abord, difficiles à modéliser pourront l'être au niveau de notre modèle de haut niveau par un assemblage de blocs du type 1oo2 et 2oo1 comme proposé dans la **figure 48**. Cet assemblage peut facilement être retrouvé en décomposant la fonctionnalité modélisée en un arbre binaire dans lequel l'information « True » sera associée à la valeur 1 et l'information « False » sera associée à la valeur 0. Les arbres de décisions binaires sont un moyen simple de décomposition d'une fonction de décision complexe en un assemblage de blocs IP.

4-3- Détermination du modèle de haut niveau de ce cas d'étude

Par analogie avec les modèles de haut niveau construits dans les paragraphes précédents (2 et 3) et en accord avec le montage donné par la **figure 47**, on peut construire le modèle de haut niveau de l'architecture considérée (fonction de sécurité pour contrôle de niveau de fluide dans un

réservoir). La **figure 49** (page suivante) prend en compte chacune des procédures de tolérance aux fautes précédemment évoquées.

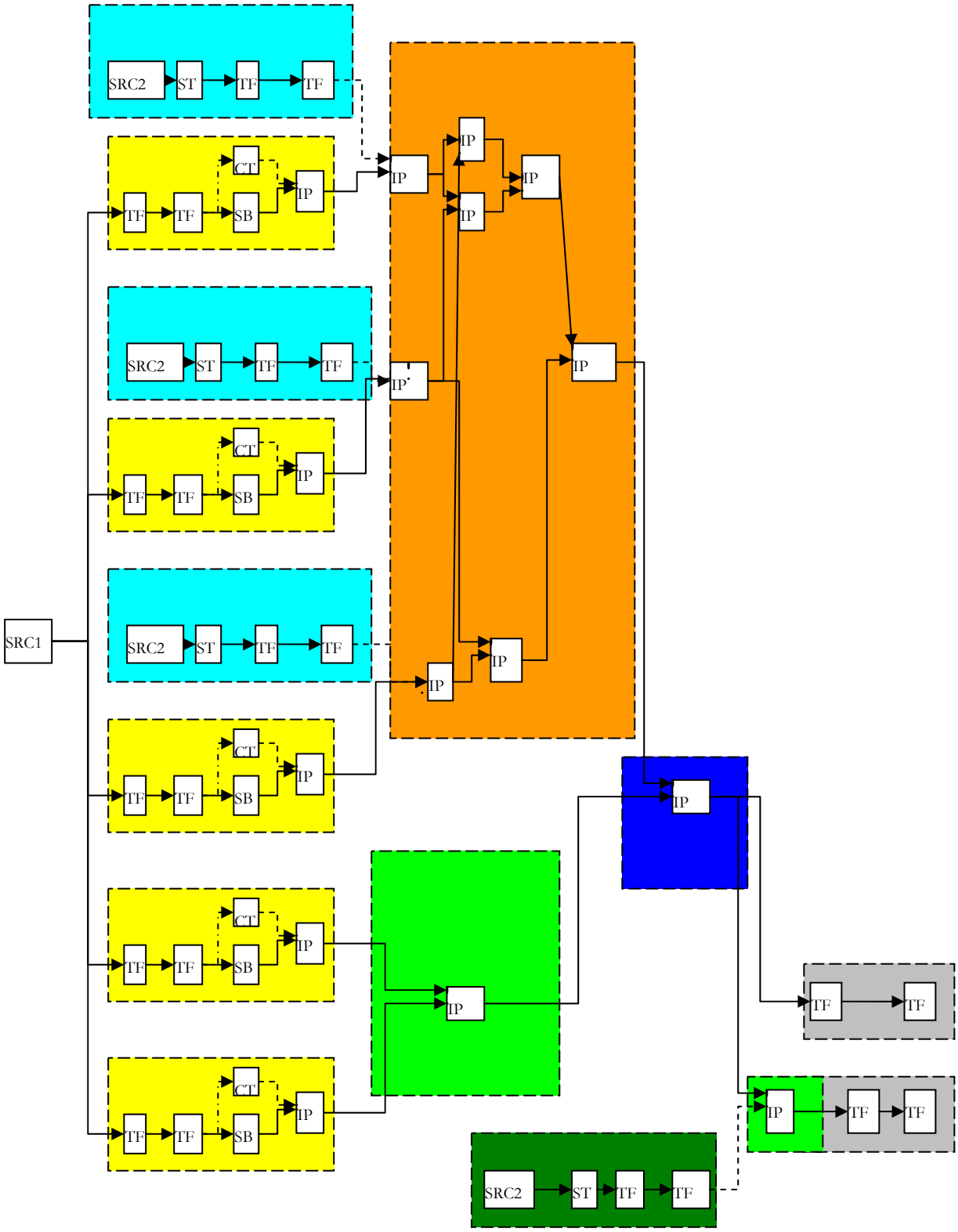


Figure 49 : Modèle de haut-

Sur cette figure, on a coloré pour une plus grande lisibilité les différentes portions du graphe.

- Les assemblages en jaune correspondent aux différentes voies d'acquisition des capteurs et de transmission de cette information acquise au PLC (avec horloge de contrôle). Il s'agit de haut en bas des voies d'acquisition correspondant aux capteurs S1, S2, S3, F1 et F2.
- Les assemblages en bleu clair correspondent aux procédés de tests périodiques des circuits d'acquisition auxquels sont reliés respectivement S1, S2 et S3 (de haut en bas) et à la transmission de cette information au PLC.
- L'assemblage en orange correspond à la prise en compte de la fonctionnalité 2oo3 et des modes dégradés rattachés (décrite en 4-2).
- L'assemblage en bleu foncé correspond à une opération du PLC de type vote 2oo1. Les deux informations d'entrées doivent être « True » pour que l'information de sortie soit « True » en l'absence de défaillances dans les entités constituant cet assemblage. Il correspond à la condition ET entre les deux sous conditions « le niveau de liquide dépasse la cote d'alerte H. » et « la vanne de contrôle C est fermée. »
- L'assemblage en vert clair correspond à une opération du PLC de type vote 1oo2.
- Les assemblages en gris correspondent au transfert des informations de contrôles de la part du PLC au niveau des vannes et à leur transformation en actions (ouverture/fermeture de vanne ou aucune action).

Nous pouvons constater au moyen de cet exemple que les opérations logiques portant sur un ensemble fini d'informations et les politiques de reconfigurations fonctionnelles qui en dépendent peuvent être modélisées par un assemblage de blocs dans un modèle de haut niveau. En cas de difficulté de construction, il suffit de se ramener à un ensemble d'arbres logique et de l'associer à un assemblage de blocs de décision (sous entités fonctionnelles).

La construction de nos listes finales se fait par la saisie manuelle des automates finis correspondant à chaque entité dans notre logiciel AFMCI (cf. chapitre 3). Sans détailler les listes obtenues, on peut affirmer une augmentation importante du nombre de mots (5234 pour la liste Ld et 2670 pour la liste La) allant d'une longueur de 1 à 17 erreurs activées.

Même si cette liste est complète, la réduction de la taille des listes est nécessaire pour assurer une évaluation numérique rapide de nos indicateurs de performances. On préconisera pour ce faire une conservation des combinaisons dont le poids en termes de probabilité moyenne est le plus important. Pour sélectionner ces combinaisons, on pourra utiliser les critères de hiérarchisation donnés au chapitre 6 (chapitre dédié aux améliorations d'architecture). La précision du calcul des indicateurs de performances dépendra de la marge d'erreur maximale que peuvent imposer ces simplifications. Dans la pratique, on peut remarquer que cette approche a le mérite d'être rapide et automatisable. En général, les ordres de grandeurs des taux de défaillances des ressources et leur maintenabilité (période, taux de couverture…), permettent de juger rapidement des combinaisons dont le poids s'avère prédominant. Cette réduction de la taille des listes augmente *a posteriori* l'erreur sur l'estimation des indicateurs de performances. Cependant elle peut être justifiée au vu des paramètres caractéristiques du modèle (taux de défaillances, temps d'inspection, MTTR…) en considérant certaines combinaisons comme ayant une probabilité négligeable.

5- Conclusion du chapitre 5

Dans ce chapitre, nous avons voulu montrer l'intérêt de notre méthode d'évaluation d'indicateurs de performance pour différents cas d'étude.

Le premier exemple, très simple, permet de conclure quant à l'utilité de cette méthode y compris pour des architectures simples. En effet, il a mis en valeur une différence en termes d'estimation du PFD_{avg} qui peut s'expliquer par la prise en compte dans notre méthode des propriétés de non cohérence de l'architecture. Ces propriétés sont dues essentiellement à la co-existence sur une même période d'erreurs matérielles de mode S et D, qui peuvent initier ou inhiber au niveau local une information erronée se propageant par la suite dans l'architecture fonctionnelle. La perturbation d'une information peut donc être un phénomène réversible le long d'un flux d'information, contrairement à la perte d'information. La prise en compte de ces réversions explique en grande partie la différence des résultats numériques.

Le second exemple, un peu plus complexe, nous permet de montrer l'intérêt de notre méthode de construction systématique pour prendre en compte les partages de ressources dans l'architecture (notamment matérielle mais aussi informationnelle, par exemple l'état de demande observée) et l'utilisation de processus de contrôle de bonne réception (horloge de contrôle) dans les protocoles de communication (ici maître-esclave). Le doublement d'une voie d'acquisition s'accompagne en général d'une augmentation de la fiabilité du système. Il est cependant intéressant de voire que l'utilisation de procédures d'horloge au niveau du bus de communication, peut permettre de diminuer l'influence des défaillances de mode I des deux voies. Il permet donc de réduire la probabilité de défaillance de la fonction de sécurité. Au niveau de notre modélisation, on pourrait ajouter, du fait du réseau un événement de perte d'information dans l'entité TF représentant le transfert d'information des modules d'entrées au PLC. La probabilité de cet événement pourrait être estimée en fonction du nombre de périphériques et d'informations échangées sur un cycle par le réseau. Nous n'entrerons pas dans le détail de cette quantification. Pour ce faire, on pourra se référer aux rapports techniques existants sur la fiabilité des réseaux de communication, comme **[Schi02]** pour les réseaux ProfiSafe.

Le troisième exemple permet enfin d'expliquer la prise en compte de test périodique pour une ressource matérielle, on prend en compte directement dans la détermination des listes du bloc source SRC2 l'ensemble des combinaisons d'erreurs détectables. Le fait que les combinaisons soient de tailles plus importantes que celles prévues lors de l'implantation du test s'explique par l'éloignement entre l'émetteur de l'ordre de contrôle et l'actionneur sollicité dans notre exemple. Il est ainsi possible que certaines combinaisons d'erreurs activées ne permettent pas la détection de l'événement redouté (« défaillance de la vanne en fermeture » pour notre exemple).

Le quatrième exemple a pour but de montrer que notre méthode de modélisation est tout à fait adaptée pour des cas d'étude réels rencontrés dans l'industrie ou dans les systèmes embarqués (ferroviaire, aéronautique…). Il permet d'expliquer en détail la prise en compte d'une logique majoritaire et des modes de fonctionnement dégradés prévus dans une architecture complexe. La longueur des listes obtenues est relativement importante, ce qui peut être handicapant en termes de temps de calcul des indicateurs de performances, et ce, même si la complétude des listes caractéristiques est un aspect important. Elle permet, en effet, de dégager l'ensemble des combinaisons d'erreurs activées nécessaires et suffisantes à l'occurrence d'un mode de défaillance du système d'étude. Nous verrons dans le chapitre 6 comment utiliser ces informations pour définir une stratégie de modification pertinente d'une architecture pour une exigence de risque résiduel maximale garantie. Le calcul pratique des indicateurs de performances peut cependant être effectué en hiérarchisant ces différentes combinaisons suivant leur poids moyens relatifs. Les critères de hiérarchisation seront les mêmes que ceux donnés dans le chapitre

6. En partant de cette liste réduite, on admet une erreur de calcul pouvant être majorée suivant le poids estimé des combinaisons écartées. On peut ensuite utiliser sans problème ces listes réduites dans des outils existants ©**Altarica** ou ©**Fault Tree+** pour permettre l'évaluation de nos indicateurs de performances en imposant : un pas de calcul T* (déterminé au chapitre 4), deux variables de somme S1 et S2 et les modèles associés à l'évolution des probabilité d'erreur de chaque ressource matérielle (cf. chapitre 4).

Dans la pratique, on remarquera que l'utilisation des sous modèles sous l'atelier ©**AltaRica** permet la prise en compte de processus markoviens homogènes, mais pas la modification des matrices de transitions au cours du temps. Il peut donc être utile de prévoir, en termes de développement, pour prendre en compte l'effet des fluctuations environnementales un couplage entre l'outil ©**CARMS** (gratuit sous Excel) qui permet l'itération de chaîne de Markov et ©**Matlab** qui permet de modifier les valeurs de la matrice de transitions suivant ses incréments temporelles (suivant des lois d'évolution de stress et de couplage des taux de pannes des composant données au chapitre 4).

Nous n'avons pas développé ces interfaces durant notre thèse du fait du développement de l'outil de construction des listes caractéristiques (**AFMCI**) qui a concentré toute notre attention, mais nous souhaitions consigner les possibilités d'utilisation d'outils existant pour automatiser complètement la méthode et rendre son utilisation plus rapide sur un large spectre de cas d'étude, sans être limité par la taille des listes caractéristiques obtenues.

Nous allons nous intéresser plus particulièrement dans le chapitre suivant à l'utilisation des listes caractéristiques pour proposer une stratégie pertinente d'amélioration architecturale d'une fonction de sécurité, permettant d'atteindre un niveau de risque résiduel garanti qui soit tolérable. Cette deuxième application des listes, en sus de l'évaluation numérique d'indicateurs de performances, permet de justifier l'effort de modélisation consenti, puisqu'elle réduit fortement le temps nécessaire à l'identification des « points faibles » de l'architecture. Elle permet d'orienter de manière pertinente les modifications architecturales, visant à l'atteinte d'objectifs sur un des deux indicateurs de performances retenus sous une contrainte de coût de mise en œuvre.

Chapitre 6

Vers une orientation pertinente d'une stratégie de modification architecturale

Chapitre 6 - Vers une orientation pertinente d'une stratégie de modification architecturale

Les précédents chapitres nous ont permis de construire un modèle d'évaluation quantitatif de nos indicateurs de performances fonctionnelles et de les évaluer pour une architecture donnée. Il est, cependant, tout à fait possible que le choix d'architecture (matérielle et fonctionnelle) et la politique de maintenance proposés pour le système étudié s'avèrent insuffisants pour ramener le risque résiduel en dessous d'un niveau de risque tolérable.

Dans ce cas, la démarche naturelle consiste à proposer une stratégie de modification pertinente de l'architecture basée sur l'augmentation de la sécurité fonctionnelle sous contrainte de coûts. On donnera priorité à l'évitement et à la prévention des combinaisons d'erreurs dont les probabilités d'existence sont les plus élevées pour un coût donné.

La hiérarchisation, en termesss de priorité, de ces modifications doit donc être fixée à la fois par

- **une estimation du niveau de réduction du risque** induit par cette modification,
- des **contraintes induites** par l'introduction de cette modification
- de la faisabilité et du **coût nécessaire** permettant sa mise en oeuvre.

La plupart des méthodes actuelles fondent la stratégie de modification architecturale sur une identification de coupes minimales **[Tan04]**. Cette identification est le plus souvent basée sur une modélisation logique du fonctionnement de l'architecture (de type arbres de causes ou arbres binomiaux). On peut ensuite hiérarchiser ces coupes par le calcul de facteurs d'importances attachés à celles-ci **[Bere05]**. Ces méthodes sont prévues, généralement, pour des systèmes cohérents. Nous allons proposer dans cette partie, d'utiliser les listes caractéristiques Ld et La, construites précédemment pour orienter de manière plus pertinente l'identification de **coupes minimales** et leur **hiérarchisation**.

Nous proposerons d'abord certaines hypothèses simplificatrices, nous permettant de nous ramener à un problème de réduction sous contraintes d'un facteur d, appelé distance à objectif, basé sur l'indicateur de performance PFD_{avg}. Nous présenterons ensuite les moyens de modifications existants et les effets attendus. Nous proposerons d'établir une stratégie d'amélioration de ce facteur, basée sur un nombre restreint de types de modifications architecturales.

Cette stratégie sera menée en deux temps, distinguant clairement le traitement des coupes minimales « détectables » de longueur supérieure à 2 de celles de longueur égale à 1. L'idée de base consiste à privilégier dans un premier temps une politique de test en ligne, basée sur certaines coupes de longueur supérieure à 2, en estimant le gain garanti pouvant être atteint par cette mesure. Si le gain garanti s'avère insuffisant, on proposera l'augmentation de la fiabilité de ressources matérielles dites critiques (i.e. pour lesquels la présence d'un état d'erreur matériel est suffisante à l'apparition d'un mode de défaillance sous demande de la fonction de sécurité), soit par une augmentation de la fréquence de maintenance de ces ressources, soit par leur remplacement par des ressources plus robustes.

Pour mettre en œuvre cette stratégie, nous proposerons de hiérarchiser la probabilité moyenne d'occurrence des coupes minimales, dans un contexte de contraintes environnementales constantes. La relation d'ordre sur laquelle sera basée cette hiérarchisation pourra attribuer à différentes coupes un ordre hiérarchique identique dans le cas où leur probabilité d'occurrence moyenne est jugée « trop proche ».

1- Vers la mise en place d'une stratégie d'amélioration architecturale pertinente

1-1- Hypothèses simplificatrices pour une stratégie de diminution du risque résiduel

La stratégie que nous tentons de définir est basée sur le cycle d'amélioration de l'**IEC61508**. Elle nécessite donc de préciser certaines hypothèses préliminaires importantes. L'idée de base est que l'architecture et la politique de maintenance rattachée à ses ressources matérielles est insuffisante pour atteindre un niveau de risque résiduel tolérable, noté R_{tol}. Nous nous plaçons, d'après l'égalité proposée au chapitre 1 (**Equation 13, p.19**) sur l'évaluation du risque résiduel, sous la condition de départ :

$$R_{res} = f.PFD_{avg}.I_{dem} + (\frac{T_{life}}{T} - f).PFS_{avg}.I_{abs} < R_{tol} \quad \textbf{(Equation 42)}$$

Dans la somme de gauche, on peut distinguer deux termes, le premier relève de la défaillance de la fonction dédiée sécurité sur demande et le second porte sur le risque engendré par le déclenchement intempestif de cette fonction. Dans le cas général, on suppose que le risque résiduel dû au non évitement de l'accident redouté est très supérieur à celui dû au déclenchement intempestif de la fonction de sécurité. Si on note PFD_{avg}(init.) et PFS_{avg}(init) les indicateurs de performances de notre système d'étude, avant qu'une quelconque modification ne soit entreprise sur son architecture ou sa politique de maintenance, on en déduira que :

$$f.PFD_{avg}(init).I_{dem} >> (\frac{T_{life}}{T} - f).PFS_{avg}(init).I_{abs} \quad \textbf{(Equation 43)}$$

On admet que le moyen le plus efficace pour réduire le niveau de risque résiduel consiste à modifier les choix architecturaux et la politique de maintenance du système de départ pour réduire l'indicateur de performances PFD_{avg}, tout en conservant PFS_{avg} à son niveau actuel.

Afin d'atteindre un niveau de risque tolérable pour notre installation, nous cherchons à garantir pour le système modifié que les nouveaux niveaux de performances, notés PFD_{avg}(fin.) et PFS_{avg}(fin), vérifient les deux inégalités suivantes :

$$f.PFD_{avg}(fin.).I_{dem} \leq R^*_{tol} \quad \textbf{(Eq.44)} \qquad \text{avec} \qquad R^*_{tol} = R_{tot} - (\frac{T_{life}}{T} - f).PFS_{avg}(init).I_{abs} \quad \textbf{(Eq.45)}$$

$$\text{et} \qquad PFS_{avg}(fin.) \leq PFS_{avg}(init). \quad \textbf{(Eq.46)}$$

En notant α, la valeur définie comme étant égale au rapport $\frac{R^*_{tol}}{I_{dem}.f}$, on devra donc garantir que :

$$PFD_{avg}(fin.) \leq \alpha \quad \textbf{(Eq.47)} \qquad \text{et} \qquad PFS_{avg}(fin.) \leq PFS_{avg}(init). \quad \textbf{(Eq.48)}$$

On appellera distance à objectif, notée d, la différence entre la valeur de l'indicateur de performance PFD_{avg} du système proposé et α. La proposition d<0 sera équivalente à la condition de validité de l'égalité **(Eq.47)**. On veut donc en modifiant l'architecture et les choix de maintenance réduire cette distance jusqu'à une valeur nulle, ou éventuellement négative, tout en assurant que nous pouvons justifier pour chaque modification l'égalité **(Eq.48)**.

Nous allons décrire rapidement les leviers de modifications existants et les contraintes qu'ils induisent. Nous regrouperons ces leviers suivant trois classes :

- la **mise en position sûre du système**, suite à la détection d'une combinaison d'erreurs induisant un mode de défaillance sous demande du système (**diagnostic en ligne**)

- la modification de l'architecture fonctionnelle par **ajout de voie redondante** à un processus de traitement d'information existant (flux de F- ou D-information) et utilisation à des fins de décision en ligne (voteur,…).
- l'augmentation des propriétés de **fiabilité de ressource matérielle**

1-2- Leviers d'amélioration : discussion sur leurs conséquences au niveau local et global.

Chaque levier d'amélioration architecturale peut être utilisé de manière complémentaire dans le cadre d'une stratégie de modification architecturale, visant à réduire notre indicateur de performance PFD_{avg} . Nous supposons que notre stratégie de modification architecturale peut être basée sur l'ajout de sous fonctions visant à l'augmentation des propriétés de tolérance aux fautes matérielles du système, et également sur la réduction de la probabilité de défaillance de certaines ressources matérielles, jugées comme critiques.

1-2-1- Notion de prévention et de tolérance aux fautes.

La propriété de **tolérance aux fautes** est définie pour un système comme la capacité pour celui-ci en présence de fautes d'assurer un niveau de service correct.

Nous serons ici intéressés par la détection d'un ensemble non vide d'erreurs matérielles, ayant pour conséquence un mode de défaillance sur demande de la fonction de sécurité. Cette détection doit entraîner la mise en position « sûre » (en préférera le terme « secure ») du système, c'est-à-dire l'exigence dans ce cas de figure d'une activation des moyens de réaction de la fonction de sécurité.

On peut définir deux moyens principaux permettant la **détection d'erreurs** **[Jaco00]**:

- le premier est basé sur le **test en ligne** d'un ensemble fini de ressources matérielles dont l'ensemble des états d'erreurs est suffisant à l'apparition du mode de défaillance sur demande du système.
- le second est basé sur l'**ajout d'une voie de redondance** dans le traitement d'information (F ou D) permettant une comparaison de deux signaux, jugés a priori équivalents, mais qui perdent cette propriété d'équivalence en présence de combinaisons d'erreurs redoutées.

Après cette détection, plusieurs solutions sont possibles vis-à-vis du traitement de ces erreurs.

La première consiste à recouvrer l'erreur détectée, c'est-à-dire à baser la suite du processus de décision du système d'étude sur un ensemble non vide d'informations, jugées saines. Le recouvrement impose donc une garantie sur la validité de ces informations en présence d'erreurs. On notera que cette validité peut être impossible à démontrer du fait des contraintes existantes en termesss de partage de ressource qui peuvent induire des règles de dépendances. Dans ce cas, l'influence de ces couplages sur les deux indicateurs de performances doit être estimée.

La seconde consiste à compenser l'information erronée en ajoutant un point de contrôle (entité IP) sur les flux d'information en aval de l'utilisation des ressources matérielles défaillantes. On va donc influer sur l'état de certaines informations pour forcer la mise en position « sûre » de la fonction dédiée sécurité. Cette approche consiste à inhiber la propagation d'une information erronée en présence de demande. Nous avons donc plusieurs possibilités dans le choix de la localisation de ce point de contrôle. Nous choisirons une coupure directe (ajout de relais) du circuit d'alimentation de contrôle des actionneurs. Ce procédé appelé mise hors énergie (de-energizing process) a pour conséquence directe un ordre de mise en position « sûre » de l'ensemble des actionneurs (activation des moyens de réaction). Il a pour avantage d'agir

directement au niveau des actionneurs, indépendamment de l'état des ordres émis par le PLC aux actionneurs.

Trois notions semblent donc importantes pour la mise en place de moyens de tolérance aux fautes :

- la **détectabilité de la combinaison d'erreurs redoutée**,
- les vulnérabilités imposées par le recouvrement ou la reconfiguration,
- les contraintes de **mise en œuvre du test** et le diagnostic.

Nous privilégierons **l'ajout de test en ligne** et un mécanisme de **compensation d'erreur** plutôt que l'augmentation du niveau de redondance dans le traitement d'information. A notre avis, ce choix est justifié par le fait que l'ajout d'une voie de redondance peut modifier de manière importante le modèle de haut niveau (chapitre 3) de notre architecture et se traduit en général par un ensemble de contraintes architecturales supplémentaires (partage de ressource, partage d'information). Nous supposerons donc l'ajout d'un « superviseur » **[Jens03][Cass99]** permettant, à chaque cycle T,

- la sollicitation d'un ensemble des ressources matérielles dont les combinaisons d'erreurs entraînent un mode de défaillance sur demande de la fonction dédiée sécurité,
- l'observation du signal de sortie généré par cet ensemble suite à cette sollicitation.

Ce procédé de sollicitation/observation ne sera retenu que si l'ensemble des ressources matérielles, dont on veut tester les états d'erreurs, possède des propriétés de commandabilité et d'observabilité, permettant de conclure de manière univoque en un temps raisonnable de la présence ou l'absence de la combinaison redoutée d'erreurs matérielles. On dira alors que la combinaison d'erreur est **entièrement testable.**

Une fois le signal de sortie observé, le « superviseur » pourra par comparaison de **signature** du test **[Zwin95]**, conclure à l'existence d'une combinaison d'erreurs sur l'ensemble des ressources matérielles testées. Dans le cas d'une combinaison d'erreurs matérielles détectée, induisant de manière suffisante un mode de défaillance du système d'étude, le superviseur transmettra directement (canal de transmission indépendant en termes de ressources matérielles) un ordre de coupure au circuit de contrôle des actionneurs qui doit exiger leur mise en position « sûre ».

Nous utiliserons l'**ajout de redondance** uniquement si un ensemble de ressources matérielles, utilisé pour une opération de transformation de signal clairement identifiée (blocs TF en série), ne peut être testé directement par l'ajout d'un « superviseur » et possède une probabilité de défaillance élevée. Nous proposerons, dans ce cas, une redondance active et un montage de type 1oo2. Ce cas de figure est le seul, où nous accepterons la mise en place d'une stratégie de mode dégradé par ajout de redondance. En effet, le doublement d'une voie de traitement proposée pour des redondances matérielles (active, passive… **[Flor03]**) est coûteux et uniquement utilisé dans le cas où aucune autre stratégie de tolérance ou d'augmentation de fiabilité de ressources matérielles ne permet de réduire de manière significative et à moindre coût la probabilité de défaillance de la voie redondée. L'utilisation de redondance mixte (logicielle/matérielle) implique, quant à elle, des contraintes de partage (matérielles ou informationnelles) entre voies de traitement redondantes pouvant réduire fortement le bénéfice d'une telle mesure sur les performances fiabilistes du système.

1-2-2- Réduction de la probabilité d'existence d'erreurs matérielles critiques

En sus des procédures de diagnostic en ligne et de mise en position sûre au niveau actionneur, il semble possible d'augmenter les performances fiabilistes d'une fonction dédiée sécurité par

augmentation de la fiabilité de certaines ressources matérielles. On parle souvent de « ressource matérielle critique », lorsqu'un état d'erreur d'une ressource suffit à produire la défaillance sur demande de la fonction de sécurité (ou **erreur matérielle d'ordre 1)**.

Il convient à présent d'identifier les erreurs matérielles d'ordre 1, dont la probabilité d'existence moyenne contribue de manière significative à la valeur du PFD_{avg}. Deux approches sont alors possibles pour réduire cette probabilité d'erreur matérielle.

La première consiste à réduire le **taux de défaillance** de ces ressources matérielles (notées « Γ.. » dans le chapitre 4) en remplaçant la ressource matérielle par une autre, fonctionnellement équivalente (possédant les mêmes spécifications fonctionnelles) dont le taux de défaillance est moindre au vu de l'environnement d'utilisation de la ressource. On **réduit ainsi la probabilité d'occurrence des erreurs matérielles globalement** sur tout le temps d'utilisation opérationnelle de la ressource. Le coût de mise en œuvre de cette modification sera considéré comme égal à la différence de prix des deux ressources matérielles, dans le cas d'un remplacement, ou au prix du moyen de protection, dans le cas d'isolement au stress de la ressource existante. Le coût d'un remplacement de ressource matérielle est variable. Il est fréquent pour un intégrateur, qu'une ressource matérielle (capteur, actionneur,…) soit remplacée par un COTS, c'est-à-dire une ressource sur étagère. L'utilisation actuelle de ces composants de remplacement est largement répandue, cependant elle doit impliquer des vérifications sur l'équivalence de leurs fonctionnalités.

La seconde consiste à introduire une procédure de maintenance préventive, à réduire la période d'inspection de la ressource matérielle critique, notée TI, ou à augmenter la qualité de cette procédure de maintenance. Nous restreindrons notre étude à la technique dite d'augmentation de la **fréquence de maintenance préventive** d'une ressource matérielle. On supposera que la qualité de la procédure de maintenance préventive est décrite par un facteur γ (avec $0 \leq \gamma \leq 1$), donnant la probabilité conditionnelle de succès de cette opération.

L'effet d'une telle modification peut être facilement mise en évidence sur l'**exemple simple d'une ressource matérielle, uniquement maintenue de manière préventive périodique** (pas de maintenance corrective), avec la période TI. Si on suppose que cette ressource matérielle est soumise à des facteurs de stress constants, en période de vie utile, et qu'elle ne présente qu'un mode de défaillance possible, on écrira que son évolution est donnée par une courbe exponentielle (de coefficient λT) pour TI>>TR>>T. Alors on pourra approximer le profil d'évolution temporel de la probabilité d'erreur matérielle, notée P2, par les deux courbes suivantes avec Γ=200 FIT=2^{E}-7/h, T=1 jour, TI= 2ans, Tlife = 10 ans, γ=0,7, grâce au modèle de la partie 4 (**Figure 50**).

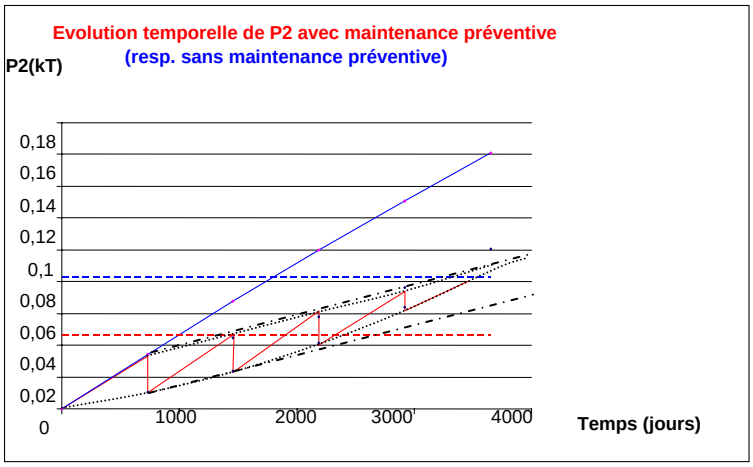


Figure 50: Comparaison de l'évolution de la probabilité d'erreur matérielle
- Impact de la maintenance préventive

On s'aperçoit qu'en comparant les moyennes des probabilités d'erreurs matérielles de la ressource (P2) que le rapport de celles-ci est égal à 0,55 soit de loin supérieur au rapport T_{life}/TI pris égal à 0,2. Ce décalage est du à l'imperfection de la procédure de maintenance (hypothèse γ=0,7). Mais on peut aussi noter que l'évolution des pics n'est pas strictement linéaire contrairement à l'hypothèse simplificatrice proposée par Goble et Bukovski **[Buko00]**, qui prévoit une relation de linéarité entre deux maxima locaux successifs de la courbe avec maintenance. Cependant, on remarque que la courbe d'évolution de P2, est comprise entre deux droites parallèles. La première passe par le premier max, noté P_m (valeur de P2 juste avant la première tâche de maintenance périodique de la ressource matérielle). La seconde passe par le premier minimum local, c'est-à-dire la valeur de P2 après cette première action de maintenance préventive. Ces deux droites « enveloppes » ont pour pente $\frac{(1-\gamma)P_m}{TI}$. On peut donc majorer la probabilité moyenne de P_2, par le produit de trois termes :

- la probabilité moyenne de P_2 évaluée en absence d'action de maintenance préventive
- le facteur $\frac{T_{Life}}{TI}$
- le facteur correctif $1+\frac{(1-\gamma)(T_{life}-TI)}{2.TI}$

Dans le cas où γ=1 (maintenance préventive parfaite dite « as good as new »), on s'approchera du profil d'inspection parfaite proposé par Bukovski **[Buko00]** et que dans ce cas le rapport des moyennes de P2 s'approchera du ratio TI/T_{life}=0,2. Au vu de ce profil (**figure 51**), on peut affirmer que l'utilisation de fréquence plus élevée pour une politique de maintenance préventive parfaite (« as good as new ») diminue la hauteur des maxima des probabilités de défaillances dans le cas le plus favorable (maintenance préventive parfaite) d'un rapport TI/Tlife et divise la moyenne par le même ratio.

Dans le cas d'une maintenance imparfaite, on peut démontrer par récurrence que la courbe précédemment évaluée pour γ<1 est toujours comprise entre deux droites enveloppes passant respectivement par le premier max et le premier min (juste avant et juste après la première procédure d'inspection préventive) et de pente égale à $\gamma.\Gamma/T$. La valeur moyenne de P2 est donc majorée par $\left[1+\frac{(1-\gamma)(T_{life}-TI)}{2.TI}\right]P^*_{2,avg}\frac{T_{life}}{TI}$ **(Eq.49)**

avec TI la période d'inspection préventive,

γ la probabilité de succès de l'opération de réparation

$P^*_{2,avg}$, la valeur de P2, en l'absence de procédure de maintenance préventive

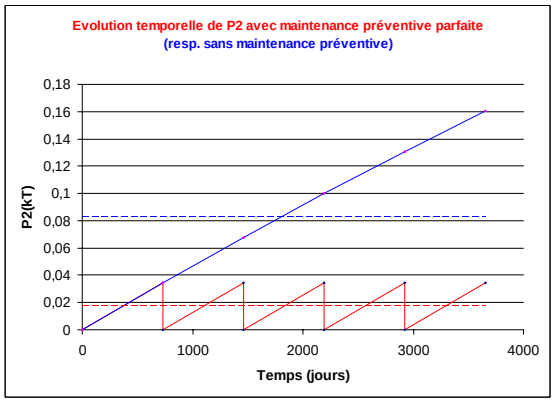


Figure 51: Comparaison de l'évolution de la probabilité d'erreur matérielle - Impact d'une maintenance préventive parfaite

On peut donc en connaissant γ proposer une majoration de PFD_{avg} (γ) pour une ressource ne possédant qu'un mode de défaillance et soumise à des stress environnementaux constants si cette ressource ne possède pas de procédure de maintenance corrective. La répartition des temps d'inspection et les procédures d'inspection appliquées sur différents modules matériels et définis localement peuvent avoir une influence importante sur les propriétés générales du système, comme l'ont montré nos exemples dans **[Hami03]**. Il convient donc de se servir de cette politique de maintenance préventive, non pas comme une variable d'ajustement visant à se rapprocher sensiblement de propriétés visées, mais bien comme un véritable levier pour atteindre ces propriétés. Le coût de certaines politiques de maintenance et leur mise en œuvre peut être faible par rapport à d'autres mesures de modification de l'architecture de notre fonction de sécurité.

On a deux approches différentes possibles, pour réduire l'occurrence d'erreur matérielles d'ordre 1, ces deux options pouvant être utilisées de manière complémentaires.

La première est basée sur **le choix d'une ressource matérielle plus résistante.** Elle a, dans le cas de ressource non correctivement maintenue, l'effet de réduire la vulnérabilité aux stress environnementaux de la ressource.

La seconde est basée sur une augmentation de la **fréquence des tâches de maintenance préventive**, à qualité de maintenance constante.

1-3- Contraintes de modifications architecturales et intérêts.

La stratégie de modification architecturale avancée va s'appuyer sur les leviers précédemment évoqués. Contrairement aux approches classiques qui privilégient toujours soit l'aspect fonctionnel, basé sur l'introduction de diagnostic et de compensation d'erreurs d'information, soit un aspect matériel, basé uniquement sur l'amélioration de la fiabilité des ressources matérielles critiques, nous nous proposons de prendre en compte ces deux aspects de manière complémentaire dans la détermination d'une stratégie de modification architecturale.

Du fait de la complexité de ce problème, nous allons proposer les hypothèses suivantes :

[H1] L'architecture fonctionnelle de base de notre système d'étude ne pourra être modifiée que par l'ajout de procédure de diagnostic et de reconfiguration en ligne, visant à augmenter les propriétés de tolérance aux fautes du système.

[H2] Une modification matérielle du système ne pourra être considérée que si l'état d'erreur d'une ressource matérielle est d'ordre 1, c'est à dire qu'elle suffit à l'apparition d'un mode de défaillance sur demande de la fonction dédiée sécurité.

[H3] Le remplacement d'une ressource matérielle par une autre ne se fera pas au détriment des spécifications fonctionnelles de la ressource (hypothèse nécessaire pour l'utilisation de COTS)

[H4] Les procédures de test de ressources en ligne déjà existantes ne pourront être modifiées.

[H5] Les procédures de reconfigurations fonctionnelles ou de modes dégradés existants ne seront pas supprimées, ni modifiées.

[H6] Une modification de la période de maintenance préventive est, a priori, envisageable pour toute ressource matérielle prise dans son ensemble.

[H7] Les opérations de maintenance préventives seront jugées comme étant de coûts fixes et strictement périodiques, les probabilités d'échec de ces opérations étant supposées constantes et estimées sur la durée d'utilisation du système.

[H8] Les contraintes environnementales seront supposées constantes (profil pire cas) pour la définition de la stratégie d'amélioration architecturale.

[H9] L'ajout de voie de redondance sur l'architecture existante ne peut être considéré comme une solution satisfaisante que si la séquence d'entités redondée ne peut être testée de manière correcte par le « superviseur ».

[H10] La procédure de mise en position sûre par le superviseur est considérée comme indépendante du système observé et se traduit par une action directe sur le circuit de contrôle des actionneurs de la fonction dédiée sécurité.

[H11] L'ajout de procédure de diagnostic et de mise en position sûre au niveau du superviseur est de coût négligeable, si on suppose des combinaisons d'erreurs, portant sur un ensemble de ressources testables.

Ces hypothèses sont en accord avec la plupart des approches industrielles, qui vont avoir tendance à ajouter des procédures de détection et de recouvrement dans l'architecture et à modifier la fiabilité des ressources matérielles. Nous désirons proposer néanmoins une stratégie pertinente, permettant la hiérarchisation des modifications à effectuer, pour garantir les inégalités **(Eq.47)** et **(Eq.48)**.

L'intérêt de ces propositions est de permettre une identification rapide des listes La et Ld, après le processus de modification, et donc d'évaluer facilement les indicateurs de performances de l'architecture finale. Pour ce faire, il suffira de modifier les listes en fonction des combinaisons d'erreurs détectées par le superviseur, et de modifier dans les processus stochastiques, modélisant l'évolution des probabilité d'erreurs matérielles, les taux de transition et la valeur des horloges TI, des ressources matérielles. On pourra ensuite relancer le calcul général du chapitre 2 **(§3-4)**. Néanmoins, pour éviter des calculs inutiles, nous allons proposer une stratégie de modification de l'architecture, permettant de **garantir** sous l'hypothèse **H8** que la relation **(Eq.47)** est bien vérifiée. Une fois cette stratégie proposée, on pourra vérifier les valeurs des indicateurs de performances du nouveau système, par simulation dynamique (chapitre 2).

Nous allons maintenant expliquer comment cette stratégie de modification de l'architecture du système est choisie et sur quels critères sont basés les processus de hiérarchisation et de garantie en termesss du facteur de distance d. Nous allons ainsi rappeler la notion de coupe minimale et définir une stratégie de modification en deux temps, basée à la fois sur l'ajout de procédure de détection et de recouvrement d'erreur et sur la modification de propriétés de fiabilité matérielles locales.

2- Stratégie proposée à partir des listes L_d.

La stratégie d'amélioration architecturale a pour but la réduction du critère d sous une triple **contrainte** :

- une contrainte de coûts de mise en œuvre aussi faible que possible
- une contrainte d'évolution faible de la perte économique induite par le déclenchement intempestif de la fonction dédiée sécurité.
- le respect d'un ensemble de contraintes architecturales existantes.

Nous ne nous intéresserons qu'à l'évitement et qu'à la prévention de combinaisons d'erreurs matérielles.

2-1- Notions de combinaisons d'erreurs minimales.

La notion de **coupe minimale** est généralement définie comme un « ensemble de composants qui s'ils sont tous défaillants alors que les autres sont dans un état correct entraîne la défaillance

du système ». Notre événement redouté est ici la défaillance sur demande de la fonction dédiée sécurité. Cette définition introduit encore le concept de coexistence d'état défaillant. Nous voulons prendre en compte les séquences d'activation d'erreurs données par la liste Ld.

Nous proposerons donc en remplacement de cette notion de coupe minimale, la notion de **combinaison minimale d'erreurs matérielles activées**. Cette combinaison correspond à l'ensemble des erreurs matérielles activées sur un cycle T et induisant un mode de défaillance sur demande de la fonction de sécurité.

On notera $\mathbf{E_{d,mat}}$ l'ensemble des **combinaisons minimales d'erreurs <u>matérielles</u> activées (cmema)** induisant une défaillance sur demande de la fonction dédiée sécurité. Par définition de $\mathbf{L_d}$ (chapitre 3), $\mathbf{E_{d,mat}}$ est donné par le sous ensemble des mots de L_d qui ne contiennent que des événements d'erreurs matérielles activées. Il peut donc être complètement identifié à partir de L_d.

La détermination des cmema est donc basée sur une approche dynamique de construction de la liste $\mathbf{L_d}$**,** prenant implicitement en compte les contraintes de partage de ressources et de précédence de l'architecture de départ. Nous allons, après avoir construit, $\mathbf{E_{d,mat}}$ déterminer les combinaisons d'erreurs matérielles, dont la probabilité d'existence devra être réduite ou qui donneront lieu à l'ajout d'une procédure de recouvrement.

2-2- <u>Notion de combinaisons d'erreurs minimales réduites</u>

La taille de l'ensemble $\mathbf{E_{d,mat}}$ peut s'avérer un obstacle au choix des modifications architecturales à apporter. Aussi on va réduire cet ensemble en définissant l'ensemble réduit des combinaisons minimales, notées $\mathbf{F_{d,mat}}$. Cet ensemble est en fait un sous ensemble de $\mathbf{E_{d,mat}}$, construit par réunion des sous ensembles C^i_d définis comme suit.

On note L le cardinal de la plus grande combinaison de $\mathbf{E_{d,mat}}$. On rappelle à ce titre que L est fini et que toutes les combinaisons de $\mathbf{E_{d,mat}}$ sont constituées d'erreurs activées lors du cycle T, portant sur des ressources matérielles deux à deux distinctes (comme sous ensemble de $\mathbf{L_d}$). On notera Card(.) l'opérateur qui à toute combinaison de $\mathbf{E_{d,mat}}$ associe le cardinal de cette combinaison.

On définit C^1_d comme l'ensemble des combinaisons de $\mathbf{E_{d,mat}}$ de cardinal 1, c'est à dire constitué d'un unique état d'erreur matériel.

On définit, par récurrence de j=1 à j=L-1, l'ensemble C^{j+1}_d, comme l'ensemble des combinaisons de $\mathbf{E_{d,mat}}$ de cardinal j+1 et ne contenant aucune sous combinaison d'éléments de l'ensemble

$$H^j = \bigcup_{k=1..j} C^k_d \qquad \textbf{(Eq.50)}$$

On définit ensuite $\mathbf{F_{d,mat}}$, par l'égalité, $F_{d,mat} = \bigcup_{J=1..L} C^J_d$ **(Eq.51)**

L'ensemble $\mathbf{F_{d,mat}}$ **est appelé ensemble réduit des combinaisons minimales (ERCM)**. Son calcul pouvant être automatisé à partir de $\mathbf{E_{d,mat}}$ en parcourant ses combinaisons suivant un cardinal croissant.

Chacun de ces éléments est une combinaison minimale d'erreurs matérielles. La réalisation de celles-ci est une condition suffisante à l'existence de toute autre combinaison minimale d'erreurs matérielles (ensemble générateur). Ces combinaisons ont, en outre, pour intérêt qu'aucune d'entre elles n'est contenue par les autres. On parle souvent dans la littérature d'impliquant premiers d'une défaillance sous demande. La détection d'une des combinaisons de $\mathbf{F_{d,mat}}$ restera donc sans effet sur les détections éventuelles d'autres combinaisons de cet ensemble. Elle pourra, par contre, entraîner la détection de toutes les combinaisons de $\mathbf{E_{d,mat}}$ la contenant.

Nous proposons de travailler sur la réduction d'occurrence ou la détection des combinaisons d'erreurs de $\mathbf{F_{d,mat}}$ pour améliorer de manière effective le PFD_{avg} de notre système d'étude suivant une stratégie en deux temps.

2-3- Orientation stratégique en deux temps.

Comme nous l'avons précédemment souligné, nous allons appuyer notre stratégie de modification architecturale sur deux principes : d'une part sur le principe de détection de combinaisons d'erreurs minimales et de mise en position sûre du système et d'autre part sur celui de réduction de l'occurrence d'erreur matérielle d'ordre 1. Ce second principe sera basé sur l'augmentation de la fiabilité des ressources matérielles critiques, pour lesquelles l'activation d'un mode de défaillance est une condition suffisante à une défaillance sous demande du système. Cette augmentation pourra être effectuée soit par remplacement de cette ressource par une ressource plus robuste, soit par l'augmentation de sa période de maintenance préventive.

Nous allons privilégier, dans un premier temps, le principe de détection des **ERCM** et de mise en position sûre sur l'ensemble de combinaisons

$$\bigcup_{J=2..L} C_d^J = F_{d,mat}/C_d^1 \quad \textbf{(Eq.52)}$$

Nous choisirons donc le sous ensemble de $\bigcup_{J=2..L} C_d^J$ dont les combinaisons peuvent être directement activées par une sollicitation du superviseur (commandables) et qui se manifeste par une défaillance observable de la séquence de ressources matérielles sollicitées. Ce sous ensemble, correspond aux combinaisons d'erreurs détectables par le superviseur. Ainsi on peut prévoir une politique de détection de certaines combinaisons et de mise en position sûre du système en présence de celles-ci. Ce sous ensemble sera noté $F_{d,mat}(det)$.

Le choix de la mise en place de diagnostic sur les combinaisons de $F_{d,mat}(det)$, dépendra essentiellement du poids des probabilités moyennes des combinaisons le constituant. On devra évaluer le gain sur d de chacune de ces modifications et privilégier la détection de combinaisons de $F_{d,mat}(det)$ ayant un poids non négligeable sur la valeur du PFD_{avg}(init). Le critère de sélection des combinaisons d'erreurs à détecter s'effectuera par hiérarchisation des combinaisons minimales suivant leurs probabilités estimées $F_{d,mat}(det)$ et par troncature de cette liste. Une fois ces procédures de détection et de mise en position sûre effectuées, nous serons en mesure de déterminer une distance garantie, atteinte grâce à ces modifications.

Si celle-ci est supérieure à 0 (non garantie de l'inégalité **(Eq.47)**), nous envisagerons d'appliquer une stratégie d'amélioration de la fiabilité de ressources matérielles, dites critiques, c'est-à-dire dont un mode de défaillance se traduit de manière nécessaire et suffisante à une défaillance sous demande de la fonction de sécurité.

La mise en œuvre de cette stratégie nécessite la hiérarchisation des combinaisons de $F_{d,mat}$ suivant leur influence sur l'indicateur PFD_{avg}. Nous allons donc dans la partie suivante expliquer comment ces combinaisons sont hiérarchisées et quels critères nous permettent d'évaluer le gain garanti apporté par la prévention de ces combinaisons.

3- Hiérarchisation des combinaisons minimales et évaluation du gain garanti

3-1- Notion de distance à objectif et mise en œuvre de notre stratégie

Toute stratégie de modification d'architecture a pour but d'atteindre un niveau de probabilité de défaillance sur demande garanti d'une fonction de sécurité, en se basant sur le critère de distance à objectif, noté d et de distance garantie notée d*.

Pour tout système, le critère de distance à objectif d est défini comme la différence entre la probabilité moyenne de défaillance sous demande du système et le seuil prescrit, noté α, dans la relation (1) (chapitre 6, 1-1). Au début du processus de modification, nous avons fait l'hypothèse que ce critère d(init), calculé pour le système de départ est strictement positif.

Nous avons proposé une stratégie en deux temps. La première étape s'appuie sur les combinaisons de $F_{d,mat}(det)$. Par l'ajout d'un superviseur, on suppose que les combinaisons d'erreurs de cet ensemble sont détectables et peuvent donc permettre la mise en position sûre du système. Nous voulons savoir quelles combinaisons de cet ensemble seront à détecter par le superviseur et quels seront les effets d'une telle procédure de détection/mise en position sûre sur la distance d.

Pour ce faire, nous devons hiérarchiser ces combinaisons selon leur influence sur le critère d et être en mesure d'estimer le gain que peuvent entraîner l'ajout de procédures de détection/mise en position sûre appliquées aux combinaisons dont le poids a été jugé prédominant. Nous devrons donc estimer une métrique de **gain garanti**, notée g*, permettant de calculer la réduction garantie de la distance d induite par un ensemble fini de modifications architecturales. Ce point sera traité dans la paragraphe 3-2. Une fois ce gain calculé après l'ajout de procédures de détection/ mise en position sûre sur un sous ensemble choisi de $F_{d,mat}(det)$, deux cas sont envisageables :

Soit g*≥d(init), nous pourrons alors affirmer que nous avons atteint notre objectif et qu'il est donc possible par l'application d'une stratégie de détection/mise en position sûre portant sur cet ensemble de combinaisons d'atteindre l'égalité (1). Cette stratégie sera donc jugée comme pertinente et suffisante.

Si g*<d(init), nous ne serons pas en mesure d'affirmer que nous avons atteint notre objectif. Nous compléterons donc ces procédures de détection/mise en position sûre par des mesures portant sur les ressources matérielles critiques (équivalent aux éléments de l'ensemble C_d^1), c'est à dire dont un mode de défaillance induit de manière suffisante un mode de défaillance sous demande du système. On tentera d'augmenter la fiabilité de certaines de ces ressources (remplacement de ressource matérielle et/ou augmentation de la fréquence de maintenance préventive) afin de réduire la distance de la valeur D=d(init)-g*. Nous choisirons les modifications les plus pertinentes selon trois critères:

- la probabilité moyenne d'existence de ces modes de défaillances p*,
- le coût d'une modification portant sur cet élément (opération de maintenance ou remplacement), noté c
- et le facteur de gain garanti de la probabilité de cette modification, noté r.

Nous hiérarchiserons les modifications possibles et proposerons de compléter la stratégie précédente (détection/mise en position sûre) afin d'atteindre les objectifs affichés en partant d'un système modifié incluant les procédures de détection et de mise en position sûres choisies dans l'étape précédente.

Nous allons dans les parties suivantes expliquer les critères de hiérarchisation et la notion de gain garanti pour des combinaisons minimales. Nous distinguerons le cas de combinaisons minimales d'erreurs détectables de longueur supérieures à 2 (éléments de $F_{d,mat}(det)$) et expliquerons le calcul du gain garanti g*, avant de traiter la hiérarchisation des erreurs de C_d^1 et la mise en place de la stratégie d'amélioration pour g*<d(init)

3-2- Critères de hiérarchisation des combinaisons d'erreurs détectable de longueur supérieur à 2 et calcul de g*

Dans cette sous partie, nous sommes intéressés par la première étape de la stratégie d'amélioration de notre système, c'est à dire par la détection de combinaisons d'erreurs de $F_{d,mat}(det)$ et la mise en position sûre dans le cas d'une détection par un superviseur.

Nous pouvons identifier en partant de la définition donnée en 2-3- un ensemble fini de combinaisons d'erreurs matérielles détectables de longueurs supérieures ou égales à 2.

Nous nous placerons, dans l'hypothèse de contraintes environnementales constantes et d'absence de phénomènes d'usure portant sur les composants de nos ressources matérielles. Nous supposerons, en outre, que les probabilités de succès dans les opérations de maintenance sont parfaites pour chaque ressource matérielle. Sous ces hypothèses, l'évolution des probabilités d'erreurs de chaque ressource matérielle peut être donnée par une chaîne de Markov homogène de pas d'itération T*. La probabilité moyenne de chaque erreur matérielle peut ainsi être évaluée. Ces calculs seront effectués grâce à l'outil ©CARMS. Celui-ci permet une simulation rapide de l'évolution de ces chaînes dans le cas homogène.

Les combinaisons de $F_{d,mat}(det)$ sont constituées d'un nombre fini d'erreurs matérielles, portant sur des ressources distinctes. Soit m un élément de $F_{d,mat}(det)$ et L le nombre d'erreurs matérielles la constituant. Il existe L erreurs matérielles, notées $e_1(m), e_2(m), \ldots e_L(m)$ portant sur des ressources deux à deux distinctes, telles que $m = e_1(m).e_2(m)\ldots.e_L(m)$. On aura donc :

$$p(m,t) = \prod_{i=1..L} p(e_i(m),t) \quad \textbf{(Eq.53),} \qquad \text{pour tout k entier vérifiant } t=kT$$

La probabilité d'occurrence moyenne d'une combinaison m de $F_{d,mat}(det)$ est donnée par la moyenne temporelle du produit des probabilités d'occurrence des erreurs dont elle est constituée. Celle-ci ne correspond pas en général au produit des moyennes de chaque erreur matérielle prise indépendamment. En effet, on aura en général:

$$p_{avg}(m) = \frac{T}{T_{life}} \sum_{k=0\cdots N} p(m,kT) = \frac{T}{T_{life}} \sum_{k=0\cdots N} \left\{ \prod_{i=1..L} p(e_i(m),kT) \right\} \neq \frac{T}{T_{life}} \prod_{i=1..L} \left\{ \sum_{k=0\cdots N} p(e_i(m),kT) \right\} = \prod_{i=1..L} p_{avg}(e_i(m)) \quad \textbf{(Eq.54)}$$

avec $N = E\left(\frac{T_{life}}{T}\right)$

Nous nous heurtons ainsi à une difficulté. En effet, on voit bien mathématiquement que l'intégrale d'un produit de fonction dépendant du temps, n'est pas égal au produit d'intégrales de chacune de ces fonctions. Il semble donc hasardeux d'approximer $p_{avg}(m)$ par le produit des termes $p_{avg}(e_i(m))$.

Nous allons donc proposer une approche permettant d'estimer au moyen d'une borne inférieure et supérieur la valeur de $p_{avg}(m)$ au moyen des valeurs des $p_{avg}(e_i)$, supposées estimées. On supposera pour ce faire que le profil d'évolution des probabilités d'erreurs $d_i(m)$ peut être ramené (sous l'hypothèse d'un temps moyen de réparation faible devant TI) au profil linéaire simplifié donné par la figure 52.

On suppose que la probabilité d'erreur croît de manière linéaire avec le temps entre deux périodes de maintenance préventive (suivant la pente α_i), que cette probabilité est ramenée à 0 après l'inspection et qu'elle est maintenue à une valeur, notée $c(e_i(m))$ durant l'inspection. Dans la pratique, $c(e_i(m))$ vaudra 0 pour des erreurs matérielles de type S ou D, et 1 pour des erreurs matérielles de type I.

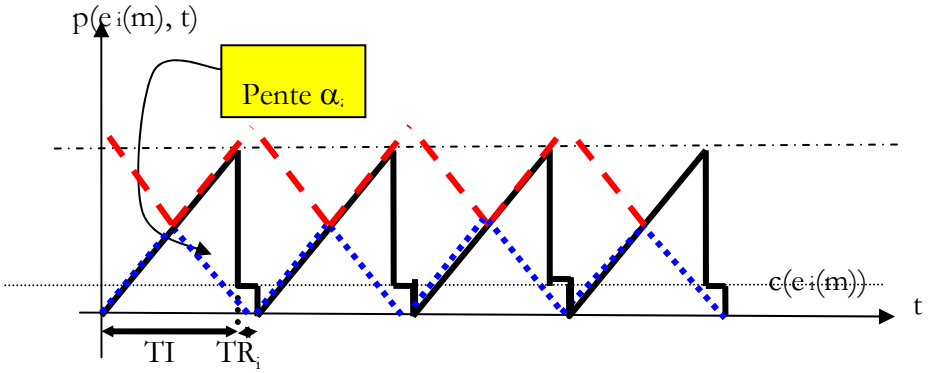


Figure 52 : Encadrement du profil d'évolution de la probabilité d'une erreur

Sur la **figure 52**, on va majorer la courbe par la courbe en triangle périodique (rouge) et la minorer par la courbe en triangle périodique bleue. La décomposition en série de Fourrier de ces courbes (TI+TR périodiques), nous donnera pour la courbe minorante l'équation :

$$f_1(t) = \frac{\alpha_i TI}{4} + \frac{\alpha(TI+TR)}{4}\sum_{k=0}^{\infty}\frac{(-1)^k}{(2k+1)^2}.\sin\left\{(2k+1)\pi\frac{(t-\frac{TI}{4}+\frac{TR}{2})}{(TI+TR)}\right\} \quad \textbf{(Eq.55)}$$

et pour la courbe majorante l'équation :

$$f_2(t) = \frac{3\alpha TI}{4} + \frac{\alpha(TI+TR)}{4}\sum_{k=0}^{\infty}\frac{(-1)^k}{(2k+1)^2}.\sin\left\{(2k+1)\pi\frac{(t-\frac{3TI}{4}+\frac{TR}{2})}{(TI+TR)}\right\} \quad \textbf{(Eq.56)}$$

Nous allons nous placer dans le cas simple Card(m)=2. Nous pourrons ensuite aisément généraliser au cas Card(m)>2.

Nous supposons que m est constitué de deux erreurs matérielles, notées a et b, les ressources sont maintenues préventivement suivant les périodes d'inspection respectives TI_a et TI_b et de temps d'inspection respectifs TR_a et TR_b. On suppose en outre que $TI_a<TI_b$ et que $TR_a<TR_b$. On note α_a et α_b les pentes des deux profils et c(a) et c(b) les deux constantes correspondant aux probabilités de ces modes durant l'inspection (chacune égale à une valeur constante égale à 0 ou 1).

On utilise le fait qu'une fonction produit de fonctions positives est majorée par le produit de fonctions majorant ses termes, et qu'elle est minorée par le produit de fonctions minorant ses termes. On en déduit que notre fonction p(m,t) est comprise entre les bornes $f_{min}(t)$ pour la courbe minorante et $f_{max}(t)$ pour la courbe majorante. On va uniquement développer les calculs sur la borne minorante, l'autre calcul se faisant de manière similaire.

$$f_{\min}(t) = \gamma_1 + \gamma_2.\sum_{n=0}^{\infty}\frac{(1)^n}{(2n+1)^2}\sin\left\{(2n+1)\pi\frac{\left(t-\frac{TI_a}{4}+\frac{TR_a}{2}\right)}{(TI_a+TR_a)}\right\} + \gamma_3.\sum_{n=0}^{\infty}\frac{(1)^n}{(2n+1)^2}\sin\left\{(2n+1)\pi\frac{\left(t-\frac{TI_b}{4}+\frac{TR_b}{2}\right)}{(TI_b+TR_b)}\right\} + \gamma_4.\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{(1)^{i+j}}{(2i+1)^2(2j+1)^2}.$$
$$\cos\left\{\pi\left[(2i+1)\frac{\left(t-\frac{TI_a}{4}+\frac{TR_a}{2}\right)}{(TI_a+TR_a)} - (2j+1)\frac{\left(t-\frac{TI_b}{4}+\frac{TR_b}{2}\right)}{(TI_b+TR_b)}\right]\right\} + \gamma_5.\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{(1)^{i+j}}{(2i+1)^2(2j+1)^2}\cos\left\{\pi\left[(2i+1)\frac{\left(t-\frac{TI_a}{4}+\frac{TR_a}{2}\right)}{(TI_a+TR_a)} + (2j+1)\frac{\left(t-\frac{TI_b}{4}+\frac{TR_b}{2}\right)}{(TI_b+TR_b)}\right]\right\}$$

(Eq.57)

avec $\gamma_1 = \frac{\alpha_a \alpha_b TI_a TI_b}{16}$, $\gamma_2 = \frac{\alpha_a \alpha_b (TI_a + TR_a) TI_b}{16}$; $\gamma_3 = \frac{\alpha_a \alpha_b (TI_b + TR_b) TI_a}{16}$ et $\gamma_4 = \frac{\alpha_a \alpha_b (TI_a + TR_a)(TI_b + TR_b)}{32}$

En intégrant cette fonction par morceau sur des intervalles de largeur T*, où T* est le plus petit commun multiplicateur des termes (TI_a+TR_a) et (TI_b+TR_b), et en sommant chacun de ses résultats, on obtient comme valeur de l'intégrale de $f_{min}(t)$ sur [0 ;T_{life}], la fonction $I_{\min}(T_{life}) = \gamma_1 T_{life} = \frac{\alpha_a \alpha_b TI_a TI_b}{16} T_{life}$. Ce résultat provient des propriétés de périodicité des fonctions sinus et cosinus de la fonction à intégrer.

On en déduit que $\frac{\alpha_a \alpha_b TI_a TI_b}{16} = \frac{1}{4} p_{avg}(a).p_{avg}(b).$ minore la valeur de $p_{avg}(m)$ recherchée **(Eq.58)**. Par la même approche, on prouvera que $\frac{9\alpha_a \alpha_b TI_a TI_b}{16} = \left(\frac{3}{2}\right)^2 p_{avg}(a).p_{avg}(b).$ majore la valeur de $p_{avg}(m)$ recherchée **(Eq.59)**.

On peut remarquer que cette démarche peut être généralisée pour tout mot de longueur m>2, en utilisant les formules trigonométriques usuelles et les propriétés de l'opérateur « plus petit commun dénominateur » (notamment sa distributivité sur l'ensemble des entiers).

Si on note : $\gamma_m = \prod_{i=1..L} p_{avg}(e_i(m))$ **(Eq.60)**, on aura donc :

$$\left(\frac{1}{2}\right)^L \gamma_m \le p_{avg}(m) \le \left(\frac{3}{2}\right)^L \gamma_m \quad \text{avec L= Card(M)} \qquad \textbf{(Eq.61)}$$

On notera par la suite, par soucis de simplification : $\delta_m = \left(\frac{1}{2}\right)^L \gamma_m$ **(Eq.62)**

On peut donc hiérarchiser les combinaisons minimales de longueurs ≥2 suivant la valeur de δ_m. On dira qu'une combinaison minimale m_a de longueur L_a a, de manière certaine, un poids plus important qu'une combinaison minimale m_b de longueur L_b.si $\left(\frac{1}{2}\right)^{L_b} \gamma_{m_b} \ge \left(\frac{3}{2}\right)^{L_a} \gamma_{m_a}$ **(Eq.63)**. On lui attribuera dans ce cas un ordre d'importance plus élevé.

Les combinaisons de $F_{d,mat}(\det)$ seront ainsi classées hiérarchiquement suivant leur poids relatifs sur la probabilité de défaillance de notre système. La détection d'une de ces combinaisons m et la mise en position sûre du système par un superviseur l'observant à chaque cycle T permettent donc de garantir une réduction de la probabilité de défaillance sous demande de la fonction d'une valeur garantie $\delta_m = \left(\frac{1}{2}\right)^L \gamma_m$ **(Eq.64)**, appelée gain unitaire.

Dans le cas où l'équation 63 n'est pas satisfaite, ce qui revient à dire que $\left(\frac{1}{2}\right)^{L_b} \gamma_{m_b} \ge \left(\frac{3}{2}\right)^{L_a} \gamma_{m_a}$ ou $\left(\frac{1}{2}\right)^{L_a} \gamma_{m_a} \ge \left(\frac{3}{2}\right)^{L_b} \gamma_{m_b}$, les bornes proposées ne nous ne permettent pas de conclure quand à la prédominance relative d'une combinaison d'erreurs sur l'autre. On attribuera donc, dans ce cas, un même niveau hiérarchique aux deux combinaisons.

Notre stratégie consiste à détecter et à éviter le plus grand nombre de combinaisons d'erreurs sur une période T en privilégiant les combinaisons de niveaux hiérarchiques élevés. On va donc déterminer un niveau seuil, noté S_d, sous lequel le ratio $\frac{\delta_m}{PFD(init)}$ est supposé négligeable et on va proposer de détecter, via le superviseur toute combinaison dont la valeur de δ_m permet de dépasser ce seuil.

En notant E(Sd) ce sous ensemble de $F_{d,mat}(det)$ (éléments de $F_{d,mat}(det)$ de plus haut niveau hiérarchique), on peut affirmer que la détection de l'ensemble des combinaisons de E(Sd), si elle est effectuée sur un cycle T par le superviseur, se traduira par un gain garanti g*, donné par la relation : $g^* = \sum_{m \in E(Sd)} \delta_m$.**(Eq.65)**

On a alors deux cas,

- soit g*>d(init), on peut conclure que cette stratégie de détection/mise en position sûre portant sur les combinaisons de E(Sd) est suffisante à l'atteinte de nos objectifs en termesss de PFD_{avg} (cas 1)
- soit g*<d(init), et on ne peut conclure quand à la suffisance de cette stratégie pour atteindre ces objectifs (cas 2)

Dans le cas 2, on cherchera à compléter cette stratégie par une augmentation de la fiabilité des ressources matérielles critiques. On notera D=d(init)-g*. On cherche donc une stratégie portant sur l'augmentation de la fiabilité d'un ensemble fini de ressources matérielles critiques, permettant de réduire le PFD_{avg} du système au-delà de la valeur D.

3-3- Critères de hiérarchisation des erreurs de C_d^1 et mise en place d'une stratégie dans le cas 2

Dans la sous partie précédente, nous avons expliqué comment hiérarchiser nos coupes minimales testables de longueur supérieure à 2. Nous avons proposé une stratégie de détection et de mise en position sûre sur un ensemble fini de combinaisons d'erreurs matérielles de $F_{d,mat}(det)$, que nous noterons E(Sd), en partant de cette hiérarchisation.

Dans le cas où g*<d(init), il nous est impossible de garantir au vu des égalités proposées, que cette stratégie de prévention sur ces combinaisons est suffisante à l'atteinte de notre objectif, donné par l'égalité (**Eq.47**). On peut cependant affirmer que pour la nouvelle architecture (incluant cette supervision), on aura nécessairement :

$$PFD_{avg} \leq PFD_{avg}(init) - g^* \quad \textbf{(Eq.66)}$$

et que $PFS_{avg} \leq PFS_{avg}(init) + h^*$ **(Eq.67)** avec $h^* = \sum_{m \in E(Sd)} \left(\frac{3}{2}\right)^L \gamma_m$

On peut remarquer que l'ajout du superviseur peut s'est accompagné d'une augmentation de la probabilité de mise en position sûre du système qui peut porter incidence sur le niveau de risque résiduel. On introduit une nouvelle métrique appelée gain garanti d'ordre 1 et noté G(int). On pose :

$$G(\text{int}) = d(init) - g^* + h^* . \frac{(\frac{T_{life}}{T} - f).I_{abs}}{fI_{dem}} \quad \textbf{(Eq. 68)}$$

Nous cherchons donc en modifiant la fiabilité moyenne d'un nombre fini de ressources matérielles de C_d^1 (soit par remplacement, soit par augmentation de leur période de maintenance

préventive), de réduire la valeur de la métrique PFD_{avg} de l'architecture avec superviseur d'une valeur au moins égale à G(int). Cet objectif nous permettra de garantir les inégalités de départ **(1)** et **(2)**.

On appellera dans toute la suite de notre travail, gain unitaire du à une modification i, noté g(R,i), un niveau de réduction garantie du PFD_{avg} de l'architecture due à l'augmentation des propriétés de fiabilité d'une ressource matérielle R appartenant à C_d^1. Nous allons d'abord rappeler les hypothèses nécessaires à cette évaluation et proposer un majorant de g(R,i) pour chaque type de modification pouvant porter sur une ressource matérielle R.

3-3-1- Evaluation de g(i,R)

Si on considère une ressource matérielle R de C_d^1, nous savons qu'il existe un état d'erreur X de cette ressource (avec X appartenant à {S,D,I}), nécessaire et suffisant, à l'existence d'un mode de défaillance sous demande du système d'étude. Nous noterons d(X,R) cet état d'erreur. Nous désirons réduire la probabilité moyenne de d(X,R) au cours du temps. Avant toute modification, nous supposons cette probabilité moyenne, notée $p_{avg}(d(X,R))$ comme connue.

Deux solutions seront envisagées : soit le remplacement de la ressource par une ressource matérielle de taux de défaillance $\Gamma_{o=>X}$ plus faible, soit la réduction de sa période de maintenance préventive d'un rapport, noté φ_R. Dans le cas d'une absence de procédure de maintenance préventive, on proposera la valeur par défaut de la période d'inspection de la ressource R, $TI_R = T_{life}$.

En nous appuyant sur les inégalités de la sous partie 1-2-2 de ce chapitre, et en supposant connu le taux de succès de l'opération de maintenance préventive γ_R, on peut écrire que dans le cas d'une augmentation de la fréquence de la maintenance préventive:

$$g(i,R) = p_{avg}(d(X,R)).\left(1-\frac{1}{\varphi_R}.\left[1+\frac{(1-\gamma).(\varphi_R-1)}{2}\right]\right) \quad \textbf{(Eq. 69)}$$

Dans le cas d'un changement de ressource matérielle, on notera ϕ_R le rapport entre la valeur minimum de $\Gamma_{o=>X}$ de la ressource remplaçante et de la valeur maximum de $\Gamma_{o=>X}$ de la ressource remplacée lors de leur utilisation. Dans ce cas, on est sûr que le niveau de réduction garanti est inférieur à $p_{avg}(d(X,R)).(1-\Phi_R)$

On prendra donc

$$g(i,R) = p_{avg}(d(X,R)).(1-\Phi_R) \quad \textbf{(Eq. 70)}$$

Dans le cas d'une modification basée à la fois sur le remplacement de la ressource et l'augmentation de sa fréquence d'inspection, on aura :

$$g(i,R) = p_{avg}(d(X,R)).(1-\Phi_R)\left(1-\frac{1}{\varphi_R}.\left[1+\frac{(1-\gamma).(\varphi_R-1)}{2}\right]\right) \quad \textbf{(Eq. 71)}$$

3-3-2- Notion de pertinence d'une stratégie modification sur C_d^1

La mise en place d'une stratégie par amélioration de la fiabilité locale sur ressources matérielles d'ordre 1 ne se heurte pas aux mêmes problèmes que celle à base de superviseur vue précédemment.

Dans la partie 3-2-, nous étions intéressés par l'identification et la hiérarchisation de combinaisons de longueur supérieure ou égale à 2 détectable. Le choix des combinaisons

diagnostiquées dépendaient essentiellement de leur poids relatif en termesss d'occurrence et des contraintes de temps liées à leur détection.

Le problème principal tient au fait qu'une amélioration de la fiabilité locale de ressources matérielles est liée à des contraintes fortes, en termesss de coût de remplacement et de coût induit par l'augmentation des procédures de maintenance préventive. Il est donc nécessaire d'introduire un indicateur permettant d'assigner au mieux les ressources financières pour atteindre notre objectif de gain garanti G(int).

Pour ce faire, on va associer à chaque ressource matérielle de C_d^1 d'une part la fonction gain, associée à un spectre limité de modification considéré, et d'autre part la fonction coût. La fonction gain a été précédemment donnée au chapitre 3-3-1 suivant la nature de la modification, à savoir remplacement de la ressource ou augmentation de la fréquence de maintenance préventive. La fonction coût est une estimation du coût de modification. Elle aura une valeur fixée dans le cas d'un remplacement de la ressource et une valeur dépendant de la période d'inspection TI dans le cas d'une augmentation de la fréquence d'inspection. On la supposera proportionnelle aux nombres des opérations de maintenance (et donc à TI).

Le coût d'une modification i sur une ressource R pourra donc s'écrire

$$c(i,R) = \Delta c_R + \varphi_R \Delta c_{m,R} N \qquad \textbf{(Eq. 72)}$$

avec Δc_R, le prix du remplacement de la ressource, $\Delta c_{m,R}$, le coût d'une opération de maintenance et N le nombre d'inspections préventives de la ressource dans l'architecture initiale (avant modification).

On dira qu'une stratégie est pertinente, si elle permet d'atteindre l'objectif de gain garanti G(int) pour un coût réduit, en termesss de mise en œuvre et de maintenance. Nous désirons identifier cette liste de modifications.

3-3-3- Mise en place de la stratégie sur ressources matérielle d'ordre 1

Après avoir identifié, les erreurs de C_d^1, leurs probabilités sont estimées en supposant les contraintes environnementales comme constantes sur [0 ;T_{life}] par résolution numérique de la chaîne de Markov associée. Pour chaque ressource matérielle R d'ordre 1, c'est-à-dire dont un état d'erreur appartient à C_d^1, on associe 2 indicateurs :

- La métrique A(Replace,R) correspond au max du rapport $\frac{p_{avg}(d(X,R)).(1-\Phi_R)}{\Delta c_R}$ pour les politiques de remplacements de la ressource matérielle. Elle permet de donner une estimation du gain g(R,i) ramenée sur le coût d'un remplacement matériel **(Eq. 72)**.
- La fonction B(Maint,R,φ_R) correspond au ratio $\frac{p_{avg}(d(X,R)).\left(1-\frac{1}{\varphi_R}.\left[1+\frac{(1-\gamma)(.\varphi_R-1)}{2}\right]\right)}{\varphi_R \Delta c_{m,R} N}$. Elle permet de donner une estimation du gain g(R,i) ramenée sur le coût d'un remplacement matériel **(Eq. 73)**.

On choisira le cas simple, où l'on augmente la fréquence de maintenance préventive d'origine **par un facteur entier** (2,3,4…). La modification consistant à augmenter la fréquence de maintenance préventive n'est donc, sous ces conditions, plus avantageuse que le remplacement que si $\Delta c_{m,R} N(1-\Phi_R) \geq \left(\Delta c_R\left[\frac{(1+\gamma)}{8}\right]\right)$ **(Eq. 74)**. On dira alors que l'erreur matérielle est de classe « Maint ». Dans le cas inverse, elle sera dite de classe « Replace ».

On va, pour chaque ressource matérielle d'ordre 1, associer les indicateurs A(Replace,R), B(Maint,R, $\varphi_R = 2$), une classe calculée suivant la formule précédente (« Maint » ou « Repl ») et deux attributs C(R) et g(R,i) pris initialement nuls.

Notre stratégie sera basée sur une modification itérative de l'architecture jusqu'à atteindre le gain garanti G(int).

Etape (G1)

On initie les variables : G à G(init) et C à 0 et on prévoit une liste vide pour annoter les modifications architecturales à effectuer

Etape (G2)

Si G>0,

On calcule l'argument R du max des indicateurs A(Replace,R) et B(Maint,R, $\varphi_R = 2$), pour les ressources matérielles R d'ordre 1.

On obtient en notant F0, l'ensemble des ressources matérielles dont un état d'erreur appartient à C_d^1,

$$R_i = Arg_{R \in F0}(Max(A(\mathrm{Re}\,place, R), B(Ma\,\mathrm{int}, R, \varphi_R = 2) \quad \textbf{(Eq. 75)}$$

On détermine ensuite la classe (« Maint » ou « Rep ») de R_i

1- Si la classe est « Rep »,

A- on ajoute un remplacement de la ressource Ri dans la liste des modifications et on ajoute à C la valeur correspondant au coût et on retranche à G le gain $g(i,R) = p_{avg}(d(X,R)).(1-\Phi_R)$ de cette modification

B- on remplace A(Replace,R) par 0 et on change la classe de « Rep » à « Maint »

C- on repart de l'**étape G1**

2- Si la classe est « Maint »,

A- on cherche φ_{R_i} vérifiant l'égalité $G = p_{avg}(d(X,R_i)).\left(1 - \frac{1}{\varphi_{R_i}}.\left[1 + \frac{(1-\gamma_i)(\varphi_{R_i}-1)}{2}\right]\right)$

B- on multiplie la fréquence de maintenance préventive de la ressource R_i par le φ_{R_i} calculé précédemment

C- on ajoute cette tâche de maintenance sur la ressource Ri dans la liste des modifications et on ajoute à C la valeur correspondant à ces actions $\varphi_R \Delta c_{m,R} N$ et on attribue à G la valeur 0.

Si G≤0,

On arrête la recherche des modifications et on en déduit la liste des modifications à apporter pour atteindre de manière sûre G(init) au moindre coût puis passage à l'**étape G3.**

Etape (G3)

La liste des modifications contient toutes les modifications nécessaires pour garantir la réduction de notre indicateur de performance PFD_{avg} d'au moins G(int). Elle sera donc retenue comme liste finale de modification sur les ressources matérielles et leur politique de maintenance préventive.

4- Bilan sur la stratégie complète.

Nous avons proposé une stratégie de modification en deux temps basée d'une part sur l'utilisation d'un superviseur et, complémentairement et d'autre part, sur une augmentation de la fiabilité moyenne de certaines ressources matérielles, jugées critiques. Elle se base sur

- le choix d'un nombre fini de combinaisons d'erreurs détectables par un superviseur et forçant la mise en position sûre de la fonction de sécurité (3-2)
- le choix de modifications matérielles (remplacement de ressources ou augmentation de la fréquence de certaines tâches maintenance préventive), permettant de réduire l'occurrence d'erreurs matérielles d'ordre 1, si possible à moindre coût.

Cette stratégie nous permet d'assurer un niveau de risque garanti pour notre installation, puisque la réduction du PFD_{avg} due à chaque modification a été minorée pour permettre une estimation pessimiste de la distance à objectif restant (d). Ce caractère pessimiste, même si il nous permet d'affirmer l'atteinte d'objectifs en termesss de risque résiduel, ne garantit aucunement une évaluation précise de nos nouveaux indicateurs de performances.

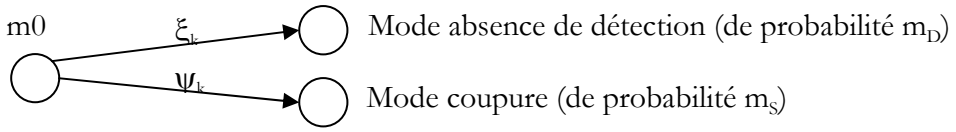
Le modèle d'évaluation de départ est essentiellement constitué de trois niveaux : le niveau environnement, auquel nous n'apporterons aucune modification, le niveau ressource matérielle et les deux listes caractéristiques La et Ld. Ces descriptions sont connues puisque ce modèle a été évalué et jugé insuffisant en termesss de fiabilité (hypothèses de départ de la partie 6).

Pour ce faire nous supposerons que la défaillance de la couche de supervision est assimilée à deux probabilités p1(t=kT) et p2(t=kT), très faibles devant les valeurs de nos métriques de performances. Ces deux événements ont pour conséquences respectives:

- une absence de détection des combinaisons d'erreurs testées sur un cycle T, mode m_D
- une mise en position sûre du circuit de contrôle des actionneurs (coupure), mode m_S.

L'état non erroné de la couche de supervision est donné par la probabilité de l'événement m_0.

Leur évolution est donnée par une chaîne de Markov non homogène simple et supposée connue. Cet ensemble étant supposé non testable et non réparable, on le modélisera par une chaîne de markov du type.



Nous pouvons alors identifier les listes La et Ld de l'architecture modifiée par rapport à celle de l'architecture de départ, que nous noterons par la suite $L_{init,a}$ et $L_{init,d}$.

La liste La est donnée par la réunion de :

- toute combinaison de E(Sd)
- l'événement m_S
- toute combinaison de $L_{init,a}$

La liste Ld est donnée par la réunion de :

- les combinaisons de E(Sd) auxquelles sont ajoutées l'événement m_D
- les combinaisons de $L_{init,d}$ ne contenant pas de sous combinaisons incluses dans E(Sd).

On peut donc très rapidement construire La et Ld du système modifié et les réutiliser en vue d'une évaluation des nouveaux indicateurs de performances, une fois que les modèles ressources auront été modifiés.

Ces changements n'ont aucune incidence sur la taille du pas de calcul du modèle général, si on démontre que les taux de défaillance des ressources de remplacement sont peu sensibles aux changements environnementaux sur T* (cf fin du chapitre 4), et si on conserve la relation TI>>T*. On peut donc par de simples modifications du modèle d'évaluation de l'architecture de départ, évaluer les indicateurs de performances de notre architecture modifiée. Cet aspect est un intérêt important de la stratégie de modification retenue qui permet d'économiser fortement les temps nécessaires à la modification et à la validation du système.

La nécessité d'une évaluation des indicateurs de performances de l'architecture est significative. Elle va, en effet, nous permettre d'estimer avec plus de précision :

- l'influence des contraintes environnementales partagées et des fluctuations de ces niveaux de stress sur les performances fiabilistes du nouveau système (les gains garantis étant basés sur une hypothèse « pire-cas »)
- une estimation du niveau de risque résiduel permettant de promouvoir la suppression de certaines modifications non nécessaires (cas d'un dépassement important d'objectif)
- une estimation de la perte économique induite sur l'installation (possibilité d'ajustement par réduction de certains niveaux de détection)

Cette étape est donc obligatoire pour valider la nécessité de la totalité des choix architecturaux avancés. Elle peut déboucher sur une suppression des dernières modifications, qui ne s'avèreraient pas nécessaire à l'atteinte de l'objectif de risque résiduel.

5- Bilan du chapitre 6

Dans ce chapitre, nous avons voulu montrer l'intérêt de la construction des listes La et Ld, dans la mise en place d'une stratégie de modification architecturale, visant à réduire le niveau de risque résiduel induit par la fonction de sécurité. Nous avons avantagé volontairement certaines techniques (détection, mise en position sûre, remplacement de ressources matérielles, augmentation de la fréquence de maintenance préventive), sans nous attacher aux modifications pouvant porter sur des politiques de recouvrement fonctionnels ou des augmentations de niveaux de redondance.

Ce choix délibéré nous permet de modifier rapidement le modèle d'évaluation de performances pour valider le bien-fondé de ces modifications et d'inscrire notre stratégie dans une approche complémentaire de prévention et d'évitement de combinaisons d'erreurs critiques.

Cette stratégie nous a permis de garantir un niveau de **risque résiduel**, tout en s'attachant à l'attribution de ressources suivant des points jugés prioritaires. La recherche de garantie ne nous permet pas de proposer la stratégie minimale, puisqu'elle sous-évalue nécessairement l'impact de certaines modifications. Cependant, elle permet, après évaluation des indicateurs de performances de l'architecture finale, d'évaluer la différence entre performances atteintes et les objectifs affichés. Si les performances atteintes dépassent largement les objectifs affichés, il est possible de renoncer aux modifications dont l'impact est le plus faible.

Elle montre donc bien l'intérêt d'une méthode structurée pour les phases de modifications architecturales. Cette méthode s'appuie non plus sur une vision statique de

l'architecture fonctionnelle mais bien sur une modélisation dynamique (chapitre 2) permettant de mettre plus aisément en lumière les combinaisons d'erreurs activées menant à une défaillance fonctionnelle.

La construction des listes La et Ld par une méthode informatique comme celle proposée au chapitre 3 permet une bonne manipulation de ces listes, un calcul rapide des ensembles C_d^1 et $F_{d,mat}(det)$ et une mise en œuvre rapide des algorithmes permettant le choix de la stratégie de modification architecturale.

Néanmoins, on peut remarquer que le choix délibéré de distinguer deux cas suivant la longueur des combinaisons minimales (dichotomie), nous a été, à l'origine, imposé par le contexte industriel de thèse. Siemens voulait, en effet, privilégier une approche diagnostic/mise en position sûre jugée moins coûteuse à une approche d'évitement d'erreurs matérielles critiques (du fait des coûts induits en changement de matériel).

Dans le cas, où on doit recourir à un évitement d'erreur matériel pour garantir l'atteinte des objectifs de performance (recours à la phase 2 d'amélioration), il peut donc être intéressant de déterminer, si le diagnostic sur certaines combinaisons minimales d'ordre 2 dont les probabilités moyennes sont les plus faibles, n'est pas superflu. Dans le cas de figure, où le poids maximum de ces combinaisons est inférieur à la marge garanti (différence entre niveau de PFD_{avg} garanti et objectif), on pourra se passer des détections portant sur ces combinaisons.

Conclusions et Perspectives

Chapitre 7 - Conclusions et perspectives

Les fonctions électroniques et programmables dédiées à la sécurité sont devenues, ces dernières années, de plus en plus complexes. Le développement d'architectures distribuées et tolérantes aux fautes a accru le partage de ressources dans l'architecture fonctionnelle. L'architecture ne peut, de ce fait, plus être considérée comme une structure statique. En conséquence, nous avons établi, pour ce type de fonction, un nouveau modèle dynamique d'évaluation des performances fiabilistes.

La plupart des architectures EP sont aujourd'hui des systèmes fonctionnellement reconfigurables et matériellement réparables. Elles échangent différents types d'informations (données et diagnostic). **Leur décomposition fonctionnelle hiérarchique reste difficile voire impossible** puisque ces systèmes peuvent être considérés comme bouclés. Nous avons donc construit un modèle dynamique qui considère l'architecture fonctionnelle comme un ensemble de procédés de traitement d'informations qui suivent des règles déterministes. Les entités constitutives de ce modèle, de nature sous- fonctionnelles, utilisent de manière séquencée un ensemble fini de ressources matérielles. Elles peuvent donc partager à la fois des informations et des ressources matérielles (bus de communication, mémoire,…).

Nous avons pris en compte dans ce modèle les procédures de test et de reconfiguration. L**es procédures de test en ligne** permettent de contrôler le niveau d'intégrité de ressources matérielles. Elles ne peuvent dès lors plus être considérées comme indépendantes du système testé. Les ressources testées supportent certaines sous fonctions entrant dans les processus de traitement d'information. La réalisation de ces tests dépend aussi de contraintes temporelles pouvant retarder leurs résultats. L'ajout d'événements « échec de test » a permis de refléter l'impossibilité d'une détection due aux contraintes de temps de test.

Les politiques de recouvrement et de reconfiguration peuvent rendre l'architecture vulnérable aux fautes. Leurs mises en œuvre peuvent créer de nouveaux mécanismes de propagation d'informations erronées entre sous fonctions. Ils peuvent alors déboucher sur une défaillance du système. Nous avons donc pris en compte ces procédures et **leurs influences** sur les modes de défaillance de la fonction de sécurité **au cours du temps**.

Les modèles actuels prennent peu en compte les **défaillances multiples** et leurs conséquences. Nous avons donc préféré nous appuyer sur la notion d'activation d'erreurs. Cette notion permet d'expliquer les mécanismes de perturbation et de perte d'information par les entités sous fonctionnelles. Elle permet d'identifier les séquences d'erreurs activées menant aux défaillances du système. On peut ainsi distinguer l'évolution temporelle des probabilités d'erreurs, des effets de leurs activations.

Nous avons donc proposé un modèle d'évaluation (**chapitre 2**) basé sur la distinction entre deux échelles de temps.

Le temps global, à horizon fini T_{life} de pas d'échantillonnage T permet, d'une part, de modéliser l'évolution des taux d'apparitions d'erreurs pris constants sur chacune de ces périodes et, d'autre part, l'évolution des probabilités d'erreurs (modèles stochastiques du **chapitre 4**). Pour ce faire, on considère qu'une ressource matérielle conserve le même état durant une période T, hypothèse justifiée par l'ordre de grandeur de T et des taux d'apparitions de pannes.

La fenêtre de référence T permet de modéliser notre architecture fonctionnelle (fonction de test et de recouvrement incluses) par un modèle de flux d'information. On détermine, pour chaque entité constitutive, une représentation par automate d'états finis. Cette représentation montre, en fonction des erreurs activées, les déviations possibles du comportement de l'entité. Ces deux niveaux de représentation (modèle de flux d'information et automates finis) permettent

d'identifier l'ensemble des séquences d'erreurs activées (La, Ld) menant à chaque mode de défaillance du système (**chapitre 3**). Le processus de construction de ces séquences a été automatisé (logiciel AFMCI, Java) et utilisé sur les cas d'étude du **chapitre 5**.

Notre approche s'appuie sur une distinction forte entre les concepts d'**apparition**, d'**existence** et **d'activation d'erreurs**. Notre modèle (**chapitre 2**) permet de considérer implicitement :

- l'influence de contraintes environnementales sur l'évolution de l'état de différentes ressources matérielles (couplage entre taux de transitions et contraintes environnementales proposées au **chapitre 4** (1-2)) ;
- les contraintes architecturales de type partage de ressources et ordonnancement des sous fonctions (par la construction des listes La et Ld -**chapitre 3-**) ;
- l'existence d'erreurs matérielles sur une même période T (évolution en parallèle des processus stochastiques donnant les probabilités d'erreurs -**chapitre 4**-).

Elle aboutit à l'estimation des indicateurs de performances de la fonction de sécurité (PFD_{avg}, PFS_{avg}).

L'utilisation du modèle est possible à la fois pour les architectures cohérentes et non cohérentes, en termes de structure fiabiliste. Ceci revient à segmenter judicieusement les ressources matérielles pour assurer à chacune de ces ressources une propriété de cohérence.

Cette utilisation a clairement été illustrée au travers du premier exemple du **chapitre 5**, par la prise en compte des modes de défaillance S au niveau de la vanne (blocage en position fermée) et de son circuit de contrôle (technologie de mise en position « sûre » par relais actif). Elle a permis de mettre en évidence les différences sur l'estimation des PFD_{avg} donnés par notre méthode d'évaluation et par une méthode classique (bloc de fiabilité/graphe de Markov). Cette dernière ne prend pas en compte la possibilité pour le système de passer d'un mode défaillant à un mode de fonctionnement correct du fait de l'apparition d'une erreur matérielle supplémentaire dans l'architecture. Les écarts entre les résultats obtenus, de faible amplitude sur cet exemple simple, peuvent devenir plus importants dans une architecture complexe, comme celle donnée à la fin du **chapitre 5**.

La méthode proposée considère, en outre, <u>trois aspects supplémentaires</u> non traités actuellement et les intègre directement à l'estimation des indicateurs de performances :

- la possibilité qu'une contrainte de temps de test ne permette pas la détection d'une faute (introduction de l'événement « échec de test » dans les automates d'états finis) ;
- la possibilité d'altération d'un signal de diagnostic et ses conséquences ;
- l'existence de défaillances dues au transfert ou au stockage de l'information (erreur environnementale).

Enfin, elle permet la prise en compte de certains modes de défaillance qui posent actuellement problème (défaillance de cause commune, mode latent) en les intégrant à la classe générale des défaillances simultanées. Elle distingue, d'une part, les couplages environnementaux portant sur les phénomènes d'apparitions et de disparitions d'erreurs matérielles et d'autre part, les couplages architecturaux qui permettent de juger des conséquences d'activation d'erreurs dans l'architecture fonctionnelle. Cette distinction est permise grâce à l'hypothèse selon laquelle une ressource matérielle présentera le même état d'erreur (combinaison de composants défectueux) sur une période de référence T. Cette hypothèse peut être considérée comme raisonnable au vu des ordres de grandeurs des taux de pannes, de réparation et du temps caractéristique T.

L'utilisation d'une méthode dynamique de construction de listes caractéristiques s'avère donc avantageuse pour une identification complète des séquences d'erreurs activées sur un cycle T induisant un mode de défaillance. Ces séquences peuvent d'ailleurs être mises à profit pour orienter les stratégies de modifications éventuelles de l'architecture, comme proposé au **chapitre 6.** On s'appuie à la fois sur une couche de diagnostic et de mise en position « sûre » pour les combinaisons minimales d'ordre >2 et, en complément, sur une augmentation de la fiabilité de certaines ressources matérielles jugées critiques. Notre méthode a donc le double intérêt de permettre l'estimation d'indicateurs de performances fiabilistes et l'orientation pertinente des modifications architecturales visant à atteindre un niveau de risque résiduel acceptable.

Malgré les avantages de notre méthode au niveau des avancées théoriques, on peut relever quatre problèmes.

Le premier réside dans la mise en œuvre pratique de ces propositions dans l'industrie. La mise en place d'un prototype de calcul des listes caractéristiques (AFMCI) a montré la possibilité d'un calcul automatique de ces listes à partir de la saisie manuelle du modèle (chapitre 3). L'étude a été développée pour des fonctions dédiées sécurité simple se basant sur une logique décisionnelle par vote logique, voire reconfiguration de ces votes. Ce spectre recouvre un nombre important de fonctions dédiées sécurité implantés dans l'industrie et les applications de transport. Ce procédé de construction peut s'avérer long. Cet inconvénient peut facilement être réglé par l'implantation de librairies de blocs (automates d'états finis) et d'assemblage de blocs (cf. figure 49, chapitre 5). Il est cependant clair que pour des systèmes plus complexes, nous aurions tout intérêt à étendre ces propositions par une approche objet, permettant de faciliter la conception des modèles, à la fois de haut et de bas niveaux. Le couplage des listes obtenus à un module de calcul logique (FT) rend possible l'évaluation numérique directe. En outre, la longueur des listes caractéristiques obtenues pour des architectures complexes impose un couplage automatique avec un outil d'évaluation numérique performant. Ce couplage peut être effectué sur **Aralia** dans le cas de taux de pannes constants. La prise en compte de l'évolution dynamique des taux de pannes des composants matériels imposerait, en sus, le développement d'interface dynamique (Aralia/ Carms).

Le deuxième problème est la génération automatique de la stratégie d'amélioration architecturale à partir de nos listes caractéristiques. Cette génération s'appuie sur les algorithmes proposés au **chapitre 6**.

La qualité des données de base utilisées pour le modèle d'évaluation numérique peut, en outre, générer des incertitudes importantes. Elles peuvent donc fortement biaiser les estimations des indicateurs de performances et ce, bien que nous ayons fortement réduit les incertitudes du modèle. Il pourrait donc être intéressant à partir des listes caractéristiques de proposer un calcul d'incertitude sur nos indicateurs de performances, afin de déterminer le niveau de qualité des taux de pannes utilisés.

Enfin, nous nous sommes intéressés pour le moment à des exigences portant sur des probabilités moyennes de modes de défaillance de la fonction de sécurité (pire cas). Il pourrait être judicieux de considérer conjointement l'introduction de plusieurs fonctions de contrôle-commande visant à éviter ou à réduire l'impact d'un accident. On pourrait ainsi imaginer la possibilité de collaboration entre différentes fonctions dédiées sécurité implantées. On se placerait alors dans une approche dynamique entre fonctions de sécurité et environnement (fiabilité dynamique). Si on conservait un certain nombre de vecteurs instantanés {PFD(kT); PFS(kT)} avec k fixé, en partant de la méthode d'estimation donnée au chapitre 2, il serait possible de simuler un tel couplage. On pourrait donc simuler l'évolution de l'environnement au cours du temps **[Lab00]**. Cette proposition pourrait aboutir à des évaluations plus réalistes du niveau de risque et des pertes économiques de l'installation (estimation du risque) que les équations proposées au chapitre 1. Elle permettrait, en outre, de prendre en compte des stress environnementaux communs lors de la détermination des profils pour les différentes fonctions

dédiées sécurité. Elle ne pourra cependant être appliquée que si ces différentes fonctions ne partagent aucune ressource matérielle et n'échangent aucune information (F- ou D-informations). En effet, la détermination des profils temporels {PFD(kT); PFS(kT)} pour chaque fonction se base sur des modèles informationnels de haut niveau (**chapitre 3**) supposés indépendants.

En plus de son intérêt pour une estimation plus précise des indicateurs de performances fiabilistes d'une fonction dédiée sécurité, les perspectives ouvertes par ce travail sont multiples. Elles vont du domaine de la fiabilité dynamique à l'étude systématique des incertitudes paramétriques et peuvent avoir des retombées importantes sur la validation de systèmes de contrôle - commande dédiés sécurité. Nous espérons donc que cette méthode débouchera sur des projets de développement d'outils informatisés sur les axes précédemment décrits.

REFERENCES

Chapitre 1

[98/37/CE]

Directive 98/37/CE concernant le rapprochement des législations des états membres relatives aux machines, publié au Journal Officiel C.E. n° L207, 23 juillet 1998, CE, Bruxelles, pp. 1-46

[Allo01]

ALLOCO, Mike, "Computer and Software Safety Considerations in Support of System Hazard Analysis", 21st International Safety Society Conférence, 4-9 août 2003, Ottawa, p.1030-1038, ISBN 0-9721385-3-6

[Aubr87]

AUBRY, Jean-François, "Conception des systèmes de commande numériques des convertisseurs électromécaniques : vers une méthodologie intégrant la sûreté de fonctionnement", Thèse d'Etat, Institut National Polytechnique de Lorraine, 1987, Nancy.

[Bace94]

BACELLI, François, BALBO, Gianfranco et all., "Annotated Bibliography on Stochastic Petri Nets", Performance Evaluation of Parallel and Distributed Systems: Solution Methods vol. 105, Centrum voor Wiskunde en Informatica O.J. Boxma and G.M. Koole(eds), pages 25-44, 1994, Amsterdam, 1994. disponibilité **http://webdiis.unizar.es/CRPetri/papers/jcampos/94_BBCC_QMIPS.ps.gz**

[Bart03]

BARTLETT, Lisa M., "Progression of the Binary Decision Diagram Conversion Methods",21st International Safety Society Conférence, Ottawa, 4-9 août 2003, p.116-125, ISBN 0-9721385-3-6.

[Behn00]

BEHNIA, Sonia, "Test de modèles formels en B : Cadre théorique et critères de couverture", Thèse de doctorat, Institut National Polytechnique, Toulouse, 27 Octobre 2000, disponibilité **http://www2.laas.fr/laas/1-4266-Publications.php** (Rapport LAAS 1709)

[Beou93]

BEOUNES; Christian, "SURF-2: A Program for Dependability Evaluation of Complex Hardware and Software Systems", 23rd International. Symposium on Fault-Tolerant Computing, Toulouse, France, pp. 668-673, 1993

[Berg04]

BERGER, Pavol, THIRIET Jean-Marc, ROBERT Michel "Dynamic Reliability and Availability Evaluation and Validation of distributed Control Systems" 19th IEEE Instrumentation and Measurement Technology Conference (IEEE/IMTC2002), Anchorage (Alaska, United States), 21-23th May 2002

[Beto00]

BETOUS-ALMEIDA, Claudia, "Construction et affinements de modèles de sûreté de fonctionnement- Application aux systèmes de contrôle- commande" ; Thèse de doctorat, Institut National Polytechnique, Toulouse, 11 Juin 2002, 159p., disponibilité **http://www2.laas.fr/laas/1-4266-Publications.php**

[Bier03]

BIER, Vicky, "Illusions of Safety", Nuclear Safety- Engineering, Dpt of Industrial Enginering, University of Wisconsin

[Bore96]

BORELL Marie, KANOUN Karama., "Dependability of Fault-Tolerant Systems – explicit modelling of interactions between HW and SW components", IEEE Transactions on Reliability, Vol.49, N°4, pp.363-376, décembre 2000, disponibilité: **http://www2.laas.fr/laas/1-4266-Publications.php** (Rapport LAAS 96046)

[Boui01]

BOUISSOU, Marc, BON, Jean-Louis, "A new formalism that combines advantages of fault trees and Markov models: Boolean logic driven Markov processes", Reliability Engineering and System Safety, Volume 82: Issue 2, pp. 149-164, novembre 2003., disponibilité **http://www.sciencedirect.com**

[Brab02]

BRABAND, Jens, BREHMKE, Bernd, "Application of Why-Because Analysis Graphs to Railway Near-Misses", , Workshops on Investigation and Reporting of Incidents and Accidents (IRIA), C.W.Johnson(ed), pp.26-32, Glasgow, 2002, disponibilité **http://www.dcs.gla.ac.uk/~johnson/iria2002/IRIA_2002.pdf**

[EN1050]

Standard Européen EN-1050, "Safety of machinery. Principles for risk assessment", 22p., 15 mars 1997, ISBN 058027153 6

[CEN01]

CENELEC; "Railway applications Systematic allocation of safety integrity requirements", Comité Européen de Normalisation Electrotechnique; Standard; réference R009-004:2001, 8 June 2001, disponibilité **www.cenelec.org**

[Chab98]

CHABOT, Jean-Luc, DUTUIT, Yves, SIGNORET, Jean Louis, "Simulation hybride, méthode de modélisation intégrant phénomènes continus et discrets", actes de conférence, MOSIM 99' ; Annecy, 6-8 octobre 1999

[Char02]

CHARPENTIER, Philippe, "Architecture d'automatisme en sécurité des machines: Etudes des conditions de conception liées aux défaillances de mode commun", Thèse de doctorat, Institut National Polytechnique de Loraine (CRAN), octobre 2002, 129p

[Coco97]

COCOZZA-THIVENT, Christiane, "Processus stochastiques et fiabilité des systèmes", Collection Mathématiques et Applications, . Springer, n°28, 430p., 1997, disponibilité: www.springerlink.com

[Dear00]

DEARDEN, Andy, HARRISON, Michael, WRIGHT, Peter, "Allocation of function: scenarios, context and the economics of effort", International Journal of Human-Computer Studies, Vol. 52, Issue 2, pp. 289-318(30), February 2000, disponibilité: **www.sciencesdirect.com** (Elsevier)

[Demm02]

DEMMOU; Hamid, KHALFAOUI, Sarhane, GUILHEM, Edwige, VALETTE, Robert, "Critical scenarios derivation methodology for mechatronic systems", Reliability Engineering & System Safety, Vol.84, N°1, pp.33-44, Avril 2004, disponibilité: **http://www2.laas.fr/laas/1-4266-Publications.php** (Rapport LASS 03046)

[DO178B]

RTCA/DO-178B, **"**Software Considerations in Airborne Systems and Equipment Certification", published by RTCA (Radio Technical Commission for Aeronautics), 01/12/1992, disponible: **www.rtca:org**

[Duga99]

MANIAN, Ragavan, COPPIT, David, SULLIVAN, Kevin J., BECHTA DUGAN, Joanne, "Bridging the gap between systemsand dynamic fault tree models," Proc. of the 1999 Reliability and Maintainability Symposium pp 105-111, Janvier 1999.

[Dutu02]

DUFOUR, François, DUTUIT, Yves, "Dynamic Reliability, a new model", actes de conferences ESREL02/ Lambda Mu 13, Volume 1, p.350-358, 18-21 Mars 2002, Lyon

[ECSS-Q-80-03]

ECSS-Q-80-03: "Guidelines for Software Dependability and Safety Techniques", Comité Européen de Normalisation, ECSS-ESA Requirements and Standards Division, 08/2004, disponibilité **http://www.estec.esa.nl/ecss/wg/drafts/publicreview/ecss-q-80-03draft1 (8April2004).pdf**

[Fite03]

FITERMANN, Charles, "Bases et orientations des politiques de risque", Rapport Conseil Economique et Social, section activités productives de la recherche et de la technologie, Annexe 2, mars 2003. "**http://www.ces.fr/rapport/doclon/03031207.PDF**"

[EN50126]

EN50126:1999 ; "Railway applications. The specification and demonstration of reliability, availability, maintainability and safety (RAMS)"norme européenne (EN), published on 15/12/99, 74p., , juin 2002, ISBN 058035694 9

[Ever00]

EVERDIJ, M.H.C., BLOM, H.A.P, "Piecewise Deterministic Markov Processes represented by Dynamically Coloured Petri Nets", Rapport NLR-TP-2000-428, National Aerospace Laboratory, Pays Bas, 2000

[Fleu02]

FLEUROT, Joelle, EVRARD, Jean-Marc, CHAUMONT, Benoît, "Les études d'évaluation des termes sources sur les REP" IRSN, Rapport scientifique et technique 2002, figure 2, p.13, http://www.irsn.fr/net-science/liblocal/docs/docs_DIR/RST2002/Chap02_art2.pdf

[Geri02]

GERICKE, Jörg, LIGGESMEYER, Peter, "Eine Erweiterung der Unified Modeling Language zur Verfolgung von Software-Anforderungen in sicherheitskritischen Systemen", Informatik Forschung und Entwickung, ed. Springer, Vol.17, Issue 2, pp.60-67, 2002. **www.spingerlink.com**.

[Gobl00]

GOBLE, William M., BROMBACHER Aarnout (AC), BUKOWSKI, Julia V., "Using Stress-Strength Simulations to Characterize Common Cause", Probability Safety Assessment and Management, PSAM 04, Mosleh. A., Bari R. (ed), Springer, 1998/09, pp.399-448, ISBN:3540762620. **www.spingerlink.com**

[Gond80]

GONDRAN, Michel, PAGES, Alain, "Fiabilité des systèmes", livre, édition Eyrolles, 1980, ISBN :2-212-01582-8, disponibilité: **www.eyrolles.com**

[Hami03]

HAMIDI, Karim, MALASSE, Olaf, AUBRY, Jean-Francois, "SILKEY: A Tool for the automatic evaluation of safety and availability of multi-level Redundancies Architecture", 21st International Safety Society Conférence, 4-9 août 2003, Ottawa, p.1092-1102, ISBN 09721385-36

[Haye94]

HAYES-ROTH, F., "Architecture Based Acquisition and Development of software (ARPA Domain Specific Software Architecture Program)" Palo Alto, CA Teknowledge Federal Systems, 1994

[Holg02]

GIESE Holger, "Safety Critical Computer Systems", Cours, Université Paderborn, année 2002-2003, disponibilité **http://wwwcs.upb.de/cs/agschaefer/Lehre/Lehrveranstaltungen/Vorlesungen/SafetyCriticalComputerSystems/ WS0203/SCCS-II-25-77-6x.pdf**

[Holn01]

HOLNAGEL Erik, „Anticipating Failures: What Should Predictions Be About?", RTO Meeting Proceeding "The Human Factor in System Reliability –Is Human Performance Predictable?"- RTO-MP-032, January 2001, disponibilité **http://www.rta.nato.int/Pubs/RDP.asp?RDP=RTO-MP-032**

[IEC50191]

Vocabulaire Electrotechnique International, Chapitre 191 – Sûreté de fonctionnement et qualité des services – 1990

[IEC61069]

CEI 61069 Norme internationale- Mesure et commande dans les processus industriel- Appréciation des propriétés d'un systèmes en vue de son évaluation –Partie 5 Evaluation de la sûreté de fonctionnement d'un système.

[IEC61508]

IEC 61508:1998 "Functional safety of electrical/electronic/programmable electronic safety-related systems", International Electrotechnical Commission, 170p, juin 1998

[IEC61511]

EN 61508:2002 "Functional safety – Safety instrumented systems for the process industry sector", International Electrotechnical Commission, mars 2002

[IEC61513]

EN 61513:2003, "Nuclear power plants - Instrumentation and control for systems important to safety - General requirements for systems", International Electrotechnical Commission, 22 mars 2003

[IEC62061]

IEC 62061:2002 " Safety of machinery Functional safety of safety-related electrical, electronic and programmable electronic control systems"; International Electrotechnical Commission, 204 pages, mai 2002.

[IEC62278]

CEI/IEC 62278: "Applications ferroviaires, spécification et démonstration de la fiabilité, de la disponibilité, de la maintenabilité et de la sécurité (RAMS)", CENELEC R009-2004, sept. 2002

[ISO8402]

ISO 8402, "Management de la qualité et assurance de la qualité – Vocabulaire", International Standard Organisation, août 1995

[Joly04]

JOLY C., VALLEE A, "Analyse des risques et prévention des accidents majeurs, synthèse des attente vis-à-vis des études de dangers ", rapport INERIS DRA AVa/CJl/SCa - P46054, p.69, disponibilité **www.ineris.fr/recherches/download/62.pdf**

[Jong02]

DE JONGH, Johannes F.C., "Share Scheduling in Distributed Systems", Thèse de doctorat, Université de Deft (Pays Bas), février 2002, ISBN 90-90151-28-1, disponibilité **http://www.pds.ewi.tudelft.nl/pubs/ph_d/dejongh.pdf**

[Kauf75]

KAUFMANN, Arnold, GRONCHKO, G., CRUON, R., "Modèles mathématiques pour l'étude de la fiabilité des systèmes", livre, éditeur Masson, 1975, ISBN 22-25-4031-55, disponibilité **www.masson.fr**

[Klet92]

KLETZ, Trevor, "Hazop and Hazan: Identifying and Assessing Process Industrial Hazards". Institution of Chemical Engineers, third edition, 1992, ISBN 0-85295-285-6

[Kvam98]

KVAM, Paul, "A parametric Mixture-Model for common-cause failure data", IEEE Transactions on Reliability, Vol. 47, No. 1, p.30–34, 1998, disponibilité **http://ieeexplore.ieee.org/iel4/24/15138/00690894.pdf?arnumber=690894**

[Labe00]

LABEAU, Pierre Etienne, SMIDTS, Christophe, "Dynamic Reliability, Toward an integrated platform for probability risk assessment", Reliability Engineering and System Safety (68), Elsevier, p.219-254, 2000, disponible: **www.sciencesdirect.com**

[Lapr96]

LAPRIE ; Jean-Claude et all., "Guide de la sûreté de fonctionnement", Cepadues Edition, 1996, ISBN 2-85428-382-1, disponibilité **http://www.cepadues.com/livre_details.asp?l=339**

[Lapr04]

AVIZIENIS Algirdas, LAPRIE ; Jean-Claude, RANDELL, Brian et all. "Basic Concepts and Taxonomy of Dependable and Secure Computing", IEEE Transactions on Dependable and Secure Computing, Volume 1, Number 1, pp. 11-33, January-March 2004, disponibilité **www.computer.ord/tdsc/**

[Leve00]

LEVESON, Nancy, "Evaluating Accident Models Using Recent Aerospace Accidents", NASA Report, Software Engineering Research Laboratory MIT, p.2-18, juin 2001

[Leve02]

LEVESON; Nancy, "A new Approach To System Safety Engineering", electronically Book (**http://sunnyday.mit.edu/book2.pdf**), June 2002

[Litt01]

LITTLEWOOD, Bev, POPOV, Peter, STRIGINI, Lorenzo, "Design Diversity: an Update from Research on Reliability Modelling", Proceedings Safety-Critical Systems Symposium 2001, ed Springer, Bristol, UK, 2001. disponibilité **www.springer.de**

[Loi03]

Loi n° 2003-699 du 30 juillet 2003 relative à la prévention des risques technologiques et naturels et à la réparation des dommages, Chapitre 5, article 21.

[Maur00]

MAURI, Giuseppe, „Integrating Safety Analysis techniques, Supporting Identification of Common Cause Failures", part 2.3 Common Cause Failure Analysis (pp.50-70), Thèse de doctorat, Université de York (HISE), septembre 2000, p.245, disponibilité **http://www.cs.york.ac.uk/ftpdir/reports/YCST-2001-02.pdf**

[Moll82]

MOLLOY, Michael K, "Performance analysis using Stochastic Petri Nets", IEEE Transactions on computers, vol. C31, n°9, p. 913-917, 1982, disponibilité **http://ieeexplore.ieee.org/**

[Mosl88]

MOSLEH, A., FLEMING, K.N., "Procedure for treating common cause failures in safety and reliability studies :procedural framework and examples", Pickard, Lowe, and Garrik, Inc, NUREG/CR-4780-V1, Jan. 1988

[Noye90]

NOYES, Daniel, ALDANONDO, Michel, "Dependability and Fault Tolerance Concepts for monitoring transportation systems in FMS", Advances in Information Systems Research, Ed. GE. Lasker, T.Koizumi, J.Pohl - Publ.IIASSRC, pp.176-182, 1991

[NUR00-02]

OCDE (organosation de coopération et de développement économique)

"ICDE Project Report on Collection and Analysis of Common-Cause Failures of Emergency Diesel Generators", Rapport NEA/CSNI/R2000(20), mai 2000

"ICDE Project Report on Collection and Analysis of Common-Cause Failures for Safety Valves and Relief Valves", Rapport NEA/CSNI/R2002(19), octobre 2002

disponibilité **http://www.nea.fr/html/jointproj/icde.html**

[Page04]

PAGETTI, Claire, "Extension temps-réel d'AltaRica", Thèse de doctorat, IRCCyN, Ecole centrale de Nantes and Université de Nantes, chapitre 2, Avril 2004, disponibilité **ftp://altarica.labri.fr/pub/publications/pagetti-phd.ps.gz**

[Pala98]

PALAMIDESSI; Castucia, GLASER, Hugh., MEINKE Karl, "Principles of Declarative Programming", 10th International Symposium, PLILP'98, Springer-Verlag, Berlin, Pisa, Italy, Sep 1998, ISBN 3-540-65012-1, disponibilité **www.springer.de**

[Perp00]

PERPEN, Jean, RUET, Magali, "Raisonnement à base de cas dans une structure de capitalisation des connaissances orientée objet", 8ème Atelier de Raisonnement à Partir de Cas (RàPC'2000), Toulouse, France, 9 mai 2000.

[PHA03]

"Process Hazards Analysis", Sutton Technical Books, livre électronique, disponibilité http://www.suttonbooks.net/downloads/hazop-sample-chapter.pdf

[Poin00]

POINT, Gerald, "AltaRica : Contribution à l'unification des méthodes formelles et de la sûreté de fonctionnement", Thèse de doctorat, LaBRI - Université Bordeaux I, Janvier 2000, disponibilité **ftp://altarica.labri.fr/pub/publications/these-point-2000.ps.gz**

[Purw01]

PURWANTORO, Yudi, BENETT, Stuart, "Hardware-Software Interaction in Dependability Modelling of Fault Tolerant System", acts of 17th Annual UK Performance Engineering Workshop (UKPEW'01), University of Leeds, UK, 18-19 juillet 2001

[Rauz97]

RAUZY, Antoine, DUTUIT, YVES, "Exact and Truncated Computations of Prime Implicants of Coherent and non-Coherent Fault Trees within Aralia", Reliability Engineering and System Safety, Vol.58, p.127–144, 1997

[Sato00]

SATO, Yoshinobu , KATO; Eiichi, "SIL Model for IEC61508, Examination of Modes of Operation",IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences, vol.E83-A, no.5, pp.863-865, 5 May 2000, disponibilité **http://www.ieice.org/eng/index.html**

[Sche03]

SCHERRER, Christoph, STEINIGER, Andreas, "Dealing with dormant fault in an embedded Fault-Tolerant Computer System", IEEE Transactions on Reliability, Vol.52, N°4, pp.512-522, December 2003, disponibilité **http://ieeexplore.ieee.org/xpl/abs_free.jsp?arNumber=1260601**

[Schn85]

SCHNEEWEISS, Winfred G., "Reliability Modeling", LiLoLe-Verlag GmbH, Hagen, Germany, 2001, disponibilité **http://www.amazon.de/exec/obidos/ASIN/393444704X/infoline-21/028-2865430-9571755**

[Scho02]

SCHOENIG, Raphael, "Définition d'une méthodologie de conception des systèmes mécatroniques sûres de fonctionnement", Thèse de doctorat, Institut National Polytechnique de Lorraine (CRAN), Nancy, 26 oct. 2004.

[Seri96]

SERICOLA, Bruno, "Availability analysis and stationary regime detection of Markov Models", IRISA (Institut de recherche en informatique et systèmes aléatoires) Rapport 2886, 17pp., 1996, disponibilité **ftp://ftp.irisa.fr/techreports/1996/PI-1013.ps.gz**

[SILVER]

SILVER – Outil pour la calculation de PFDavg proposé par Exida (**www.exida.com**) basé sur diagramme de fiabilité/ graphes de Markov homogènes.

[SINTEF]

"Offshore Reliability Data Handbook", editor SINTEF (Scandinavia), 4th Edition, 2002, disponible **http://www.sintef.no/static/tl/projects/oreda/handbook.html**

[Smet99]

DE SMET Olivier, COUFFIN, Stefane et all., "Safe programming of PLC using formal verification methods", actes de l'ICP'00, Utrecht, The Netherlands, , p.73-78, octobre 2000

[Tich02]

TICHY, Michael, „Durchgängige Unterstützung für Entwurf, Implementierung und Betrieb von Komponenten in offenen Softwarearchitekturen mittels UML“, Thèse de doctorat, Université de Paderborn; Allemagne, juillet 2002.

[Triv94]

FRICKS, Ricardo M., TRIVEDI; Kishor, “Modelling Failure Dependencies in Reliability using GSPN”, actes de l’European Simulation Multi-conference (ESM '97), Istanbul, Juin 1997, disponible **http://www.ee.duke.edu/~kst/spn_papers/fail.pdf**

[Vill87]

VILLEMEUR; Alain, “Sûreté de fonctionnement des systèmes industriels “, édition Eyrolles, 03/1997, ISBN 2-212-01615-8, disponibilité **http://www.eyrolles.com/**

[Z61-102]

AFNOR Norme Z61-102, “Norme vocabulaire Qualité Logicielle“, Association Francaise de Normalisation, 10 p., 1995.

[Zann95]

ZANNE, Christian, “Contribution à la conception des dispositifs de commande pour les systèmes dynamiques hybrides“, Rapport pour l’habilitation à diriger les recherches, Institut National Polytechnique de Lorraine, Automatique, 1995.

[Zwin95]

ZWINGELSTEIN, Gilles, “Diagnostic des défaillances - Théorie et pratique pour les systèmes industriels“, Traité des Nouvelles Technologies, série Diagnostic et Maintenance, Hermès, Paris, 1995, ISBN 2-86601-463-4, disponibilité **http://www.editions-hermes.fr/**

Chapitre 2

[Auch04]

AUCHET; Olivier; RIEDINGER, Patric, MALASSE Olaf; ”Glass furnaces, simplified modelling for control and real-time simulation”, International Symposium on Industrial Electronics, Ajaccio, France, 2004.

[Bore96]

BORELL Marie, KANOUN Karama., “Dependability of Fault-Tolerant Systems – explicit modelling of interactions between HW and SW components“, IEEE Transactions on Reliability,

Vol.49, N°4, pp.363-376, décembre 2000, disponibilité: **http://www2.laas.fr/laas/1-4266-Publications.php** (Rapport LAAS 96046)

[Boui01]

BOUISSOU, Marc, BON, Jean-Louis, "A new formalism that combines advantages of fault trees and Markov models: Boolean logic driven Markov processes", Reliability Engineering and System Safety, Volume 82: Issue 2, pp. 149-164, novembre 2003., disponibilité **http://www.sciencedirect.com**

[Char02]

CHARPENTIER, Philippe, "Architecture d'automatisme en sécurité des machines: Etudes des conditions de conception liées aux défaillances de mode commun", Thèse de doctorat, Institut National Polytechnique de Lorraine (CRAN), 2002

[Chen00]

CHEN, Xi, "Contribution to EMI/EMC Modelling and Characterization of CMOS Integrated Circuits", Thèse de doctorat, INSA Toulouse, France, novembre. 2000

[Duga99]

MANIAN, Ragavan, COPPIT, David, SULLIVAN, Kevin J., BECHTA DUGAN, Joanne, "Bridging the gap between systemsand dynamic fault tree models," Proc. of the 1999 Reliability and Maintainability Symposium pp 105-111, Janvier 1999.

[EN954-1]

EN954-1:1997 , "Safety of machinery. Safety related parts of control systems. General principles for design", Comité Européen des Normes, EN, 28p., 15 juin 1997

[FIDES]

"FIDES et Ingénierie fiabilité en électronique", EADS/Thales, Actes de conférence Lambda Mu 04, 12-19/10/2004, Bourges, France

[Holl98]

HOLLNAGEL, Erik, "Cognitive Reliability and Error Analysis Model- CREAM", book, Oxford- Elsevier, p.302; 1998, disponibilité **www.elsevier.com**

[INERIS03]

Rapport INERIS, "Formalisation du savoir et des outils dans le domaine des risques majeurs (DRA-35)- Rapport Ω-7 - Outils d'analyse des risques générés par une installation industrielle"; direction des risques accidentels, mai 2003, disponibilité **www.ineris.com**

[Lapr96]

LAPRIE ; Jean-Claude et all., "Guide de la sûreté de fonctionnement", Cepadues Edition, 1996, ISBN 2-85428-382-1, disponibilité **http://www.cepadues.com/livre_details.asp?l=339**

[Maur00]

MAURI, Giuseppe, "Integrating Safety Analysis techniques, Supporting Identification of Common Cause Failures", part 2.3 Common Cause Failure Analysis (pp.50-70), Thèse de doctorat, Université de York (HISE), septembre 2000, p.245, disponibilité **http://www.cs.york.ac.uk/ftpdir/reports/YCST-2001-02.pdf**

[Popo01]

POPOV, Peter, STRIGINI, Lorenzo, "The reliability of Diverse Systems: A contribution using Modelling of the Fault Creation Process",acts of International Conference on Dependable Systems and Networks (DSN'01), IEEE-Computer Society, p. 5, July 2001. Goteborg, Sweden.

[Purw01]

PURWANTORO, Yudi, BENETT, Stuart, "Hardware-Software Interaction in Dependability Modelling of Fault Tolerant System", acts of 17th Annual UK Performance Engineering Workshop (UKPEW'01), University of Leeds, UK, 18-19 juillet 2001.

[Rauz99]

RAUZY, Antoine, DUTUIT, Yves, SIGNORET, Jean-Pierre, "Evaluation of systems reliability by means of binary decision diagram", Proceedings of the Probabilistic Safety Assessment Conference, PSA'99, volume 1, pages 521-528, American Nuclear Society, 1999. ISBN 0-89448-640-3

[Rasm97]

RASMUSSEN, Jens, "Risk management in a dynamic society : a modelling problem", Safety Science, Elsevier Science Ltd., vol. 27, nos. 2/3, pp. 183-213, 1997 **www.sciencedirect.com**

[Swai83]

SWAIN, A. D., Guttmann H.E., "Handbook of Human Reliability Analysis with Emphasis on Nuclear Power Plant Application", USNRC Report, NUREG/CR-1278. Washington, DC: U.S. Nuclear Regulatory Commission, 1983.

[Triv94]

FRICKS, Ricardo M., TRIVEDI; Kishor, "Modelling Failure Dependencies in Reliability using GSPN", actes de l'European Simulation Multi-conference (ESM '97), Istanbul, Juin 1997, disponible **http://www.ee.duke.edu/~kst/spn_papers/fail.pdf**

Chapitre 3

[Bräu96]

BRÄUNER, Torben, "Introduction to Linear Logic", Report BRICS-LS-96-6, Department of Computer Science, University of Aarhus, Norvège, 1996, disponibilité **http://www.brics.dk/LS/96/6/BRICS-LS-96-6.pdf**

[Cass99]

CASSANDRAS, Christos G, LAFORTUNE; Stéphane; "Introduction to Discrete Event Systems", book, Kluwer academics Publisher, 848pp, September 1999, ISBN-0-7923-8609-4, disponibilité **http://www.eecs.umich.edu/~stephane/Book/**

[Deha00]

DEHAIS, Frédéric, PASQUIER, Philippe, "Approche générique du conflit", actes de conférence Ergo- IHM 2000, Biarritz, France, 3-6 octobre, 2000, disponibilité **http://damas.ift.ulaval.ca/~pasquier/Publications/Pasquier-dehais-2000.pdf**

[Demm02]

DEMMOU; Hamid, KHALFAOUI, Sarhane, GUILHEM, Edwige, VALETTE, Robert, "Critical scenarios derivation methodology for mechatronic systems", Reliability Engineering & System Safety, Vol.84, N°1, pp.33-44, Avril 2004, disponibilité: **http://www2.laas.fr/laas/1-4266-Publications.php** (Rapport LASS 03046)

[Gira87]

GIRARD, Jean-Yves, "Linear Logic", Theoretical Computer Science, London Mathematical 50:1, pp. 1-102, 1987, disponibilité **http://iml.univ-mrs.fr/~girard/linear.pdf**

[Gira97]

GIRAULT, François, "Formalisation en logique linéaire du fonctionnement des réseaux de Pétri" Thèse de doctorat Université Paul Sabatier, Informatique industrielle, Toulouse, décembre 1997, disponibilité: **http://www2.laas.fr/laas/1-4266-Publications.php** (LAAS Rapport 97557)

[Hoc87]

HOC. Jean-Michel, "Psychologie cognitive de la planification", livre, Presses Universitaires de Grenoble, 1987, 200 p., ISBN 2 7061 0280 2, disponibilité **http://www.pug.fr/Titre.asp?Num=389**

[Jaul90]

JAULENT, Patrick, "Génie Logiciel : les méthodes", chapitre 3, Ed. Armand Colin, 1990, Paris, ISBN : 2200420021

[Mell85]

MELLOR, Stephane, WARD, Paul, "Structured Development for Real-Time Systems", book, Yourdon Press, 1985, ISBN 0917072 -51-0, pp.468.

[Pirb87]

HATLEY, Derek J.; PIRBHAI, Imtiaz A., "Strategies for Real-Time System Specification"; Dorset House, New York, 1987

[Rauz99]

RAUZY, Antoine, DUTUIT, Yves, SIGNORET, Jean-Pierre, "Evaluation of systems reliability by means of binary decision diagram", Proceedings of the Probabilistic Safety Assessment Conference, PSA'99, volume 1, pages 521-528, American Nuclear Society, 1999. ISBN 0-89448-640-3

[Reto02]

RETORE, Christian, "Logique linéaire et syntaxe des langues", Rapport d'Habilitation à Diriger les Recherches, Université de Nantes/INRIA, janvier 2002.

Chapitre 4

[Akme01]

AKHMEDJANOV, Farid M., "Reliability Databases State of Arts and Perspectives", Riso National Laboratory Internal Report (Danemark), August 2001 (38p), disponibilité **http://www.risoe.dk/rispubl/SYS/ris-r-1235.htm**

[Buko00]

BUKOWSKI, Julia V., "Modelling and Analyzing the Effects of Periodic Inspection on the Performance of Safety-Critical Systems", IEEE Transactions on Reliability, Vol.50, Issue 3, September 2001.

[Coco97]

COCOZZA-THIVENT, Christiane, "Processus stochastiques et fiabilité des systèmes", Collection Mathématiques et Applications, Springer, n°28, 430p., 1997, disponibilité: www.springerlink.com

[Dutu02]

DUFOUR, François, DUTUIT, Yves, "Dynamic Reliability, a new model", actes de conferences ESREL02/ Lambda Mu 13, Volume 1, p.350-358, 18-21 Mars 2002, Lyon

[FIDES04]

EADS/Thales, "FIDES et Ingénierie fiabilité en électronique", Actes de conférence, Lambda Mu 04, 12-19/10/2004. Bourges, France

[Gobl98]

GOBLE, William M., "Control Systems Safety Evaluation and Reliability", livre., publié par l'ISA 1998, 513 p.

[Hass95]

HASSET, T., DIETRICH D.L, "Time varying Failure Rates in the availability and Reliability Analysis of Repairable Systems", IEEE Transactions on Reliability Vol 44/1, p-155-160, 1995

[Héli01]

HELIAS, Arnaud, "Agrégation/Abstraction de modèle pour l'analyse et l'organisation de réseaux de flux", Thèse de doctorat Ecole Nationale Supérieure agronomique de Montpellier, Génie des procédés, partie III-2-2, p.45-56, décembre 2003.

[IEC61508]

IEC 61508:1998 "Functional safety of electrical/electronic/programmable electronic safety-related systems", International Electrotechnical Commission, 170p, juin 1998

[Kaff01]

KAFFEL, Hédi, "La maintenance distribuée: concept, évaluation et mise en oeuvre", Hédi Kaffel, Thèse de doctorat Université de Laval, Québec, p.13-15, octobre 2001, disponibilité **www.theses.ulaval.ca/2001/19524/19524.html**

[Mart04]

MARTIN, Philippe A, "Physique statistique des processus irréversibles", cours DEA de physique statistique, 2001-2004, 25p., Ecole Normale Supérieure Lyon.

[Maur00]

„Integrating Safety Analysis techniques, Supporting Identification of Common Cause Failures", Giuseppe Mauri, 2000, Thesis York University (HISE) p.60-62

[MIL-HDBK]

MIL-HDBK-217F, "Reliability prediction of electronic equipment", Department of Defense/ Reliability Analysis Center and Rome Laboratory at Griffiss AFB, NY, USA, December 1991

[Popo01]

POPOV, Peter, STRIGINI, Lorenzo, "The reliability of Diverse Systems: A contribution using Modelling of the Fault Creation Process",acts of International Conference on Dependable Systems and Networks (DSN'01), IEEE-Computer Society, p. 5, July 2001. Goteborg, Sweden.

[Rouv00]

Rouvroyes, Jan L., "Enhanced markov analysis as a method to assess safety in the process industry", Thèse de doctorat Technische Universiteit Eindhoven, 157p., mai 2001, disponibilité **http://alexandria.tue.nl/extra2/200111552.pdf**

[Sato03]

SATO, Yoshinobi, ZHANG, Tieling, LONG, Wie, "Availability of systems with self-diagnostic components, applying Markov models to IEC61508-6"; Reliability Engineering and System Safety 80, p.133-141, 2003

[Schn85]

SCHNEEWEISS, Winfred G., "Reliability Modeling", LiLoLe-Verlag GmbH, Hagen, Germany, 2001, disponibilité **http://www.amazon.de/exec/obidos/ASIN/393444704X/infoline-21/028-2865430-9571755**

[Scho02]

SCHOENIG, Raphael, "Définition d'une méthodologie de conception des systèmes mécatroniques sûres de fonctionnement", Thèse de doctorat, Institut National Polytechnique de Lorraine (CRAN), Nancy, 26 oct. 2004.

[Seri96]

SERICOLA, Bruno, "Availability analysis and stationary regime detection of Markov Models", IRISA (Institut de recherche en informatique et systèmes aléatoires) Rapport 2886, 17pp., 1996, disponibilité **ftp://ftp.irisa.fr/techreports/1996/PI-1013.ps.gz**

[Shar93]

SHARIT, Joseph, "Human reliability modelling" extracted from „New trends in System Reliability Evaluation", Elsevier p.369-410, Amsterdam, 1993

[SN29500]

"Siemens SN29500" Siemens internal Data Base for failure rate evaluation of electronically components, dernière actualisation mai 2005

[Snoo05]

SNOOK, Ian, MARSHALL; John, "Physics of Failure as an Integrated Part of Design for Reliability", acts of Conference ARTS'16 (Advances in Reliability Technology Symposium); Loughborough, U.K, 14-17 Mai 2005.

[Tebb04]

TEBBI, Ouahiba, GUERIN, Fabrice, DUMON, Bernard, "Estimation de la fiabilité par les essais accélérés"; LASQUO EA 3858 : Angers, Mécanique & Industries 6, p.155-167, 2005.

[Zhao03]

ZHAO Y.X., "On preventive maintenance policy of a critical reliability level for system subject to degradation", Reliability Engineering and System Safety 79 (2003) 301-308, disponibilité **http://www.paper.edu.cn/scholar/download.jsp?file=zhaoyongxiang-6**

Chapitre 5

[Schi02]

SCHILLER, Frank.; MATTES, Teo, "An efficient Method to Evaluate CRC-Polynomials for Safety-Critical Industrial Communication", .System Modelling Control-2005, Zakopane, Poland, 17.-19.10.2005, Byczkowska-Lipinska, L. et al, 2005, Warschau, 2005, Akademicka Oficyna Wydawnicza EXIT, 2005, p. 269-274

Chapitre 6

[Bere05]

BERENGUER, Christophe, DIEULLE, L., "Comparaison et évaluation des méthodes de calcul des facteurs d'importance fiabiliste pour les systèmes dynamiques", actes du congrés Qualita 2005, Bordeaux, avril 2005.

[Buko00]

BUKOWSKI, Julia V., "Modelling and Analyzing the Effects of Periodic Inspection on the Performance of Safety-Critical Systems", IEEE Transactions on Reliability, Vol.50, Issue 3, September 2001.

[Flor03]

FLORIN, Gérard, "La tolérance aux pannes dans les systèmes répartis", cours du CNAM, 1995, laboratoire Cédric, disponible **http://rangiroa.essi.fr/cours/systeme2/08-95-slides-tolerance-pannes.pdf**

[Hami03]

HAMIDI, Karim, MALASSE, Olaf, AUBRY, Jean-Francois, "SILKEY: A Tool for the automatic evaluation of safety and availability of multi-level Redundancies Architecture", 21st International Safety Society Conférence, 4-9 août 2003, Ottawa, p.1092-1102, ISBN 09721385-36

[Jaco00]

JACOBSON, Jean, HERARD, J., „STARSCES Annexe 7, Method for fault detection“, extrait (partie 7) du projet européen STARSCES, Standards for Safety Related Complex Electronic Systems, European Commission, DGXI, Contract SMT 4CT97-219; 32 pages, août 2000

[Jense03]

JENSEN, Rune M., „DES Controller Synthesis and Fault Tolerant Control, a survey of Recent Advances“, IT University of Copenhagen, TR-2003-40, ISSN 1600-61000, 2003

[Tan04]

TAN, Zhiuha, DUGAN Jean-Baptiste, “Minimal Cut Set/Sequence Generation for Dynamic Fault Trees”, Annual Reliability and Maintainability Symposium 2004 Proceedings, LA, January 2004

[Zwin95]

ZWINGELSTEIN, Gilles, “Diagnostic des défaillances - Théorie et pratique pour les systèmes industriels“, Traité des Nouvelles Technologies, série Diagnostic et Maintenance, Hermès, Paris, 1995, ISBN 2-86601-463-4, disponibilité **http://www.editions-hermes.fr/**

.

Glossaire

Par ordre alphabétique

A - B

Absence de demande situation où les paramètres environnementaux observables au niveau des capteurs d'une fonction de sécurité, vérifient (états réels) une condition ne nécessitant pas la réaction de la fonction de sécurité (situation non dangereuse au vu de son cahier des charges). L'événement associé à cette acquisition est noté AbsD dans notre travail.

Accident événement résultant d'une dérive de l'installation et se traduisant par des dommages, c'est à dire des blessures physiques ou atteintes à la santé affectant des personnes soit directement soit indirectement comme conséquence à un dégât causé aux biens ou à l'environnement **[CEI1050]**

[Situation] accidentogène (ou situation dangereuse (hazard)): situation potentiellement dangereuse car pouvant entraîner un accident, en absence de la mise en œuvre effective d'actions permettant son évitement.

Accessibilité propriété pour un graphe orienté, correspondant à l'existence d'au moins un chemin, c'est à dire une séquence de transition, permettant le passage d'un état initial fixé à un état but. L'état but est alors dit « accessible ».

Actionneurs redondants montage d'un nombre fini, strictement supérieur à 1, d'actionneurs, permettant de mener à bien l'action d'évitement de l'accident prévue dans le cahier des charges de la fonction dédiée sécurité.

Activation d'erreur matérielle sollicitation d'une ressource matérielle sous un mode pour lequel l'état d'erreur de la ressource induit sa défaillance

Activation d'erreur « environnementale » perturbation lors de sa transmission ou de son stockage d'un signal du fait de contraintes environnementales (EMI, température…).

Actualisation terme signifiant qu'une donnée a été renouvelée. Ce terme peut être utilisé dans plusieurs contextes. On parlera, par exemple, d'actualisation de registres d'entrée pour signifier la prise en compte de nouvelles données d'entrée de référence, ou d'actualisation logicielle pour signifier l'implantation d'un logiciel de remplacement (totalement modifié ou augmenté). On parle de période d'actualisation pour désigner la durée maximale (contrainte) entre deux observations de l'état de demande effectuée par le système d'étude.

(Graphe) acyclique propriété permettant d'affirmer qu'un graphe orienté ne contient pas de cycle, c'est à dire que quelque soit l'état initial et l'état marqué choisi pour ce graphe, le langage marqué généré est fini (remarque il peut dans certain cas être vide).

Agrégation d'état méthode de simplification d'un modèle consistant à fusionner plusieurs états, dont une propriété est jugée équivalente. Cette opération aboutit à la réduction de la complexité

du modèle de départ (nombre d'état plus faible) et doit permettre d'estimer l'évolution des nouveaux états résultant de cette opération.

ALARP "as low as reasonably possible", terme anglo-saxon, – Stratégie de réduction des risques basée sur une hiérarchisation de ceux-ci. Cette stratégie est itérative. Elle identifie le niveau de risque individuel le plus haut, et lui applique la stratégie de réduction la plus efficace (risque résiduel le plus bas possible). Une fois ces modifications effectuer, elle va déterminer les nouvelles situations à risque de l'installation, identifier le risque le plus élevé et s'attacher à sa réduction. On répétera l'opération jusqu'à atteindre un niveau de risque acceptable.

Allocation fonctionnelle méthode visant par le choix de spécification et de performances fonctionnelles à garantir une qualité de service à l'installation dans laquelle ces fonctions sont intégrées.

Alphabet ensemble de caractère (ou de lettres) permettant de générer les mots d'un ensemble de langages.

Amélioration architecturale (d'une fonction de sécurité) Ensemble de changements apportés dans l'architecture opérationnelle (c'est-à-dire matérielle, fonctionnelle ou les règles de maintenance) d'une fonction de sécurité permettant l'amélioration d'une ou plusieurs de ces performances fiabilistes.

Amont (respect. aval) se dit d'une entité sous- fonctionnelle dont le signal de sortie (resp. d'entrée) peut être utilisé par (resp. provient de) l'entité sous- fonctionnelle considérée.

Approche dysfonctionelle étude des conséquences de défaillances locales sur les lois fonctionnelles d'un système.

Approche zonale Approche développée en aéronautique basée sur la décomposition de l'espace où sont implantés les ressources matérielles du système d'étude en zones d'influence pour lesquels les stress environnementaux peuvent être jugés équivalents. Cette décomposition se base sur des zones d'influences et des profils de dissipation spatiales suivant les sources de chaleur, de pression, de vibration, de concentration chimique et de radiation et sur la superposition de ces différentes influences. **[Pop96]**

Architecture distribuée Organisation décentralisée des ressources matérielles et logicielles d'une architecture, induisant la répartition des processus de traitement d'information et de décision entre des ressources matérielles distantes.

Architecture matérielle décomposition de l'architecture suivant les ressources matérielle permettant de détailler les connections physiques existants entre celles-ci, ainsi que leurs composants (électronique, électriques, mécaniques…).

Architecture fonctionnelle décomposition sous- fonctionnelle d'une architecture.

Architecture logicielle spécification abstraite consistant à décrire un ensemble de fonctions en définissant leurs opérations, leurs sources d'informations, leurs interfaces et leurs interactions **[Hayes94]**

Automate d'état fini définition usuelle (alphabet, ensemble fini d'état, fonction de transition, état initial, ensemble marqué) cf. chapitre 4 (notions de base) ou **[CasLaf97]**

Automate hybride déterministe Un automate hybride est (**Hélias01**) un 5-uplet {S, X, u ,I ,E,F}avec :

S : un ensemble fini de sommets,

X : un espace d'état continu de dimension finie,

u un vecteur d'états évoluant sur l'ensemble S × X,

A est l'ensemble des transitions (auxquelles est associées une condition de passage portant sur u, appelée Gard, et une opération de transformation des valeurs de u, appelée saut

I une fonction qui à tout sommet associe l'ensemble des variables continues invariantes

E un ensemble d'événement lié aux transitions

F un ensemble d'application qui associe à chaque sommet de l'automate une règle d'évolution des composants continus de u

Un automate hybride est dit déterministe si l'évolution du vecteur d'état u l'est.

Dans le chapitre 4, nous nous sommes intéressés á une classe d'automate hybride déterministe particulière pour laquelle aucune condition de Gard n'est requise, les événements de E sont liés à des temps de séjour sur les sommets de l'automate et où les applications de sauts portent sur le vecteur d'état P.

Barrières organisationnelles mesures organisationnelles (évacuation, ..) visant à réduire l'exposition des biens et des personnes aux conséquences d'un accident.

Entité sous fonctionnelle procédé de traitement de signal utilisant un ensemble fini de signaux d'entrées et devant délivrer un signal de sortie unique, il est basé sur l'application d'une règle d'utilisation de ressources matérielles sur l'ensemble des signaux d'entrées. Cette règle est choisie sur un ensemble fini de règles d'utilisation possibles des ressources matérielles, en regard de l'état des signaux d'entrée à traiter

Mettre un renvoi – définition propre

Boucle de sécurité architecture opérationnelle (matérielle + logicielle) visant à transformer un ensemble d'information issu de l'observation de l'état de demande au niveau des interfaces d'acquisition (capteurs) en une action adéquate (activation des moyens de réaction ou absence d'action) suivant le cahier des charges de la fonction de sécurité.

Bus de communication association d'une couche physique de transmission de données, d'un protocole permettant de contrôler les règles de communication entre éléments connectés (déterministe), et de tests permettant de contrôler l'intégrité de l'information transmise (ajout de signature de conformité, de condition en terme de contrainte de temps…). Exemples très utilisé dans l'industrie et les systèmes embarqués: bus CAN, ProfiBus, ProfiSafe et AsI/ AsI-Safe. Couche matérielle + couche logicielle permettant de transmettre avec des performances définies (fiabilité, temps…) une information d'une application vers une autre application (info).

C

Capteurs (ou actionneurs) « intelligents » (smart sensor/actuator) se dit d'un capteur (ou actionneur) possédant intrinsèquement des capacités

- de test permettant la détection de fautes internes
- de décision, permettant son adaptation fonctionnelle
- et de communication, permettant la transmission ou la réception de données de diagnostic avec d'autres modules distants.

Chemin séquence de transition, équivalent à la séquence des événements correspondant à chacune de ces transitions (étiquettes) ….

(partie) Co- accessible dans un graphe orienté marqué, ensemble des états pour lesquels il existe un chemin permettant de se rendre à un des états de l'ensemble marqué.

Concaténation opérateur de manipulation de mots. Si $u = u_1 \dots u_n$ et $v= u = v_1 \dots v_p$ (avec $u_1,\dots,u_n$ et $v_1,\dots,v_p$ lettre d'un alphabet), alors la concaténation de u et v est le mot $u.v = u_1 \dots u_n v_1 \dots v_p$. Cet opérateur est non commutatif et possède comme élément neutre le caractère ε.

Conflit présence de propositions explicite et directement contradictoire **[Dehais00]**

Combinaisons d'erreurs matérielles minimales combinaison d'erreurs matérielles révélée sur une période de référence T, se traduisant par une défaillance du système dans un mode de défaillance fixé (la défaillance sous demande de la fonction de sécurité étant considérée dans la partie 6).

Consommation d'une information utilisation et destruction d'une information par une sous- entité fonctionnelle, visant à la délivrance d'un signal de sortie.

Contrainte environnementale (stress) facteurs pouvant dégrader (stress environnementaux) ou restaurer (réparation) l'état d'une ressource matérielle, ou contribuer à l'occurrence d'une faute environnementale.

Contraintes environnementales partagées exposition de plusieurs parties du systèmes (ressources) à une contrainte environnementale. Cette contrainte est alors « partagée ».

(Evénements) contributeurs événement influençant les mécanismes de propagation d'un accident (et donc l'impact associé à celui-ci), mais non suffisants à eux seuls à son occurrence.

COTS, de l'anglais « component on the shell », ressource matérielle ou logicielle existante et pouvant être utilisée dans une architecture, fournie par un tierce. Ces composants imposent des justifications en terme de fonctionnement, de qualité et de tests.

Coupes minimales plus petite combinaison d'entités entraînant l'échec de la mission du système (elle ne contient aucune autre coupe).

Coupure (coupe-circuit = de-energizing process) procédure visant à la mise en position sûre du système par l'intermédiaire de relais de coupure sur le circuit de contrôle des actionneurs du système.

Coût d'implantation coût totale de mise en œuvre d'une modification (développement, test et matériel).

Coût de maintenance coût induits par les opérations de maintenance (coût humain + coût d'inspection + coût de remplacement) portant sur un ensemble fini non vide de ressources matérielles.

Criticité produit de la fréquence estimée d'un accident et de son impact.

D

Décision Capacité de faire des choix en fonction d'informations disponibles

Déclenchement intempestif d'une fonction de sécurité (spurious trip of a safety- related function) activation des moyens de réaction d'une fonction dédiée sécurité en l'absence de demande. Cette défaillance peut se traduire par une perte d'exploitation, une baisse de la qualité des missions de l'installation et/ou une mise en danger de l'installation et de son environnement.

Défaillance cessation de l'aptitude d'une ressource à accomplir une fonction requise (CEI61508-4)

Défaillance systématique défaillance liée de manière certaine à une cause, qui ne peut être éliminée que par modification de la conception, du procédé de fabrication, du mode d'emploi, de la documentation ou d'autres facteurs appropriés. (IEC61508)

Défaillance de cause commune (CCF) défaillance du système d'étude due à plusieurs événements de défaillances simultanées, résultat d'erreurs dont l'apparition simultanée peut être attribuée à une cause unique (partagée).

Délivrance d'une information émission par une sous- entité fonctionnelle d'un signal de sortie.

Détectabilité possibilité de détection d'un ensemble fini non vide de combinaisons d'erreurs (erreurs matérielles, informations erronées..) par un test, quand celles-ci existent.

Disponibilité capacité d'un système ou d'une entité de fournir un service correct quand celui-ci est demandé (notion instantanée).

Diversité modification d'une architecture permettant d'affirmer que ses entités constitutives sont moins exposées aux causes communes de défaillance. On parle, en général, de voies de redondances diversifiées, c'est-à-dire pour lesquelles, on veut éviter au maximum les contraintes de partage de ressource et l'exposition à des contraintes environnementales pouvant générer la défaillance simultanée des deux voies.

Défectueux (composant matériel) état d'un composant matériel le rendant inapte à remplir correctement l'ensemble de ses spécifications fonctionnelles.

Données éléments bruts, de nature mesurable, n'ayant pas été interprété

Durée de test nombre moyen de cycle (de largeur T) permettant la réalisation complète d'un test (balayage de son spectre de détection).

E

EMI de l'anglais, interférences électro- magnétique, type de contraintes environnementales pouvant affecter l'état d'un signal transmis ou stocké. On parle en général d'EMC (compatibilité électro- magnétique) quand on considère un support de transmission ou de stockage d'information est robuste vis-à-vis d'un niveau d'interférence magnétique estimée au regard des interférences auxquels il pourrait être soumis du fait de son environnement architectural (matérielle) ou de sources externes.

Environnement ensemble extérieur au système d'étude, incluant l'installation industrielle et les contraintes qu'elle peut faire porter sur l'évolution temporelles des contraintes environnementales.

Erreur Ecart ou discordance entre une valeur ou une condition calculée, observée ou mesurée, et la valeur vraie, prescrite ou théoriquement correcte.

Erreur de conception Ecart le cahier des charges fonctionnel ou sous- fonctionnel et le fonctionnement de l'architecture proposée en l'absence de défaillances matérielles.

Erreur humaine Ecart entre les actions menées et les actions prescrites (par exemple dans les tâches de maintenance).

Information erronée. Existence d'un écart entre l'état d'une information et l'état attendue de celle-ci sous des hypothèses de bon fonctionnement du système, en regard de l'état de demande (présence/absence) observable ayant induit sa délivrance.

Erreur matérielle Ecart entre les règles de fonctionnement d'une ressource matérielle et celle attendu, dans le

cas où aucun de ses composant n'est défectueux. On peut classer les erreurs matérielles suivant les modes de défaillances par lesquelles elles se manifestent.

Erreur environnementale Conséquence d'une faute environnementale (situation de stress) induisant lors de l'échange ou le stockage la génération d'une erreur sur un signal et la perturbation de l'information lui étant rattachée.

Erreur de type « commande » Erreur dont l'existence disparaît avec sa cause.

Erreur matérielle d'ordre 1 Erreur matérielle suffisante à l'apparition d'un mode de défaillance du système d'étude.

Essais aggravés Techniques visant à tester la robustesse d'un composant par exposition à un stress amplifié, permettant de détecter ses faiblesses dans un but d'amélioration en production et de d'estimer un profil de défaillance dans le cas d'une utilisation classique, en utilisant des règles de transposition entre le profil de contraintes utilisés lors du test et ses spécifications d'utilisation.

Etat de demande présence ou (exclusif) absence de demande au moment de l'acquisition par le système de paramètres environnementaux (T,P…) permettant une prise de décision quand à la réaction de la fonction de sécurité.

Etat d'erreur présence ou absence d'une erreur (matérielle, environnementale…) sur un cycle T donné.

Etat « puit » état d'un automate d'état ne possédant aucun arc sortant.

Etude de sensibilité étude de l'influence des différents stress environnementaux sur le taux de panne d'un composant matériel.

Etude préliminaire de danger (PHA : preliminary hazard analysis) Approche visant à identifier les situations potentielles d'accidents et à estimer leur probabilité d'occurrence et leurs conséquences en terme d'impact pour une installation existante

Evénement initiateur événements permettant de figurer la réception par une entité sous-fonctionnelle d'un ou plusieurs signaux d'entrées.

Evénement « certain » événement dont la probabilité est égale à 1.

(Evénements) mitigatifs événements dont l'occurrence à tendance à réduire la vitesse du processus de destruction des biens ou des personnes, voir à l'arrêter, et réduit de ce fait l'impact d'un accident.

Evénement « échec de test » événement permettant de figurer, sous l'hypothèse d'une réalisation normale d'un test en ligne la non- détection par celui-ci, sur un cycle T du système d'étude, d'une combinaisons d'erreurs appartenant pourtant á son spectre de détection.

Exemplaire abstraction permettant d'attribuer á différents objets une propriété équivalente

F

Faute toute cause adjugée ou hypothétique (événement, circonstance) de la présence d'une erreur.

Faute environnementale passage à un niveau de stress environnemental (interférences électromagnétiques, température) induisant l'altération d'un signal. ….

Fiabilité Aptitude d'une entité à accomplir une fonction requise, dans des conditions données, pendant un intervalle de temps donné.

Fiabilité dynamique discipline permettant de prendre en compte l'interaction entre l'installation, en y incluant le système d'étude, et leur environnement et de simuler l'évolution de celui-ci par des modèles (automates hybrides, réseau de Pétri) dont les conditions de transition sont simulées par tirs probabilistes, suivant des profils de réaction de l'installation.

Flux élémentaire d'information transfert d'une information de sortie d'une sous- entité fonctionnelle à l'entrée de la sous- entité fonctionnelle suivante.

Flux d'information toute séquence de sous- entité fonctionnelles pouvant être reliées par des flux d'information élémentaire (chemin).

(Approche) FMDS (RAMS en anglais) acronyme correspondant au sigle fiabilité, disponibilité, maintenabilité, sécurité. Approche visant à mettre en évidence les liens entre ces attributs et d'intégrer une approche d'allocation d'objectifs en terme de performances fonctionnelles fiabilistes, basées sur l'estimation des conséquences des modes d'interactions fonctionnels et une évaluation de ceux-ci en fonctions des choix architecturaux (matériels, fonctionnels) et les politiques de maintenance préconisées.

FTA arbre de défaillance (fault tree analysis)

FME(C)A(D) failure mode and effect analysis. Analyse de conséquences par exploration des défaillances locales du système, détermination de leur conséquence directe (en absence d'autres défaillance). Elle peut être complété par une étude exploratoire des moyens de détections attachées aux erreurs en étant la cause (ajout de la lettre D) et d'une analyse de criticité de ces conséquences (ajout de la lettre D).

Fonction dédiée sécurité SDSA fonction dédiée à la sécurité observant un état de demande et possédant une unique action de réaction (single demande/single action) pouvant être, le cas échéant, assurée par un montage d'actionneurs redondants.

Fonction EP dédiée sécurité fonction permettant la détection d'une situation accidentogène et visant à éviter l'occurrence de l'accident par l'activation dans un délai suffisant d'un ensemble de moyens de réaction

Fonction temps réelle fonction devant satisfaire des contraintes de temps de réponse explicites et bornées, le non-respect de ces bornes entraînant l'apparition de dégradations de performances et de mauvais fonctionnement **[Pal98]**.

Fréquence d'un accident en absence de la fonction de sécurité (f) fréquence estimée de l'accident avant l'ajout d'une fonction de sécurité.

G...K

(échelle de) Gravité mesure de l'impact d'un accident, en fonction des dégâts occasionnés de manière direct ou indirect, suivant une échelle de référence, appelée échelle de gravité.

Hiérarchisation ordonnancement suivant une relation d'ordre partielle ou complète.

HAZOP Analyse de risque et d'opérabilité. Analyse exploratoire basée sur l'identification d'événements potentiellement dangereux, la recherche de toutes les causes de ces événements, l'analyse des conséquences liées à ces dérives. L'étape d'identification est basé sur une approche de type « mots-clefs », comme le montre des travaux existants sur l'utilisation d'une telle méthode sur un process chimique et aéronautique (Université de York).

(Processus Markovien) homogène processus de Markov dont les taux de transitions sont supposés comme constants au cours du temps.

Horloge de contrôle (Watch Dog) dispositif permettant le contrôle de la bonne réception d'une information sous une contrainte de temps, dite d'acceptation, ou de la bonne réalisation d'un ensemble de fonction. Il permet en cas de non- validation de cette condition, de générer un signal de diagnostic pouvant être utilisé par des fonctions de recouvrement fonctionnelles ou d'affectation par défaut de variables.

ICDE « Event: Impairment of two or more entities (with respect to performing a specific function) that exists over a relevant time interval and is the direct result of a shared cause" **[QSA-30-08]** – Défaillances d'au moins deux entités lors d'un même intervalle de temps, résultant d'une même cause (ESA)

Indépendance (terminologie probabiliste) deux événements sont dits « indépendants » si la probabilité de la combinaison de ces deux événements est différentes de celles du produit de ces événements pris séparément.

Impact conséquences en terme de dégâts matériel, humain et/ou environnementaux directe, indirecte ou différée d'un accident, mesuré en fonction d'une échelle de gravité préalablement choisie comme référentiel.

Indicateurs de performances critères permettant de juger des performances comportementales d'un système (fiabiliste).

Information signal auquel a été adjoint une signification.

F- information signal devant refléter l'état de demande d'une fonction de sécurité et dont l'état peut être déterminé dans le cas d'un fonctionnement normatif (absence de défaillances) du système si l'on connaît l'état de demande lors de l'acquisition ayant permis sa synthèse.

D- information signal devant refléter la présence ou l'absence d'une combinaison d'erreurs détectable (matérielle, informationnelle et/ou non réalisation (cas d'une horloge de contrôle)).

Initiation d'erreur d'information délivrance par une entité sous- fonctionnelle d'une information erronée alors que l'ensemble de ses informations d'entrées ne le sont pas

Inhibition d'erreur d'information délivrance par une entité sous- fonctionnelle d'une information non erronée alors qu'au moins une de ses informations d'entrée est erronée.

Installation application industrielle ou de transport dans laquelle nous avons décidé d'implanter la fonction de sécurité dans un but de réduction du niveau de risque industriel attaché à un scénario d'accident, incluant tous les autres dispositifs de sécurité.

Intégrité propriété d'absence d'altération pour une entité

L

Label (ou étiquette) caractère désignant un événement de transition dans le graphe d'événement associé à un automate d'état fini (un caractère peut être associé à une fonction et faire figurer ces attributs dans une chaîne alpha- numérique, cette chaîne sera désignée sous le terme caractère).

Langages marqués langage constitué de l'ensemble des séquences d'événements (représentés par leur caractère) permettant le passage de l'état initial d'un automate d'état fini à au moins un de ses états marqué.

(Erreur) latente erreur non activée sur un cycle T du système d'étude.

Listes caractéristiques combinaisons d'erreurs dont l'existence est nécessaire et suffisante à l'apparition d'un mode de défaillance du système d'étude et dont l'activation est certaine au vue des choix architecturaux de celui-ci.

La liste caractéristique pour un déclenchement intempestif de la fonction dédiée sécurité

Ld liste caractéristique pour une défaillance sous demande de la fonction dédiée sécurité

Logique linéaire Logique, issue de la logique intuitioniste, basée sur les concepts de présence ET de validité d'une proposition, et se distinguant de ce fait de la logique classique, attaché à des conditions de vérité. Elle peut être vue comme un raffinement et une symétrisation de la logique intuitionniste. Une définition précise de sa sémantique de base est donnée par **[Girard87]**. Un énoncé s'y interprète comme l'ensemble de ses démonstrations, et une implication A => B est vue comme l'ensemble des fonctions qui transforment les démonstrations de A en démonstrations de B…..

Loi exponentielle loi de dégradation pour laquelle la fonction de survie d'un composant à un profil exponentiel par rapport au temps.

Loi de Weibull loi de dégradation, souvent utilisée pour les composants mécaniques, basée sur trois paramètres estimée et permettant de prendre en compte l'usure du composant et donc l'effet de l'accumulation de contrainte dans celui-ci au cours du temps (équation donnée au chapitre 4).

M

Maintenance Combinaison de toutes les actions techniques et administratives, y compris les opérations de surveillance, destiner à maintenir ou à remettre en état une entité (matérielle ou logicielle) dans un état lui permettant d'accomplir une fonction requise.

Maintenance préventive Maintenance effectué à des intervalles prédéterminés ou selon des critères prescrits et destiné à réduire la probabilité de défaillance ou la dégradation de fonctionnement d'une entité matérielle ou logicielle. On s'intéresse dans notre étude aux procédures de maintenance préventive sur ressources matérielles.

> Maintenance préventive planifiée (= entretien programmé) Maintenance effectué à des intervalles prédéterminés, selon une fonction ne dépendant pas d'opérations éventuelles de maintenance corrective
>
> Maintenance préventive adaptative (ou conditionnelle) Maintenance effectué à des intervalles prédéterminés sous condition de non occurrence de certaines actions de maintenance corrective préventive sur cet intervalle.
>
> Maintenance préventive parfaite (« as good as new ») processus de remplacement ou de réparation d'une ressource matérielle ou logicielle, sous l'hypothèse d'un état parfait (comportement conforme au cahier des charges) de la ressource remplacée à l'instant de sa première utilisation

Maintenance corrective Maintenance effectuée après une détection de pannes effectuée automatiquement (test en ligne, contrôle de réception, comparaison de signaux équivalents) et destiner à remettre l'entité dans un état lui permettant sa fonction requise

Maintenabilité Aptitude d'une entité, dans un contexte d'utilisation fixé, à être maintenue ou rétablie dans un état dans lequel elle peut remplir une fonction requise, sous des conditions de respect des procédures et moyens prescrits à cet effet

Maîtrise des risques démarche visant à d'estimer le niveau de risque de son installation, prendre les mesures qui s'imposent pour les détecter au plus vite, déclencher des actions d'évitement de l'accident ou de réduction de son impact, ayant pour but de garantir un niveau de risque tolérable à une installation

Graphes de Markov processus markovien basé sur une échelle de temps continu.

Chaîne de Markov processus markovien basé sur une échelle de temps discrète.

Mise en danger apparition d'une situation accidentogène vu comme la conséquence directe de l'occurrence d'un ou plusieurs événements générés par l'installation, le système d'étude ou une interaction de ces deux systèmes.

Mise en position « sûre » (abus de langage, traduction anglo-saxonne de « safe state ») activation forcée des moyens de réaction du système d'étude visant à un retour à une situation non-dangereuse de l'installation suite à la détection d'une défaillance de celui-ci.

Mode de défaillance ensemble des effets, dans un contexte de sollicitation fixée, par lequel une défaillance est observée **[Zingelstein85]**

Mode de défaillance latent mode de fonctionnement du système se traduisant par un service correct de celui-ci qui induirait suite à l'occurrence d'une défaillance locale supplémentaire une défaillance du système alors que cet événement n'est pas une condition suffisante d'apparition d'une défaillance du système.

Mode de fonctionnement dégradé du système d'étude mode de fonctionnement incomplet du système, qui en présence de demande ne permet pas d'éviter un accident mais en atténue les conséquences (se traduisant donc par une réduction de son impact)

Moyen de protection moyen technique ou organisationnel visant à réduire l'impact d'un accident, soit en réduisant l'exposition des biens menacés, soit en réduisant ou en ralentissant sa propagation

Moyen de réaction actionneurs d'une fonction de sécurité dont l'activation doit permettre d'éviter un accident en présence de demande

Mot séquence de caractères (labels)

N..O

Niveau de demande (= mode de sollicitation) notion introduite dans l'ICE61508, distinguant les prescriptions en terme de performances fiabilistes pour un niveau de SIL suivant la fréquence moyenne d'accident f. La norme introduit deux niveaux (faible mode de sollicitation/haut mode de sollicitation). Cette distinction est à approcher de l'égalité liant le risque résiduel et les indicateurs de performances, développée dans la partie 1.

Occurrence de demande passage d'un état d'absence de demande à un état présence de demande

Opération de réduction de langage opération visant à réduire la taille d'un langage (par l'exclusion de mots permettant d'éviter les séquences conflictuelles) ou de ses constituants (par suppression de caractères inutiles pour le but recherché)

Opération de croisement de langage opération sur les mots de deux langages, introduit dans le chapitre 3 permettant de prendre en compte l'utilisation par une entité décisionnel (bloc IP) d'informations soumise à des sources ou à des influences non indépendantes (partage de ressources éventuelles (de nature informationnelle ou matérielle))

P-Q

Pas d'itération d'une chaîne de Markov largeur du pas d'incrémentation temporelle pour une chaîne de Markov

Panne (en anglais fault) Etat d'un composant matériel (électrique, électronique ou mécanique) le rendant inapte à accomplir sa mission

Partage de ressource matérielle utilisation par deux entités sous- fonctionnelles distinctes d'une même ressource matérielle pendant un intervalle de référence T

Partage d'information utilisation par deux entités sous- fonctionnelles distinctes d'une même informationpendant un intervalle de référence T

Performance « fiabiliste » d'une fonction de sécurité critère permettant de juger de la fiabilité du système d'étude

Période d'acquisition (du système) temps séparant deux observations successives, par l'ensemble des interfaces capteurs, de l'état de demande par le système d'étude. Chaque observation se traduisant par une prise de décision du système (activation/ non activation des moyens de réaction). Cette période est notée T et correspond au temps de cycle du système d'étude.

Période de vie utile(d'un composant électronique) durée pendant laquelle un composant est supposé conservé un taux de pannes quasi-constant (cf. courbe en baignoire au chapitre 4)

Période d'inspection (TI) durée séparant deux actions de maintenance préventive successive sur une ressource matérielle fixée (ce critère peut être variable)

Période de test durée (en nombre de cycles T) séparant deux réalisation successives d'un tests en ligne

Perte économique induite (C_{ind}) coût estimé sur la durée de vie du système, causé par la dégradation de la qualité de service de l'installation et induit par les déclenchement intempestifs de la fonction de sécurité.

Perturbation d'information passage d'une information d'un état correct à un état erronée ou d'un état erronée à un état correct

PFD(t) ou PFD (t=kT) probabilité de défaillance sur demande, probabilité pour une fonction de sécurité de ne pas réagir (activation des moyens de réaction), suite à l'apparition d'une situation accidentogène qu'elle est sensée détectée au temps t =kT (avec k un entier et T la période d'acquisition). Sa valeur moyenne temporelle est notée PFD_{avg}.

PFS(t) ou PFS (t=kT) probabilité de déclenchement intempestif de la fonction de sécurité (réaction en l'absence de demande) au temps t =kT (avec k un entier et T la période d'acquisition). Sa valeur moyenne temporelle est notée PFS_{avg}.

Phase de jeunesse d'un composant phase précédent l'implantation durant laquelle un composant matériel est amélioré, elle vise à réduire son taux de panne lors de sa durée d'utilisation future, par une maîtrise des politiques de production et de test de celui-ci

Politique de maintenance description des relations entre les échelons de maintenance (niveau de responsabilité organisationnelle), les niveaux d'intervention (niveau de subdivision des tâches) et les niveaux de maintenance (corrective/préventive sur les sous- entités constitutives) des différentes ressources matérielles et logicielles maintenues

Présence de demande existence d'une situation accidentogène détectable par la fonction de sécurité étudiée (en accord avec le cahier des charges de sa fonction de base)

Prévention d'un accident Approche visant à la détection au plus tôt de situation accidentogène et à l'évitement d'un accident par l'ajout de moyens de réaction

Profil de mission profil estimé de l'évolution des contraintes environnementales en absence de demande, dérivé des tâches accomplies par l'installation. Pour certaines installations, ce profil est périodique ou pseudo- périodique.

Propagation d'erreurs d'information transmission d'une information erronée le long d'un flux d'information

Propagation de langage utilisation d'un langage généré par une entité sous- fonctionnel comme événement initiateur de l'automate d'état fini représentant l'entité fonctionnelle suivante (avale le long du flux d'information)

Qualité d'une action de maintenance confiance accordée dans la bonne réalisation d'une procédure de maintenance

Qualité de service l'ensemble de ses caractéristiques justifiant de l'aptitude d'un système **durant sa durée d'utilisation** à satisfaire à la fois les besoins exprimés (i.e. capacité d'évitement d'accident

pour notre système d'étude) et implicites (i.e. non perturbation des missions du système en absence de situation dangereuse pour notre système d'étude) **ISO84102**

R

Réaction action exigée par la fonction de sécurité et visant à éviter un accident suite à l'occurrence d'une demande

RBD (diagramme de fiabilité) Modèle d'une entité ou d'un système par une approche diagramme bloc de type « tout ou rien ». qui se base sur l'indépendance des modes de défaillances attribué à chaque bloc, et vise à évaluer la fiabilité de l'entité représentée par des opérations simples d'additions et de multiplications des probabilités de défaillances de chaque bloc (supposées constantes pour une approche RBD classique).

RdPs = GSPN extension des réseaux de Pétri classiques, réseau de Pétri dont les transitions sont conditionnés par des événements déterministes et aléatoires

Reconfiguration fonctionnelles mécanisme visant après la détection d'une faute à assurer un mode de fonctionnement correct, en s'appuyant sur un changement d'utilisation des ressources existantes

Réparation action visant à la remise en état d'une ressource matérielle ou logicielle (remplacement éventuel de tout ou partie de la ressource). On définit le MTTR (temps moyen de réparation), comme l'espérance mathématique de la somme de la durée d'inspection nécessaire au choix de cette action et de la durée de mise en œuvre effective

Redondance* doublement d'une voie de traitement d'information dans le but affirmé d'augmenter la fiabilité locale en s'appuyant sur les informations délivrées par ces deux voies (jugées équivalente) de manière comparative ou complémentaire (type de redondance : Redondance active Redondance passive …)

Ressource chose qui peut être consommé, utilisée et produite

> Ressource matérielle ensemble de composants électronique et programmable, pouvant remplir une fonctionnalité unique (et donc non reconfigurable) durant la durée de vie du système étudiée, dont la réparation est menée globalement et pour lequel ces procédures de réparation (préventive et ou corrective) implique une indisponibilité de l'ensemble des composants durant la période de réparation ….
>
> Ressources matérielles critiques ressource matérielle dont la défaillance induit de manière suffisante celle du système

Ressource informationnelle signal auquel est adjoint une signification et ayant pour but de permettre de manière directe ou indirecte la réalisation de la fonction de sécurité. On distingue les ressources informationnelle de classe 1 (F- et D- informations), des ressource informationnelle de classe 2 (instructions, code)

Risque mesure d'un danger associant une mesure de l'occurrence d'un événement indésirable et une mesure de ses effets ou conséquences, défini en général comme le produit de la probabilité d'occurrence de cette situation et de son impact

Risque résiduel somme des risques possible suite à l'occurrence de la situation accidentogène i ou au déclenchement intempestif des fonctions de sécurité basé sur sa détection

Risque tolérable (ou acceptable) niveau d'acceptation du risque (global pour un installation, individuel sur le risque résiduel d'une situation accidentogène précise)

Robustesse capacité de résistance vis-à-vis de contraintes environnementale d'une ressource matérielle ou d'une entité lui permettant de continuer à offrir un service correct ou suffisant au regard de son cahier des charges fonctionnel.

S

Scénario d'accident évolution d'une situation d'impact non négligeable au vu d'un contexte initial fixé et d'un nombre restreint, identifiable et daté d'événements perturbateurs.

Sécurité- confidentialité propriété de garanti de l'absence d'altération de modifications non autorisé d'information

Sécurité- innocuité (appelée par abus sécurité dans notre étude) propriété permettant de justifier l'absence de conséquences catastrophiques sur l'utilisateur ou son environnement

Séquence d'activation d'erreurs séquence d'événements d'activations d'erreurs (matérielles, environnementale) et/ou d'événements « échec de test »

Signal ensemble structuré de données

SIL (Safety Integrity Level) niveau de confiance (introduit par l'IEC61508) vis-à-vis d'une fonction EP dédiées sécurité basé sur l'application de normes de qualité, dans les phases de définition, de conception et de vérification, mais aussi de contrôle de bonne mise en œuvre des politiques de suivi et de maintenance d'une architecture dédiée sécurité la vérification des objectifs de performances fiabilistes d'une telle fonction par l'évaluation quantitative de son seul PFD_{avg}.

(défaillances) simultanées occurrence de deux événements de défaillance sur un même cycle T

(erreurs) simultanées existence de deux erreurs (matérielles ou environnementales) sur un même cycle T

Stratégie Manière d'organiser et de coordonner une série d'actions, un ensemble de conduites en fonction d'un résultat.

Superviseur (sens Lafortune) organe de contrôle permettant de tester un ensemble fini de ressources (par injection de séquences de test), de détecter certaines combinaisons d'erreurs matérielles (par comparaison de signatures) et d'agir directement sur le système en émettant un ordre de mise en position sûre sur le circuit de contrôle (coupe-circuit) des moyens de réaction du système d'étude

Sûreté de fonctionnement ensemble des propriétés permettant de placer une confiance justifiée dans le service délivré par un système au vu de ses utilisations

Synchronisation d'automate cf **[CasLaf97]**

Système (d'étude) ensemble des ressources matérielles (éventuellement maintenus correctivement et/ou préventivement) et logicielles visant à **assurer correctement** la fonction de sécurité et, ce quelque soit l'état de demande

Système non cohérent (au sens de fonction de structure fiabiliste) système pour lequel une défaillance survenant sur un composant du système défaillant annule les effets d'une défaillance antérieure et permet ainsi au système de fonctionner à nouveau

T

Taux de panne d'un composant matériel limite du quotient de la probabilité conditionnelle d'apparition d'une panne d'un composant matériel sur un intervalle de temps, sachant que ce composant est non défectueux au début de l'intervalle, sur la largeur de cette intervalle quant cette valeur tend vers 0 (par valeur positive).

Temps critique de réaction Tc temps maximal pour lequel, suite à l'occurrence d'une demande, la non- activation de contre-mesures ne débouche pas sur un accident.

Temps caractéristique, T durée séparant deux observations successives de l'état de demande par le système d'étude

Temps de cycle (≤ T) temps maximal admis entre l'acquisition d'un état de demande par une boucle de sécurité et la réception par le circuit de contrôle des ses actionneurs en résultant

Temps de panne latente: Intervalle de temps entre une défaillance et la détection de panne qui en résulte.

Temps inertiel Ti Temps maximum entre la réception par l'ensemble des moyens de réactions (actionneurs) d'un ordre de contrôle et l'action correspondante en absence de défaillance de celui-ci.

Temps de réaction temps maximal admis entre l'acquisition d'un état de demande par une boucle de sécurité et la réception par le circuit de contrôle des ses actionneurs en résultant

Temps de réponse Temps maximal nécessaire à la réaction de la fonction de sécurité pris comme la somme du temps de cycle T et du temps inertiel Ti.

Temps (moyen) de réparation (MTTR, TR) temps moyen (espérance mathématique) du temps nécessaire à l'inspection hors-ligne et à la réparation d'une ressource lors d'une opération de maintenance. On distingue la durée moyenne pour une opération de maintenances corrective, noté MTTR (mean time to repair) et la durée moyenne d'une opération de maintenance préventive, noté TR.

Temps d'utilisation du système T_{life} (aussi appelé temps de vie utile use- life time) temps de fonctionnement prévue d'une fonction de sécurité dans une installation, en général pris égal à la durée de vie estimée de l'installation avant démembrement. Il permet d'estimer la fiabilité du système d'étude sur un horizon d'utilisation fini.

Test « en ligne » test permettant la détection d'erreur ou de combinaisons d'erreurs matérielles. Il est basé sur la sollicitation d'un ensemble de ressources matérielles et l'observation et l'analyse des signaux de sorties en résultant.

Tolérance aux fautes Propriété correspondant à la capacité d'une entité d'accomplir une fonction malgré les pannes (fault) de certains de ses constituants.

U ...Z

UML (Unified Modeling Language, traduisez "langage de modélisation objet unifié") est une notation permettant de modéliser un problème de façon standard. Elle est née de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 90 : OMT, Booch et OOSE Unified.

Usure sur stock altération, détérioration d'une ressource matérielle lors de son stockage avant implantation (pour les ressources destinées au remplacement de ressources jugées défectueuses)

Utilisation d'une ressource matérielle manière de tirer parti d'une ressource matérielle pour mener à bien une opération de traitement de signal

Vitesse de propagation d'un accident fonction reflétant le taux d'accroissement de dégâts par unité de temps pour un scénario d'accident fixé.

Vecteurs de propagation d'un accident Ensemble de facteurs ou d'événements influençant la vitesse de propagation d'un accident.

Annexe A

Prise en compte de modes dégradés pour deux actions collaboratives

Cas d'une fonction de classe SDMA

On s'intéresse dans cette annexe à l'utilisation d'une fonction EP dédiée sécurité de classe SDMA (une demande – multi- actions). On prend l'exemple d'un arrêt d'urgence d'une enceinte sous pression de type four à verre. Si un seuil de température limite est dépassé, la fonction de sécurité exige:

1- la fermeture de l'alimentation du réacteur au moyen de deux valves séries (action A1 vraie)
2- l'ouverture du circuit de refroidissement du réacteur.

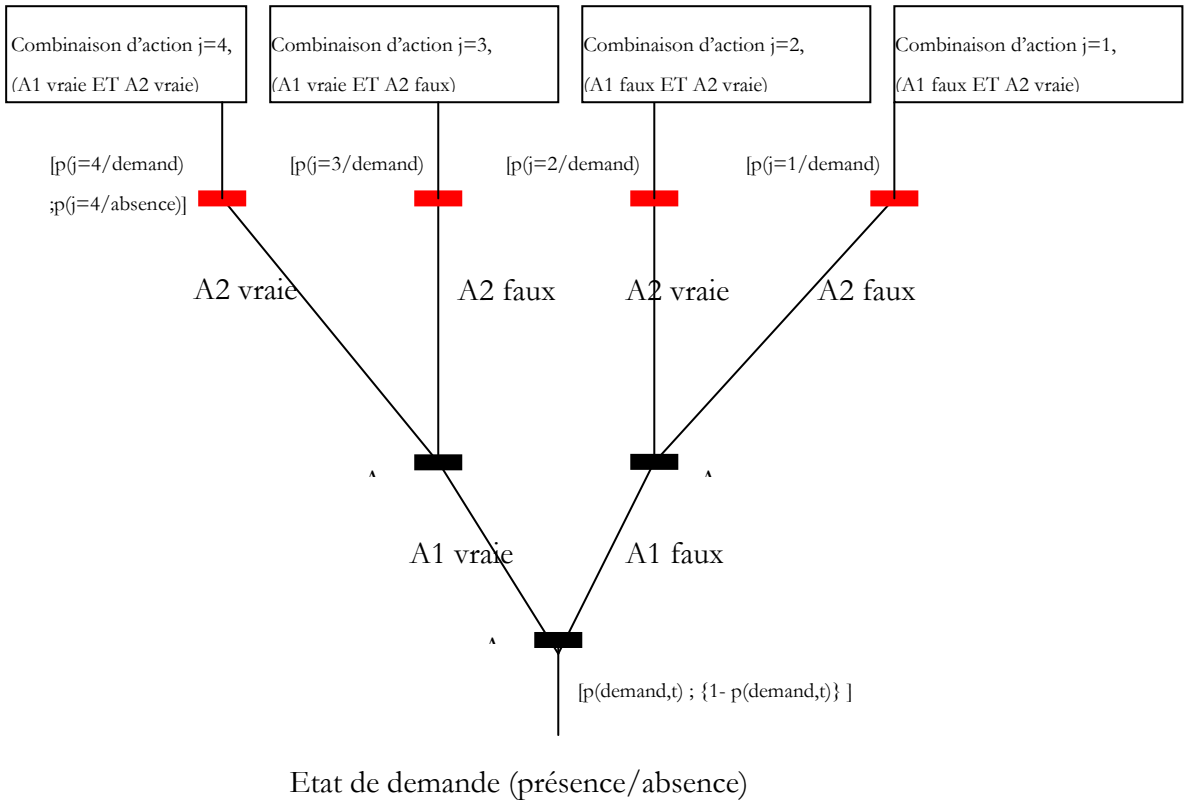


Figure 1 –Annexe A : Arbres binomial d'impact et de coûts

On peut symboliser les effets (impacts et coûts induits) des différents scenarii en présence et en absence de demande par la figure suivante (**figure 1/annexe A**). On rappelle (chapitre 1) qu'on associe à toute combinaison d'actions ou d'absence d'actions j:

- une métrique d'impact de l'accident induite par cette combinaison d'action au temps t+Tc sous l'hypothèse d'une demande au temps t (valant 0 en absence d'accident), noté $I_{dem}(j)$
- une métrique d'impact de l'accident induit par cette combinaison d'action au temps t+Tc sous l'hypothèse d'une absence de demande au temps t (valant 0 en absence d'accident), noté $I_{abs}(j)$ (correspondant aux scenarii de mise en danger du système par déclenchement intempestif des moyens de réaction)
- une métrique de perte économique induite par cette combinaison d'action au temps t+Tc sous l'hypothèse d'une absence de demande au temps t (valant 0 en absence d'accident), noté $C_{abs}(j)$ (correspondant aux pertes économiques de l'installation causés par le déclenchement intempestif des moyens de réaction)

j variera de 1 à 2^N avec N le nombre d'action de réaction (ici 2). Par convention, on notera j=1, la combinaison correspondant à une absence de réaction (aucune activation d'action de réaction) et j=2^N la combinaison correspondant à une activation de toutes les actions de réactions de la fonction de sécurité.

On notera, en outre, p(j/demand,t) la probabilité de la combinaison d'actions j à t+Tr si il y a une présence de demande à l'instant d'acquisition t et p(j/abs,t) son complémentaire

On peut facilement démontrer en prenant chaque combinaisons d'actions ets ous les mêmes hypothèses qu'en 3-1- pour le calcul du risque pour une action unique (fonction de sécurité de classe SDSA) que :

$$R_{res} = \frac{1}{T_{life}} \sum_{j=2}^{2^N} \sum_{k=0..N} \left(\int_{t=kT}^{t=(k+1)T} p(dem,t)dt. \right) p(j/dem,kT).I_{dem}(j) + \frac{1}{T_{life}} \sum_{j=1}^{2^N-1} \sum_{k=0..N} \left(\int_{t=kT}^{t=(k+1)T} (1-p(dem,t))dt. \right) p(j/abs,kT).I_{abs}(j)$$

et

$$C_{ind} = \sum_{j=1}^{2^N-1} \sum_{k=1..N} \{[1-p(demand,kT)].P(j/abs,t=kT).C(j,abs)\}$$

On en déduit donc d'après l'équation 12 (chapitre 1) que :

$$R_{res} = f \sum_{j=2}^{2^N} p_{avg}(j/dem).I_{dem}(j) + (\frac{T_{life}}{T} - f) \sum_{j=1}^{2^N-1} p_{avg}(j/abs).I_{abs}(j) \qquad \textbf{Eq. 13a}$$

et

$$C_{ind} = (\frac{T_{life}}{T} - f) \sum_{j=1}^{2^N-1} p_{avg}(j/abs).C_{abs}(j) \qquad \textbf{Eq. 16a}$$

avec

$$p_{avg}(j,abs) = \frac{T}{T_{life}} \sum_{k=0\cdots N} p(j/abs,kT) \quad \textbf{(Eq. 14a)}$$

$$p_{avg}(j,pres) = \frac{T}{T_{life}} \sum_{k=0\cdots N} p(j/pres,kT) \quad \textbf{(Eq. 15a)}$$

Les 2.(2^N-1) indicateurs données par les équations 14a et 14b sont les indicateurs de performances dans le cas d'une fonction dédiée sécurité de classe SDMA. On ajoute donc au service incorrect (absence de réaction ou déclenchement intempestif de toutes les actions), les autres modes dégradés qui peuvent correspondre à des scenarii d'accidents différents.

Il faut souligner que ces indicateurs correspondent à des moyennes (temporelles) de probabilités d'actions suivant l'état de demande observé. Ils diffèrent donc des produits des moyennes des probabilités de chaque action suivant ce même état de demande. Il s'agit en effet d'une même fonction dédiée sécurité, ce qui implique le partage d'un ensemble non vide de sous- fonctions d'acquisition et de décision entre les processus de traitement d'information décidant de ces actions.

Annexe B

Cas d'étude

Annexe B1

Cas d'étude : arrêt d'urgence de réacteur (1-)

B11: Automate d'état fini (cas simple)

B12 : Graphes de Markov

B13 : Désignation des ressources matérielles et taux de défaillance

B14 : Listes caractéristiques (cas simple)

Annexe B2

Cas d'étude : arrêt d'urgence de réacteur (2-)

B21: Automate d'états finis (cas 2- seulement ceux non décrits dans 1-)

B22 : Listes L1,L2 et L3 pour sortie du bloc 11

B11- Automate d'état fini (cas simple)

Les automates d'états finis sont donnés sous la forme de leur matrice de transitions, tout événement X contenu dans la ligne i et la colonne j est équivalent à l'existence entre les états i et j du graphe d'une transition ayant pour origine l'état i et comme destination l'état j, conditionné par l'événement X.

Les automates d'états finis sont donnés suivant la numérotation des blocs (entité sous-fonctionnelle) de la **figure 40**. Les blocs de même classe portant la même numérotation entre les **figures 40**, **43** et **46** possèderont les mêmes automates d'états finis représentant leur loi comportementale. La liste des ressources matérielles (désignation des xi) est donnée dans l'**annexe B13**.

Blocs SRC1

L1 ={d(xo,S)}

L2 = {d(xo,D)}

L3 = ∅

Bloc 1

	1	2	3	4	5	6	7	8	9	10(E1)	11(E2)	12(E3)
1		Init(true)				Init(false)						d(x1,I)
2			d(x1;0)									d(x7,I)
3				d(x7;0)								
4					d(x7,0)				d(x7,D)			
5										Fin(True)		
6							d(x1;0)					d(x1,I)
7								d(x7;0)				d(x7,I)
8					d(x3,S)				d(x3,0)			
9											Fin(False)	
10												
11												
12												

Bloc 2

	1	2	3	4	5	6	7	8 (E1)	9 (E2)	10 (E3)
1		Init(true)			Init(false)					
2			d(x5,0)							d(x5,I),
3				ε			bf(e1)			
4								Fin(True)		
5						d(x5,0)				d(x5,I),
6				bf(e2)			ε			
7									Fin(false)	

Bloc 3

	1	2	3	4	5	6 (E1)	7 (E2)
1		Init(True)	Init(False)				
2				ε			
3					ε		
4						Final(True)	bf(e4)
5						bf(e3)	Final(False)
6							
7							

Bloc 5

	2	3	4	5	6 (E1)	7 (E2)
1	Init(true)	Init(false)				
2			d(x10,0)	d(x10,D)		
3			d(x10,S)	d(x10,0)		
4					Final(True)	
5						Final(False)
6						
7						

Bloc 11

	1	2	3	4 (E1)	5 (E2)	6 (E3)
1		Init (1,1), Init(1;2) Init (2,1)	Init (2,2)			
2				d(x8,0).d(x11,0)	d(x8,D)	d(x8,I)
3				d(x8,S)	d(x8,0).d(x11,0)	d(x8,I)

Bloc 12

	1	2	3	4	5	6	7	8 (E1)	9 (E2)	10 (E3)
1		Init(true)			Init(false)					
2			d(x5,0).d(x7,0)							d(x5,I), d(x7,I)
3				ε			bf(e7)			
4								Fin(True)		
5						d(x5,0).d(x7,0)				d(x5,I), d(x7,I)
6				bf(e8)			ε			
7									Fin(false)	

Bloc 13

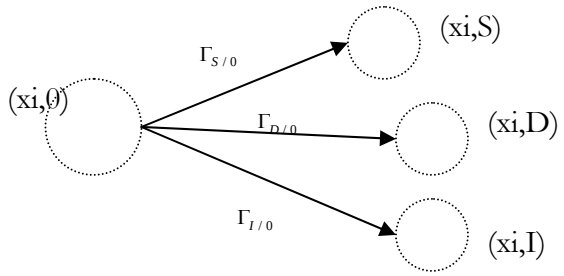
	2	3	4	5	6	7	8	9	10(E1)	11(E2)	12(E3)
1	Init(true)				Init(false)						d(x15,I)
2		d(x15;0)									d(x7,I)
3			d(x7;0)								
4				d(x7,0)				d(x7,D)			
5									Fin(True)		
6						d(x15; 0)					d(x15,I)
7							d(x7;0)				d(x7,I)
8				d(x3,S)				d(x3,0)			
9										Fin(False)	
10											
11											
12											

B12 Chaînes de Markov – Erreurs des ressources matérielles

A - Cas de ressources préventivement mais non correctivement maintenues

(Ressources x1, x2, x3, x10, x11, x12, x13, x15, x16, x17)

- **Hors des périodes d'inspections**



Si on note $P(t)=\begin{bmatrix} p(d(xi,0),t) \\ p(d(xi,S),t) \\ p(d(xi,D),t) \\ p(d(xi,I),t) \end{bmatrix}$ et $A=\begin{bmatrix} 1-\Gamma_{S/0}-\Gamma_{D/0}-\Gamma_{i/0} & 0 & 0 & 0 \\ \Gamma_{S/0} & 1 & 0 & 0 \\ \Gamma_{D/0} & 0 & 1 & 0 \\ \Gamma_{I/0} & 0 & 0 & 1 \end{bmatrix}$, on a : $P(t+T)=A.P(t)$

- **Pendant les périodes d'inspections**

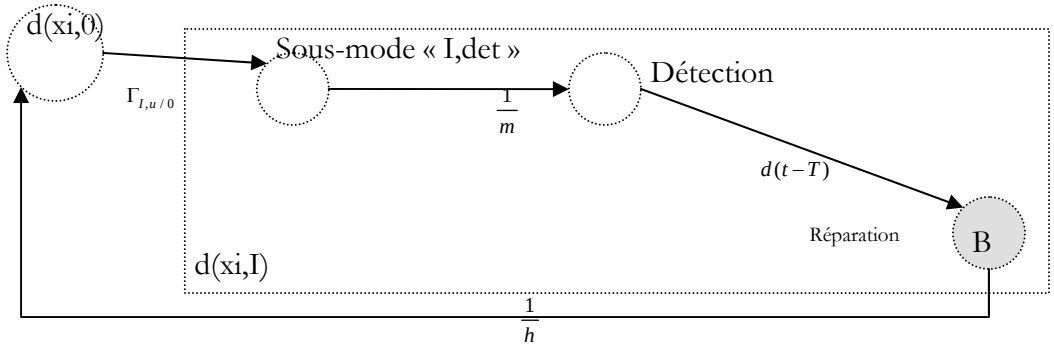
On a : $P(t)=\begin{bmatrix} p(d(xi,0),t) \\ p(d(xi,S),t) \\ p(d(xi,D),t) \\ p(d(xi,I),t) \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}=P_I$

On a retenu une maintenance préventive parfaite pour notre cas d'étude, on aura donc dans l'automate d'état fini de la figure 36 (chapitre 4, 1-4), $R_L=\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

B - Cas de ressources correctivement mais non préventivement maintenues

(Ressources x5 ;x7 ;x8)

Pour notre cas d'étude, les ressources x5, x7 et x8 ont seulement un mode de défaillance I. On peut donc simplifier la **figure 35** (cas général, donné au chapitre 4, 1-3) par la figure :



B13- Désignation des ressources matérielles et taux de défaillance

Rappel d'unité 1FIT = 10^{-9}/h

Comme les contraintes environnementales sont supposées constante, on calcule directement les taux de transitions des chaînes Markoviennes suivant ces valeurs de stress et la valeur d'un temps limite (en nombre de cycle N) et d'une loi d'usure consécutive à ce temps (facteur alpha). On supposera dans un premier temps non pris en compte le facteur d'usure, sauf pour l'actionneur, pour mener à bien la comparaison des méthodes exposées (hypothèses de départ identiques). On supposera en outre, les procédures de réparation comme parfaite, c'est-à-dire que l'état d'intégrité de la ressource sera assurée après chaque action de maintenance.

On prend dans toute l'étude T=500 ms et T_{life}=10 ans

Ressource matérielle	Désignation	{$\Gamma_{S/0}$, $\Gamma_{D/0}$, $\Gamma_{I/0}$} en FIT/T	{MTTR(h), T_{test} (en nombre de cycle T)} + condition de bonne transmission du diagnostic	{TI, TR}	Erreurs environnementales (désignation et probabilité)
Capteur de température 1 + connexion au module d'acquisition	x1	{0, 0, 528}	{0,0}	{1 an, 0,5h}	0
Capteur de température 2 + connexion au module d'acquisition	x2	{0, 0, 678}	{0,0}	{1 an, 0,5h}	0
Module d'entrée 1	x3	{125,80, 145}	{0,0}	{5 ans, 1h}	0
Module d'entrée 2	x4	{125,80, 145}	{0,0}	{5 ans, 1h}	0
Module de communication (IP) + Bus	x5	{0, 0, 240}	{0,5, 1}, si horloge et bus non défaillant	{0,0}	{bf(e1), 5^E-10} {bf(e2), 7^E-11} (codage 4 bits)
Horloge de contrôle (Watch Dog)	x6	{0,0,65}	{0,0}	{0,0}	0
Alimentation 1 (PS1) pour modules entrées-sorties	x7	{0,0,632}	{1h, 8400} visuel	{0,0}	0
Alimentation 2 (PS2) pour PLC	x8	{0,0,500}	{1h, 4200} visuel	{0,0}	0
Fonction comparateur dans PLC (pour seuil de température)	x10	{47, 67,0}	{0,0}	{0,0}	0
Vote 1oo2 (porte NAND)	x11	{12,22, 0}	{0,0}	{0,0}	0
Registres d'entrées (2 en tout) – adressage fixe sans test	x12 x13	{12,22, 0}	{0,0}	{0,0}	{bf(e3), 5^E-14} {bf(e4), 9^E-16} {bf(e5), 5^E-14} {bf(e6), 9^E-16} (codage 4 bits) indépendance stricte des chocs entre registres.
Module de sortie 1	x15	{85,50, 145}	{0,0}	{5 ans,1h}	0
Circuit de contrôle	x16	{85, 0,35}	{0,0}	{0,0}	0

Valve (moteur + alimentation)	x17	{920,450, 1400}	{0,0}	{2 ans, 24h}	0

B14- Listes caractéristiques La et Ld (cas simple)

Pour des raisons de commodité d'écriture, on a sous-entendu dans chaque mot la suite de terme d(xi,0) correspondant à une absence d'erreur de ressource. On prend le produit de ces termes en compte pour le calcul numérique effectué dans la partie 5.

La={d(x3,S) d(x10,0) d(x15,0),
bf(e2),
d(x3,S) bf(e1) d(x10,S),
d(x10,S),
d(x3,S) d(x10,D) d(x15,S),
bf(e2) d(x10,D) d(x15,S),
d(x3,S) bf(e1) d(x15,S),
d(x15,S),
d(x17,S),
d(x3,S) d(x15,D),
bf(e2) d(x15,D),
d(x3,S) bf(e1) d(x10,S) d(x15,D) d(x16,S)
d(x3,0) d(x10,S) d(x15,D) d(x16,S) ,
d(x3,S) d(x10,D) d(x16,S),
d(x3,0) bf(e2) d(x10,D) d(x16,S)
d(x3,S) bf(e1) d(x16,S),
d(x3,0) d(x16,S),
}

Ld={d(x3,D) d(x5,0).bf(e2),
bf(e1) d(x10,S),
d(x3,D) d(x10,S),
d(x10,D) d(x15,S),
d(x3,D) bf(e2) d(x10,D) d(x15,S),
bf(e1) d(x15,S),
d(x3,D) d(x15,S),
d(x15,D) d(x16,S),
d(x3,D) bf(e2) d(x15,D) d(x16,S)
bf(e1) d(x15,D) d(x16,S),
d(x3,D) d(x10,S) d(x15,D) d(x16,S),
d(x10,D) d(x16,S) d(x17,0),
d(x3,D) bf(e2) d(x10,D) d(x16,S),
bf(e1) d(x16,S),
d(x3,D) d(x8,0) d(x16,S),
d(x7,I),
d(x3,D) bf(e2) d(x7,I),
bf(e1) d(x10,S) d(x7,I),
d(x3,D) d(x10,S) d(x7,I),
d(x15,I),
d(x3,D) bf(e2) d(x15,I),
bf(e1) d(x10,S) d(x15,I),
d(x3,D) d(x10,S) d(x15,I),
d(x8,I),
d(x3,D) bf(e2) d(x8,I),
bf(e1) d(x8,I),
d(x3,D) d(x8,I),
d(x1,I),
d(x7,I)
d(x3,I)
d(x5,I),
d(x3,D) d(x5,I),
d(x17,I)
}

B21- Automates d'états finis (cas 1-2)

Les automates d'états finis décrivant les entités sous- fonctionnelles 1, 2, 12 ,13 ont déjà été donnés. Nous donnerons dans cette partie les automates d'états finis des entités sous-fonctionnelles 3,4,5,11. On remarque que l'architecture est symétrique, les automates des entités {8,9,10} seront donc obtenus par analogie avec ceux des automates des entités {3,4,5}.

Bloc 3

	1	2	3	4	5	6 (E1)	7 (E2)
1		Init(True)	Init(False)				
2				d(x12,0)	d(x12,D)		
3				d(x12,S)	d(x12,0)		
4						Final(True)	bf(e4)
5						bf(e3)	Final(False)
6							
7							

Bloc 4

	1	2	3 (E1)	4	5(E2)
1		Init(True)		Init False	
2			d(x6,0).d(x8,0)		
3			d(x6,I), d(x8,I)		
4					Final(True)
5					

Bloc 5

	2	3	4	5	6	7	8	9	10 (E1)	11 (E2)
1	Init(false,true)	Init(false,false)	Init(true,true)	Init(true, false)						
2					d(x10,0)	d(x10,D)				
3					d(x10,S)	d(x10,0)				
4							d(x10,0)	d(x10,D)		

5							d(x10,S)	d(x10,0)		
6									Final(True)	
7									d(x11,D)	d(x11,0)
8									d(x11,0)	d(x11,S)
9										Final(False)
10										

Bloc 11

	1	2	3	4 (E1)	5 (E2)	6 (E3)
1		Init (True,True), Init(True;False) Init (False,True)	Init (False,False)			
2				d(x8,0).d(x11,0)	d(x8,D)	d(x8,I)
3				d(x8,S)	d(x8,0).d(x11,0)	d(x8,I)

B22- Listes caractéristiques (cas 2-2)

L1={d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,0) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,0) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,0) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,0) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,S) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,S) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,S) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).d(x12,S) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,D) bf(e8) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,D) bf(e8) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,D) bf(e8) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,D) bf(e8) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,0) d(x8,0) d(x10,S) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,0) d(x8,0) d(x10,S) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,0) d(x8,0) d(x10,S) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0). d(x12,0) d(x8,0) d(x10,S) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,D) d(x8,0) d(x10,S) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,D) d(x8,0) d(x10,S) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,D) d(x8,0) d(x10,S) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,D) d(x8,0) d(x10,S) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,S) bf(e7) d(x8,0) d(x10,S) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,S) bf(e7) d(x8,0) d(x10,S) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,S) bf(e7) d(x8,0) d(x10,S) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0). d(x12,S) bf(e7) d(x8,0) d(x10,S) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,0) d(x8,0) d(x10,D) d(x7,0) d(x15,S),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,0) d(x8,0) d(x10,D) d(x7,0) d(x15,S),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,0) d(x8,0) d(x10,D) d(x7,0) d(x15,S),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,0) d(x8,0) d(x10,D) d(x7,0) d(x15,S),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,S) d(x8,0) d(x10,D) d(x7,0) d(x15,S),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,S) d(x8,0) d(x10,D) d(x7,0) d(x15,S),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,S) d(x8,0) d(x10,D) d(x7,0) d(x15,S),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).d(x12,S) d(x8,0) d(x10,D) d(x7,0) d(x15,S),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,D) bf(e8) d(x8,0) d(x10,D) d(x7,0) d(x15,S),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,D) bf(e8) d(x8,0) d(x10,D) d(x7,0) d(x15,S),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,D) bf(e8) d(x8,0) d(x10,D) d(x7,0) d(x15,S),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,D) bf(e8) d(x8,0) d(x10,D) d(x7,0) d(x15,S),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,0) d(x8,0) d(x10,0) d(x7,0) d(x15,S),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,0) d(x8,0) d(x10,0) d(x7,0) d(x15,S),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,0) d(x8,0) d(x10,0) d(x7,0) d(x15,S),

d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0). d(x12,0) d(x8,0) d(x10,0) d(x7,0) d(x15,S),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,D) d(x8,0) d(x10,0) d(x7,0) d(x15,S),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,D) d(x8,0) d(x10,0) d(x7,0) d(x15,S),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,D) d(x8,0) d(x10,0) d(x7,0) d(x15,S),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,D) d(x8,0) d(x10,0) d(x7,0) d(x15,S),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,S) bf(e7) d(x8,0) d(x10,0) d(x7,0) d(x15,S),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,S) bf(e7) d(x8,0) d(x10,0) d(x7,0) d(x15,S),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,S) bf(e7) d(x8,0) d(x10,0) d(x7,0) d(x15,S),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0). d(x12,S) bf(e7) d(x8,0) d(x10,0) d(x7,0) d(x15,S),
}

L2={
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,0) d(x8,0) d(x10,0) d(x7,0) d(x15,D),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,0) d(x8,0) d(x10,0) d(x7,0) d(x15,D),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,0) d(x8,0) d(x10,0) d(x7,0) d(x15,D),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,0) d(x8,0) d(x10,0) d(x7,0) d(x15,D),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,S) d(x8,0) d(x10,0) d(x7,0) d(x15,D),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,S) d(x8,0) d(x10,0) d(x7,0) d(x15,D),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,S) d(x8,0) d(x10,0) d(x7,0) d(x15,D),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).d(x12,S) d(x8,0) d(x10,0 d(x7,0) d(x15,D),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,D) bf(e8) d(x8,0) d(x10,0) d(x7,0) d(x15,D),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,D) bf(e8) d(x8,0) d(x10,0) d(x7,0) d(x15,D),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,D) bf(e8) d(x8,0) d(x10,0) d(x7,0) d(x15,D),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,D) bf(e8) d(x8,0) d(x10,0) d(x7,0) d(x15,D),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,0) d(x8,0) d(x10,S) d(x7,0) d(x15,D),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,0) d(x8,0) d(x10,S) d(x7,0) d(x15,D),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,0) d(x8,0) d(x10,S) d(x7,0) d(x15,D),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0). d(x12,0) d(x8,0) d(x10,S) d(x7,0) d(x15,D),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,D) d(x8,0) d(x10,S) d(x7,0) d(x15,D),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,D) d(x8,0) d(x10,S) d(x7,0) d(x15,D),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,D) d(x8,0) d(x10,S) d(x7,0) d(x15,D),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,D) d(x8,0) d(x10,S) d(x7,0) d(x15,D),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,S) bf(e7) d(x8,0) d(x10,S) d(x7,0) d(x15,D),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,S) bf(e7) d(x8,0) d(x10,S) d(x7,0) d(x15,D),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,S) bf(e7) d(x8,0) d(x10,S) d(x7,0) d(x15,D),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0). d(x12,S) bf(e7) d(x8,0) d(x10,S) d(x7,0) d(x15,D),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,0) d(x8,0) d(x10,D) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,0) d(x8,0) d(x10,D) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,0) d(x8,0) d(x10,D) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,0) d(x8,0) d(x10,D) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,S) d(x8,0) d(x10,D) d(x7,0) d(x15,0),

d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,S) d(x8,0) d(x10,D) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,S) d(x8,0) d(x10,D) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).d(x12,S) d(x8,0) d(x10,D) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,D) bf(e8) d(x8,0) d(x10,D) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,D) bf(e8) d(x8,0) d(x10,D) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,D) bf(e8) d(x8,0) d(x10,D) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,D) bf(e8) d(x8,0) d(x10,D) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,0) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,0) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,0) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0). d(x12,0) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,D) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,D) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,D) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,D) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,S) bf(e7) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,S) bf(e7) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,S) bf(e7) d(x8,0) d(x10,0) d(x7,0) d(x15,0),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0). d(x12,S) bf(e7) d(x8,0) d(x10,0) d(x7,0) d(x15,0),

}

L3={
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,0) d(x8,0) d(x10,0) d(x7,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,0) d(x8,0) d(x10,0) d(x7,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,0) d(x8,0) d(x10,0) d(x7,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,0) d(x8,0) d(x10,0) d(x7,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,S) d(x8,0) d(x10,0) d(x7,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,S) d(x8,0) d(x10,0) d(x7,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,S) d(x8,0) d(x10,0) d(x7,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).d(x12,S) d(x8,0) d(x10,0 d(x7,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,D) bf(e8) d(x8,0) d(x10,0) d(x7,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,D) bf(e8) d(x8,0) d(x10,0) d(x7,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,D) bf(e8) d(x8,0) d(x10,0) d(x7,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,D) bf(e8) d(x8,0) d(x10,0) d(x7,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,0) d(x8,0) d(x10,S) d(x7,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,0) d(x8,0) d(x10,S) d(x7,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,0) d(x8,0) d(x10,S) d(x7,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0). d(x12,0) d(x8,0) d(x10,S) d(x7,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,D) d(x8,0) d(x10,S) d(x7,I),

d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,D) d(x8,0) d(x10,S) d(x7,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,D) d(x8,0) d(x10,S) d(x7,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,D) d(x8,0) d(x10,S) d(x7,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,S) bf(e7) d(x8,0) d(x10,S) d(x7,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,S) bf(e7) d(x8,0) d(x10,S) d(x7,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,S) bf(e7) d(x8,0) d(x10,S) d(x7,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0). d(x12,S) bf(e7) d(x8,0) d(x10,S) d(x7,I), *
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,0) d(x8,0) d(x10,0) d(x15,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,0) d(x8,0) d(x10,0) d(x15,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,0) d(x8,0) d(x10,0) d(x15,I),,
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,0) d(x8,0) d(x10,0) d(x15,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,S) d(x8,0) d(x10,0) d(x15,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,S) d(x8,0) d(x10,0) d(x15,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,S) d(x8,0) d(x10,0) d(x15,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).d(x12,S) d(x8,0) d(x10,0 d(x15,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,D) bf(e8) d(x8,0) d(x10,0) d(x15,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,D) bf(e8) d(x8,0) d(x10,0) d(x15,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,D) bf(e8) d(x8,0) d(x10,0) d(x15,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,D) bf(e8) d(x8,0) d(x10,0) d(x15,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,0) d(x8,0) d(x10,S) d(x15,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,0) d(x8,0) d(x10,S) d(x15,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,0) d(x8,0) d(x10,S) d(x15,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0). d(x12,0) d(x8,0) d(x10,S) d(x15,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,D) d(x8,0) d(x10,S) d(x15,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,D) d(x8,0) d(x10,S) d(x15,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,D) d(x8,0) d(x10,S) d(x15,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,D) d(x8,0) d(x10,S) d(x15,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,S) bf(e7) d(x8,0) d(x10,S) d(x15,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,S) bf(e7) d(x8,0) d(x10,S) d(x15,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,S) bf(e7) d(x8,0) d(x10,S) d(x15,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0). d(x12,S) bf(e7) d(x8,0) d(x10,S) d(x15,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,0) d(x8,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,0) d(x8,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,0) d(x8,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,0) d(x8,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,S) d(x8,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,S) d(x8,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,S) d(x8,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).d(x12,S) d(x8,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,D) bf(e8) d(x8,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,D) bf(e8) d(x8,I),

d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,D) bf(e8) d(x8,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,D) bf(e8) d(x8,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,0) d(x8,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,0) d(x8,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,0) d(x8,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0). d(x12,0) d(x8,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) d(x12,D) d(x8,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) d(x12,D) d(x8,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0).bf(e2) d(x12,D) d(x8,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0).bf(e2) d(x12,D) d(x8,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0) bf(e1) d(x12,S) bf(e7) d(x8,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,0) bf(e1) d(x12,S) bf(e7) d(x8,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,0) d(x12,S) bf(e7) d(x8,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,0). d(x12,S) bf(e7) d(x8,I),
d(x1,I)d(x2,I),
d(x1,0)d(x2,0)d(x7,I)
d(x1,0)d(x2,0)d(x7,0) d(x3,I)
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,S) d(x5,I),
d(xo,S) d(x1,0)d(x2,0)d(x7,0) d(x3,D) d(x5,I),
d(xo,D) d(x1,0)d(x2,0)d(x7,0) d(x3,0) d(x5,I)}