Contribution à l'évaluation quantitative de la sureté de fonctionnement des systèmes d'automatisation en phase de conception

Blaise Conrard

UFR Ecole Supérieure des Sciences et
Technologies de l'Ingénieur de Nancy
Ecole Doctorale IAE+M
DFD Automatique et Production automatisée

Thèse

Présentée pour l'obtention du titre de

Docteur de l'Université Henri Poincaré, Nancy 1

Spécialité Automatique

Par **Blaise CONRARD**

CONTRIBUTION A L'EVALUATION QUANTITATIVE DE LA SURETE DE FONCTIONNEMENT DES SYSTEMES D'AUTOMATISATION EN PHASE DE CONCEPTION

Soutenue publiquement le 24 septembre 1999 devant la commission d'examen

Membres du jury :

Président :	M. GABRIEL	Professeur, Université Henri Poincaré Nancy 1
Rapporteurs :	M. BAYART	Professeur, Université des Sciences et Technologies de Lille
	D. NOYES	Professeur, Ecole Nationale d'Ingénieurs de Tarbes
Examinateurs :	A. CHOVIN	Ingénieur, Société CROUZET Automatismes
	J. CICCOTELLI	Institut National de Recherche et de Sécurité, Directeur de Recherche associé CNRS
	M. ROBERT	Professeur, Université Henri Poincaré Nancy 1
	J.-M. THIRIET	Maître de Conférences, Université Henri Poincaré Nancy 1

Centre de Recherche en Automatique de Nancy – Equipe Instrumentation Intelligente
2, rue Jean Lamour – 54500 VANDOEUVRE

A Abel et Delphine

Remerciements

Ce mémoire est le résultat de travaux menés au Centre de Recherche an Automatique de Nancy (CRAN-CNRS UPRESA 7039) dans l'équipe "Instrumentation Intelligente" située dans les locaux de l'ESSTIN (Ecole Supérieure des Sciences et Technologies de l'Ingénieur de Nancy).

Je remercie vivement :

Monsieur **Michel ROBERT**, Professeur à l'Université Henri Poincaré Nancy 1 et Monsieur **Jean-Marc THIRIET**, Maître de Conférences à l'Université Henri Poincaré Nancy 1 pour la direction scientifique de ces travaux, et aussi pour leur constante disponibilité, les nombreux conseils prodigués, mais surtout pour l'amitié qu'ils m'ont toujours témoignée,

Monsieur **Claude HUMBERT**, Professeur à l'Université Henri Poincaré Nancy 1 et directeur de l'ESSTIN pour m'avoir accueilli dans son laboratoire,

Madame **Mireille BAYART**, Professeur à l'Université des Sciences et Technologies de Lille et Monsieur **Daniel NOYES**, Professeur à l'Ecole Nationale d'Ingénieurs de Tarbes pour avoir accepté la lourde tâche de rapporteurs,

Monsieur **Joseph CICCOTELLI** de l'Institut National de Recherche et de Sécurité, Directeur de Recherche associé CNRS, Monsieur **André CHOVIN**, Ingénieur de la Société CROUZET Automatismes et Monsieur **Marc GABRIEL**, Professeur à l'Université Henri Poincaré Nancy 1 pour avoir accepté de participer à mon jury.

Je tiens également à remercier les membres du groupe CIAME/Sûreté de fonctionnement et Instrumentation Intelligente pour les nombreuses discussions fructueuses sur des aspects connexes aux travaux présentés.

Enfin mes remerciements vont aussi à tous mes collègues de laboratoire qui m'ont témoigné leur amitié et m'ont apporté à tout moment leur contribution, en particulier :

Mademoiselle Marie-Christine SUHNER, et Messieurs Frédéric HERMANN, Damien LUTTENBACHER, Jean-Michel RIVIERE, Dominique LARCHER, Eric GNAEDINGER, Jean-Luc NOIZETTE, Jean-Marie ORY et Philippe THOMAS.

SOMMAIRE

Chapitre I : Le Système d'Automatisation à Intelligence Distribuée et la Sûreté de Fonctionnement

Chapitre II : Modélisation d'un Système d'Automatisation

Chapitre IV : Intégration de la Sûreté de Fonctionnement en phase de conception des Systèmes d'Automatisation

Chapitre V : Applications de la méthode de conception

Introduction

L'emploi de composants dits intelligents et de réseaux de terrain dans les systèmes d'automatisation a fortement influencé aussi bien leur structure que leur conception. Ces nouvelles technologies permettent d'évoluer vers des architectures réparties dans lesquelles les traitements sont désormais distribués sur différents composants. L'évolution des moyens disponibles a, de plus, modifié le besoin des utilisateurs qui attendent du système d'automatisation d'autres fonctions que celles relatives à la conduite, telles que par exemple le diagnostic, la surveillance, la reconfiguration, la tolérance aux fautes...

La conception des systèmes d'automatisation fait l'objet de nombreux travaux depuis plusieurs années et une des approches souvent employée consiste à considérer le système d'automatisation comme un ensemble de tâches à répartir sur une architecture matérielle. La minimisation de la charge de traitement des composants et des supports de communication est par exemple l'objectif visé lors de cette répartition. Les aspects relatifs à la sûreté de fonctionnement y sont abordés essentiellement au niveau du choix des tâches à répartir et en conséquence, ne participaient pas directement à l'activité de distribution.

Les travaux présentés ici visent à mieux intégrer la sûreté de fonctionnement afin de concevoir des système dits "sûrs", définis comme des systèmes qui réalisent ce pourquoi ils ont été conçus, sans incident mettant leur rentabilité en question et sans accident mettant la sécurité en jeu[1]. Ainsi, selon ce point de vue, la sûreté de fonctionnement est considérée ici comme un ensemble de critères mettant en balance le profit que l'on pourrait tirer du système face aux incidents susceptibles de survenir. Par l'estimation a priori de ces critères, ceux-ci peuvent alors participer à l'activité de répartition en vue de déterminer la meilleure architecture du système d'automatisation à concevoir.

Par architecture, il faut entendre une organisation des tâches sur une organisation de composants. Ainsi, ces travaux proposent de profiter de l'activité de répartition déterminant l'organisation des tâches pour participer aux choix de l'organisation des composants. C'est à ce double objectif que ces travaux tentent de proposer des solutions.

Ce mémoire est structuré en cinq chapitres :

Le premier chapitre rappelle les propriétés et les caractéristiques des systèmes d'automatisation et de leurs composants (instruments intelligents et réseaux de terrain). Le cycle de vie et plus particulièrement les étapes liées à la conception de tels systèmes y sont présentés. Enfin, les concepts de base de la sûreté de fonctionnement sont expliqués.

[1] A. LEROY, J.P. SIGNORET, Le risque technologique, Collection Que Sais-Je ?, 1992.

Le deuxième chapitre concerne les différents modèles utilisés pour représenter les systèmes d'automatisation à différentes phases de leur conception. Une première partie traite de l'architecture fonctionnelle employée pour établir une organisation de tâches ou traitements élémentaires. Une seconde partie concerne l'architecture matérielle destinée à accueillir ces traitements et pour laquelle la construction est soumise à un ensemble de contraintes technologiques ayant une influence directe sur la réalisation des systèmes à concevoir.

Le troisième chapitre examine les différentes méthodes dont on dispose pour l'étude et la quantification des différents attributs de la sûreté de fonctionnement. L'objectif est alors d'en choisir une permettant une évaluation globale et a priori du système. Ainsi à partir du comportement supposé connu des composants, il s'agit de déterminer celui du système selon deux aspects. Le premier cherche à étudier la capacité du système à accomplir sa mission malgré les défaillances. Le second aspect est relatif à l'estimation des probabilités d'occurrences d'événements dont leurs conséquences variées (arrêt intempestif, accident...) risquent de nuire à l'installation. Enfin, la méthode choisie doit être susceptible de s'intégrer à une méthode de conception afin d'en faciliter la mise en œuvre.

Le quatrième chapitre expose de manière globale la méthode de conception et d'optimisation des systèmes d'automatisation prenant en compte la sûreté de fonctionnement. On y définit les informations complétant la description fonctionnelle et matérielle et destinées à la quantification des critères relatifs à la sûreté de fonctionnement. Par l'association de ces critères à travers une métrique unique, une grandeur peut être établie et rendre compte de la robustesse de chaque solution envisagée vis-à-vis des moyens à mettre en jeu (nombre de composants, de redondances...). Enfin des algorithmes d'optimisation sont présentés et peuvent être employés pour déterminer la meilleure solution en terme de répartition des traitements élémentaires et d'architecture matérielle.

Le dernier chapitre met en œuvre la méthodologie proposée sur deux applications. La première est destinée à la régulation de la température et du niveau d'un fluide dans une cuve. A partir de la décomposition fonctionnelle du système d'automatisation correspondant, on recherche la meilleure solution à partir de différentes possibilités d'architectures matérielles, de redondances et de mécanismes de sécurité. Cette étude est ensuite approfondie par la modification de la pondération des critères et fait alors apparaître d'autres solutions correspondant à d'autres niveaux de sécurité. La seconde application concerne la partie commande d'un processus thermique où le schéma général de l'architecture matérielle est défini et pour lequel les connexions entre instruments et automates sont recherchées. Dû au nombre important de solutions possibles, un algorithme d'inspiration génétique est utilisé.

Chapitre I

Le Système d'Automatisation à Intelligence Distribuée et la Sûreté de Fonctionnement

Chapitre I
Le Système d'Automatisation à Intelligence Distribuée et la Sûreté de Fonctionnement

Ce premier chapitre expose les principales notions relatives à la conception des systèmes d'automatisation à intelligence distribuée et relatives à la sûreté de fonctionnement. Les systèmes d'automatisation sont d'abord étudiés en terme de fonctions et en terme de composants ; puis les activités liées à leur conception sont ensuite décrites. Enfin, les différentes notions relatives à la sûreté de fonctionnement et applicables à ce type de système sont présentées.

1 SA, SdA, SAID

Ce premier sous-chapitre définit ce qu'on entend aujourd'hui par système automatisé (SA) et par système d'automatisation (SdA, composant du premier). Ce dernier est vu tout d'abord selon un aspect fonctionnel pour ensuite être étudié comme une organisation de composants matériels.

1.1 Système automatisé

Un système automatisé peut se définir de la manière suivante :

> Ensemble d'éléments organisés pour accomplir une fonction, en minimisant les interventions humaines. [DEL 89]

L'accomplissement d'une fonction est généralement équivalent à la délivrance d'un service (comportement d'un système tel qu'il est perçu par son ou ses utilisateurs [LAP 95]). La minimisation des interventions humaines caractérise l'automatisation.

Dans le cas d'un système automatisé de production, sa fonction s'exprime en terme de valeur ajoutée. Delcuvellerie propose la définition suivante :

> Système destiné à augmenter la valeur ajoutée de produits conformément à des objectifs de productivité et de qualité. [DEL 89].

L'apport de l'automatisation réside dans l'amélioration des objectifs à atteindre qui peuvent avoir trait à [HER 97] :

- l'accroissement de la productivité dans le but d'améliorer la rentabilité et en conséquence la compétitivité,
- l'amélioration de la qualité par une meilleure répétabilité, de meilleures "régulations" et une meilleure surveillance,
- l'augmentation de la sécurité relative aux opérateurs, à l'environnement ou au système lui-même.
- faciliter les tâches de l'opérateur particulièrement pour les travaux pénibles ou en environnements hostiles.

Ceci sous-entend qu'un certain nombre de tâches est accompli plus efficacement par des "automatismes" que par des opérateurs humains.

Les définitions précédentes sont plutôt liées au domaine de l'automatisation, auquel est accolé le vocable système. Ce terme système porte en lui-même une sémantique qu'il nous semble nécessaire de préciser dans un contexte industriel.

Un système est un ensemble d'éléments en interaction dynamique organisés en fonction d'un but [DEL 89]. Cette définition est récursive et certains éléments peuvent à leur tour être considérés comme des sous-systèmes composés d'éléments plus élémentaires. Un critère d'arrêt lié au point de vue employé, fixe une décomposition limite.

La première décomposition possible distingue pour les systèmes automatisés de production (SAP) les entités suivantes [STA 94] :

- l'homme : il participe aux diverses tâches de conduite, maintenance, gestion technique, gestion de production, entretien et évolution du système. Le SAP fournit une partie des informations nécessaires à ces activités [LAP 95]. L'utilisateur d'un système est un autre système (humain ou physique) qui interagit avec le système considéré [LAP 95]. Un opérateur est un homme réalisant une tâche en interaction avec un système technique dans une situation de travail.
- le processus : "Ensemble des opérations d'élaboration d'un produit selon un procédé déterminé, au moyen d'unités de traitements et de transformation". On distingue :
  - le processus de production qui est le support de la transformation de matière d'œuvre et d'énergie en biens de consommations,
  - le processus de gestion technique qui permet de réceptionner, d'entreposer et de manutentionner la matière d'œuvre et l'énergie pour alimenter la production.
- le système d'automatisation : il est l'interface entre l'homme et le processus. Il commande et contrôle le processus par l'intermédiaire de capteurs et d'actionneurs, en fonction des tâches qui lui ont été assignées.

L'organisation globale de ces entités est choisie en fonction du procédé : "Méthode à suivre pour obtenir un résultat. On y explique les opérations nécessaires pour obtenir le produit résultant" [HER 97].

Une autre décomposition communément utilisée s'appuie sur la nature des flux traités (informationnelle ou matière d'œuvre). Elle distingue ainsi une partie commande (ou système d'information) où sont effectuées les tâches de coordination et une partie opérative qui regroupe l'ensemble des dispositifs et actionneurs qui concourent à l'obtention de la valeur ajoutée [GRE 85].

La première décomposition caractérise mieux ce qu'est actuellement l'activité d'automatisation car elle intègre l'implication croissante des capteurs et actionneurs à l'activité de contrôle/commande favorisée par l'instrumentation intelligente.

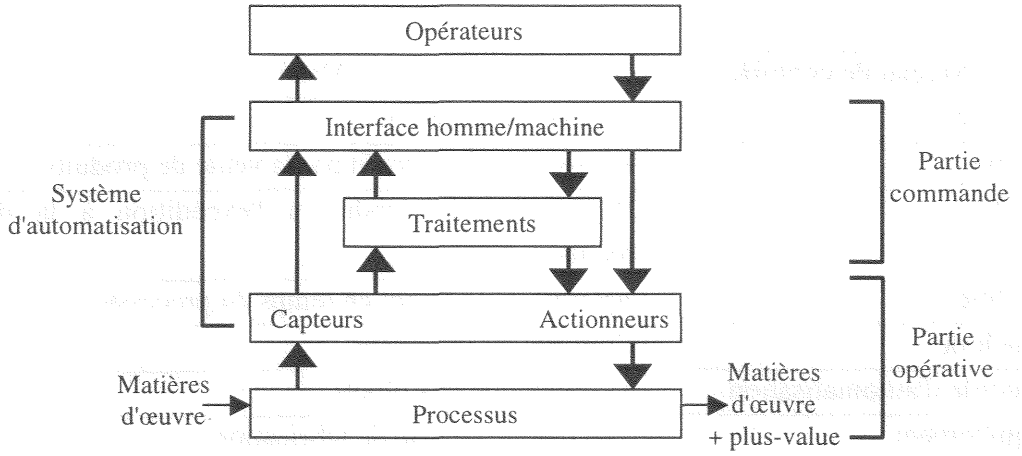


Figure I.1 : Décomposition d'un système automatisé [STA 95]

1.2 Système d'automatisation

Comme tout système, le système d'automatisation peut être décomposé. La figure I.1 en propose une décomposition sous la forme d'une association de trois composants. Selon les points de vue employés, d'autres sont possibles dont les plus communément rencontrées sont présentées dans ce sous-chapitre.

1.2.1 Aspect conceptuel

1.2.1.1 Décomposition hiérarchisée

Selon le contexte, les expressions suivantes sont employées pour dénommer le (sous)système d'automatisation : système de contrôle/commande ou système informatique temps-réel.

La décomposition la plus commune du système d'automatisation s'appuie sur un critère hiérarchique. Entre chaque niveau, elle intègre une notion de contrôle (équivalent à "qui dirige") et/ou une notion de découpage spatial (équivalent à "se compose de"). Chaque niveau doit pouvoir être décrit en terme de missions.

Parmi les modèles disponibles (modèle NBS [JON 86], modèle de Williams [WIL 87], Computer Integrated Manufacturing (CIM) [MEY 87]), le plus complet est le modèle de Biemans [BIE 88] [BIE 86a] [BIE 86b].

Niveau de contrôle	Mission
Société	Gestion corporative
Usine	Réalisation d'un profit par la vente de produits
Atelier	Préparation de produit à l'expédition à la date requise
Cellule	Approvisionnement en temps du processus
Machine	Contrôle du processus
Module d'automatisation	Modification des objets
Equipement	Suivi de la gamme de fabrication
Régulateur	Etablissement de la consigne
Capteur et Actionneurs	Configuration des paramètres

Tableau I.1 : décomposition par le modèle de Biemans

1.2.1.2 Décomposition fonctionnelle et topologique

Le modèle "3 axes" [VER 89] permet de mieux distinguer les différents aspects possibles de la modélisation. Il y est ainsi représenté un axe topologique (représentatif du parcours du flux de produits), un axe hiérarchique (représentatif de la décomposition) et un axe fonctionnel (représentatif des principales fonctions attendues au sein du système).

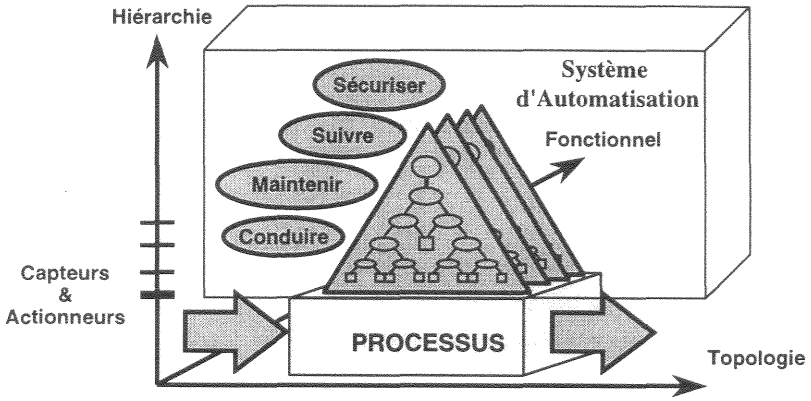


Figure I.2 : décomposition selon le modèle 3 axes

Les tâches qui composent le système d'automatisation au sein de ce modèle sont classées en quatre catégories [SIM 92] correspondant à des finalités distinctes :

- La fonction Conduire : fonction de base destinée à contrôler le comportement futur et immédiat du processus de production, pour atteindre les objectifs de production exprimés en terme de qualité et de productivité. Cette fonction recouvre aussi bien les automatismes séquentiels que les asservissements ou les régulations.
- La fonction Maintenir : fonction de base destinée à assurer une disponibilité maximale du processus de fabrication. Pour atteindre cet objectif, il est nécessaire d'intégrer un niveau de sûreté de fonctionnement satisfaisant.
- La fonction Suivre : fonction de base destinée à recueillir et à synthétiser les informations relatives à l'état présent et passé du processus ainsi qu'à l'état et à la position des produits.

- La fonction Sécuriser : fonction de base destinée à assurer la non occurrence de défaillances catastrophiques qui pourraient mettre en danger le personnel d'exploitation, l'environnement, les installations de production, les produits. Cette fonction s'intègre également aux aspects liés à la sûreté de fonctionnement.

1.2.1.3 Les différentes formes de flux de données

Ces modélisations ne se limitent pas à une simple décomposition. Lors de leurs établissements, on a cherché à définir les relations qui lient les tâches. Globalement il s'agit de flux de données aux caractéristiques particulières à chaque niveau de relation.

Une classification possible des flux de données par niveau est fournie dans [BOI 91] :

Niveau	Volume des flux	Aspect temporel
Machine (capteurs, actionneurs)	Faible taille	Flux périodiques, temps réel obligatoire
Cellule (commande)	Messages de type commande Données de type fichier	Temps réel moins obligatoire
Atelier (supervision et surveillance)	Volume important Messages de type ordre	Le temps différé est possible
Gestion de production	Volumes de messages importants	Pas d'impératif de temps au sens temps réel

Tableau I.2 : Classification des flux de données

Quels que soient les modèles de référence pour les systèmes d'automatisation, ils mettent en évidence qu'utiliser une décomposition hiérarchisée est assez naturelle pour identifier, d'une part, les différentes tâches que doit accomplir le système et, d'autre part, les flux informationnels qui les lient.

1.2.2 Composants et architecture matérielle

De manière simple, l'architecture matérielle d'un système d'automatisation est l'ensemble organisé de machines et de moyens de communication [SIM 95]. Elle résulte donc de choix de conception.

Indépendamment de l'objectif à réaliser, différentes organisations sont possibles qui peuvent être classifiées de la manière suivante [BOI 91]:

- architecture centralisée : un unique calculateur dirige l'ensemble des fonctions informatiques,
- architecture décentralisée : les fonctions sont réparties sur un ensemble de calculateurs de capacité plus réduite avec une faible interaction. Ce phénomène est induit par l'apparition de micro-ordinateurs à faible coût.
- architecture répartie : l'ensemble des équipements est capable de communiquer. Cette structure est permise par l'apparition d'un nombre croissant de composants numériques ce qui a permis, par exemple, d'envisager la simplification des supports de communication

afin de faciliter la maintenance ou la mise en place des redondances matérielles dans le but d'améliorer la disponibilité globale.

Le choix entre ces différentes architectures dépend essentiellement des technologies disponibles.

1.3 Système d'Automatisation à Intelligence Distribuée

Si les évolutions technologiques ont influencé l'architecture des systèmes d'automatisation, il en est de même pour les composants élémentaires. Ainsi, les capteurs ou actionneurs sont devenus "intelligents" par l'adjonction de moyens de traitements.

1.3.1 Capteurs intelligents

L'introduction de la micro-électronique au sein des capteurs a modifié leur rôle de simples transducteurs [ROB 93].

Le premier progrès réside dans l'amélioration de la qualité de la mesure [GEH 94]. Il s'agit dans une première étape de perfectionner les fonctions auparavant réalisées de manière analogique (ex : filtre, compensation...). Le traitement numérique des non-linéarités et la meilleure prise en compte des grandeurs d'influence sont par exemple facilités par l'utilisation de ces supports. Une activité locale de détection de défauts a pu être adjointe et valide les conditions de mesures.

Tous ces perfectionnements visent principalement la fonction mesure. Cependant, la prise en compte de l'intégration du capteur au sein d'un système a incité à déporter certaines fonctions de gestion qui, traditionnellement, étaient dévolues au calculateur central dans l'hypothèse d'une architecture centralisée. Citons par exemple :

- le diagnostic : grâce à la validation de ces composants,
- la maintenance : grâce à la mémorisation d'information de gestion des équipements (identificateur, date d'étalonnage),
- la sécurité : grâce à la possibilité de participer à des arcs réflexes aptes à mettre promptement le système dans des positions de repli,
- la disponibilité : en permettant une coopération entre équipements redondants.

Enfin la possibilité de dialoguer grâce à un système de communication évolué accroît la qualité globale de chacune de ces fonctions.

Ce double aspect est à l'origine des appellations retenues dans la littérature de "smart sensors" (capteurs dotés de capacités de traitements) et de "intelligent sensors" (capteurs destinés à participer à un système dans lequel tous les composants sont coopératifs). [KLE 91]

1.3.2 Actionneurs intelligents

La vocation initiale de l'actionneur est de pouvoir agir sur certaines variables du processus physique. Il s'agit en fait d'une modulation, d'une conversion et d'une transmission d'énergie [STA 94] conformément à un ordre. Ainsi, l'actionneur se limite au niveau bas du pôle conduite (en référence au modèle 3 axes).

L'association de capteurs intégrés permet d'accroître la participation de l'actionneur à l'activité de conduite et aux autres pôles du système d'automatisation. Ces fonctionnalités permettent en outre :

- de surveiller la chaîne de transfert d'énergie et de détecter les éventuelles défaillances ou de les anticiper (ex : détecter une surchauffe),
- d'exécuter une partie des actions de conduite. (ex : commande arrêt par des fins de course intégrés à l'actionneur).

Ces capacités peuvent être étendues par l'emploi de traitements numériques. Ainsi il devient possible :

- de mettre en œuvre une contre réaction locale (ex : régulation en vitesse d'un moteur couplée à une génératrice tachymétrique),
- de participer aux activités de gestion technique et de maintenance par des fonctions d'enregistrement (ex : mesure des durées de fonctionnement).

Enfin, la possibilité de connexion à un système de communication numérique en font un actionneur intelligent par sa capacité à participer à l'ensemble des activités du système d'automatisation en s'intégrant à la base de données globale du système.

1.3.3 Instrument intelligent

L'instrument intelligent (qu'il soit capteur, actionneur ou unité de commande) se caractérise par sa capacité à accueillir localement un certain nombre de traitements. Ces derniers n'ont pas seulement pour objectif la commande mais participent à l'ensemble des fonctions attendues dans un système d'automatisation (commande, conduite, maintenance, gestion de production...).

Son architecture matérielle repose ainsi sur [BOU 97]:

- des moyens de traitement permettant la réalisation des fonctions attendues,
- des moyens de mémorisation aptes à accueillir les informations de conduite, de maintenance, de gestion technique...
- des moyens de communication avec l'ensemble du système d'automatisation pour assurer la diffusion des informations élaborées localement et recevoir les données utiles à ses actions (en provenance des opérateurs ou des autres composants).

Selon la nature des traitements exécutables par les instruments intelligents, on distingue [AKA 96] :

- les instruments "fermés" pour lesquels un ensemble de traitements a été pré-implanté par le constructeur. Ils correspondent généralement à la finalité de l'instrument. L'utilisateur n'a comme seule option que son paramétrage. Ce sont principalement des capteurs ou des actionneurs.
- les instruments "ouverts" pour lesquels les traitements sont programmables. Ils nécessitent avant emploi d'y implanter les tâches à réaliser. Ils correspondent actuellement surtout à des équipements de commande tels que des automates.

L'activité de distribution des traitements se limite souvent à une répartition des traitements aux instruments "ouverts", l'architecture matérielle ayant été préétablie. Les travaux présentés dans ce mémoire tentent de rapprocher les étapes du choix de l'architecture matérielle et de la répartition des tâches.

1.3.4 Système de communication

La notion de système d'automatisation distribuée est intrinsèquement liée à celle de système de communication. Son rôle a minima est la valorisation des fonctionnalités des instruments intelligents en leur permettant un échange efficace de leurs informations.

Cela se traduit par :

- la mise à disposition d'un "canal" de communication pour chaque information entre son producteur et les consommateurs concernés,
- un respect des contraintes temps réel imposées par le processus. Si l'on considère la chaîne mesure/décision/action, tous les maillons doivent maintenir la continuité temporelle.

Ceci pourrait être réalisé physiquement par de simples liaisons "point à point". Cependant, limiter le système de communication à ce rôle passif n'est pas optimum pour le système général. Les missions suivantes permettent de le rendre actif à l'ensemble des missions du système d'automatisation :

- faciliter sa maintenance (détection des défauts, faciliter la réparation par une simplification du câblage),
- participer à la (re)configuration (réorganisation des échanges suite à la défaillance d'instrument ou d'une partie du réseau, ou à une évolution du processus par l'adjonction de nouveaux composants),
- informer des éventuelles défaillances du support de communication (permettant des mises en repli plus rapide) [CON 98],
- accepter un nombre suffisant de transmissions pour permettre le contrôle et la surveillance à distance (améliorant les conditions de travail pour les opérateurs)...

Le réseau de terrain répond à ces spécifications grâce au multiplexage de données sur un même bus et à la diffusion de ces données entre abonnés [ROB 98].

2 L'activité de conception

Concevoir, c'est "définir les caractéristiques d'un objet ou d'une procédure recherché(e)" [BON 92]. Cette définition peut être étendue à "définir les spécifications du produit, génériques ou propres à un client particulier, développer des prototypes et certifier la conformité" [ROB 93].

2.1 Problème de la conception

La conception est ici envisagée dans le cadre d'une relation entre un demandeur (le client) et le concepteur.

Le concepteur est confronté aux problèmes suivants [CAL 90] :

1. Gérer la relation avec le client

Fondamentalement, il s'agit de satisfaire le client. Il doit ainsi, tout d'abord par le dialogue, traduire les besoins exprimés le plus souvent de manière très informelle. Cette transposition est souvent délicate car le client n'a pas nécessairement une connaissance complète des services que peut rendre l'automatisation (ce qui justifie généralement l'appel à un tiers pour résoudre son problème de conception). Par ailleurs, il exprime un ensemble de contraintes supplémentaires qui consistent à proposer une solution à un coût convenable, à

respecter des délais imposés et à pouvoir prouver (autant que faire se peut) la validité de la solution proposée.

2. Offrir une réponse adéquate pour des systèmes de plus en plus complexes.

Le bon fonctionnement n'est plus le seul objectif à atteindre aujourd'hui. La multiplicité des fonctions attendues pour un système d'automatisation augmente la complexité des systèmes.

3. Profiter de la technologie la plus adaptée

L'évolution constante des techniques diversifie les solutions envisageables. Ainsi, le concepteur doit envisager et comparer un nombre croissant d'architectures et sélectionner la meilleure.

4. Satisfaire des critères variés de qualité

Le simple fait de fonctionner ne suffit pas au client. Il attend un système de bonne qualité c'est-à-dire ayant de bonnes caractéristiques de robustesse de l'installation (fiabilité, tolérance aux pannes), de maintenabilité (facilité à identifier les défaillances et à les corriger), ... C'est en partie par l'intégration appropriée de fonctions de gestion et de surveillance associées aux composants que cette qualité pourra être atteinte.

L'ordonnancement des relations client/concepteur complique la démarche [DEN 94]. En effet, le client avant de s'engager requiert une offre chiffrée. Pour pouvoir y répondre le concepteur doit envisager une solution suffisamment précise et être capable d'en estimer le coût de développement. En revanche, il ne faut pas que l'investissement pour réaliser cette pré-étude soit trop important, car il est susceptible de ne pas être couvert en cas de rejet du projet par le client.

2.2 Activités de développement

Les activité du cycle de vie d'un système d'automatisation tel qu'il est présenté dans [BAY 92] sont les suivantes :

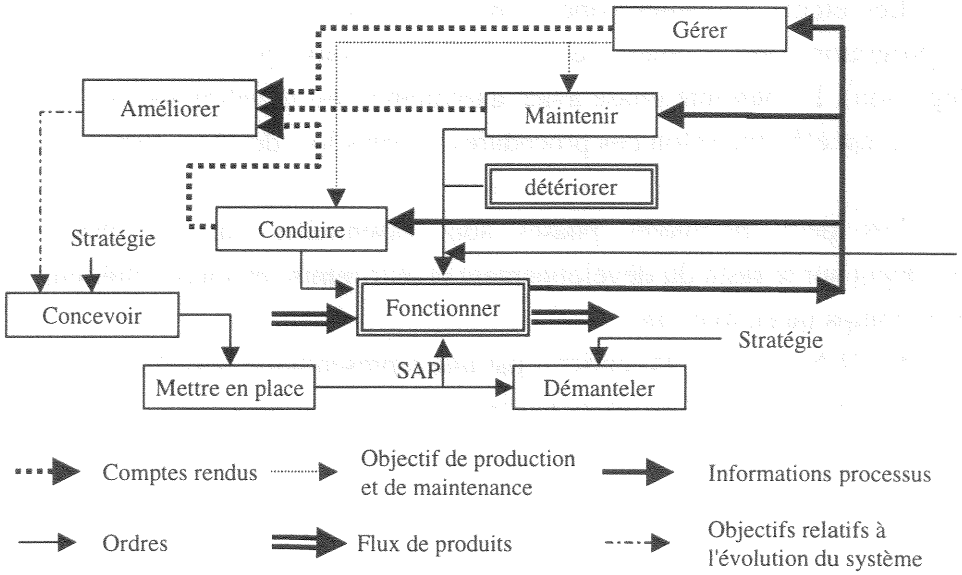


Figure I.3 : Activités du cycle de vie

Ce graphe traduit l'objectif propre du système d'automatisation, qui est de "fonctionner", tout en prenant en compte les autres fonctions que sont la maintenance ("Maintenir") et la gestion ("Gérer").

Si ce diagramme exprime clairement les liens entre les différentes activités, en revanche, il masque le déroulement de la conception comme une succession d'étapes. La représentation du développement du système par un cycle en V, définit ces étapes [AFN 92].

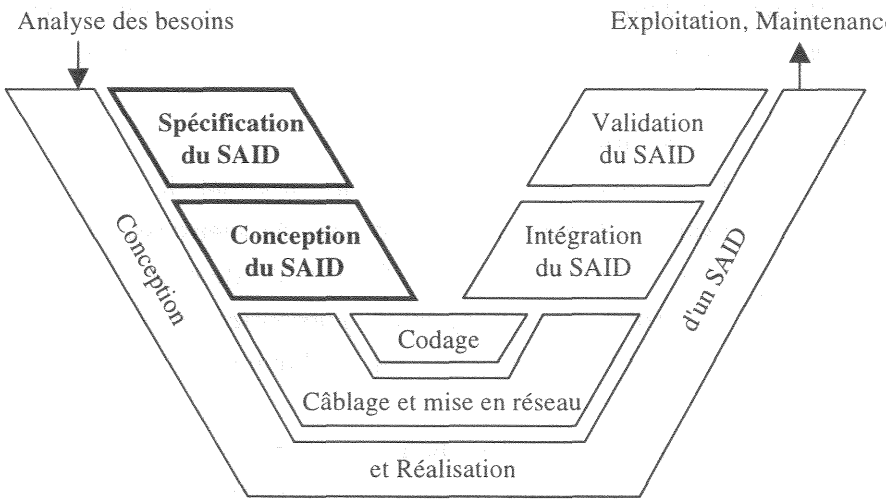


Figure I.4 : cycle de développement en V

On y distingue la phase de spécification où les fonctions attendues sont définies en faisant abstraction de toute implémentation matérielle. La phase de conception correspond à l'établissement de l'architecture matérielle (choix des équipements, du réseau...) pour laquelle l'affectation des fonctions sur ses composants est définie (choix de la distribution des fonctions). Le résultat obtenu est alors l'architecture opérationnelle. Ce sont à ces deux phases que s'intéressent les travaux présentés ici.

Les étapes suivantes concernent tout d'abord le développement des composants par leur programmation (codage) et leur câblage, suivi par leur mise en place sur le site (intégration). La dernière phase avant exploitation est la validation par une campagne de test. Tout ceci est effectué selon des procédures décrites lors des deux premières phases.

Les deux premières phases sont essentielles car les choix effectués influent directement sur le reste du développement et leur remise en cause ultérieure est coûteuse aussi bien en temps qu'en matériel.

B. DENIS [DEN 94] enrichit par une représentation SADT ces phases en y intégrant la notion d'objectif relatif à un appel d'offres ou à un développement.

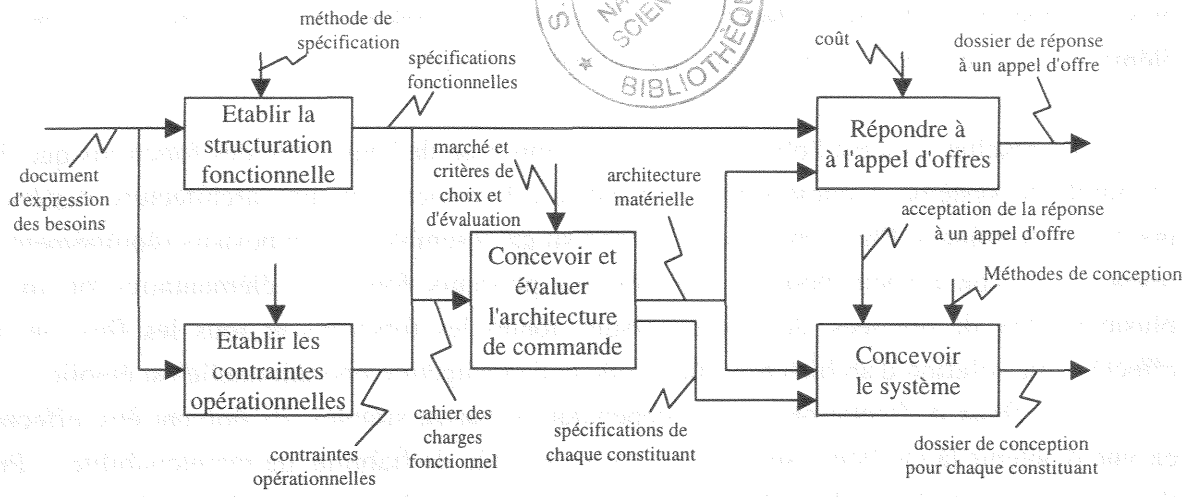


Figure I.5 : diagramme SADT de conception d'un SdA [DEN 94]

On y retrouve la démarche séquentielle d'établissement d'une spécification fonctionnelle suivie par la recherche d'une architecture matérielle/opérationnelle.

2.3 Double conception : fonctionnelle / matérielle

La conception telle qu'elle est considérée pour le développement d'un système d'automatisation distribué requiert l'établissement préliminaire d'une architecture fonctionnelle, théoriquement indépendante de tout aspect matériel.

Appelée aussi modèle d'implantation, elle a pour objectif d'exprimer l'ensemble des activités du système et leur structuration à travers un seul et unique formalisme. Elle répertorie donc toutes les fonctions caractérisées par les services qu'elles doivent rendre. Ces fonctions sont en majorité des composants logiciels mais certaines sont directement liées au processus physique (fonction de mesure ou d'actionnement). A cette description du service attendu et pour complètement définir chaque fonction, il faut préciser son comportement. Ainsi, il s'agit de définir les conditions de déclenchement qui peuvent être de type périodique ou faire suite à l'occurrence d'un événement. De plus, la nature temps réel de l'activité de conduite du processus nécessite de définir certaines contraintes temporelles (durée d'exécution, temps de réaction...) pour chaque fonction.

A cette description individuelle des fonctions, il faut adjoindre les flux qui les interconnectent. De manière similaire aux fonctions, leur nature doit être définie. Il peut s'agir de l'échange d'un événement, d'une donnée périodique (mesure, consigne, commande...), de fichiers... Une partie de ces flux est à caractère temps réel et pour chacun d'eux il est nécessaire d'y associer les contraintes temporelles exprimées généralement sous la forme d'une durée maximale de transmission.

Différentes méthodes existent pour établir cette architecture fonctionnelle à partir d'un cahier des charges. Une des façons de faire consiste en une approche dite descendante [SIM 95]. Appelée aussi démarche de planification hiérarchique [BON 92], elle consiste en une décomposition de chaque tâche en sous-tâches (ou de façon plus globale, chaque fonction

en sous-fonctions). Par raffinement successif, on aboutit finalement à des tâches suffisamment élémentaires pour être implantées sur des composants matériels.

Le résultat final est l'obtention d'une organisation de l'ensemble des fonctions que doit accomplir le système. L'étape suivante consiste à la projeter sur une architecture matérielle que l'on déterminera. Plus simplement, on choisit un ensemble de composants (équipements et réseaux) où l'on associe pour chacun une ou plusieurs fonctions élémentaires ou un ou plusieurs flux de données. A l'issue, lorsque toutes les fonctions et tous les flux ont été affectés, une solution d'architecture matérielle et d'architecture opérationnelle est établie.

Les choix d'affectation ne sont cependant pas aussi simples. Ils doivent être effectués en vue d'obtenir la meilleure solution en terme de coût, de fiabilité, de maintenabilité... Pour des systèmes complexes, l'emploi d'algorithmes de recherche de la meilleure allocation peut grandement faciliter ce travail.

La représentation SADT de DENIS [DEN 94] intègre cette démarche. La sélection entre différentes possibilités d'affectation est décrite par la composition d'une activité "évaluer" et une "choisir".

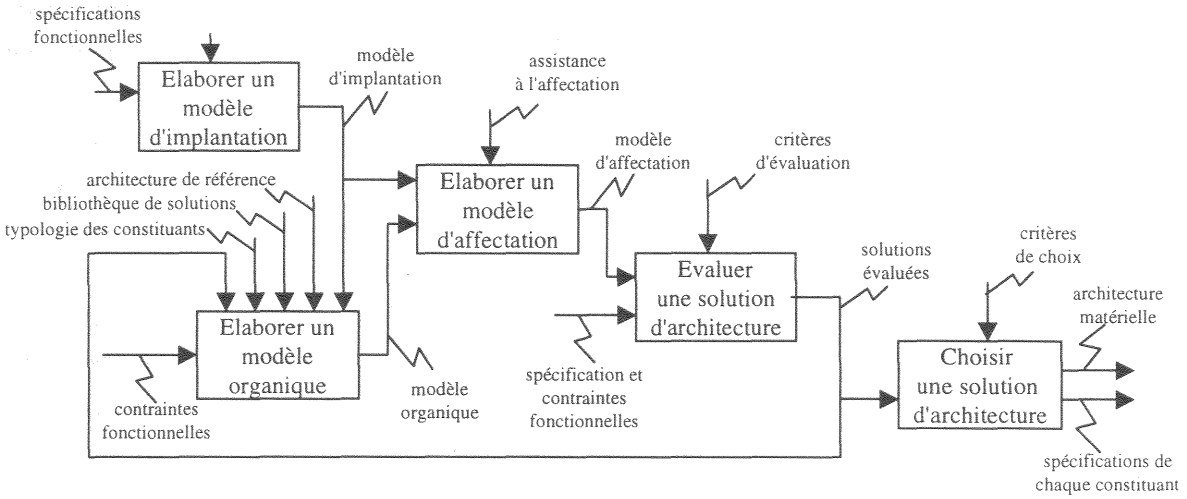


Figure I.6 : diagramme SADT de conception et évaluer l'architecture [DEN 94]

Cette démarche purement descendante est cependant trop théorique. Même si elle correspond au schéma général de conception, elle occulte de nombreux aspects. D'une part, le choix de raffinement lors de la démarche ascendante anticipe les solutions matérielles et se base en partie sur leur connaissance. D'autre part, l'établissement d'une architecture opérationnelle induit souvent l'ajout de nouvelles fonctions destinées à favoriser la gestion technique du système. Ce peut être par exemple des outils d'administration du réseau.

Ainsi, une démarche ascendante subsiste. Appelée aussi "composition d'unités de conceptions" [BON 92], elle consiste à envisager des solutions d'architecture partielles. Individuellement, elles correspondent à l'association de plusieurs fonctions élémentaires (de l'approche descendante). Une partie ou la totalité de l'architecture opérationnelle est obtenue par l'agrégation de ces solutions.

A ces deux démarches très souvent citées dans le domaine de la conception de système d'automatisation, une autre peut être adjointe. Qualifiée en psychologie cognitive d'**opportuniste** [VIS 87], elle exprime le fait que la poursuite de l'analyse n'est pas linéaire. Des interruptions dans le cycle sont effectuées par le concepteur et celui-ci "reprend les différents composants de son projet de conception plusieurs fois avant de les considérer achevés". Que la démarche générale soit descendante ou ascendante, l'avancement du projet peut porter sur plusieurs séquences simultanément. Dans la progression de l'étude, il est ainsi possible d'étudier successivement des parties très distinctes. Selon le résultat obtenu pour chacune d'elles, des remises en cause successives des autres éléments peuvent s'avérer efficaces dans la conduite de l'analyse.

En conséquence, l'utilisation d'une démarche descendante est le schéma directeur pour l'établissement d'une architecture fonctionnelle. Elle doit cependant permettre d'intégrer (même de manière informelle) des éléments résultants d'une démarche ascendante, due à sa finalité qui est d'être projetée sur une architecture matérielle. Enfin le formalisme adopté doit faciliter les remises en cause successives, en autorisant de manière transitoire l'utilisation d'éléments partiellement définis. Ceci dans le but d'être en harmonie avec la démarche d'analyse propre aux concepteurs qui sont susceptibles de revenir à plusieurs reprises sur un même élément afin d'affiner au fur et à mesure les solutions envisagées.

2.4 Critère de sélection : l'évaluation

Déjà mentionnée auparavant, la sélection de la meilleure architecture opérationnelle nécessite une phase d'évaluation.

D'un point de vue général, elle consiste à évaluer des indicateurs quantitatifs et qualitatifs relatifs à la performance a priori du système. Les critères d'appréciation employés sont particuliers au système. Ils dépendent en grande partie du cahier des charges. Ils doivent permettre d'apprécier l'adéquation entre le système envisagé et l'objectif (expression du désir des utilisateurs traduisible en termes de critères d'appréciation [DEL 89]).

Ces critères concernent principalement le coût du système et la sûreté de fonctionnement. Le premier fait référence au coût d'installation (coût des équipements, de leur installation et du développement des composants logiciels) et à celui de fonctionnement (personnel nécessaire, coût d'entretien...). La sûreté de fonctionnement renvoie à des notions de fiabilité (capacité à produire), de sécurité (capacité à éviter des événements catastrophiques), de qualité (capacité à produire conformément aux spécifications).

De par leur nature très distincte, l'obtention d'une appréciation globale est délicate. Certains critères ne sont pas quantifiables et pour ceux qui le sont, les unités dans lesquelles ils sont exprimés n'autorisent pas forcément une comparaison directe (ex : coût de programmation et taux de défaillance).

Un des travaux du concepteur consiste donc à concilier ces critères d'appréciation à caractère absolu et relatif. Certains servent à vérifier si les solutions envisagées sont satisfaisantes par rapport aux objectifs attendus (caractère absolu). D'autres doivent permettre de différencier les solutions envisagées pour en considérer une comme étant la meilleure (caractère relatif).

En conséquence, les méthodes de conception doivent intégrer des éléments aptes à permettre une évaluation des solutions envisagées. Cela implique que la représentation de l'architecture opérationnelle doit faciliter l'estimation des critères en fournissant les données nécessaires et en les ordonnant de telle sorte que leur exploitation en soit simplifiée.

Les travaux présentés ont cet objectif et sont essentiellement consacrés à la prise en compte de la sûreté de fonctionnement dans le processus d'évaluation décrit précédemment.

3 La Sûreté de Fonctionnement

Après avoir étudié les systèmes d'automatisation et leurs possibilités de distribution (première partie) puis la démarche de conception dans le cadre de systèmes industriels (seconde partie), nous abordons maintenant les différents aspects de la sûreté de fonctionnement.

3.1 Présentation générale

La Sûreté de Fonctionnement est, au sens large, la Science des Défaillances [VIL 88]. Au sens strict, elle est souvent vue comme un concept générique et est définie comme la propriété qui permet à ses utilisateurs de placer une confiance justifiée dans le service qui leur est délivré [LAP 88]. Cette notion est particulièrement complexe car un système doit rendre non pas un mais plusieurs services pour assurer sa mission.

Concernant les systèmes de production, la sûreté de fonctionnement englobe un ensemble de notions qui se distinguent par leur objectif (délivrer correctement un service, éviter des conséquences néfastes...) ou par l'activité qui peut leur être associée (maintenir les équipements en état de fonctionner). Ainsi, elle se définit par [ISO 93] :

> un ensemble de propriétés qui décrivent la disponibilité et les facteurs qui la conditionnent : fiabilité, maintenabilité et logistique de maintenance

A ces aspects purement productifs, la recherche de moyens aptes à éviter les événements catastrophiques pour les biens et les personnes s'intègre à la sûreté de fonctionnement sous la dénomination de sécurité. Son caractère particulier (on ne s'intéresse plus à la valeur ajoutée) en fait un domaine d'étude spécifique où seules la gravité des conséquences et leur probabilité d'occurrence sont prises en compte.

3.2 Faute, erreur, défaillance : Notion de fiabilité

La première entrave au bon déroulement d'une des missions d'un système d'automatisation est la défaillance. Elle est définie comme [AFN 88] :

> Cessation de l'aptitude d'une entité à accomplir une fonction requise

En terme de service, une défaillance est un événement survenant lorsque le service délivré dévie de l'accomplissement de la fonction du système [LAP 95].

Il s'agit ainsi d'une transition d'un service correct vers un service incorrect et son occurrence fait suite à un contexte particulier. On distingue la faute qui est à l'origine de la défaillance. Elle apparaît suite à un phénomène physique ou à un mauvais comportement humain. Elle peut entraîner une erreur. L'erreur est un état du système susceptible de provoquer la défaillance. Elle caractérise ainsi le contexte dans lequel la faute va pouvoir se propager en une défaillance par la modification du service délivré.

Une chaîne de causalité regroupe ainsi ces trois notions (faute → erreur → défaillance). Celle-ci peut se poursuivre à travers les relations d'interactions qui unissent les différentes fonctions du système. La défaillance d'une fonction devient alors une faute pour une autre.

La fiabilité qualifie la capacité d'un système à accomplir sa mission et est définie de la manière suivante [AFN 88] :

> Aptitude d'une entité à accomplir une fonction requise, dans des conditions données, pendant un intervalle de temps donné.

3.3 Remise en service : notion de maintenabilité

Les systèmes d'automatisation associés à un système de production sont en général des systèmes réparables (tous les systèmes ne le sont pas, ex : satellite). Ainsi dans le contexte qui nous intéresse, la défaillance n'est pas définitive, une remise en service est possible. La remise en service est définie comme [AFN 88] :

> Récupération de l'aptitude d'une entité à accomplir une fonction requise après une panne.

Cette capacité pour une entité à être rétablie appartient en partie à la notion de maintenabilité, définie de la manière suivante [AFN 88] :

> Dans des conditions données d'utilisation, aptitude d'une entité à être maintenue ou rétablie, sur un intervalle de temps donné, dans un état dans lequel elle peut accomplir une fonction requise, lorsque la maintenance est accomplie dans des conditions données, avec des procédures et des moyens prescrits.

3.4 Etats d'un système : notion de disponibilité

La défaillance et la remise en état traduisent des transitions d'un état à un autre caractérisant des comportements différents. Le comportement d'une entité ou un service peut être ainsi représenté sous forme d'état.

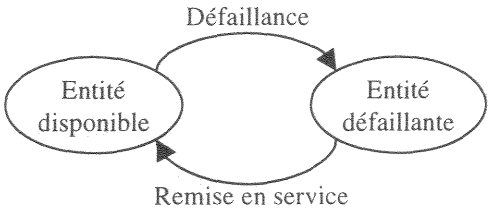


Figure I.7 : Graphe d'état disponible/défaillant

De manière simpliste, un état binaire peut qualifier le comportement d'une entité à rendre les services qui lui sont demandés. Défaillance et remise en service sont les transitions entre ces deux états qualifiés de "disponible" ou "défaillant". L'état défaillant ou panne est définie comme [AFN 88] :

> Etat d'une entité inapte à accomplir une fonction requise, dans des conditions données d'utilisation

La disponibilité est définie comme [AFN 88] :

> Aptitude d'une entité à être en état d'accomplir une fonction requise dans des conditions données, à un instant donné ou pendant un intervalle de temps donné, en supposant que la fourniture des moyens extérieurs nécessaires soit assurée.

Lorsque ce point de vue est adopté, la véritable fonction de la maintenance est de conserver l'état disponible. En complément des actions de remise en service, la maintenance doit également agir sur cet état disponible pour qu'il le reste. La maintenance se définit ainsi comme [AFN 88] :

> Ensemble des actions destinées à maintenir ou rétablir une entité dans un état dans lequel elle peut accomplir une fonction requise

Il est possible d'affiner l'analyse des causes d'indisponibilité en distinguant d'autres états que celui de panne. Exprimés en proportion du temps, ils vont rendre compte des différentes causes d'incapacité d'un système à accomplir sa mission, ceux-ci ayant une influence directe sur la productivité de ce système.

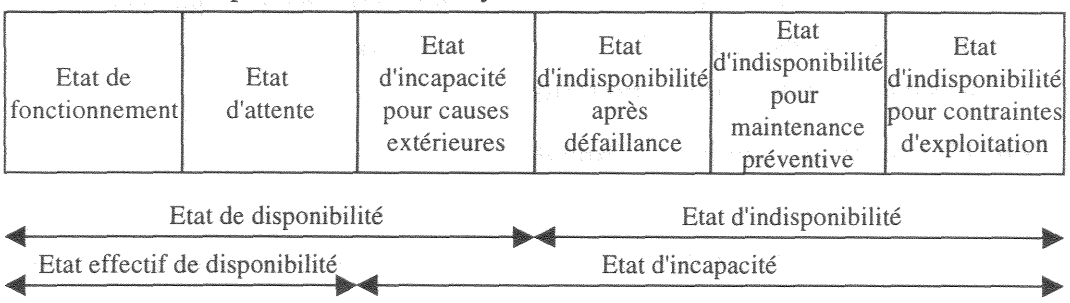


Figure I.8 : Classification des états d'une entité [AFN 88]

3.5 La redondance

La principale technique pour assurer une bonne sûreté de fonctionnement est la redondance. Elle se définit de la manière suivante [AFN 88] :

> Existence dans une entité, de plus d'un moyen pour accomplir une fonction requise

On distingue [AFN 88] :

- la redondance active où tous les moyens d'accomplir une fonction requise fonctionnent simultanément,
- la redondance passive où une partie seulement des moyens d'accomplir une fonction requise est en fonctionnement, le reste n'étant utilisé sur sollicitation qu'en cas de défaillance de la partie en fonctionnement.

Son emploi correspond à différentes finalités même si son rôle est toujours d'assurer une continuité d'un service. Dans une vision de production, il s'agit de remplacer les équipements défaillants pour poursuivre la mission. Dans une vision "sécurité", il s'agit de conserver les fonctions de protection malgré l'absence des équipements défaillants, ce qui peut être au détriment d'autres fonctions (en général de production). En conséquence, on dénomme ces deux objectifs (souvent antagonistes) sous l'expression "conflit sécurité/disponibilité". Dans le premier cas, on privilégie une architecture dite "série" où chaque équipement contrôle les actions des autres. Dans le second cas, une architecture dite "parallèle" permet à un équipement défaillant d'être remplacé par un autre.

3.6 Mesure de la SdF

Une des étapes de la conception est l'évaluation. Cela impose de posséder une métrique pour justifier qu'une solution est meilleure qu'une autre. De par la nature de la sûreté de fonctionnement, elle est plus délicate à établir car elle fait référence à de nombreuses notions difficilement conciliables. De plus, celles-ci dépendent fortement des missions et du contexte du système.

L'emploi de grandeurs offre des possibilités d'association. Ainsi, la plupart des concepts liés à la sûreté de fonctionnement ont la possibilité d'être quantifiés et peuvent alors être utilisés indirectement comme des critères à caractère économique.

Seuls deux aspects de la sûreté de fonctionnement (disponibilité et fiabilité), seront utiles par la suite et sont développés ci-après.

3.6.1 Mesure de disponibilité

La disponibilité (instantanée) est mesurée par [AFN 88]:

> La probabilité qu'une entité soit en état de disponibilité dans des conditions données à un instant donné en supposant que la fourniture des moyens extérieurs nécessaire soit assurée.

Elle est noté [VIL 88] :

$A(t) = P$ [entité non défaillant à l'instant t]

Cette grandeur exprimée sous la forme d'un taux offre un premier outil de quantification. En effet, il devient possible d'évaluer à partir de ce taux et de la durée de vie estimée (D_v) du système, quelle pourrait être la durée de fonctionnement (D_f) du système et indirectement quelle serait sa production possible.

$$D_f = \int_0^{D_v} A(t)dt$$

ou, sous l'hypothèse d'une disponibilité constante (A) : $D_f = D_v \times A$

Concernant les systèmes, il peut être trop restrictif de ne considérer que les deux états "disponible"/"indisponible". En effet, l'architecture de certains systèmes peut permettre des modes de fonctionnement dégradés c'est-à-dire où la production est moindre en quantité ou en qualité. Ainsi, plusieurs taux doivent être alors utilisés pour estimer la production possible. De même, différents états d'indisponibilité peuvent être identifiés car ils n'induisent généralement pas les mêmes répercussions économiques. Ainsi pour une chaîne de fabrication, il est préférable d'être en arrêt plutôt que d'avoir une production non conforme et invendable due à la défaillance de certains composants. La "Total Productive Maintenance" (TPM [RIC 96]) différencie les états productifs des états non-productifs [NAK 89]. Par leur association à des durées, ils rendent compte indirectement des profits que l'on peut tirer du système.

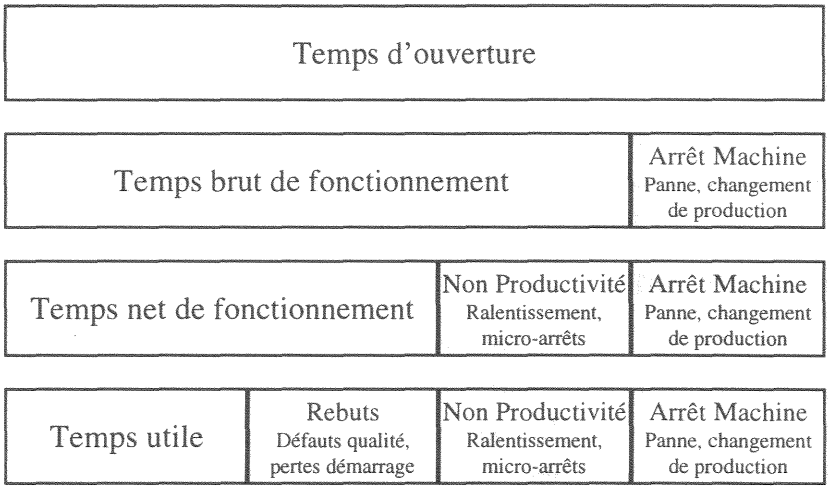


Figure I.9 : Classification des états d'une entité [PIM 91]

Un des objectifs de la conception va donc consister à évaluer la part de chacun de ces états pour des architectures envisagées de systèmes d'automatisation afin de choisir parmi celles qui offrent la plus grande disponibilité.

3.6.2 Mesure de la fiabilité

La fiabilité est mesurée par [AFN 88] :

> La probabilité pour qu'une entité accomplisse une fonction requise, dans des conditions données, pendant un intervalle de temps donné [0,t].

Elle est aussi noté [VIL 88] :

R(t) = P [entité non défaillante sur [0,t]]

Cette grandeur dépend directement du taux (instantané) de défaillance $\lambda(t)$ c'est-à-dire la probabilité instantanée pour qu'une défaillance apparaisse. Ainsi :

$$R(t) = \exp\left(-\int_0^t \lambda(u)\, du \right)$$

ou sous l'hypothèse d'un taux de défaillance constant λ : $R(t) = e^{-\lambda t}$

L'utilisation des ces mesures (fiabilité et taux de défaillance) peut être trop restrictive pour être utilisée directement pour l'évaluation globale d'un système d'automatisation. En effet, on ne s'intéresse alors qu'à la cessation du service tandis que la façon dont se produit cette interruption est tout aussi importante. Par exemple, à l'occurrence de défaillances entraînant une indisponibilité, le système peut soit prendre une position dite de repli, soit continuer dans un fonctionnement non productif (production non conforme) voire dangereux.

Ainsi les conséquences ou les effets d'une défaillance ou d'une association de défaillances doivent être prises en compte au même titre que leur probabilité d'apparition. La notion de criticité introduite dans les analyses AMDEC (Analyse de Mode de Défaillance et de leurs Effets et de leurs Criticités, étudiée dans le chapitre III), intègre ces deux aspects [AFN 86]. En conséquence, il peut être plus judicieux d'étudier les aspects relatifs à la fiabilité à l'aide du couple probabilité-gravité et ce pour chaque mode de défaillance du système étudié.

Enfin, pour un système d'automatisation, étant donné que lors de la conception on envisage toute la vie de ce système, il peut être plus intéressant de considérer les défaillances par leur nombre d'occurrences auquel un coût financier peut être associé. Ainsi, ces mesures à caractère économique et relatives à la fiabilité peuvent être comparées ou intégrées aux autres critères d'évaluation.

4 Conclusion

Ce chapitre s'est attaché à définir le rôle et la fonction du système d'automatisation. De l'étude des composants actuellement utilisables, il est apparu que son architecture matérielle peut permettre une distribution des traitements. Ces systèmes peuvent être dénommés Système d'Automatisation à Intelligence Distribuée.

Afin de prendre en compte cette spécificité dans la réalisation de ce type de système, leur conception a été étudiée. La démarche la plus traditionnelle consiste en une première étape à effectuer une analyse fonctionnelle. Etablie par décompositions successives, elle aboutit à un recensement de fonctions élémentaires et de flux les interconnectant. Ces "briques" élémentaires vont pouvoir dans une seconde étape être employées pour déterminer l'architecture matérielle susceptible de les supporter.

Une des difficultés de la conception réside dans le fait que pour un même système, de nombreuses possibilités d'architectures peuvent remplir la mission souhaitée. Un des travaux du concepteur est donc de pouvoir évaluer les solutions envisagées afin d'en déterminer une qui soit satisfaisante. Les critères d'évaluations peuvent lui offrir un outil pour effectuer cette sélection.

Intégrer les aspects relatifs à la sûreté de fonctionnement en phase de conception revient à les considérer comme des critères d'évaluation. Deux mesures de la sûreté de fonctionnement ont été ainsi présentées. L'une est relative à la disponibilité et caractérise l'aptitude du système à accomplir sa mission. L'autre est relative à la fiabilité et rend compte des dysfonctionnements possibles qui pourraient apparaître et de leurs conséquences.

5 Bibliographie

[AFN 86] AFNOR, Techniques d'analyse de la fiabilité des systèmes - Procédures d'analyse des modes de défaillance et de leurs effets (AMDE), X 60-510, fascicule de documentation, 1986.

[AFN 88] AFNOR, Terminologie relative à la fiabilité - Maintenabilité – Disponibilité, X 60-500, norme expérimentale, 1988.

[AFN 92] AFNOR, Projet : Génie Automatique : Représentation des systèmes de contrôle et de commande des systèmes automatisés (norme expérimentale), Pr Z 68-901, janvier 1992

[AKA 96] J. AKAICHI, Systèmes Automatisés de Production à Intelligence Distribuée. Des stratégies de Répartition Basées sur une Approche de Classification, Thèse de doctorat de l'Université des Sciences et Technologies de Lille, 1996.

[BAY 92] M. BAYART, M. STAROSWIECKI, Coherence of distributed applications under critical time constraints, IFAC/IFIP Worshop on Real Time Programming WRTP 92, Bruges, juin 1992.

[BIE 86a] F.P. BIEMANS, Reference model of production control systems, IECON'86, 1986.

[BIE 86b] F.P. BIEMANS, The design of distributed transport systems as a major standard interface in Computer Integrated Manufacturing, Computers in Industry, vol. 7, n°4, 1986.

[BIE 88] F.P. BIEMANS, Computational tasks in robotics and factory automation, Computers in Industry, vol. 10, n°2, juillet 1988.

[BOI 91] S. BOIS, Intégration de la gestion des modes de marche dans le pilotage d'un système automatisé de production, Thèse de doctorat de l'Université des Sciences et Techniques de Lille Flandres Artois, novembre 1991.

[BON 92] N. BONNARDEL, Le rôle de l'évaluation dans les activités de conception, Thèse de doctorat de l'Université d'Aix en Provence, juillet 1992.

[BOU 97] A. BOURAS, Contribution à la conception d'architectures réparties : modèles génériques et interopérabilité d'instruments intelligents, Thèse de doctorat de l'Université des Sciences et Technologies de Lille, juillet 1997.

[CAL 90] J.-P. CALVEZ, Spécification et conception des systèmes : une méthodologie, édition Masson, 1990.

[CON 98] B. CONRARD, J.M. THIRIET, M. ROBERT, A tool for the evaluation of fieldbuses reliability for intelligent distributed automation system, in S.G. TZAFESTA: Advanced in manufacturing, decision, control and information technology, Springer, ISBN 1-85233-126-7, (Sélection d'articles de EURISCON'98, Athens, Juin 1998), 1999.

[DEL 89] J.L. DELCUVELLERIE, Ingénierie des systèmes d'automatisation de production, Thèse de doctorat de l'I.N.P.L. Université de Nancy I, mars 1989.

[DEN 94] B. DENIS, Assistance à la conception et à l'évaluation de l'architecture de conduite des systèmes de production complexes, Thèse de l'Université de Nancy I, février 1994.

[GEH 94] A.L. GEHIN, Analyse fonctionnelle et modèle générique des capteurs intelligents : application à la surveillance de l'anesthésie, Thèse de doctorat de l'Université des Sciences et Technologies de Lille, janvier 1994.

[GRE 85] GREPA : Le GRAFCET – de nouveaux concepts, Cépaduès-éditions, 1985.

[HER 97] F. HERMANN, Contribution à la répartition des traitements et des données sur une architecture distribuée équipée d'un réseau de terrain FIP : Application à un processus thermique pilote, Thèse de l'Université Henri Poincaré, Nancy I, novembre 1997.

[ISO 93] Norme internationale, Norme pour la gestion de la qualité et l'assurance de la qualité, Partie 4 : Guide de gestion du programme sûreté de fonctionnement, ISO 9000-4, 1993.

[JON 86] A.T. JONES, C.R. Mc LEAN, A proposed Hierarchical Control Model for Automated Manufacturing System, Journal of Manufacturing System, vol. 5, n°1, 1986.

[KLE 91] P. KLEINSCHMIDT, F. SCHMIDT, How many sensors does a car need ?, Eurosensors V, Roma, pages 1 à 13, 1991.

[LAP 88] J.-C. LAPRIE, Sûreté de Fonctionnement et tolérance aux fautes : concepts de base, rapport LAAS n°88.287, paru dans les Techniques de l'ingénieur, 1988.

[LAP 95] J.-C. LAPRIE et al., Guide de la Sûreté de Fonctionnement, Cépaduès-éditions 1995.

[MEY 87] W. MEYERE, S. APPEL, ESPRIT 932 : Knowledge Based Real-Time Supervision in CIM, 18 months report 3/1, supplt. A1, Task 1 : dedicated AIP modules for CIM, Philips Hamburg, Août 1987.

[NAK 89] S. NAKAJIMA, La maintenance productive totale (TPM) Mise en œuvre, Edition afnor gestion, 1989.

[PIM 91] PIMOR, TPM la Maintenance Productive, Masson, 1991.

[RIC 96] RICHET D., GABRIEL M., MALON D., BLAISON G., Maintenance basée sur la fiabilité, Masson, 1996.

[ROB 93] M. ROBERT, M. MARCHANDIAUX, M. PORTE, Capteurs intelligents et méthodologie d'évaluation, édition Hermès, 1993.

[ROB 98] M. ROBERT, B. CONRARD, Réseaux de terrain et sûreté de fonctionnement des systèmes d'automatisation distribuée, Semaine de l'électronique et de la physique 98, SEP 98, p1-12, Paris, Septembre 1998.

[SIM 92] F. SIMONOT-LION, C. VERLINDE, Importance d'un cadre de référence dans la mise en place d'une démarche de développement d'un système automatisé de production, Acte de la conférence, Automatisation Industrielle, vol I, Montréal, juin 1992.

[SIM 95] F. SIMONOT-LION, La démarche de conception des systèmes d'automatisation, Journées d'Etude SAPID, Paris, Mai 1995.

[STA 94] M. STAROSWIECKI, M. BAYART, Actionneurs intelligents, Hermès, Paris, 1994.

[VER 89] C. VERLINDE, Contribution à l'étude des architectures de systèmes automatisés, Thèse de doctorat de l'I.N.P.L. Université de Nancy I, 1989.

[VIL 88] A. VILLEMEUR, Sûreté de fonctionnement des systèmes industriels, Edition Eyrolles, Paris, 1988.

[VIS 87] W. VISSER, Abandon d'un plan hiérarchique dans une activité de conception, Acte de COGNITIVA'87, Paris, 1987.

[WIL 87] T.J. WILLIAMS, The Integrated Approach to Industrial Control, IFAC 87, Munich, 1987.

Chapitre II
Modélisation d'un Système d'Automatisation

Chapitre II
Modélisation d'un Système d'Automatisation

La conception d'un système d'automatisation s'appuie généralement sur l'utilisation de plusieurs modèles aptes à répondre à différents objectifs intermédiaires utiles lors de la démarche d'analyse. Certains de ces modèles visent à identifier les services ou traitements que doit accomplir le système (point de vue fonctionnel). D'autres s'attachent à définir ou à expliciter le comportement des fonctions (point de vue comportemental). Enfin, les derniers cherchent à représenter l'architecture capable de réaliser les missions attendues (point de vue matériel et "exécutionnel"). Durant la démarche d'analyse, ces modèles coopèrent et chacun enrichit au fur et à mesure l'image du système à réaliser. Parallèlement, les informations issues de ces différents modèles servent lors d'activités d'évaluations pour juger les architectures envisagées (matérielles et opérationnelles) afin d'en déterminer une comme étant la meilleure.

Ce chapitre présente ainsi les différentes représentations utiles lors de la conception et sur lesquelles la méthodologie proposée s'appuiera.

1 Modélisation fonctionnelle

De nombreux ouvrages [CAL 90] [JAU 90] recensent les méthodes de développement de systèmes les plus communément utilisées. On peut les classer en trois catégories [LUT 97] : la première dite "orientée fonctions" cherche à identifier les fonctions et leurs interactions ; la deuxième "orientée données" s'intéresse à modéliser les informations utilisées par le système étudié ; enfin la dernière "orientée objets" considère que les fonctions et les données sont étroitement liées et sont étudiées simultanément.

Nous aborderons essentiellement l'approche "orientée fonctions" car c'est sur celle-ci que reposent les travaux présentés.

1.1 Méthodes orientées fonctions

La plupart des méthodologies de conception de systèmes d'automatisation s'appuient sur la construction préalable d'une décomposition fonctionnelle. Il s'agit de caractériser l'ensemble des fonctions internes au système. L'expression du besoin décrit le service ou la fonction que doit rendre le système. L'activité de conception fonctionnelle consiste à décomposer cette fonction complexe en fonctions plus simples, puis celles-ci en sous-fonctions encore plus simples, et à nouveau jusqu'à aboutir à des fonctions élémentaires. Un critère d'arrêt sera présenté ultérieurement. Cette démarche s'apparente à la décomposition d'un problème en sous-problèmes. Pour être bien menée, elle se doit d'être indépendante des techniques de réalisation, c'est-à-dire séparer l'étude et la compréhension du problème de la description de ses solutions [RIV 95].

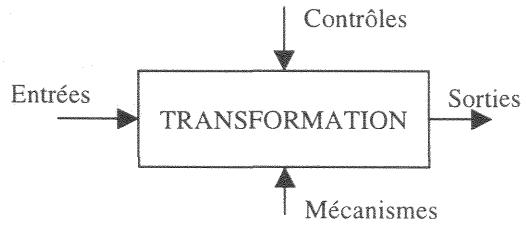
Cette démarche fournit en final une structuration hiérarchique de la mission du système en fonctions. Cependant, elle doit être complétée par une identification des flux de données entre ces fonctions pour aboutir à une architecture fonctionnelle. Ainsi, selon sa nature, chaque fonction consomme ou produit des données (variable ou événement) en

provenance ou à destination d'autres fonctions. Cette description explicite de ces flux de données (aboutissant à l'obtention de graphes) exprime indirectement des relations d'interconnexions entre fonctions.

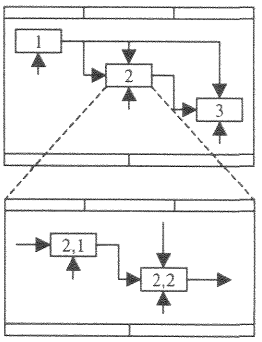
L'emploi de représentations graphiques facilite la réalisation et l'utilisation de ces décompositions fonctionnelles. En effet, ces représentations peuvent apporter un gain en concision et en non-ambiguïté, si elles sont bien utilisées. Les méthodes ou outils suivants proposent des règles d'écriture aptes à permettre cette analyse.

1.1.1 La représentation SADT

La méthode SADT [IGL 89] cherche à identifier par décomposition des activités (équivalentes aux fonctions recherchées) et à représenter leur enchaînement. Chaque activité dite "de transformation" modélise la modification des caractéristiques de produits ou d'informations. Elle se caractérise par des flux d'entrée (données ou produits consommés), des flux de sortie (données ou produits résultant de la transformation) et des flux de contrôle (paramètres de la transformation). Enfin cette représentation permet de définir le ou les équipements chargés d'accomplir chaque transformation. Ce dernier point même s'il semble antinomique à la conception fonctionnelle indépendante de toute solution, est utile pour la progression du projet. Au début de la conception préliminaire, il permet de définir le type d'équipement (capteur, actionneur, automate, ordinateur...) et à la fin il permet d'exprimer les composants choisis pour l'affectation des fonctions.



Représentation d'une boîte de transformation avec les positions des différents flux



Représentation hiérarchisée des différents niveaux de décomposition utilisée par SADT.

Figure II. 1 : Actigramme et décomposition SADT

Les faiblesses souvent énoncées de la méthodologie SADT concernent l'absence de distinction entre les différentes natures des flux. Ainsi un seul type de flèche représente aussi bien des flux de produits que de données et aucune différence n'est faite entre des flux de synchronisation de ceux contenant des informations. De plus, les aspects temporels n'y sont pas explicités. Enfin on reproche parfois à cette méthode d'être "lourde" et fastidieuse d'emploi.

Malgré ces quelques faiblesses, la méthode SADT est couramment utilisée, car elle impose une certaine rigueur. Une application de cette méthode est proposée dans la thèse de HERMANN [HER 97] et traite du système d'automatisation d'un processus thermique.

1.1.2 La représentation SART par diagramme de flot de données

La dénomination S.A.R.T. (Structured Analysis Real Time) désigne en fait deux méthodologies d'analyse de système. La première (notée WM) est à l'initiative du Dr P. Ward et S. Mellor [WAR 86], la seconde (notée HP) de D. Hatley et I. Pirbhai [HAT 91]. Ces méthodologies s'appuient sur plusieurs représentations dont une seule sera développée ci-après. Il s'agit des diagrammes flots de données et de contrôle qui appartiennent dans ces méthodologies au modèle des besoins et qui dans notre contexte peuvent être employés dans le cadre de la spécification des fonctions.

Conceptuellement proches de la méthode SADT, ces diagrammes représentent à l'aide de graphes les différents traitements (nommés processus) du système et les relations qui les lient. Ces relations sont essentiellement des flots de données (schématisés par des flèches continues) et représentent alors le chemin d'un processus à un ou plusieurs autres d'une donnée ou d'un regroupement de données. Certaines extensions distinguent par un symbolisme particulier (apparence de l'extrémité de la flèche) les flots continus et discrets. D'autres relations expriment par des flots d'événement (représentés par des flèches pointillées) le contrôle des processus, c'est-à-dire des ordres d'activation ou de désactivation. Les processus (représentés en pointillé pour WM ou par une barre verticale pour HP) chargés de gérer ces flots apparaissent de manière distincte. Enfin une distinction graphique est utilisée pour indiquer des unités de stockage de données (à l'aide d'une double barre) [PER 90].

La même possibilité de raffinement itératif que pour SADT est proposée. Ainsi un processus peut représenter un diagramme de flots de données à un niveau inférieur comportant des sous-processus.

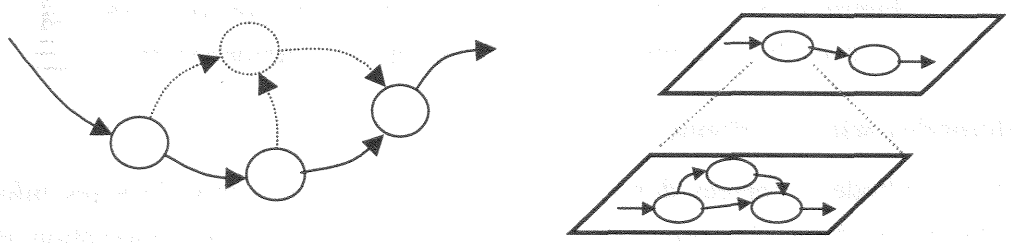


Figure II. 2 : Diagramme de flots de données et leur décomposition

Cette représentation plus souple que celle employée par SADT, offre l'intérêt de représenter plus facilement les différents traitements qui composent un système. De plus, les plus nombreux symboles graphiques disponibles des diagrammes de flots de données fournissent des représentations plus riches en information en décrivant la nature des flux ou des processus schématisés. En contrepartie, tous les aspects relatifs aux interactions du système avec un processus physique sont plus difficilement modélisables qu'avec SADT, et facilitent moins le travail de projection sur une architecture matérielle.

L'intérêt de ces deux représentations réside pour le concepteur dans le fait que :

- les fonctions du système sont vues selon une approche "boîte-noire" [CAL 90],
- les différentes interactions fonctionnelles sont décrites selon deux aspects, le premier hiérarchique (fonction décomposée en sous-fonctions), le second flux de données (organisation de fonctions),
- leur simplicité et leur nature intuitive les rendent facile d'emploi,
- les règles de représentations graphiques offrent des documents plus facilement lisibles.

1.1.3 Aspect lié à la sûreté de fonctionnement

Bien que ces représentations permettent de représenter différentes formes de redondance, elles n'intègrent pas d'information utile à l'évaluation de la sûreté de fonctionnement. Un des objectifs des travaux présentés est de pallier ce manque. Ainsi à la modélisation de chaque niveau de décomposition du système, on cherchera à adjoindre des paramètres utiles à l'étude du comportement de l'installation en présence de défaillance, tout en laissant au concepteur la liberté du choix du formalisme concernant la description fonctionnelle.

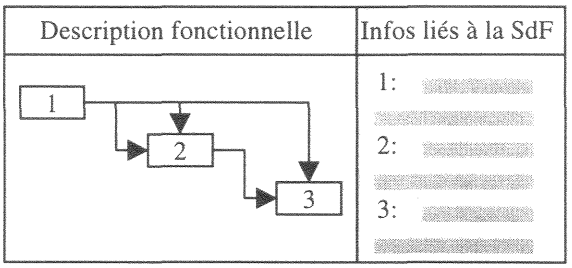


Figure II. 3 : Enrichissement des décompositions fonctionnelles avec des informations liées à la sûreté de fonctionnement

1.2 Méthodes orientées données

Les méthodes orientées données visent principalement à modéliser les informations traitées par un système. Le principal outil est le diagramme entité-association (Objets + attributs + associations ou relations) [CHE 76]. La méthode Merise en est inspirée [TAR 87].

Ces diagrammes définissent les entités comme des éléments distincts. Ils sont regroupés au sein de classes entre lesquelles des relations sont identifiées. Plusieurs attributs peuvent être utilisés et complètent les classes et les relations.

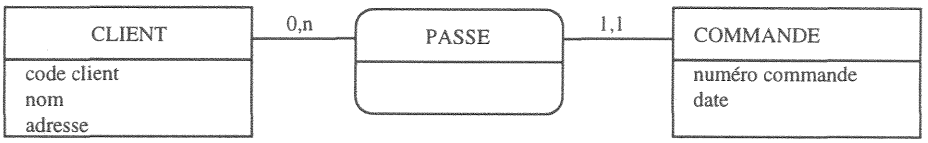


Figure II. 4 : Modèle entité-association

On reproche souvent à l'emploi de telles modélisations leurs difficultés à fournir une vision globale et structurée du système et à identifier clairement les flux de données [DEL 89]. Ce dernier aspect est primordial dans le cas d'une structure distribuée où plusieurs équipements doivent échanger des informations. Ce type de méthode peut cependant être

complémentaire aux autres approches, particulièrement quand l'utilisation d'une base de données suffisamment conséquente est nécessaire.

1.3 Méthodes orientées objets

L'approche orientée objet est une technique de développement qui intègre de façon cohérente aussi bien les aspects données que traitements [RUM 95].

Elle s'appuie sur les notions suivantes. Un objet est la représentation d'une entité indépendante à laquelle sont associés des données et un comportement exprimé au travers de méthodes. Il est susceptible d'accomplir un ensemble d'opérations. Celles-ci sont les seuls éléments visibles de l'extérieur et au travers d'une interface. Cette technique offre de nombreuses facilités de développement. Ainsi, l'emploi du mécanisme d'héritage permet de ne définir qu'une seule fois les similitudes que peuvent présenter plusieurs objets, regroupés alors au sein d'une classe.

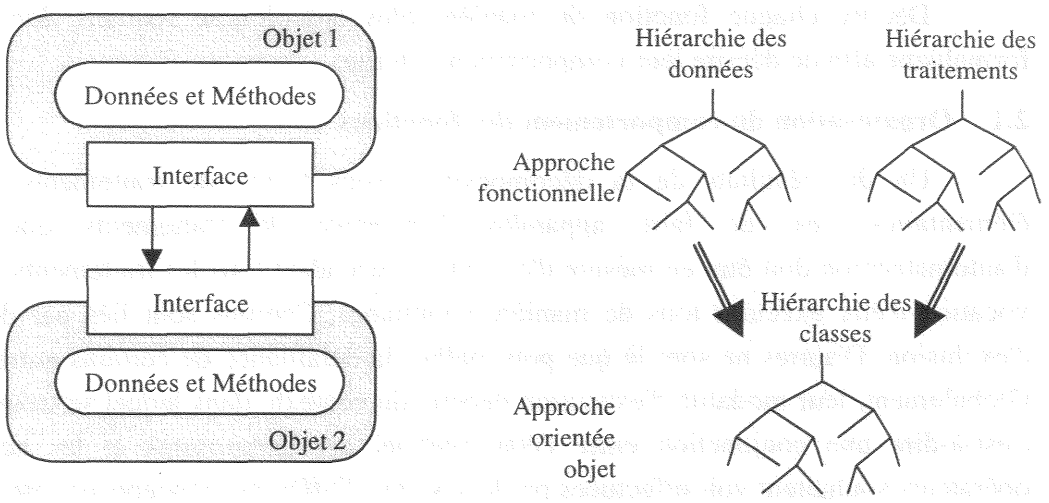


Figure II. 5 : communications entre objets et hiérarchie des structures [LUT 97]

Les méthodologies objet présentent l'avantage de pouvoir intégrer le développement informatique à partir de l'organisation ou modélisation du système en un ensemble d'objets. De plus, elles semblent mieux adaptées aux évolutions possibles du besoin qui pourraient survenir durant la conception ; ainsi les modifications seraient moins nombreuses qu'avec des méthodes à base de flots de données, dû au fait que la structuration en objets s'appuie plus sur le domaine d'application que sur les besoins fonctionnels du système à concevoir [RUM 91].

Une des difficultés de l'emploi des méthodes objet pour le problème de la répartition des traitements, réside dans l'identification des traitements élémentaires à distribuer. En effet, cette modélisation incite plutôt à distribuer les objets et non leurs méthodes. Il devient alors nécessaire de définir des objets suffisamment élémentaires pour pouvoir être distribués. Ceci risque alors de revenir à un diagramme flot de données, perdant ainsi une partie des atouts des techniques objets.

De plus, l'association d'informations relatives à la sûreté de fonctionnement y est moins aisée qu'avec les diagrammes flots de données. En effet, l'enrichissement proposé s'appuie sur la possibilité d'avoir une représentation hiérarchique par décomposition des

différents éléments du système (fonction ou objet). Or les techniques objets ne nécessitent pas cette hiérarchisation et l'étude de la sûreté de fonctionnement telle qu'elle est proposée n'est plus applicable.

Cependant les nombreux avantages des méthodes objets (plus grande facilité de développement informatique, capacité de réutilisation...) en font un outil puissant pour la conception et elles peuvent être utilisées sous certaines conditions dans la méthodologie présentée.

2 Représentation comportementale

Le degré maximal de décomposition des fonctions est atteint dès que les considérations technologiques doivent être prises en compte. Les fonctions élémentaires obtenues sont alors caractérisées par trois propriétés [MES 95], une identité fonctionnelle (un service unique est rendu), une identité géographique (un composant unique la réalise) et une identité d'exécution (la fonction ne peut être exécutée partiellement).

Décrire chaque fonction de manière plus approfondie requiert donc un nouveau formalisme afin de décrire leur comportement interne.

2.1 Organisation du comportement des fonctions

Un des résultats de la décomposition sous forme de traitements (ou fonctions élémentaires), est de faire apparaître l'ensemble de traitements que le système d'automatisation doit être en mesure d'exécuter. Cependant tous les traitements n'ont pas pour vocation d'être exécutés tous de manière simultanée. Certains sont liés par des contraintes d'exclusion. D'autres ne sont là que pour pallier la défaillance de certains autres traitements. Globalement, leur modalité d'exécution dépend du contexte dans lequel se trouve le système, c'est-à-dire une conjonction entre l'état matériel des composants et les actions que les opérateurs souhaitent voir effectuées par le système. Différents concepts ont été proposés pour représenter et organiser leurs exécutions.

2.1.1 Les modes d'utilisation

Les modes d'utilisation forment une première structuration de l'organisation des traitements. Plus évolué que le GEMMA (Guide d'Etude des Modes de Marches et d'Arrêts [ADE 79]), ils visent à classifier de manière cohérente les services offerts sans imposer une structure arbitraire ; celle-ci dépend de la nature de l'équipement ou par extension du système.

A un instant donné, un équipement est dans un mode d'utilisation (unique) qui définit le sous-ensemble de services disponibles. Ceux-ci ont été choisis par leur compatibilité. Ainsi, le mode d'utilisation "conduite automatique" rejette tout ordre relatif à une conduite manuelle. Le rôle des modes d'utilisation est ainsi en premier lieu de regrouper par catégorie les services compatibles.

Si chaque mode d'utilisation définit les services disponibles, Staroswiecki [STA 94] propose de leur associer des droits d'accès. Ainsi pour un mode et un service considéré, seul un ensemble prédéfini d'opérateurs pourra demander son exécution. D'un point de vue pratique, cela autorise, par exemple, uniquement l'opérateur de conduite à utiliser le service "commande manuelle" en mode "conduite manuelle", alors qu'en mode "maintenance" seul

l'opérateur de maintenance peut utiliser ce service. Les modes d'utilisation permettent également de contrôler la bonne coordination des opérateurs. Ceci nécessite en contrepartie une procédure d'authentification où seuls les opérateurs compétents ont accès aux modes dont ils ont besoin.

Globalement les modes d'utilisation permettent de :

- structurer les services pour rejeter les ordres contradictoires
- contrôler les droits d'action de chaque opérateur.
- contrôler la bonne utilisation particulièrement en terme de procédure de mise en œuvre ou en marche.

Ils accroissent la sûreté de fonctionnement en évitant les utilisations incohérentes ou menées par des opérateurs non qualifiés.

2.1.2 Les modes de marches et les modes de fonctionnement

Les modes d'utilisation tels qu'ils sont évoqués précédemment sont relativement indépendants de l'état matériel du système. Les défaillances ne sont prises en compte que par l'indisponibilité de certains services relatifs au changement de mode. Globalement, ils ne rendent cependant pas compte de l'état de dégradation du système. La notion de mode de marche a été introduite dans cette optique.

Un service n'est disponible de façon nominale que si l'ensemble prescrit des ressources (informationnelles et matérielles) pour son exécution est disponible. Lorsqu'une des ressources vient à manquer et si le concepteur a prévu un traitement de remplacement, le service reste disponible mais sous une version dégradée.

La version du traitement à utiliser pour exécuter un service, est spécifiée par le mode de marche. Ce dernier dépend d'un part des actions forcées, c'est-à-dire des ordres provenant des opérateurs (ex : changement de mode d'utilisation) et d'autre part des actions spontanées (ex : perte d'une ressource). Ainsi, le mode de marche est défini comme la combinaison d'un mode d'utilisation (voulu par l'opérateur) et de l'état du système (subi et dû aux défaillances).

		Etats de l'entité (subi)		
		…	Etat j	…
Mode d'utilisation (voulu)	…	…	… ↓	…
	MU i →	…	Mode de marche ij	…
	…	…	…	…

Table II. 1 : Correspondance entre états et modes d'utilisation en modes de marche [STA 94]

Les modes de fonctionnement offrent une autre forme pour caractériser l'état d'une entité ou d'un système. Un mode de fonctionnement est défini comme [AFN 88] :

> Une combinaison de fonctions d'une entité simultanément requises.

Cette notion est ainsi relativement proche des modes de marche. Elle définit pour un système quels traitements sont mis en œuvre pour accomplir un service demandé. Le mode de fonctionnement d'un système dépend ainsi à la fois des demandes des opérateurs concernant le service demandé et des états des composants.

2.1.3 Aspect hiérarchique

Les considérations précédentes s'adressent essentiellement à un équipement et décrivent son comportement sous la forme d'un automate à état fini. Chaque état correspond à un mode (d'utilisation, de marche ou de fonctionnement) et selon le point de vue adopté est équivalent à des états "marche" ou "arrêt" (pour la commande), "disponible" ou "indisponible" (pour la sûreté de fonctionnement)...

En considérant le système d'automatisation comme un système hiérarchique, on obtient à chaque niveau l'équivalent d'un composant d'automatisation. Il est en effet chargé d'accomplir une fonction. Des modes de fonctionnement peuvent lui être attribués et ils précisent alors sa capacité à rendre les services demandés et la façon dont ils le sont. Ce sous-système n'est cependant que la composition d'éléments d'automatisation plus élémentaires et n'est physiquement qu'une association de composants coopérants.

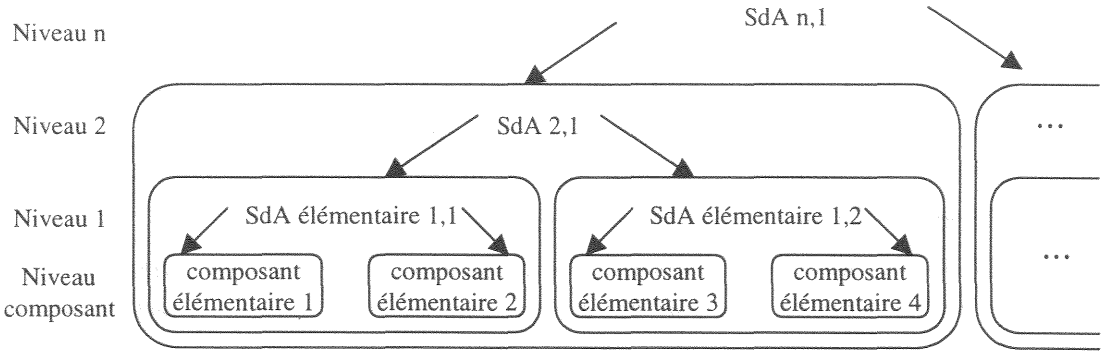


Figure II. 6 : association hiérarchisée de sous-systèmes d'automatisation

Dans cette organisation hiérarchisée, les modes (d'utilisation, de marche ou de fonctionnement) sont liés les uns aux autres. Une relation définit le mode de chaque système selon celui de ses sous-systèmes ou celui du système (plus global) auquel il appartient. Cette relation peut avoir une propagation aussi bien ascendante que descendante. Ainsi la défaillance d'un composant (équivalent à la perte d'un service au niveau composant) se propage de manière ascendante par des changements de mode de marche aux niveaux supérieurs. Inversement les ordres des opérateurs sont appliqués généralement au sommet de la hiérarchie et vont se diffuser de manière descendante avec par exemple des changements de modes d'utilisation.

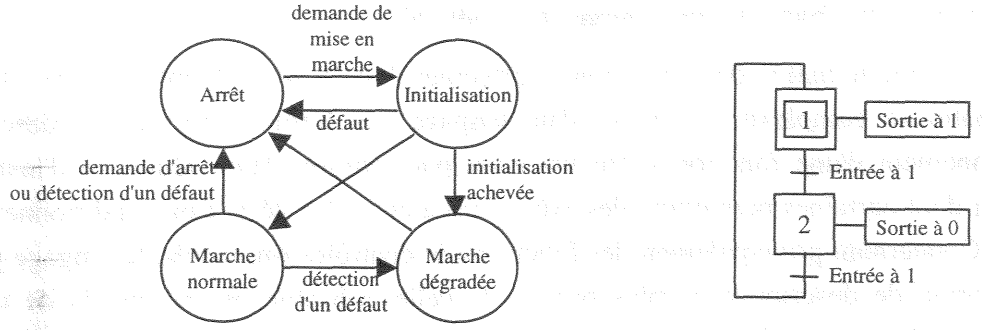
Ces modes permettent de modéliser les comportements d'un composant, d'une fonction ou d'une association de composants ou de fonctions. Les paragraphes ci-après s'intéressent à la modélisation et à la spécification d'une fonction unique, considérée alors comme une entité indépendante.

2.2 Formalisme basé sur des graphes d'état

Un grand nombre de fonctions peut être représenté graphiquement à l'aide de graphes du type état/transition permettant une modélisation à états discrets. Différentes formes sont possibles : Grafcet, réseau de Petri, "statecharts" ... Elles mettent en évidence des séquences d'états et/ou d'événements. [CAL 90]

Les modes de marche ou de fonctionnement décrivent des macro-étapes [BOI 91]. Ils représentent le comportement attendu de la fonction selon les ordres des opérateurs et selon l'état physique des composants. Afin de pouvoir implanter les fonctions sur un système d'automatisation, la description de ces fonctions doit être complétée par celle de leurs macro-étapes. Pour de nombreuses fonctions et particulièrement les fonctions de commande, des graphes peuvent permettre cet enrichissement.

Le Grafcet est alors souvent utilisé. Il permet de définir l'état des signaux de sortie selon ceux disponibles en entrée. Il est particulièrement simple d'emploi quand les sorties suivent une séquence prédéfinie où les étapes sont franchies suite à un changement déterminé des signaux d'entrée ou à une temporisation.



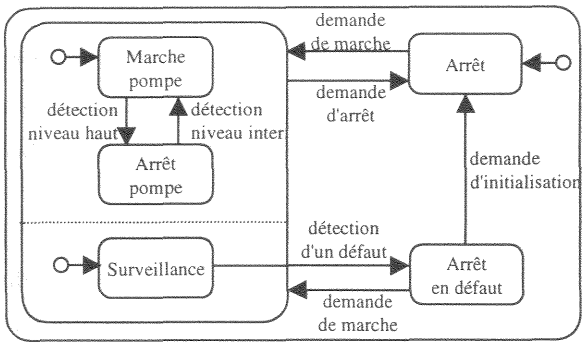
Graphe de macro-étapes

Grafcet pour un mode de fonctionnement

Figure II.7 : Descriptions par graphe d'une fonction

Le graphe des macro-étapes (issues des modes de fonctionnement) ne fournit qu'un schéma directeur pour la réalisation de la fonction et son implantation. Certaines transitions entre modes sont ainsi gérées au sein de ces macro-étapes. Elles proviennent de la mise en place de détections de défauts (ex : contrôle du changements des signaux d'entrée dans un temps déterminé) ou de la lecture de signaux d'entrées contrôlés par les opérateurs.

L'intégration de plusieurs mécanismes (détections de défauts, commandes des opérateurs...) au sein d'une même fonction, peut parfois trop complexifier l'élaboration d'un diagramme de type Grafcet. Les statecharts [RUM 95] offrent une autre forme de représentation réduisant les limitations d'un simple graphe. Ils autorisent ainsi, sous un même graphe la représentation de plusieurs "automates" simultanés ou imbriqués les uns dans les autres.



Ce statechart est la spécification d'une fonction chargée de maintenir le niveau d'un fluide dans une cuve. Il y décrit quels sont les ordres transmis à une pompe selon les signaux perçus par des détecteurs de niveau, les ordres fournis par les opérateurs et les messages de détection de défauts émis éventuellement par d'autres fonctions de surveillance.

Figure II.8 : Exemple de statechart

Lorsque la complexité des opérations décrivant une fonction devient trop importante, l'emploi d'un langage procédural est préférable aux graphes. C'est particulièrement le cas d'algorithmes complexes chargés de traiter des nombreuses données analogiques (ex : algorithme adaptatif, utilisation d'un réseau de neurone...). Des formes mixtes existent aussi où certains états des graphes sont associés à l'exécution d'un algorithme.

2.3 Formalisme basé sur un langage procédural

Lorsque la nature des opérations à effectuer devient trop compliquée (ex : traitements mathématiques complexes), l'emploi d'un langage procédural s'impose. La description du fonctionnement d'une fonction s'effectue alors par l'édition d'une séquence d'instructions à exécuter dont certaines permettent des exécutions à caractère itératif ou conditionnel.

Concernant principalement les fonctions de contrôle/commande, le langage procédural est la forme de description la plus proche de celle utilisable par les unités de traitements numériques programmables qui emploient un jeu d'instructions élémentaires restreint. Mais l'emploi de compilateurs permet une retranscription d'algorithmes écrits dans des langages plus évoluées et/ou portables entre différentes machines. Il est à noter que les diagrammes d'états/transitions peuvent être traités de la même façon en les convertissant en un programme compréhensible par les unités de calcul chargées de les exécuter.

Quelle que soit la manière de décrire le fonctionnement des fonctions, les informations que le concepteur en extrait pour son activité de répartition sont relatives :

- au type d'équipement susceptible de supporter la fonction (micro-contrôleur, automate, ordinateur...)
- aux ressources nécessaires (mémoire, vitesse, entrées/sorties...)

3 Modélisation matérielle d'un système d'automatisation

Les modélisations précédentes cherchent essentiellement à décrire les fonctions et traitements que doit réaliser le système d'automatisation et de quelle manière ils sont réalisés. Elles doivent être complétées pour intégrer le second aspect de la conception de ces systèmes, leur réalisation matérielle. Cette dernière impose de nombreuses contraintes dues aux limitations des composants physiques (capacité de traitement limité) et est la source essentielle des dysfonctionnements.

3.1 Architecture matérielle et opérationnelle

L'architecture matérielle (ou architecture support) pour les systèmes répartis est définie comme l'ensemble des ressources matérielles et logicielles permettant la réalisation d'une application [BOU 97]. Pour une application particulière, c'est ainsi une organisation de composants interconnectés. Une telle architecture est donc spécifiée par :

- la liste des instruments et de leurs interconnexions. En considérant le système de communication comme un instrument, cette liste recense l'ensemble des composants (réseaux, liaisons, capteurs, actionneurs...) et leur agencement les uns par rapport aux autres.
- la spécification de chaque instrument. C'est pour chaque composant, l'ensemble des informations caractérisant son comportement et ses possibilités d'action.

L'architecture matérielle résulte de choix (dimensionnement, topologie...) effectués en vue de pouvoir supporter l'architecture fonctionnelle établie préalablement.

La projection d'une architecture fonctionnelle sur une architecture matérielle forme l'architecture opérationnelle [MES 95]. C'est ainsi la mise en relation d'une architecture matérielle et d'une répartition déterminée des différents traitements identifiés dans l'architecture fonctionnelle.

Déterminer l'architecture matérielle revient donc, en partie, à choisir un ensemble organisé de composants capables de supporter les traitements attendus (l'autre partie de cette activité est d'optimiser ce choix [CON 99]). Il est donc nécessaire de posséder un modèle des propriétés des composants permettant de valider l'allocation des traitements.

3.2 Composant d'automatisation pour l'architecture opérationnelle

3.2.1 Modélisation du composant d'automatisation

On modélise classiquement un composant d'automatisation comme l'association de quatre entités [GEH 94]. Ce sont :

- l'entité Interface Processus. Elle se compose d'organes d'acquisition des signaux du processus et/ou permettant d'agir sur celui-ci.
- l'entité Interface Réseau. Elle assure l'échange d'information avec les autres composants du système.
- l'entité Application. Elle réalise des traitements plus ou moins complexes sur les données qu'elle reçoit (ou émet) des (ou vers les) interfaces.
- l'entité Gestion. Elle assure la gestion coordonnée des autres entités.

En utilisant comme représentation des traitements élémentaires les blocs exécutifs [GEH 94], le composant doit partager (ou distribuer) ses ressources afin de pouvoir accueillir ces blocs.

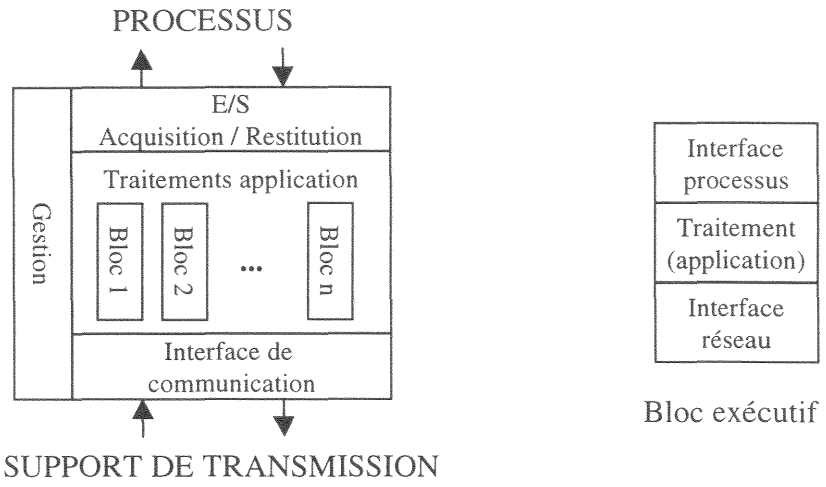


Figure II.9 : Modélisation d'un composant et de ses blocs exécutifs

Le respect des contraintes matérielles impose de vérifier que l'ensemble des ressources demandées par les blocs exécutifs n'excède pas les capacités du composant et ce pour les 3 entités (les 2 interfaces et le traitement).

Cette représentation du composant et de l'allocation des traitements est cependant trop restrictive dans la mesure où elle ne cherche à modéliser que les instruments intelligents. Il est possible de l'étendre à tous les composants du système d'automatisation que ce soit les instruments ou le réseau.

En considérant l'entité gestion comme un bloc exécutif particulier, la décomposition d'un composant d'automatisation se limite aux trois entités : interface processus, interface de communication, application. Chacune possède un nombre limité de ressources.

Pour l'interface processus, ces ressources sont le nombre d'entrées et de sorties du système d'automatisation avec le processus ou bien les opérateurs. Ces ressources sont typées dans la mesure où chacune est destinée à un type de donnée particulière. On peut ainsi distinguer les interfaces homme/machine, les organes de mesure (corps d'épreuve) ou d'actionnement. Donc outre le nombre de ressources disponibles, leur nature est à prendre en compte lors de l'affectation de traitements et peut être assimilée à une contrainte.

Pour l'interface de communication, les ressources s'expriment en nombre de points de connexion. C'est généralement un pour un instrument, deux pour une liaison "point-à-point" (ex. : ligne 4-20mA), plus de deux pour un bus de terrain. Bien évidemment, leur nature (comme précédemment) est une contrainte limitant les interconnexions possibles. Il s'agit ainsi de vérifier si deux équipements sont interopérables. L'interopérabilité d'un ensemble d'équipement est définie comme leur capacité à échanger des informations et coopérer pour satisfaire les besoins de l'utilisateur final [BEN 95].

Pour l'entité application, les ressources sont relatives à la mémoire disponible, aux capacités de traitements... Ce dernier aspect concerne aussi bien la quantité de traitements (ex. : nombre d'opérations possibles à la seconde) que leur nature (ex. : capable d'utiliser ou non des nombres flottants). Enfin les instruments "fermés" (cf. § I.1.2.3.) ne supportent qu'un

nombre limité de traitements pré-implantés mais paramétrables et contraignent encore plus le choix des traitements supportables.

			Capteur (simple)	Capteur intelligent	Actionneur (simple)	Actionneur intelligent	Automate / Calculateur	Liaison pt-à-pt	Réseau de terrain
Ressources	Interface processus	Entrée processus	≥ 1	≥ 1	≥ 0	≥ 0	0	0	0
		Sortie processus	0	0	≥ 1	≥ 1	0	0	0
		IHM (E/S)	≥ 0	≥ 0	≥ 0	≥ 0	≥ 0	0	0
	Interface de communication (points de connexion)		≥ 1	≥ 1	≥ 1	≥ 1	≥ 1	2	≥ 2
	Application	Pré-implanté (paramétrable)	≥ 1	≥ 0	≥ 1	≥ 0	≥ 0	≥ 1	≥ 1
		Programmable	0	≥ 1	0	≥ 1	≥ 1	0	0

Table II. 2 : Classification des composants selon leurs ressources

La modélisation des composants en vue de vérifier, d'une part, la structure de l'architecture matérielle et, d'autre part, l'allocation des traitements, peut être effectuée à l'aide d'une décomposition selon trois entités. Entre composants, cela revient à s'assurer que leur interface de communication possède chacune, un point de connexion interropérable. Pour chaque composant pris individuellement et pour chacune de ses entités, il s'agit alors de contrôler qu'il comporte un nombre suffisant de ressources pour soutenir les traitements, et que la nature de ces traitements est compatible avec les possibilités du composant.

Si l'on cherche à représenter l'architecture matérielle finale d'un système à l'aide de cette représentation des composants, on obtient la forme suivante :

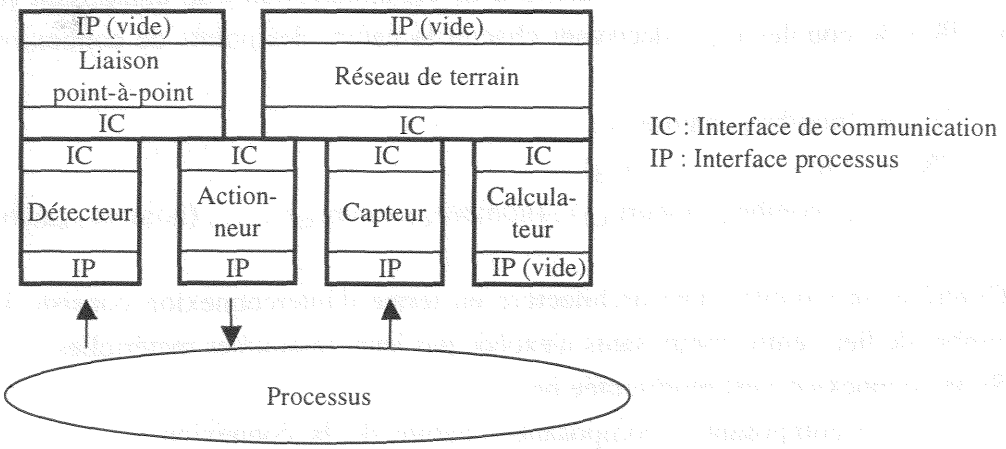


Figure II.10 : Exemple architecture matérielle utilisant la modélisation des composants en trois entités

Cet exemple d'architecture matérielle peut correspondre à un système chargé de maintenir un niveau d'eau donnée dans une cuve. Une première chaîne de commande est ainsi constituée d'un capteur de niveau qui transmet sa mesure au calculateur par l'intermédiaire du réseau de terrain. Ce calculateur établit un degré d'ouverture de vanne et transmet cette consigne à l'actionneur via à nouveau le réseau de terrain. Par ailleurs, un détecteur de niveau "trop haut" peut demander la fermeture de la vanne grâce à la liaison "point-à-point" et éviter

un éventuel débordement dû à la défaillance du calculateur, du capteur de niveau ou du réseau.

3.2.2 Evaluation et vérification des contraintes

Les paragraphes précédents exposent la manière dont les composants peuvent être modélisés. Mais la vérification du respect des contraintes mérite d'être mieux formalisée, principalement dans l'objectif de pouvoir être traitée de manière automatique et donc être informatisée.

Contrainte structurelle

Il s'agit de vérifier que l'organisation des composants et leurs interconnexions sont réalisables. Il s'agit donc de décrire les paramètres définissant l'interface de communication d'un composant.

Chaque point de connexion d'un composant est décrit par sa nature. Pour qu'un autre composant puisse y être connecté, il doit posséder un point de connexion de même nature. Actuellement, les types de connexion les plus courants sont des ports 0-10V, 4-20mA ou RS485 ; pour les bus de terrain ce sont des ports FIP, CAN, ASI...

Par ailleurs, un composant peut posséder plusieurs points de connexions de même nature. Il s'agit donc de préciser leur nombre. C'est particulièrement le cas des concentrateurs et des automates où plusieurs entrées et sorties sont présentes. Mais, il en va de même pour un bus de terrain (considéré comme un composant) où pour des raisons dues au protocole de communication ou à l'adaptation d'impédance un nombre limité de composants peut être raccordé.

Globalement, on peut définir l'interface de communication d'un composant (c) par un ensemble (IR_c) de couples ($ir_{c,j}$) décrivant chacun la nature des points de connexion et leur nombre.

$$ir_{c,j} = (nombre_{c,j}, nature_{c,j})$$

$$IR_c = \{ ir_{c,1} , ir_{c,2} , \dots , ir_{c,n} \}$$

$$= \{ (nombre_{c,1}, nature_{c,1}) , (nombre_{c,2}, nature_{c,2}) , \dots , (nombre_{c,n}, nature_{c,n}) \}$$

Contrôler la validité d'une architecture en terme d'interconnexion consiste à vérifier que le nombre de liens entre composants n'excède pas leurs ressources matérielles.

Si une connexion i est représentée par

$$c_i = (composant_{1,i} , composant_{2,i} , nature\_de\_la\_connexion_i)$$

et l'ensemble des connexions par l'ensemble C

$$C = \{ c_1 , c_2 , \dots , c_i , \dots , c_m \}$$

l'architecture est valide si :

$$\forall c \ \forall j \ \ nombre_{c,j} \leq card \{ c_i \in C / nature\_de\_la\_connexion_i = nature_{c,j} \text{ et } ((composant_{1,i}=j) \text{ ou } (composant_{2,i}=j)) \}$$

Ainsi on vérifie que l'installation ne requiert pas plus de connexions entre composants que ce qu'ils peuvent matériellement supporter.

Contrainte de charge

Quels que soient les composants employés dans le système, leurs capacités sont limitées. Ainsi chacun de ces composants ne peut supporter qu'un nombre fini de blocs exécutifs. Il s'agit donc de vérifier que les choix concernant l'affectation des blocs exécutifs respectent les contraintes physiques des composants. Deux types de contraintes sont présentées ci-après : celles relatives aux ressources et celles liées aux temps d'exécution.

- Charge relative aux ressources

Un type de contraintes à vérifier pour chaque composant est relative aux ressources requises par les blocs exécutifs. Suivant la nature de l'équipement (instrument intelligent, automate, calculateur), ces contraintes peuvent concerner le nombre d'entrées/sorties (cf. sous-chapitre précédent), la quantité de mémoire requise ou encore l'espace de stockage disponible.

Ainsi, on peut définir les ressources d'un composant (c) par un ensemble (RE_c) de couples ($re_{c,j}$) décrivant chaque ressource du composant par sa nature et sa capacité :

$$re_{c,j} = (nature_{c,j}, capacité_{c,j})$$

$$RE_c = \{ re_{c,1} , re_{c,2} , \dots , re_{,n} \}$$

$$= \{ (nature_{c,1}, capacité_{c,1}) , (nature_{c,2}, capacité_{c,2}) , \dots , (nature_{c,n}, capacité_{c,n}) \}$$

Les blocs exécutifs alloués à ce composant (c) est l'ensemble (BL_c) où chaque bloc exécutif ($bl_{c,i}$) est défini par l'ensemble des ressources (nature et quantité) nécessaires à son exécution.

$$bl_{c,i} = \{ (nature_{c,i,1}, quantité_{c,i,1}) , \dots , (nature_{c,i,m}, quantité_{c,i,m}) \}$$

$$BL_c = \{ bl_{c,1} , bl_{c,2} , \dots , bl_{c,n} \}$$

Vérifier que les capacités limites d'un composant (c) n'ont pas été dépassées, revient à vérifier les inéquations suivantes :

$$\forall j \quad capacité_{c,j} \leq \sum_{\substack{i,m \\ nature_{c,j}=nature_{c,i,m}}} quantité_{c,i,m}$$

En vérifiant pour tout le système que les ressources requises par les blocs exécutifs n'excèdent pas les possibilités de composants, on valide en partie l'architecture opérationnelle.

- Charge relative aux temps d'exécution

Certains blocs exécutifs (principalement ceux correspondant à un algorithme de traitement de données) doivent être exécutés périodiquement et nécessitent un certain temps de calcul. Une caractérisation possible des blocs [DEN 94] passe par l'emploi d'un couple formé de la fréquence d'exécution et du nombre d'instructions utilisés pour son exécution. Ces paramètres peuvent être adaptés pour certains types de composants. Ainsi, pour les réseaux et pour la transmission d'une donnée (équivalent à un bloc exécutif), on ne considère pas un nombre d'instructions mais la taille des informations à transmettre. Quoi qu'il en soit, la plupart des blocs exécutifs peuvent être définis par un couple fréquence/quantité.

Ce couple peut être souvent ramené à une valeur unique de charge de traitement grâce à un produit. Elle définit alors la charge de traitement induite par le bloc exécutif.

$$c_h(i) = f(i) \times q(i)$$

où $c_h(i)$ est la charge de traitement induit par le bloc exécutif i
$f(i)$ est la fréquence d'exécution du bloc i
$q(i)$ est la quantité de traitement requise par le bloc i

Pour beaucoup de composants, ne pas dépasser leur ressources de traitement revient à leur affecter un nombre de blocs tels que la somme des charges de traitement ne dépasse pas la capacité du composant qui les accueille. Cette limite est exprimée en nombre d'instructions par unité de temps (équivalent pour un réseau, à un débit utile, c'est-à-dire à la quantité d'informations transmises par unité de temps). Ceci est équivalent à la relation suivante :

$$l(c) \leq \Sigma\, c_h(i)$$

où i est un bloc exécutif affecté au composant c
$l(c)$ est la limite de la charge de traitement du composant c

L'installation n'est valide que si cette inéquation est vérifiée pour tous les composants.

Les automates constituent un cas particulier où tous les blocs exécutifs qui leur sont affectés, sont exécutés avec une même période. La période étant trop dépendante du nombre de blocs exécutifs affectés, seuls ceux dont la fréquence est une limite et n'est pas strictement imposée, peuvent être alloués à ce type de composant. C'est le cas de nombreux blocs utilisés dans les processus manufacturiers où un temps minimal de réaction est imposé. Ce n'est pas le cas pour les régulations où la période dite d'échantillonnage doit être strictement respectée (une exception est possible dans la mesure où les paramètres temporels de la régulation peuvent être adaptés à la fréquence du composant).

La validation des contraintes de charge est alors soumise aux inégalités suivantes :

$$f(i) \leq f_c(c)$$

où i est un des bloc exécutif affecté au composant c
$f_c(c)$ la fréquence d'exécution des traitements du composant c

avec $f_c(c) = F_c(\Sigma\, q(i))$

où F est une fonction fournissant la fréquence d'exécution du composant c selon la quantité de traitement à réaliser par cycle.

Différentes expressions de F_c sont disponibles dans la littérature [DEN 94].

L'emploi de réseau de terrain ayant une unique période de diffusion (ex. : AS-Interface) est un cas similaire et peut être étudié de la même façon.

Par la vérification de ces contraintes, on valide la capacité du système à accomplir dans les temps impartis les fonctions qui lui ont été assignées.

4 Conclusion

Dans ce chapitre, nous avons abordé différentes modélisations des systèmes d'automatisation utiles dans la démarche de conception. En effet, par raffinement, ils

participent à la démarche d'analyse du système à concevoir en enrichissant progressivement sa description.

Ainsi globalement, la conception revient tout d'abord à déterminer les fonctions que doit accomplir le système, puis par décompositions successives une organisation de fonctions élémentaires. La description de ces fonctions élémentaires peut alors être complétée en précisant leur fonctionnement (paramètres, séquencements, algorithmes...). Ensuite, une architecture matérielle peut être établie à partir des besoins des fonctions élémentaires. Finalement, ces fonctions (équivalentes à des blocs exécutifs) sont affectées aux composants de l'architecture matérielle.

Une activité de validation est effectuée parallèlement. Elle consiste en particulier, à vérifier que l'architecture matérielle supporte l'ensemble des fonctions qui lui est assigné.

Le chapitre suivant étudie un autre aspect de la conception qui consiste à évaluer les solutions envisagées et ce en intégrant le point de vue de la sûreté de fonctionnement.

5 Bibliographie

[ADE 79] ADEPA, Le GEMMA "guide des modes de marches et d'arrêts", 1979.

[AFN 88] AFNOR, Terminologie relative à la fiabilité - Maintenabilité – Disponibilité, X 60-500, norme expérimentale, 1988.

[BEN 95] Y. BENKHELLAT, J.P. THOMESSE, L'interopérabilité et sa validation dans les réseaux et les protocoles de communications, Electronic Journal on Networks and Distributed Processing, n°1, pages 7 à 25, 1995.

[BOI 91] S. BOIS, Intégration de la gestion des modes de marche dans le pilotage d'un système automatisé de production, 28 novembre 1991, Thèse de doctorat de l'Université des Sciences et Techniques de Lille Flandres Artois.

[BOU 97] A. BOURAS, Contribution à la conception d'architectures réparties : modèles génériques et interopérabilité d'instruments intelligents, Thèse de doctorat de l'Université des Sciences et Technologies de Lille, 1997.

[CAL 90] J.-P. CALVEZ, Spécification et conception des systèmes : une méthodologie, Paris, édition Masson, 1990.

[CHE 76] P.S. CHEN, The entity-relationship Model-Towards a unified view of data, ACM transactions on database systems, vol 3, n°1, 1976.

[CON 99] B. CONRARD, F. HERMANN, J.M. THIRIET, M. ROBERT, Vers une boîte à outils pour la conception de systèmes distribués, AIAI, p 20.9-20.11, Montréal (Canada), juin 99.

[DEL 89] J.L. DELCUVELLERIE, Ingénierie des systèmes d'automatisation de production, Thèse de doctorat de l'I.N.P.L. Université de Nancy I, mars 1989.

[DEN 94] B. DENIS, Assistance à la conception et à l'évaluation de l'architecture de conduite des systèmes de production complexes, Thèse de doctorat de l'Université de Nancy I, 1994.

[GEH 94] A.-L. GEHIN, Analyse fonctionnelle et modèle générique des capteurs intelligents : Application à la surveillance de l'anesthésie, Thèse de doctorat de l'Université des Sciences et Technologies de Lille, 1994.

[HAT 91] D. HATLEY, I. PIRBHAI, Stratégies de spécification des systèmes temps réel (SA-RT), traduit de l'américain [Strategies for real-time system specification], Paris, Edition Masson, 1991.

[HER 97] F. HERMANN, Contribution à la répartition des traitements et des données sur une architecture distribuée équipée d'un réseau de terrain FIP : Application à un processus thermique pilote, Thèse de doctorat de l'Université Henri Poincaré, Nancy I, novembre 1997.

[IGL 89] IGL Technology, SADT : Un langage pour communiquer, Paris, Edition Eyrolles, 1989.

[JAU 90] P. JAULENT, Génie logiciel : les méthodes SADT, SA, SA-RT, OOD, HOOD..., Edition Armand Colin, 1990.

[LUT 97] D. LUTTENBACHER, Modélisation du concept capteur intelligent par une approche orientée objet : Application à un capteur intelligent de température, Thèse de doctorat de l'Université Henri Poincaré, Nancy I, 1997.

[MES 95] M. BAYART, F. SIMONOT-LION, Projet MESR 2033 Convention 92-p-0239, Impact de l'émergence des réseaux de terrain et de l'instrumentation intelligente dans la conception des architectures des systèmes d'automatisation de processus, février 1995.

[PER 90] J. P. PEREZ, Systèmes temps réel: Méthodes de spécification et de conception, Paris, Editions Dunod, 1990.

[RIV 95] J.-M. RIVIERE, Contribution à la définition d'un modèle de référence de capteur intelligent : application à un capteur intelligent de température, Thèse de doctorat de l'Université Henri Poincaré Nancy I, 1995.

[RUM 91] J. RUMBAUGH et al., Object-Oriented modeling and design, Prentice-Hall International Editions, 1991.

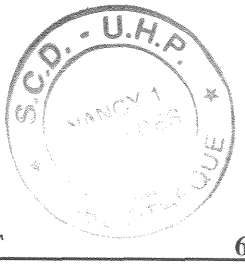
[RUM 95] J. RUMBAUGH et al., OMT : Modélisation et conception orientées objet, édition française revue et augmentée, Edition Masson et Prentice-Hall, 1995.

[STA 94] M. STAROSWIECKI, M. BAYART, Actionneurs intelligents, Hermès, Paris, 1994.

[TAR 87] H. TARDIEU et al., La méthode Merise tome 1, Edition d'Organisation, 1987.

[WAR 86] P. WARD, J. MELLOR, Structured development for real-time systems, 3 vol., Englewood Cliffs, Prentice-Hall, 1986.

Chapitre III
La Sûreté de Fonctionnement : Méthodes d'évaluation



Chapitre III
La Sûreté de Fonctionnement : Méthodes d'évaluation

Quels que soient les domaines concernés, la démarche des concepteurs (ou des analystes de la sûreté de fonctionnement) doit inclure les étapes suivantes : [SIG 96]

- Comprendre : - le fonctionnement de l'installation ;
  - la manière dont une panne survient ;
  - les effets d'une panne ;
- Calculer : - les probabilités d'occurrence des événements indésirables;
  - les conséquences des incidents ou des accidents;
- Fournir une aide à la décision par la détermination :
  - des points faibles ;
  - des actions présentant le meilleur rapport coût/efficacité à réduire les risques ;

De nombreuses méthodes ont été développées afin de répondre à ces problèmes. Elles s'appuient sur une modélisation des systèmes à étudier, c'est-à-dire une représentation simplifiée du réel utilisable par l'analyste. Par ailleurs, elles proposent une démarche pour établir le modèle correspondant au problème à traiter.

La variété des méthodes provient de l'hétérogénéité des domaines concernés, du mode de raisonnement utilisé (déductif ou inductif) et de la diversité des objectifs (recherche des points faibles, évaluation des risques...).

Parmi l'ensemble des techniques et méthodes, seules celles concernant les systèmes matériels sont présentées.

1 Méthodes d'analyse de la sûreté de fonctionnement

Ce sous-chapitre s'intéresse à l'étude des dysfonctionnements et recense les différentes techniques qui peuvent être employées. Comme on pourra le constater, il n'existe pas de méthode universelle pour étudier et qualifier la sûreté de fonctionnement d'un système. Chaque méthode ne vise qu'un objectif d'étude limité. Seule l'association de plusieurs de ces méthodes peut permettre d'obtenir une image relativement complète de la sûreté de fonctionnement. Les méthodes présentées ci-après sont classées selon les différents objectifs possibles des analyses de sûreté de fonctionnement.

1.1 En terme de conséquences

Le rôle le plus souvent dévolu aux analyses de sûreté de fonctionnement concerne l'étude des effets d'une (ou des) défaillance(s). Les méthodes employées visent ainsi, d'une part, à répertorier les défaillances possibles et, d'autre part, à identifier individuellement leur(s) conséquence(s) (méthodes déductives).

Méthodes qualitatives

Pour y parvenir les méthodes dites "analyse préliminaire des risques" (APR), "analyse des modes de défaillance, de leurs effets et de leur criticité" (AMDEC [AFN 86]) et "analyse opératoire des dangers" (HAZOP – Hazard and Operability Study [LER 92] [VIL 88]) proposent l'emploi de tableaux. En premier lieu, elles cherchent à inventorier les éléments dangereux (APR [LER 92] [VIL 88]), les modes de défaillances (AMDEC) ou les dérives de paramètres physiques par rapport à leur valeur nominale (HAZOP). L'analyste estime alors les conséquences possibles, juge de leur gravité et de leur criticité et propose des mesures préventives de détection ou de correction.

Bien que les résultats obtenus soient de nature très qualitative, la simplicité de mise en œuvre rend ces méthodes utiles pour une première approche.

Désignation de l'équipement	Fonction	Repère	Mode de défaillance	Cause de défaillance	Effet de défaillance		Détection de la défaillance	Dispositifs de remplace-ment	Probabilité de défaillance	Niveau de criticité	Remaques
					Effet local	Effet final					

Tableau III.1 : Tableau d'analyse des modes de défaillances, de leurs effets et de leur criticité [AFN 86]

Méthodes quantitatives

Etablir une mesure de la criticité, combinaison de gravité, fréquence d'apparition et probabilité de non-détection (proposée par l'AMDEC), peut cependant devenir une tâche ardue. C'est principalement le cas des systèmes où pour une même défaillance un grand nombre de mécanismes de protection est susceptible d'intervenir et/ou de multiples conséquences peuvent survenir. La méthode de l'arbre des conséquences (MACQ [VIL 88]) propose d'y remédier à l'aide d'un arbre de décision. Elle consiste à établir les conséquences faisant suite à un événement initiateur et selon les successions possibles des fonctionnements ou dysfonctionnements (généralement des fonctions de sécurité).

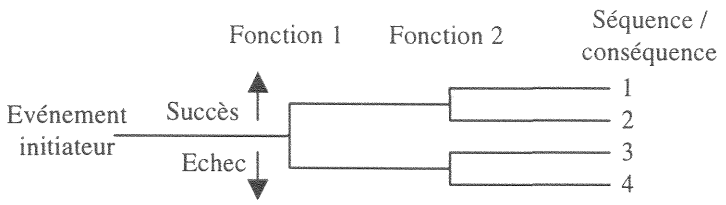


Figure III.1 : Exemple d'arbre des conséquences

L'arbre établi, l'identification des différentes conséquences et des chemins y aboutissant est relativement immédiate.

A l'aide d'un tel graphe, une extension vers une analyse quantitative est alors possible. Elle nécessite la connaissance du taux de succès et d'échec pour chaque fonction. Lorsque ces probabilités sont indépendantes, le calcul se résume à un simple produit. Il devient cependant beaucoup plus délicat en cas de fortes dépendances.

L'intérêt de cette méthode réside dans l'exploration exhaustive des séquences suite à une défaillance et dans le fait que la quantification soit possible.

1.2 En terme de causes

Dans un certain nombre de cas (particulièrement en ce qui concerne la sécurité), il peut être intéressant d'étudier un nombre limité de conséquences (généralement les plus critiques). Pour un événement redouté, identifié théoriquement à partir d'une première analyse (telle que APR), l'analyste a alors pour objectif de déduire les causes possibles de défaillance conduisant à l'événement redouté (méthodes inductives).

La méthode de l'arbre des défaillances (MAD [LIM 91]) répond à ce problème. Par la construction d'un arbre, on recherche les différentes causes ou combinaisons de causes conduisant à l'événement redouté et ce de manière itérative jusqu'à aboutir à des événements élémentaires dont le développement ne présente plus d'intérêt.

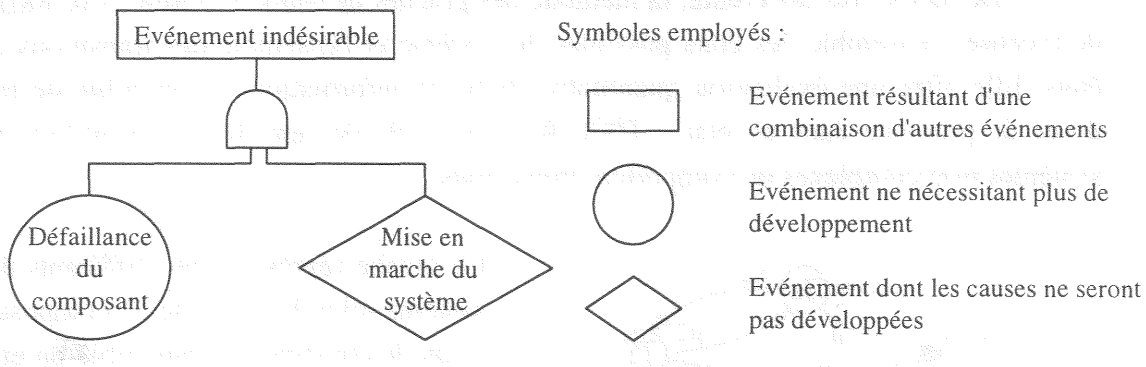


Figure III.2 : Exemple d'arbre des défaillances [VIL 88]

Adapté à une très large variété de système, cette méthode permet d'identifier les séquences susceptibles d'aboutir à l'événement indésirable (aspect qualitatif). Mais de plus, sur la connaissance des probabilités associées aux événements élémentaires, une évaluation de la probabilité d'occurrence de l'événement redouté est possible (aspect quantitatif).

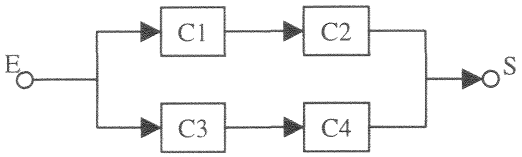
Une extension de cette méthode est la méthode du diagramme causes-conséquences (MDCC [LER 92] [VIL 88]). Elle propose d'associer arbres de défaillances et arbres des conséquences afin de représenter pour un événement à la fois les causes possibles et les conséquences éventuelles.

Les méthodes inductives sont peu adaptées au problème d'une évaluation globale de la sûreté de fonctionnement. Ceci s'explique par leur caractère inductif qui ne cherche pas à établir une vision exhaustive des conséquences possibles des défaillances.

1.3 En terme d'états

Favorisées par les outils mathématiques disponibles, d'autres approches proposent des représentations par graphes.

La méthode du diagramme de succès (MDS [VIL 88]) modélise le fonctionnement du système à l'aide de blocs correspondant à ses composants ou sous-systèmes. En établissant leurs liens, on décrit les différentes combinaisons permettant au système d'assurer sa mission.

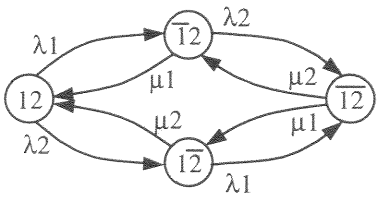


Ce diagramme représente un système de 4 composants (C1 à C4) où sa fonction n'est assurée que si les composants C1 et C2 ou C3 et C4 sont en état de fonctionner.

Figure III.3 : Exemple de diagramme de succès

Ce modèle permet relativement facilement le calcul de la disponibilité (dans la mesure où il y a indépendance entre les taux de panne des composants) ou de la fiabilité dans le cas de système non-réparable. Une transformation en arbre des conséquences (MACQ) est possible. Cette représentation est alors moins concise mais peut en faciliter la quantification.

De nature très différente, la méthode des graphes de Markov (GdM [VIL 88]) propose de recenser l'ensemble des états possibles du système et l'ensemble des transitions entre ces états. Elle offre une évaluation quantitative riche en informations (probabilité de transition, durée de présence dans un état... [PEL 92]) mais elle devient difficile à utiliser pour des systèmes trop complexes ou comportant trop d'états.



Ce graphe représente les différents états d'un système selon les états des 2 composants (1 et 2) qui le constituent (disponible ou en panne). λ et μ sont les taux de transition représentatifs du taux de défaillance et de celui de réparation pour chacun des composants.

Figure III.4 : Exemple de graphe de Markov

1.4 Relations entre les méthodes

La plupart de ces méthodes visent des objectifs différents et se limitent individuellement à un seul champ d'investigation. Effectuer une analyse de sûreté de fonctionnement requiert donc souvent l'emploi successif de plusieurs de ces méthodes. Le tableau suivant [BER 95] synthétise les relations entre ces méthodes.

Critères Méthodes	Type de démarche	Type d'évaluation	Entités de départ	Type de résultats	Formalisme utilisé	Liens entre les méthodes Amont	Aval
APR	inductive	qualitative	éléments dangereux	répertoire des situations dangereuses	tableaux	Analyse fonctionnelle	AMDE HAZOP
AMDEC	inductive	qualitative et semi-qualitative	mode de défaillance des composants	répertoire des modes de défaillance	tableaux et grilles	Analyse fonctionnelle et/ou APR	Méthodes ACQ, AD ou DCC
HAZOP	inductive	qualitative	paramètres mesurables du système	répertoire des déviations	tableaux	APR	MAD MDCC
MACQ	inductive	qualitative et semi-qualitative	événement initiateur	séquences d'événements inacceptables	arbre binaire	Analyse fonctionnelle , MDS	MAD GdM
MAD	déductive	mixte	événement redouté	coupes minimales, probabilité d'occurrence	arbre logique combinatoire	Méthodes inductives	GdM
MDCC	mixte	mixte	événement initiateur ou critique	séquences d'événement inacceptables, coupes minimales	arbre binaire, diagramme logique combinatoire	AMDE	/

Critères Méthodes	Type de démarche	Type d'évaluation	Entités de départ	Type de résultats	Formalisme utilisé	Liens entre les méthodes Amont	Aval
MDS	/	quantitative	fonctions ou composants	coupes minimales et paramètres de sûreté	diagrammes (blocs)	analyse fonctionnelle	/
GdM	déductive	quantitative	états de fonctionnement et de panne	paramètres de sûreté	graphes d'états	AMDE	événements pour MAD

Tableau III.2 : Relations entre les méthodes [BER 95]

2 Ateliers logiciels

Une part du succès des méthodes précédentes réside dans le fait qu'elles sont utilisables sans moyens de calculs importants. Les progrès de l'informatique associés à la complexité croissante des problèmes à traiter ont fait évoluer ces outils de base vers des ateliers logiciels.

2.1 Génération automatique

Une des premières limitations à la rédaction de tableaux ou de graphes, est leur taille susceptible d'être importante dès que les systèmes à étudier sont suffisamment grands. D'autres solutions basées sur des moyens informatiques (telle que FIABEX), ont été étudiées d'une part pour faciliter la saisie de la description du système, et d'autre part pour permettre la génération automatique des graphes et tableaux.

La plupart de ces logiciels s'inspirent des systèmes experts où le comportement (fonctionnel et dysfonctionnel) des composants est décrit sous forme de règles de production. Elles décrivent pour chaque composant, ses états, ses modes de défaillance, ses flux d'entrée ou de sortie... associés à des lois de propagation ou de transition. [SIG 96]

Exemple de loi de propagation :

SI état = fermé ALORS débit←nul

Exemple de loi de transition :

SI défaillance = cc ET état = fermé PENDANT 5 ALORS état←déclenché APRES 2

Par déduction à partir de ces lois, il est possible de générer automatiquement les tableaux AMDEC, des arbres de défaillance, des réseaux de Pétri. Mais l'utilisation des lois permet aussi directement d'effectuer la simulation de la propagation de défaillances au sein du système. Associées à d'autres outils de calcul, une quantification des risques de défaillance est possible.

2.2 Les méthodes de quantification analytique

L'importance du nombre de données à traiter a également une influence directe sur la quantification des résultats. L'emploi d'outils informatiques est alors indispensable pour mener ce type de travail.

L'utilisation des graphes et diagrammes destinés à une quantification s'appuie sur des outils mathématiques probabilistes. Il s'agit à partir des probabilités des états des entités ou d'occurrences d'événements de calculer les probabilités pour des associations d'états et/ou d'événements conduisant aux conséquences que l'on cherche à étudier.

Les opérateurs d'algèbre booléenne (+ et •) sont utilisés et P[X] représente la probabilité d'observer l'événement X.

P[A+B] = P[A] + P[B] – P[A•B]

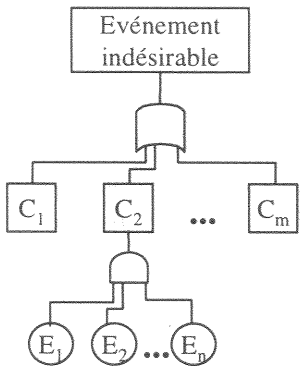
P[A•B] = P[A/B] . P[B] où A/B exprime la probabilité que A se produise si B s'est déjà produit

En cas d'indépendance de A et B :

P[A/B] = P[A]

D'où P[A•B] = P[A] . P[B]

L'application aux **arbres de défaillance** (MAC) s'effectue en établissant la relation booléenne entre l'événement redouté et les événements de base. Après réduction (simplification) de cette relation, l'emploi des relations précédentes permet d'évaluer la probabilité de l'événement final. L'informatisation de cette méthode algébrique est délicate. Une des techniques actuelles consiste à répertorier l'ensemble des coupes minimales (plus petite combinaison menant à l'événement), notés C_i, puis d'utiliser le théorème de Poincaré [VIL 88].



$$P[F] = P[C_1+C_2+\ldots+C_m]$$

$$= \sum_{i=1}^{m} P[C_i] - \sum_{i=2}^{m}\sum_{j=1}^{i-1} P[C_i \bullet C_j]$$

$$+ \sum_{i=3}^{m}\sum_{j=2}^{i-1}\sum_{k=1}^{j-1} P[C_i \bullet C_j \bullet C_k]$$

$$\ldots + (-1)^m P\left[\prod_{i=1}^{m} C_i\right]$$

Figure III.5 : Quantification des arbres de défaillance

L'application aux **arbres de conséquence** (MACQ) est plus immédiate. De par sa nature, les variables aléatoires sont pré-structurées et chaque événement final est associé à une coupe élémentaire. Le calcul est alors immédiat en effectuant le produit des probabilités des événements de base puis en sommant les probabilités des différents chemins menant au même événement. Ces arbres sont ainsi plus simples à informatiser. Cependant leur taille croissant de manière exponentielle avec le nombre de fonctions, les capacités nécessaires de mémoire et de calcul peuvent devenir vite très importantes voire dépasser les moyens actuels.

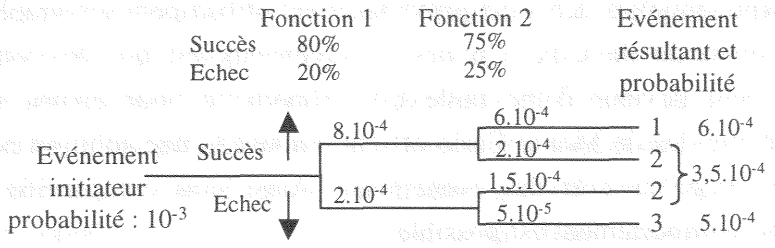


Figure III. 6 : Quantification des arbres de conséquence

L'application aux **diagrammes de succès** est plus délicate à automatiser. Ceci s'explique par le formalisme de graphe employé. Un pré-traitement est nécessaire afin d'identifier les différentes branches ayant une structure parallèle ou série. Cette phase effectuée, le calcul est relativement simple pour la fiabilité dans le cas de systèmes irréparables ou pour celui de la disponibilité. Pour les autres cas (ex : système avec des stratégies complexes de maintenance), le diagramme de succès n'est pas adapté à ce type d'analyse. Une autre façon d'effectuer la quantification, est de profiter de la transformation possible de ce graphe en arbre des conséquences.

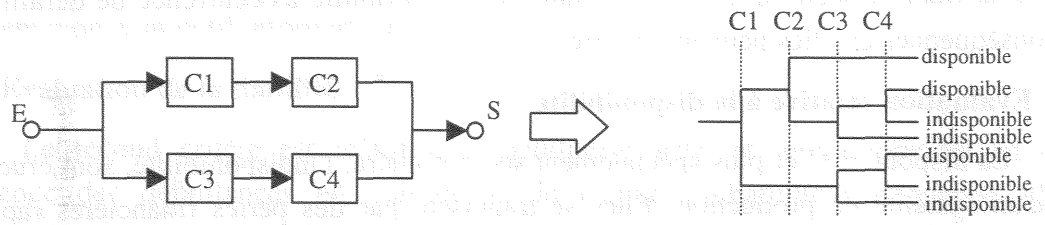


Figure III.7 : Transformation des diagrammes de succès en arbre de conséquence

L'application aux **graphes de Markov** est réalisée à l'aide de calculs matriciels. Ainsi, en utilisant une matrice composée des différents taux de transition entre les états, on peut établir pour un régime stationnaire les probabilités de présence dans chaque état, puis celles de transition entre ces états. L'utilisation de matrices peut requérir un nombre important de calculs dès que le nombre d'états devient important, mais peut être facilement informatisé. En revanche, l'étude du régime transitoire est beaucoup plus complexe et difficile à mettre en œuvre.

2.3 Les méthodes de quantification stochastique

Les précédentes méthodes analytiques souffrent souvent du nombre trop important d'opérations à réaliser dès que le système devient suffisamment grand. Une autre technique consiste à étudier le comportement du système grâce à un tirage au sort des événements. Cette méthode dite de "Monte Carlo", est applicable à beaucoup des méthodes précédentes. Elle permet même pour des systèmes de taille importante, d'aboutir à des résultats dont la précision ne dépend que du temps de calcul.

Une autre des difficultés rencontrées avec les méthodes précédentes réside dans le fait que seule la loi de défaillances exponentielle est traitée. Certaines techniques permettent cependant de transformer une loi complexe (ex : loi de Weibull) en une association de

plusieurs lois exponentielles. Ceci est principalement utilisé pour les graphes de Markov par l'emploi de la méthode de Cox. Un des inconvénients est que le système à étudier se complexifie et peut devenir d'une taille trop importante pour qu'une quantification soit réalisable. Les méthodes de Monte Carlo offrent à nouveau une solution par l'intégration des lois choisies au tirage au sort des événements. Ainsi sans complexifier le système, une simulation de son comportement est possible.

3 Application à l'évaluation d'un système d'automatisation

Toutes les méthodes précédentes sont relativement générales dans la mesure où elles peuvent s'appliquer à des domaines variés. De même, leurs objectifs sont très différents, et ne rendent compte que d'un aspect particulier pour l'évaluation de la sûreté de fonctionnement. Dans le cadre de l'étude d'un système d'automatisation, le but est de quantifier un ensemble de critères aptes à recouvrir les différents attributs de la sûreté de fonctionnement afin d'en donner une vision globale.

Deux attributs (transposables en critères) semblent importants à prendre en compte. Le premier est relatif à la disponibilité du système et traduit sa capacité à produire. Le second, relatif à la fiabilité/sécurité, est représentatif de la probabilité d'occurrence de défaillances et des conséquences qu'elles peuvent induire.

3.1 Evaluation relative à la disponibilité

La disponibilité et plus spécialement son corollaire, l'indisponibilité, sont cruciales au sein d'un système de production. Elles se traduisent par des pertes financières rapidement élevées dues à l'arrêt de production. La disponibilité qualifiant l'aptitude du système à assurer sa mission se quantifie généralement par un taux. Ce taux est le rapport entre le temps utile (pendant lequel il y a production effective) et le temps d'ouverture (pendant lequel on souhaiterait produire). Par l'estimation de ce taux et son application à une durée de production (de l'ordre de quelques années), on obtient une estimation de la production convertible en un coût.

Parmi les méthodes précédentes le diagramme de succès pourrait être un bon outil pour quantifier ces taux de disponibilité. La technique consiste alors à établir un premier diagramme où les traitements élémentaires sont représentés par les blocs. Ce premier diagramme décrit ainsi les relations entre les fonctions et est indépendant de la répartition choisie. De celui-ci, il est alors possible de tester différentes distributions des traitements. Il s'agit alors d'effectuer dans le diagramme la transposition de chaque traitement par l'équipement qui le supporte. On obtient un graphe où ne figurent que des équipements et pour lequel les méthodes de quantification de la disponibilité peuvent être appliquées.

Cependant, cette méthode ne convient pas totalement. En effet, on ne considère que deux états (disponible/indisponible) pour qualifier le système. Or dès que le système se complexifie, il est utile de considérer plusieurs modes de fonctionnement (liés en partie, aux défaillances susceptibles de se produire). Ils permettront de définir des fonctionnements dégradés pour lesquels même si le système n'accomplit pas sa mission de manière nominale, il

est capable de rendre une partie des services. Quelques cas typiques sont une quantité de production réduite ou une qualité des produits moindre, mais non destinés au rebut.

Enfin l'établissement du premier diagramme est mal adapté aux méthodes de conception. En effet, elles procèdent par décomposition successive des fonctions en sous-fonctions et ne permettent pas d'obtenir un graphe général des fonctions (ou traitements) élémentaires. Un travail de transposition délicat serait donc nécessaire pour transposer la décomposition fonctionnelle en un diagramme des fonctions.

Pour toutes ces raisons, l'emploi d'un arbre peut mieux permettre la représentation du système. Cet arbre est similaire à celui de la méthode de l'arbre des conséquences (MACQ), mais n'est pas associé à un événement initiateur. Il décrit selon l'état de chaque fonction (disponible ou indisponible), l'état du système. L'utilisation d'une telle représentation ne limite plus la modélisation du système à deux états, mais permet de considérer des modes dégradés. L'édition de cet arbre peut être faite de manière automatique en profitant de la décomposition fonctionnelle (exposé dans le chapitre suivant). Enfin des outils algorithmiques existent (présentés en fin de ce chapitre) et permettent une relativement bonne intégration de cette représentation à un outil informatique.

3.2 Evaluation de la fiabilité

Le second critère est relatif à la fiabilité/sécurité, et vise à prendre en compte l'influence des défaillances en considérant le couple probabilité/conséquence. Il s'agit d'estimer l'ensemble des défaillances susceptibles de compromettre la mission du système et de les qualifier par leur gravité et leur probabilité d'occurrence. Le premier paramètre de gravité dépend de la nature de l'événement considéré et est indépendant de la répartition. Il doit donc être évalué de manière indépendante par le concepteur. En revanche, le second paramètre relatif à la probabilité d'occurrence est fortement lié au choix de la répartition et nécessite un calcul pour chaque configuration. La transposition de ces informations vers un critère d'évaluation du système, s'effectue en considérant une période de fonctionnement. En quantifiant la gravité par un coût, il est alors possible grâce à la probabilité d'occurrence de l'événement d'évaluer son coût sur la durée de vie du système.

La méthode de l'arbre des causes (MAC) semblerait correspondre à ce besoin de quantifier les événements indésirables susceptibles d'affecter la mission du système. Ainsi pour chacun de ces événements, il faut établir l'arbre définissant son occurrence selon les défaillances des fonctions élémentaires. Puis, la quantification est effectuée après avoir associé chaque fonction à un composant dont la probabilité d'occurrence de chacun de ses modes de défaillance est connue.

Cependant, comme précédemment pour la disponibilité, l'établissement de ces arbres s'adapte mal à la décomposition fonctionnelle dressée pendant la conception.

Une autre façon de faire est l'utilisation de la méthode de l'arbre des conséquences (MACQ). Cela revient à établir pour chaque mode de défaillance (considéré comme

l'événement initiateur), un arbre décrivant les conséquences possibles pour le système selon l'état des autres fonctions. Une des difficultés réside dans le nombre d'arbres à traiter (un par mode de défaillances). En revanche, cette méthode est homogène avec l'étude précédente de disponibilité, par les outils algorithmiques utilisés. De plus, elle peut s'appuyer sur la décomposition fonctionnelle (exposée dans le chapitre suivant).

4 Outils algorithmiques de traitements des arbres

Comme on peut le constater au travers des méthodes présentées précédemment, les arbres sont très utilisés pour décrire le comportement d'un système en présence de défaillances. Si l'établissement de ces arbres est relativement simple, leur quantification l'est beaucoup moins particulièrement lorsqu'ils ont une taille importante. Il s'agit d'évaluer la probabilité d'occurrence d'événements à partir de celles connues des modes de défaillance des composants.

4.1 Présentation du diagramme de décision binaire

Les diagrammes de décision binaire (BDD pour Binary Decision Diagrams) offrent une solution au traitement de ces arbres. Apparues en 1994 [COU 94] dans le domaine de la sûreté de fonctionnement, ils permettent de traiter de grandes fonctions booléennes. Ce sous-chapitre expose les principaux principes de cette technique.

La décomposition de Shannon est la base mathématique pour l'utilisation des arbres booléens [SHA 49]. Pour une fonction $F(x_1,\ldots,x_n)$ dont les arguments x_i sont binaires, la décomposition est la suivante :

$$F(x_1,\ldots,x_i,\ldots,x_n) = \neg x_i \,.\, F(x_1,\ldots,0,\ldots,x_n) + x_i \,.\, F(x_1,\ldots,1,\ldots,x_n)$$

En réitérant cette démarche, on obtient une fonction comportant 2^n nœuds. Cette valeur peut ainsi devenir très grande et les diagrammes de décision binaire proposent une simplification apte à éviter cette explosion.

Le diagramme de décision binaire d'une fonction F, associé à l'ordre des variables σ est la représentation de l'arbre de Shannon de F, associé à l'ordre σ, par l'application des règles suivantes :

- Les sous-arbres identiques ne sont figurés qu'une seule fois,
- Les branches d'un nœud conduisant à un même sous-arbre sont confondues.

L'exemple suivant représente le cas de 4 machines dont 2 en bon état suffisent pour que le système soit disponible. La fonction F associée est donc :

$$F(x_1, x_2, x_3, x_4) = x_1.(x_2 + x_3 + x_4) + x_2.(x_3 + x_4) + x_3.x_4$$

Ses décompositions sous forme d'arbre sont les suivantes :

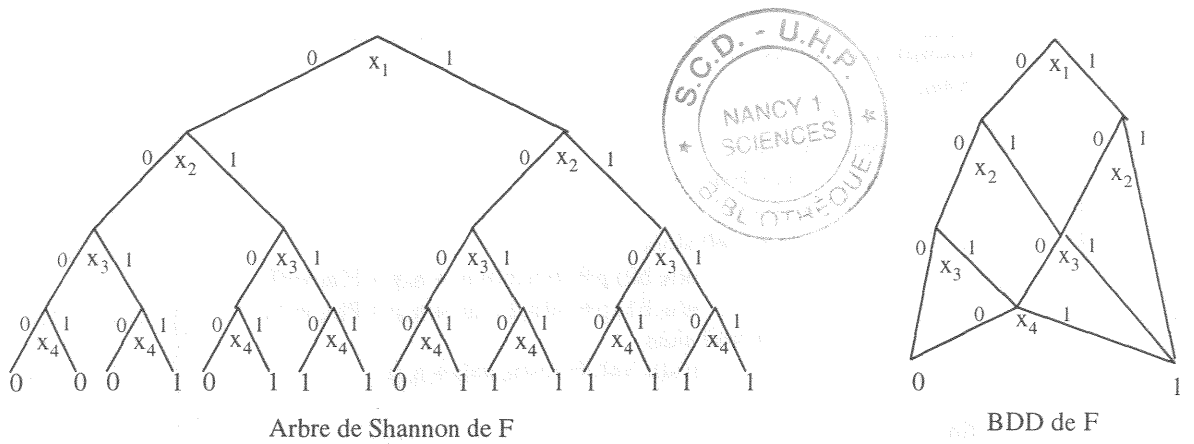


Figure III.8 : Représentation de la fonction F sous forme d'arbre

On constate par le nombre de nœuds l'économie que les diagramme de décision peuvent apporter. Ainsi, dans l'exemple précédent, le diagramme de décision comporte 6 nœuds au lieu de 15 pour la décomposition de Shannon.

Cette réduction se traduit directement par un gain de mémoire lors de l'informatisation du traitement de ce type d'arbre.

4.2 Quantification des arbres et des diagrammes

La quantification de ces arbres consiste à évaluer les probabilités des valeurs de F selon celles de ces variables.

Concernant la sûreté de fonctionnement, la représentation des variables peut être variée. La probabilité associée peut être un taux de disponibilité, une probabilité relative à l'exécution correcte d'un service... Quoi il en soit, il s'agit d'un taux exprimable dans l'intervalle [0,1].

L'algorithme pour l'évaluation de F, fonctionne selon le principe suivant. Il s'agit d'établir pour chaque nœud la probabilité que les valeurs des variables mènent à ce nœud. Ceci ne peut être fait que par une démarche descendante (du sommet vers les branches). Ainsi en connaissant la probabilité de présence dans un nœud et la probabilité de la variable associée à ce nœud, on peut calculer (par un simple produit) la probabilité d'emprunter chaque branche et par conséquent la probabilité de présence dans les nœuds de destination de ces branches. Le premier nœud a donc une probabilité de 1, les deux nœuds du second niveau de $P[x_1=1]$ et $P[X_1=0]$, puis pour ceux du troisième niveau $P[x_1=0].P[x_2=0]$, $P[x_1=0].P[x_2=1]$, $P[x_1=1].P[x_2=0]$ et $P[x_1=1].P[x_2=1]$, ainsi de suite jusqu'à atteindre les nœuds terminaux.

L'algorithme est le suivant :

Chaque nœud n_i (ou noté n(i))est défini par les champs suivants :

- v : numéro d'ordre de variable aléatoire associée au nœud, égal à 0 s'il s'agit d'un nœud terminal,
- b_0,b_1 : nœud de destination lorsque la variable v vaut 0, respectivement 1,
- p : probabilité de présence dans ce nœud,
- val : valeur (0 ou 1) du nœud terminal signalé par v égal à 0

L'arbre contient m nœuds.

```
fonction quantifie(n)
début
        n1.p←1
        pour i de 2 à m
                ni.p←0
        pour i de 1 à m
                si ni.v≠0 alors
                        n(ni.b0).p← n(ni.b0).p  + ni.p × P[ni.v=0]
                        n(ni.b1).p← n(ni.b1).p  + ni.p × P[ni.v=1]
                si ni.v=0 alors
                        res(ni.val)← res(ni.val) + ni.p
        retourne res
fin
```

Cet algorithme nécessite que l'ordre des nœuds soit descendant par rapport à l'arbre. Les résultats obtenus sont les probabilités des valeurs de la fonction, c'est-à-dire res(valeur) .

L'application de ceci à l'exemple précédent des 4 machines donne les résultats suivants. On considère que les disponibilités respectives des machines sont 0,9, 0,8, 0,7 et 0,6.

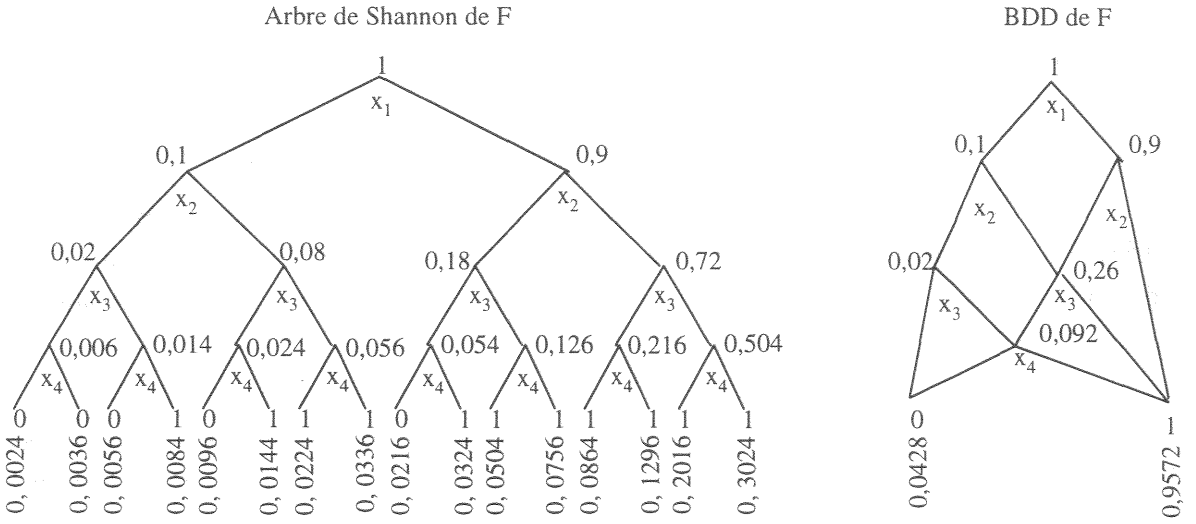


Figure III.9 : Quantification des arbres

Les résultats finaux sont P[F=1]=0,9572 et P[F=0]=0,0428.

A nouveau, les diagrammes de décision montrent leur efficacité en réduisant le nombre d'opérations arithmétiques. Le tableau suivant fournit le nombre d'additions et de multiplications flottantes utilisés par chaque représentation.

	Nombre d'additions	Nombre de multiplications
Arbre de Shannon	14	30
Diagramme de décision	5	12

Tableau III.3 : Comparaison entre un arbre de Shannon et un diagramme de décision

Outre l'évaluation de la disponibilité, ce type d'arbres peut être utilisé pour le calcul des probabilités d'occurrences. Il s'agit alors de l'application directe des arbres de la méthode de l'arbre des conséquences. A partir d'un événement initiateur dont on connaît la probabilité d'occurrence, on évalue celle de défaillance du système, selon les états possibles de ce système exprimés par l'arbre.

4.3 Opération entre graphes

Un autre intérêt des diagrammes de décision réside dans la possibilité d'effectuer des opérations entre arbres. Pour deux diagrammes, il est alors nécessaire que leurs variables soient ordonnées selon le même ordre.

L'algorithme suivant (inspiré de [BRY 92]) applique un opérateur op (tel qu'un "et" ou un "ou" logique) à deux diagrammes dont les nœuds aux sommet sont n1et n2. La déclaration des nœuds est la même que celle employée en 4.2.

```
fonction applique(op,n1,n2)
début
        si ((n1.v=0) et (n2.v=0)) alors
                res← nouveau_nœud_terminal( op( n1.val, n2.val ) )
        si ( ((n1.v=0) ou (n1.v>n2.v)) et (n2.v≠0) ) alors
                res← nouveau_nœud( n2.v,
                        applique(op,n1,n2.b0), applique(op,n1,n2.b1) )
        si ((n1.v≠0) et ((n2.v=0) ou (n2.v>n1.v)) ) alors
                res← nouveau_nœud( n1.v,
                        applique(op,n1.b0,n2), applique(op,n1.b1,n2) )
        si ((n1.v≠0) et (n2.v=n1.v)) alors
                res← nouveau_nœud( n1.v,
                        applique(op,n1.b0,n2.b0), applique(op,n1.b1,n2.b1) )
        retourne res
fin
```

Les fonctions nouveau_nœud_terminal(v,b0,b1) et nouveau_nœud(val) recherchent si le nœud que l'on veut créer n'existe pas déjà. Dans un tel cas elles retournent son numéro. Dans le cas contraire, elles créent ce nouveau nœud et retournent son numéro.

Cet algorithme peut être optimisé car il est susceptible de traiter plusieurs fois une même branche. Il suffit alors de mémoriser les nœuds déjà traités et en début de la fonction application(op,n1,n2) de contrôler que le nœud en cours n'a pas déjà été traité.

4.4 Ordre des variables

Il n'existe qu'un seul diagramme de décision pour une fonction donnée et pour un ordre des variables donné. Cependant cet ordre influence directement la taille du diagramme. Ainsi pour optimiser la mémoire et le temps de traitement, il est nécessaire de trouver un ordre susceptible de minimiser la taille des diagrammes.

A titre d'illustration, l'exemple suivant traite de la construction du diagramme permettant d'évaluer la disponibilité d'un système comprenant n éléments redondés chacun m fois. Seuls les états disponibles et indisponibles sont considérés. Selon l'ordre employé, N est le nombre de nœuds du diagramme fournissant l'état du système selon celui de ses entités.

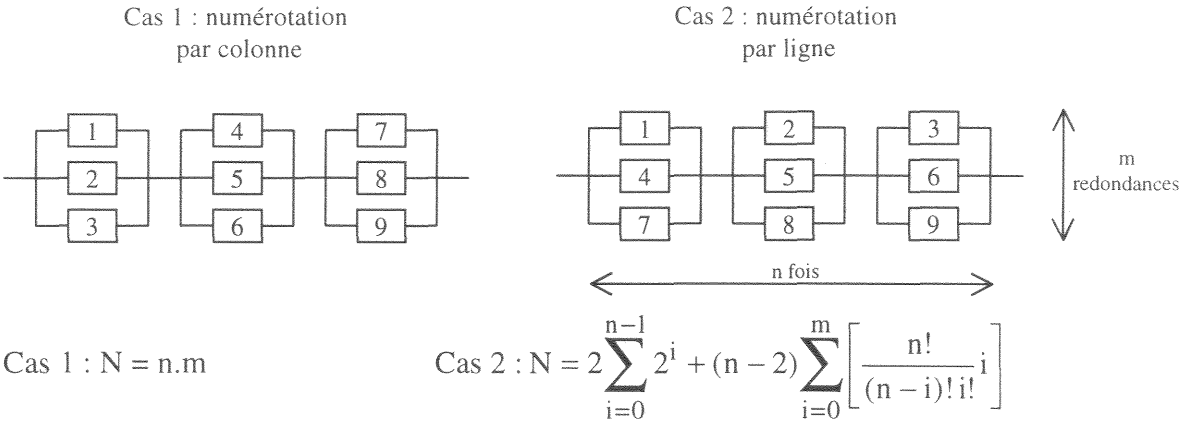


Figure III.10 : nombres de nœuds d'un diagramme selon l'ordre des variables

Le tableau suivant fournit pour quelques valeurs de n et de m, le nombre de nœuds que comprend le diagramme selon les deux ordres de numérotation envisagés.

n	m	Cas 1	Cas 2
1	1	1	1
2	2	4	6
3	3	9	26
4	4	16	94
5	5	25	302
6	6	36	894
7	7	49	2494
8	8	64	6654

Tableau III.4 : Nombre de nœuds pour différentes valeurs de n et m

Apparemment, il n'existe pas d'heuristiques capables d'ordonner les variables de manière optimale. Cependant la procédure "recherche en profondeur –à gauche– d'abord" [CHE 97] permet d'obtenir des diagrammes relativement concis.

Lorsque le diagramme est établi à partir d'une décomposition (ex : arbre des défaillances), il s'agit d'indexer les variables dans l'ordre où elles apparaissent dans la décomposition. On profite ainsi d'un pré-ordonnancement réalisé lors de la décomposition. Grâce à la façon dont elle est établie, elle regroupe indirectement les variables ayant le plus d'interactions entre elles, et limite ainsi la taille du diagramme.

4.5 Extension à des variables non-binaires

L'utilisation des diagrammes de décision peut être étendue à des variables non-binaires. Si pour l'évaluation d'états du type "disponible"/"indisponible", l'emploi de variables booléennes est suffisant, il ne l'est plus lorsque l'on s'intéresse à un nombre plus important de modes. Considérer des modes dégradés en est un exemple.

Cette extension peut s'appliquer aussi bien aux variables et/ou qu'aux fonctions considérées. Les domaines doivent cependant être des ensembles finis pour que les méthodes précédentes (quantification, opération) soient toujours applicables.

5 Conclusion

De nombreuses méthodes ont été élaborées pour fournir aux ingénieurs des outils leur permettant d'étudier la sûreté de fonctionnement avec rigueur. Cependant, les objectifs visés et les résultats obtenus par ces méthodes ne concerne qu'un aspect limité de la sûreté mais différent pour chacune d'elles. Or l'estimation de la robustesse globale des solutions d'architecture envisagées ne peut être obtenue qu'à partir d'un ensemble de critères dont seule une association de ces méthodes permettrait leur évaluation.

Par ailleurs, on retiendra que si l'on ne considère que les aspects relatifs à la disponibilité et à la fiabilité comme étant les deux principaux critères d'évaluation, un même outil mathématique peut être utilisé. Il s'agit des diagrammes de décision qui sont susceptibles de traiter des arbres de taille relativement importante, et dont l'informatisation est relativement économe en ressource mémoire ou de traitement.

Le chapitre suivant propose une démarche de conception intégrant ces outils.

6 Références bibliographies

[AFN 86] AFNOR, Techniques d'analyse de la fiabilité des systèmes, Procédures d'analyse des modes de défaillance et de leurs effets (AMDE), X 60-510, fascicule de documentation, 1986.

[BER 95] M. BERGOT, L. GRUDZIEN, Sûreté et diagnostic des systèmes industriels. Principaux concepts, méthodes, techniques et outils, Collection Diagnostic et sûreté de fonctionnement, Vol n°5, n°3, pages 317 à 344, Hermes, 1995.

[BRY 92] R.E. BRYANT, Symbolic boolean manipulation with ordered binary-decision diagrams, ACM Computing Surveys, n°24, vol 3, pages 293 à 318, 1992.

[CHE 97] M. CHEVALIER, et al., "Arbres de défaillances et diagrammes de décision binaires", actes du congrès Qualité et sûreté de fonctionnement, Compiègne , tome 2, page 47 à 56, 1994.

[COU 94] O. COUBERT, J.C. MADRE, MetaPRIME: an iteractive fault tree analyser, IEEE Transactions on Reliability, n°43, vol 1, pages 121 à 127, 1994.

[LER 92] A. LEROY, J.-P. SIGNORET, Le risque technologique, Collection Que Sais-Je ?, PUF, 1992.

[LIM 91] N. LIMNIOS, Arbres de défaillances, Collection Traité des Nouvelles Technologies, Série Diagnostic et Maintenance, Hermès, 1991.

[PEL 92] J. PELLAUMAIL, Graphes simulation L-matrices : Applications aux files d'attente, Collection Traité des Nouvelles Technologies, Série Mathématiques appliquées, Hermès, 1992.

[SHA 49] C. E. SHANNON, The synthesis of two-terminal switching circuits, Bell Syst. Tech. J., 28, page 59 à 98, 1949.

[SIG 96] SIGNORET J.-P., Modélisation et simulation dans le domaine de la Sûreté de Fonctionnement, Revue REE, n°8, pages 40 à 46, 1996.

[VIL 88] A. VILLEMEUR, Sûreté de fonctionnement des systèmes industriels, Edition Eyrolles, Paris, 1988.

Chapitre IV
Intégration de la Sûreté de Fonctionnement en phase de conception des Systèmes d'Automatisation

Chapitre IV
Intégration de la Sûreté de Fonctionnement en phase de conception des Systèmes d'Automatisation

L'objectif de la méthode proposée est d'enrichir les modèles utilisés pour la conception des systèmes d'automatisation par des données permettant l'étude de la sûreté de fonctionnement, ceci afin de permettre l'obtention d'une architecture distribuée plus sûre. Cette démarche doit ainsi permettre l'évaluation probabiliste de la capacité du système à accomplir sa fonction en présence de défaillances et son comportement lors de leur occurrence.

1 Enrichissement de l'architecture fonctionnelle par des informations de SdF

L'architecture fonctionnelle, telle qu'elle a été présentée précédemment, définit les fonctions du système et leur organisation. Ces fonctions sont architecturées en une décomposition hiérarchique où les fonctions complexes sont réalisées par l'association de sous-fonctions. Le critère d'arrêt de cette décomposition est l'obtention de fonctions suffisamment élémentaires pour qu'une décomposition plus poussée n'ait plus d'intérêt pour la répartition. L'objectif est de profiter de cette structure afin d'y intégrer les informations relatives à la sûreté de fonctionnement.

1.1 Aspects relatifs à la disponibilité

Comme mentionné précédemment, un des éléments de l'analyse de la sûreté de fonctionnement concerne l'étude de la disponibilité du système. De manière synthétique, il s'agit de déterminer la part du temps pendant lequel l'installation est en état d'accomplir sa mission. Cette valeur dépend essentiellement de la disponibilité propre des composants du système et de leur organisation, principalement en terme de redondances [CON 97].

L'étude de la disponibilité revient donc à estimer la répartition probabiliste des différents états du système. Ces états décrivent la capacité du système à accomplir sa mission et sont généralement de la forme "disponible" (la mission peut être assurée), "indisponible" (elle ne l'est pas). Comme il a été dit précédemment, ils peuvent être enrichis par des états de type "disponible dégradé" pendant lesquels la mission peut être réalisée mais de manière partielle (ex : production moindre).

Bien entendu, en vue de pouvoir participer aux critères d'évaluation, chaque état doit pouvoir être associé à un coût financier. Ces coûts sont globalement une projection de la plus-value que l'utilisateur espère tirer de l'installation. Il est normalement positif pour un état disponible, négatif ou nul lorsqu'il est indisponible.

Pour les fonctions élémentaires (correspondant aux traitements à répartir) qui sont directement associées à des composants matériels, la même notion d'état peut être utilisée pour caractériser la manière dont le service est rendu ou non. Bien évidemment, l'état de la fonction est lié à celui du composant qui la supporte. Ainsi, si le composant est "en bon état"

(respectivement "en panne"), la fonction est "disponible" (respectivement "indisponible"). L'intégration de la sûreté de fonctionnement consiste donc pour ces fonctions élémentaires à établir la relation liant l'état de chaque composant à celui de chaque fonction qu'il supporte. La représentation la plus simple pour exprimer cette relation est l'emploi d'un tableau comme dans l'exemple suivant.

Composant "capteur"	Fonction "mesurer"
En bon état	Disponible
Corps d'épreuve défaillant (biaisé, bruité)	Indisponible, Mesure fausse
En panne complète	Indisponible

Tableau IV.1a : Tableau de correspondance des états entre une fonction "mesurer" et un capteur

Composant "vanne tout ou rien"	Fonction "contrôler débit"
En bon état	Disponible
En panne, bloquée ouverte	Indisponible, ouvert
En panne, bloquée fermée	Indisponible, fermé

Tableau IV.1b : Tableau de correspondance des états entre une fonction "contrôler débit" et une vanne tout ou rien

La distinction de plusieurs états indisponibles peut avoir un intérêt pour la détection des fonctionnements non-conformes. Ainsi, dans l'exemple précédent, l'état "Indisponible, Mesure fausse" peut servir à identifier le cas où l'installation utiliserait des mesures fausses, et ceci en l'absence d'une fonction de validation de la mesure.

Pour les fonctions hiérarchiquement supérieures, la même notion d'états (de "disponibilité") peut être utilisée. Cependant, ces fonctions sont l'association de fonctions élémentaires, et ne sont donc pas supportées directement par des composants matériels. Leurs états dépendent alors uniquement de celui de leurs sous-fonctions. L'intégration de la sûreté de fonctionnement consiste alors à définir la relation qui lie ces états. L'emploi d'une table de vérité est un moyen de représenter cette relation.

Fonction : "Mesurer"		Sous-fonction 2 : "Mesurer II"		
		Disponible	Indisponible, Mesure fausse	Indisponible
Sous-fonction 1 : "Mesurer I"	Disponible	Disponible	Disponible	Disponible
	Indisponible, Mesure fausse	Disponible	Indisponible	Indisponible
	Indisponible	Disponible	Indisponible	Indisponible

Tableau IV.2a : Table de vérité des états pour une fonction "Mesurer" composée de deux fonctions redondantes "Mesurer I" et "Mesurer II"

Fonction : "Contrôler débit"		Sous-fonction 2 : "Contrôler débit II"		
		Disponible	Indisponible, ouvert	Indisponible, fermée
Sous-fonction 1 : "Contrôler débit I"	Disponible	Disponible	Disponible	Indisponible, fermé
	Indisponible, ouvert	Disponible	Indisponible, ouvert	Indisponible, fermé
	Indisponible, fermé	Indisponible, fermé	Indisponible, fermé	Indisponible, fermé

Tableau IV.2b : Table de vérité des états pour une fonction "Contrôler débit" composée de deux fonctions redondantes "Contrôler débit I" et " Contrôler débit II" correspondant à deux vannes montées en série

Cette représentation permet de définir facilement des formes de redondance. Ainsi dans le premier exemple la disponibilité d'au moins une sous-fonction "Mesurer x" rend disponible la fonction "Mesurer". Dans le second, le contrôle du débit dans une canalisation avec deux vannes montées en série, dépend de leurs modes de défaillance.

D'un point de vue pratique, utiliser une démarche ascendante pour l'établissement de ces relations entre fonctions et sous-fonctions semble préférable. En effet, d'une part, les états des fonctions élémentaires peuvent être facilement déduits des propriétés des composants matériels. D'autre part, concernant les fonctions hiérarchiquement supérieures, il est plus simple d'établir la relation avec ses ressources si leurs différents états sont complètement recensés.

1.2 Aspects relatifs à la fiabilité

La disponibilité ne fournit qu'un premier élément de l'évaluation de la sûreté de fonctionnement. Celle-ci doit être complétée par l'étude du comportement de l'installation lors de l'occurrence d'une défaillance. L'objectif est alors double ; d'une part, il s'agit d'identifier les événements relatifs à la mission du système et à l'altération de sa réalisation. D'autre part, leur probabilité d'occurrence ou leur nombre estimatif d'apparitions pour la durée de vie du système doit être déterminé. Ces événements ne sont que la résultante de défaillances des composants dont les effets sur la mission n'ont pas pu être corrigés par le système.

On suppose, par ailleurs, que les conséquences de chacun de ces événements peuvent être quantifiées. En effet, en vue de l'intégration de la fiabilité comme un critère d'évaluation, le passage à une métrique relative à la rentabilité nécessite de connaître en plus de la probabilité d'occurrence, une grandeur liée aux conséquences et aux préjudices subis par l'apparition de l'événement.

Comme précédemment concernant la disponibilité, il s'agit de tirer parti de la décomposition fonctionnelle. Globalement, on va ainsi chercher à définir la manière dont une défaillance va se propager au sein de la hiérarchie.

Chaque fonction élémentaire est directement associée à un composant. L'occurrence d'une défaillance sur celui-ci influe donc directement sur la capacité de la fonction à rendre son service. En utilisant la notion d'événement, l'adjonction d'informations relatives à la sûreté de fonctionnement consiste donc à définir la propagation de l'événement. Il s'agit ainsi, d'associer à chaque mode de défaillance du composant, l'événement correspondant concernant la délivrance du service des fonctions qu'il supporte. Comme précédemment, l'emploi d'un tableau permet la représentation de cette relation.

Composant "capteur" : Mode de défaillance	Fonction "mesurer" : Evénement associé
Altération du corps d'épreuve (biais, bruit)	Mesure faussée
Arrêt du capteur	Mesure interrompue

Tableau IV.3a : Tableau de correspondance des événements entre une fonction "mesurer" et un capteur

Composant "vanne tout ou rien" : Mode de défaillance	Fonction "contrôler débit" : Evénement associé
Blocage en position ouverte	Arrêt ouvert
Blocage en position fermée	Arrêt fermé

Tableau IV.3b : Tableau de correspondance des événements entre une fonction "contrôler débit" et une vanne tout ou rien

On notera que la plupart des événements peuvent être associés à un changement d'état. Cependant, l'utilisation d'événements indépendants des états permet de définir des fautes fugitives (et donc non associées à un changement d'état) ou de distinguer plusieurs transitions entre les mêmes états (mais dont les conséquences sont différentes en ce qui concerne le système). A titre d'exemple, ce peut être pour un réseau la perte d'un message ; malgré cette défaillance, la fonction "transmettre une information" reste disponible. De même, pour une vanne, deux événements (blocage ouvert, blocage fermé) sont la transition d'un état disponible à indisponible, mais les effets peuvent être différents.

Concernant les fonctions hiérarchiquement supérieures qui ne sont que l'association de fonctions plus élémentaires, la même notion d'événement est utilisée. Pour une fonction donnée, il s'agit de traduire l'effet de l'apparition d'un événement sur une de ses sous-fonctions par un autre événement. Ainsi, on établit une relation entre les événements susceptibles d'affecter une fonction et ceux de ses sous-fonctions. Avec une telle méthode, on définit globalement la propagation des événements du composant jusqu'à la fonction principale (ou mission) du système.

Cependant, la propagation d'un événement à travers une fonction est liée à son état. Selon les sous-fonctions disponibles (présence ou non de mécanisme de sécurité, de redondance...) les conséquences et donc la nature des événements induits diffèrent. Par

ailleurs, tous les événements ne génèrent pas forcément un événement correspondant au niveau supérieur (typiquement le cas où une redondance pallie la défaillance d'une des ressources). Dans un tel cas (généralement favorable), la propagation de l'événement initiateur est arrêtée et n'atteint pas la mission principale du système. Globalement cela revient à associer à chaque cas de la table d'état une table de correspondance relative à la propagation des événements.

Fonction : "Contrôler débit"		Sous-fonction 2 : "Contrôler débit II"		
		Disponible	Indisponible, ouvert	Indisponible, fermée
Sous-fonction 1 : "Contrôler débit I"	Disponible	Disponible ①	Disponible ②	Indisponible, Fermé
	Indisponible, ouvert	Disponible ③	Indisponible, ouvert	Indisponible, fermé
	Indisponible, fermé	Indisponible, fermé	Indisponible, fermé	Indisponible, fermé

	Evénement initiateur		Evénement résultant sur la fonctions globale "contrôler débit"
① : ss-f. 1:	Arrêt fermé	→	Arrêt fermé
ss-f. 2:	Arrêt fermé	→	Arrêt fermé
② : ss-f. 1:	Arrêt ouvert	→	Arrêt ouvert
	Arrêt fermé	→	Arrêt fermé
③ : ss-f. 2:	Arrêt ouvert	→	Arrêt ouvert
	Arrêt fermé	→	Arrêt fermé

Tableau IV. 4 : Table de correspondance entre les événements générés par "Contrôler débit" en fonction de ceux provenant de ses sous-fonctions.

D'un point de vue pratique, comme pour la disponibilité et pour les mêmes raisons, utiliser une démarche ascendante pour définir ces relations est préférable. En effet, les événements sont facilement déduits des modes de défaillance des composants matériels. De plus, l'identification des événements pour les fonctions nécessite de connaître ceux de leurs sous-fonctions.

Lors de l'établissement des tables de correspondance, il faut parfois tenir compte d'événements simultanés. En effet, plusieurs événements peuvent apparaître conjointement pour une même fonction. En particulier, ce cas se présente lorsque plusieurs fonctions élémentaires sont allouées à un même composant. Lors de l'occurrence d'une défaillance de celui-ci, les fonctions qu'il supporte, génèrent chacune un événement. Dus à leur propagation dans la décomposition, ces événements se rencontreront au niveau des fonctions. En conséquence, il est nécessaire dans la table de correspondance de traiter les cas d'événements simultanés.

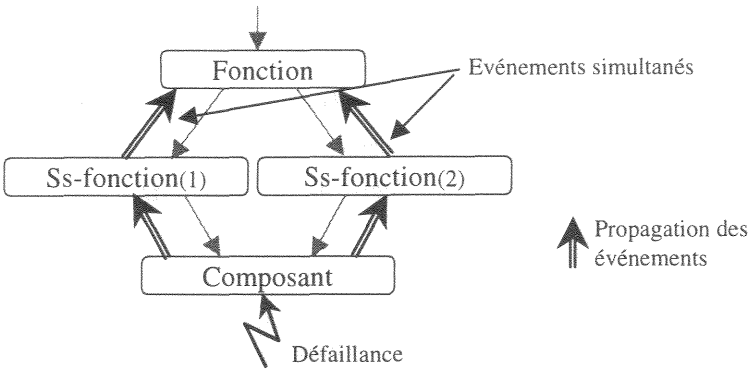


Figure IV.1 : Propagation des événements dans la décomposition

Traiter de manière indépendante les événements et les états donne une plus grande liberté au concepteur pour modéliser le comportement du système qu'il étudie. Certaines méthodes d'analyse ne l'autorisent pas (ex : les graphes de Markov associent les événements à des transitions entre états). Le principal avantage est de pouvoir identifier plusieurs modes de défaillance (que ce soit pour le composant ou pour les fonctions) sans nécessairement ajouter des états de panne qui pourraient alourdir l'étude relative à la disponibilité.

1.3 Influence des modes de fonctionnement

La propagation des événements telle qu'elle est envisagée est cependant trop simpliste pour rendre complètement compte du comportement du système. En effet, selon le fonctionnement du système, l'occurrence d'une défaillance aura des conséquences variables ; typiquement, selon que le système est en marche ou en arrêt, les événements résultants sont différents.

L'utilisation de modes de fonctionnement permet de modéliser le comportement du système. Comme précédemment, cette modélisation s'appuie sur la décomposition hiérarchique. Cependant à l'opposé des états et des événements, les modes de fonctionnement sont assujettis à une propagation descendante.

Ainsi pour chaque fonction, on définit selon son mode de fonctionnement (ex : arrêt, marche réduite, marche nominale...), celui de ses sous-fonctions. Il est possible d'y introduire un paramètre probabiliste. Cela permet de décrire le comportement d'actionneurs redondants où seul l'un d'eux fonctionne lorsqu'un ordre de marche est donné. Cette possibilité peut aussi être utile dans le cas d'une séquence de fonctionnement (ex : marche du système 1 puis du système 2...) où seule une sous-fonction est dans un état de marche à un instant donné.

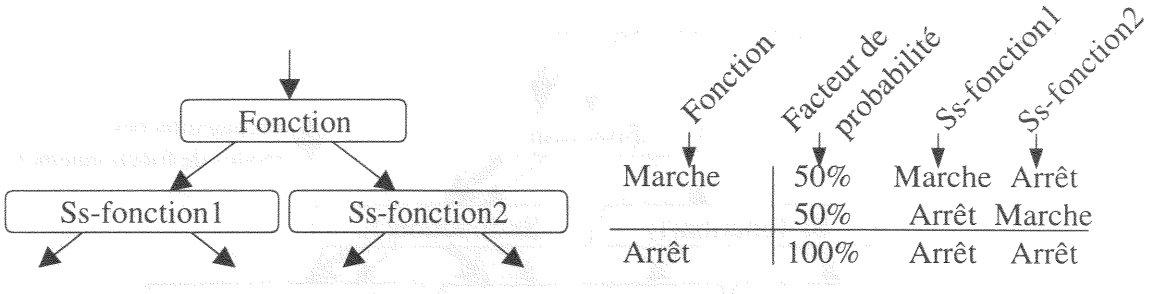


Figure IV.2 : Propagation des modes de fonctionnement dans le cas d'une redondance

Cette définition des différents modes de fonctionnement permet d'adapter les tables de correspondance des événements et de différencier les conséquences des défaillances suivant le contexte de travail du système.

Dans le cas de l'exemple précédent où deux vannes "tout ou rien" sont montées en série, on peut considérer deux modes de fonctionnement, "Ecoulement" et "Fermeture". La table de correspondance des événements peut alors être de la forme suivante :

	Evénement initiateur			Evénement résultant	
				Ecoulement	Fermeture
① :	ss-f. 1:	Arrêt fermé	→	Arrêt intempestif	Blocage fermé
	ss-f. 2:	Arrêt fermé	→	Arrêt intempestif	Blocage fermé
② :	ss-f. 1:	Arrêt ouvert	→	Blocage ouvert	Ouverture intempestive
		Arrêt fermé	→	Fermeture intempestif	Blocage fermé
③ :	ss-f. 2:	Arrêt ouvert	→	Blocage ouvert	Ouverture intempestive
		Arrêt fermé	→	Fermeture intempestif	Blocage fermé

Tableau IV.5 : Table de correspondance entre événements pour la fonction "Contrôler débit" selon le mode de fonctionnement

L'établissement de l'ensemble des tables et tableaux de correspondance des états et événements pour l'ensemble de la décomposition fonctionnelle hiérarchique fournit un modèle de description du comportement de l'installation en présence de défaillances.

1.4 Regroupement dans la décomposition

La décomposition fonctionnelle telle qu'elle est présentée ne peut être qu'arborescente. Cependant, afin d'en faciliter l'établissement et de la rendre plus concise, il peut être intéressant de pouvoir entrecroiser les branches. Ceci s'avère aussi utile lorsqu'une sous-fonction participe à plusieurs fonctions. C'est par exemple, le cas d'une sous-fonction "mesure" (indirectement associée à un capteur) qui peut prendre part à la fois à une activité de régulation et à une autre de surveillance.

L'emploi de cette technique n'a pas d'influence sur la modélisation des états de disponibilité ou de la propagation des événements.

En revanche dans un tel cas, les modes de fonctionnement ne peuvent plus être utilisés de la même façon. En effet, leur propagation descendante nécessite qu'une seule fonction

définisse le mode de fonctionnement. Ainsi, la situation incohérente où plusieurs fonctions imposeraient des modes différents, est évitée.

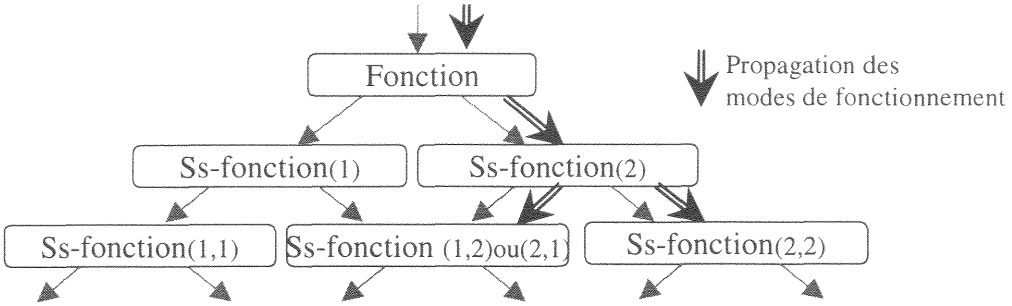


Figure IV.3 : Propagation des modes de fonctionnement dans la décomposition

1.5 Réutilisation

La modélisation par décomposition fonctionnelle permet de réemployer des éléments déjà définis lors d'autres études. En effet, grâce à son aspect modulaire, il est possible de prédéfinir des fonctions plus ou moins génériques (ex : régulation PID d'une grandeur physique...), en vue d'être assemblées lors de la conception de nouveaux systèmes. Cette technique de réutilisation permet de réduire le travail d'établissement des différentes tables et tableaux nécessaires à chaque fonction et dont le nombre peut être important dès que les systèmes envisagés deviennent suffisamment grands.

Cette fonctionnalité peut ainsi s'avérer utile à intégrer dans un logiciel de conception de systèmes d'automatisation.

2 Spécification de l'architecture matérielle

Tout ce qui précède concerne uniquement la modélisation fonctionnelle du système que l'on cherche à étudier. On définit ainsi le comportement du système face aux défaillances des fonctions élémentaires. Il s'agit donc pour permettre l'évaluation des critères de sûreté de fonctionnement, d'y associer les composants matériels.

2.1 Spécifications relatives à l'architecture matérielle

Avant de considérer l'allocation des fonctions élémentaires sur les composants, il est nécessaire d'avoir déterminé au préalable une architecture matérielle préliminaire.

2.2 Architecture matérielle préliminaire

Il s'agit de déterminer un ensemble des composants susceptibles d'accueillir le système. Ce choix est réalisé selon les traitements à accomplir et l'état de l'offre en matériel disponible au moment de l'étude. Leur organisation doit aussi être établie. Elle définit la manière dont les composants sont interconnectés et est édifiée selon les besoins en communication des traitements à implanter sur le système.

En vue d'offrir plusieurs solutions d'architectures matérielles pour permettre une optimisation de l'architecture finale, il semble intéressant de définir une architecture matérielle préliminaire surdimensionnée. Plus explicitement, pour une même fonction, il s'agit de définir plusieurs composants susceptibles de l'accueillir. Par exemple, si une fonction mesure d'une grandeur physique (ex. : température) doit être implantée en un point du

processus, on proposera différents capteurs (par ex. : fonctionnant en 4-20mA, ou connectable à un réseau de terrain). Avec cette technique, lors de l'allocation des fonctions, seuls les composants supportant au moins une fonction sont retenus et composent l'architecture matérielle finale destinée à être évaluée. Cela permet lors de l'optimisation d'envisager plusieurs solutions et indirectement de rechercher le meilleur couple allocation des fonctions/architecture matérielle selon les critères choisis.

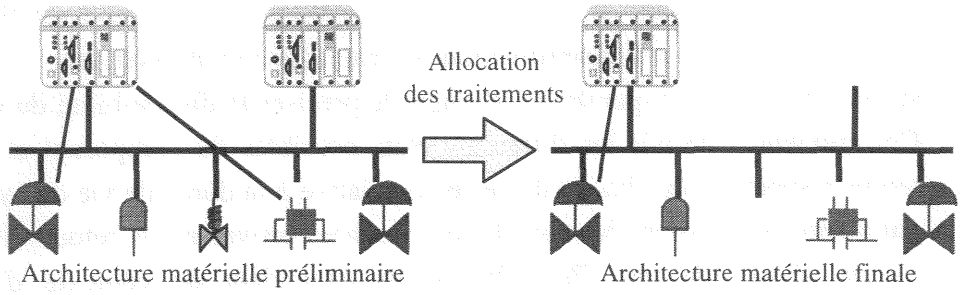


Figure IV.4 : Sélection de l'architecture matérielle

Outre la liste des composants, l'architecture matérielle préliminaire doit spécifier leurs connexions (ou leur organisation). L'objectif est alors de vérifier que les flux entre fonctions élémentaires sont physiquement réalisables grâce à des composants interconnectés.

2.3 Contraintes dues à l'architecture matérielle

La plupart des contraintes liées à l'allocation des fonctions sont essentiellement dues aux limites physiques des composants de l'architecture matérielle. Il s'agit donc lors de l'allocation des fonctions de vérifier le respect des contraintes décrites ci-dessous.

La première de ces contraintes est relative au besoin des fonctions élémentaires à communiquer entre elles. Ainsi lorsqu'un flux de données relie deux fonctions élémentaires affectées à deux composants différents, il s'agit vérifier que ce flux peut être assuré et ce grâce à la compatibilité des interfaces de communication des deux composants considérés. Ainsi, à titre d'exemple, si une fonction "mesure de température" est affectée à un capteur de température de type 4-20mA, la fonction "transporter la mesure de température" ne peut pas être supportée par un réseau de terrain car les interfaces de communication de ces deux composants sont incompatibles.

Les autres contraintes sont relatives aux charges induites par l'allocation des fonctions. D'une part, il s'agit de vérifier que les ressources (ex. : mémoire, nombre d'entrées/sorties) des composants ne sont pas dépassées. D'autre part, les contraintes temporelles doivent être respectées, celles-ci dépendant principalement du nombre de fonctions élémentaires allouées à un même composant et des capacités de traitements de ce composant. La fin du chapitre II formalise la manière dont ces contraintes peuvent être vérifiées.

Une dernière contrainte est relativement indépendante des fonctions élémentaires. Il s'agit de vérifier que les capacités d'interconnexion des composants n'ont pas été dépassées dans l'architecture matérielle finale. A titre d'exemple, certains réseaux de terrain ne pouvant supporter plus de 32 abonnés, le nombre de composants connectés ne doit pas excéder cette valeur.

L'architecture matérielle établie pour une solution d'allocation des fonctions élémentaires, l'évaluation des critères peut débuter afin de juger la solution envisagée.

2.4 Paramètres de sûreté de fonctionnement

Pour chaque composant, un ensemble des paramètres est nécessaire pour permettre l'évaluation des critères. Il faut ainsi connaître :

- le coût du composant. Il définit le coût d'utilisation du composant (installation, entretien...) et sert à la quantification du critère de coût du système,
- ses paramètres de disponibilité. Il s'agit de préciser la disponibilité du composant. Cela revient à définir les différents états possibles du composant et la part du temps respectif pour chacun de ces états relative à la durée de vie du système. Ces paramètres nécessitent de posséder des données provenant de retours d'expérience ou de phases de test. Par ailleurs, ils dépendent du contexte d'emploi du composant, correspondant à l'agressivité du milieu (environnement poussiéreux, vibration, fortes variations de températures...) et à la politique de maintenance (relative à la rapidité de remise en état),
- ses paramètres de fiabilité. Il s'agit de connaître l'ensemble des modes de défaillance et leur probabilité d'occurrence. Comme pour la disponibilité, il est nécessaire de posséder des données de retours d'expérience ou de phases de test.

Ces paramètres connus, l'évaluation quantitative des critères liés à la sûreté de fonctionnement peut être effectuée, l'allocation des fonctions élémentaires ayant été effectuée sur les composants du système. Une simple transposition des paramètres de disponibilité et de fiabilité de chaque composant vers ceux des fonctions qu'il accueille, va permettre cette quantification.

3 Evaluation et optimisation

3.1 Fusion des critères en une grandeur unique

Le dernier aspect pour l'évaluation globale d'un système, concerne la transposition des résultats de l'évaluation (répartition des états, et occurrence des événements) vers une métrique unique.

Il s'agit donc pour la fonction principale de faire correspondre :

- à chacun des états possibles une valeur équivalente aux profits (ou aux pertes) que peut en tirer l'utilisateur,
- à chacun des événements une valeur équivalente aux pertes que peut subir l'utilisateur, lors de leur occurrence.

Avec ces informations, le calcul d'un critère unique est possible. En restreignant aux trois types de critères : coût des équipements, disponibilité et fiabilité, l'évaluation peut être réalisée de la manière suivante [CON 99].

Critères	Quantification		Notation	
Coût des équipements	$-\Sigma c_i$		N_c	(<0)
Disponibilité				
Etat 1 (ex. : disponible)	c_{d1}	p_{d1}	N_{d1}	(>0)
Etat 2 (ex. : disponible dégradée)	c_{d2}	p_{d2}	N_{d2}	
Etat 2 (ex. : indisponible)	c_{d3}	p_{d3}	N_{d3}	(≤0)
...	...		...	
Fiabilité				
Evénement 1 (ex. : arrêt)	$- c_{e1}$	p_{e1}	N_{e1}	(<0)
Evénement 2 (ex. : fuite)	$- c_{e2}$	p_{e2}	N_{e2}	(<0)
...	...		...	
Evaluation globale de l'installation	Somme :		$N_{globale}$	

Tableau IV. 6 : Evaluation par l'association de critères

Les variables utilisés sont :

c_i : coût d'installation et d'utilisation du composant i

c_{di} : coût en profits ou pertes lorsque l'installation est dans l'état i

c_{ei} : coût dû à l'occurrence de l'événement i (négatif)

N_c : coût de l'ensemble des composants (valeur négative)

N_{di} : coût estimatif du profit ou des pertes dus à la présence du système dans l'état i pendant sa durée de vie (établi par $c_{di} \times p_{di}$),

N_{ei} : coût estimatif des pertes dues aux occurrences de l'événement i pendant la durée de vie du système (établi par $-c_{ei} \times p_{ei}$), (valeur négative)

$N_{globale}$: indicateur de la qualité de l'installation exprimé sous forme d'un coût

p_{di} : part estimée du temps dans laquelle l'installation se trouve dans l'état i

p_{ei} : nombre estimé de fois où s'est produit l'occurrence de l'événement i

Après avoir effectué une répartition, l'objectif est la quantification des grandeurs p_{di} et p_{ei}, pour aboutir finalement à évaluer la note globale ($N_{globale}$), obtenue pour l'architecture testée.

3.2 Traitements pour la quantification

La forme utilisée pour décrire le comportement du système relatif à la sûreté de fonctionnement permet de générer assez facilement des diagrammes de décision aptes à être quantifiés. La méthode décrite ci-après est une proposition pour y parvenir.

Disponibilité

Une fonction élémentaire est associée à un composant (celui sur lequel elle est allouée). Son état (disponible, indisponible) dépend directement de celui du composant (état correct, en panne...). Il est donc possible d'établir un diagramme décrivant selon l'état du

composant celui de la fonction. La relation utilisée pour établir ce diagramme, et qui lie ces états, a été définie dans la décomposition fonctionnelle.

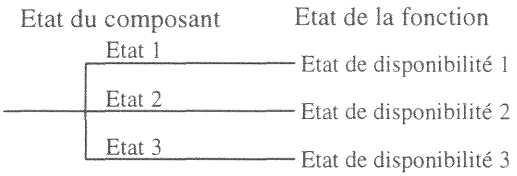


Figure IV.5 : Diagramme des états d'une fonction élémentaire

Pour l'ensemble des fonctions élémentaires, la même opération est à réaliser et détermine par des diagrammes de décision, leur états selon ceux des composants.

Pour les fonctions hiérarchiquement supérieures, il s'agit d'établir le même type de diagramme mais celui-ci peut comporter pour une même fonction plusieurs composants. En effet, ces fonctions sont l'association de fonctions plus élémentaires et donc utilisent indirectement plusieurs composants. Tels qu'ils sont définis dans la table de vérité, les états de disponibilité d'une fonction dépendent uniquement de ceux de ses sous-fonctions. Grâce à cette relation et par une démarche ascendante (par rapport à la décomposition), il est possible d'établir progressivement tous les diagrammes. Pour chacun, il s'agit de "croiser" les diagrammes des sous-fonctions en utilisant la relation qui les lie. Ceci est réalisé par l'application des méthodes d'opérations entre diagrammes présentées au chapitre précédent.

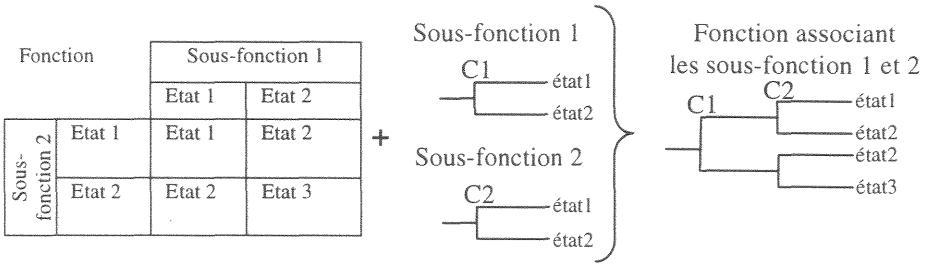


Figure IV.6 : Diagramme des états possibles d'une fonction hiérarchiquement supérieure

En appliquant cette méthode de manière itérative, on obtient finalement un diagramme décrivant la capacité du système à assurer sa mission en fonction de l'état des composants.

Grâce à celui-ci et aux paramètres de disponibilité des composants, l'évaluation de la répartition temporelle des différents états du système est réalisable.

Fiabilité

L'étude des aspects relatifs à la fiabilité consiste à évaluer la probabilité d'occurrence de chacun des événements affectant la mission du système. Ces événements ne sont que la manifestation des conséquences d'une défaillance d'un des composants. Pour y parvenir, l'emploi des diagrammes de décision est aussi proposé.

Une démarche similaire à celle de l'étude de la disponibilité est utilisée et consiste à établir successivement les diagrammes pour chaque fonction de manière ascendante. Ces diagrammes se différencient de ceux relatifs à la disponibilité par le fait qu'ils fournissent pour un événement initiateur (une défaillance) l'événement résultant selon l'état des composants. Cependant, la méthode proposée ne s'applique qu'à l'étude individuelle de chaque

mode de défaillance de chaque composant. Ainsi, il est nécessaire de réitérer l'opération autant de fois qu'il y a de composants et de modes de défaillances.

Les premiers diagrammes à établir sont ceux des fonctions élémentaires supportées par le composant étudié (et responsable de la défaillance). Pour un mode de défaillance choisi, on établit par ce diagramme les événements résultants pour chacune des fonctions et ce selon l'état du composant. Ainsi pour un capteur sur lequel une fonction "mesurer" est affectée, le diagramme correspondant à l'événement "défaillance du capteur par arrêt" est le suivant :

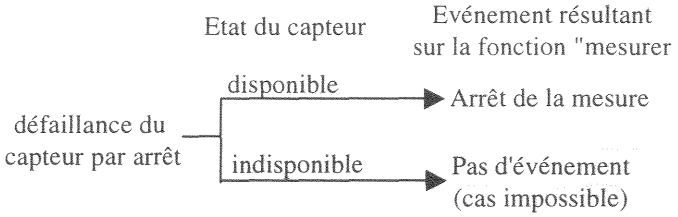


Figure IV. 7 : Diagramme des événements pour une fonction élémentaire

A partir de ces diagrammes, ceux des fonctions hiérarchiquement supérieures peuvent être établis à leur tour. Les tables de correspondance des événements (décrivant la propagation ascendante des événements) sont utilisées. Pour une fonction donnée, cette table fournit pour chaque combinaison des états des sous-fonctions, les correspondances entre événements. Le diagramme est donc construit à partir du diagramme des états de chaque sous-fonction (établi lors de l'étude de disponibilité) et de leurs diagrammes d'événements. Ainsi, pour la fonction "Contrôler débit" (exemple du paragraphe 1.2), et en considérant une défaillance influant sur la sous-fonction "Contrôler débit I", trois diagrammes sont utilisés ; deux relatifs aux états des deux sous-fonctions, un relatif aux événements de la sous-fonction "Contrôler débit I".

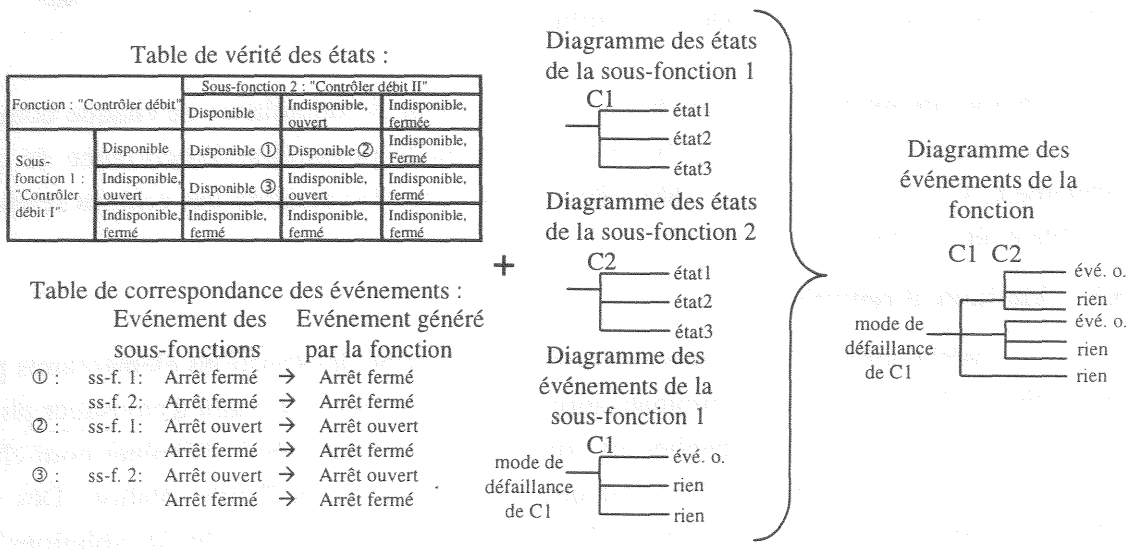


Figure IV. 8 : Diagramme des états pour une fonction hiérarchiquement supérieure

Afin de faciliter le traitement de ces tables, tableaux et diagrammes, il apparaît intéressant de préétablir un premier diagramme de décision uniquement relatif aux sous-fonctions. Edifié au début de la phase d'évaluation à partir de la table de vérité et de la table de correspondance, ce diagramme fournit l'événement résultant selon les états et les

événements des sous-fonctions. Grâce à lui, le diagramme final concernant l'étude d'un mode de défaillance d'un composant est construit assez facilement en remplaçant chaque nœud de ce diagramme préliminaire par les diagrammes correspondants des sous-fonctions. Ce diagramme préliminaire a ainsi la forme suivante :

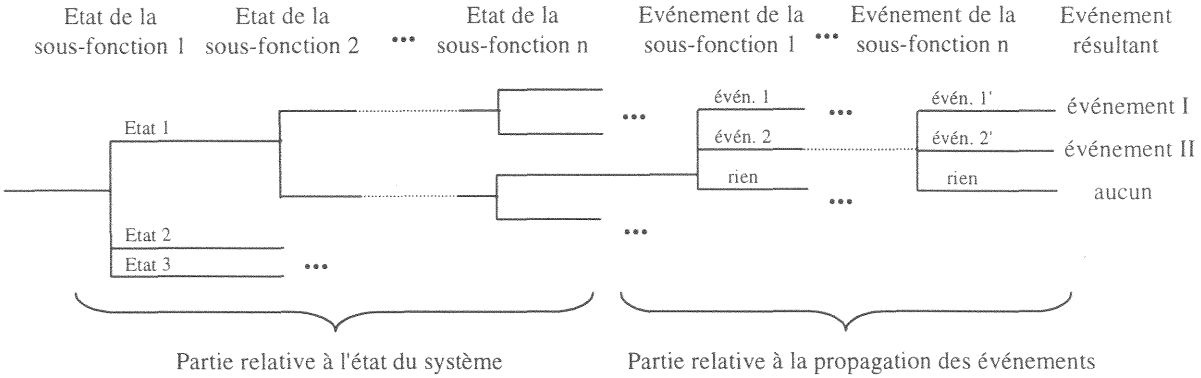


Figure IV. 9 : Diagramme préliminaire pour traiter les événements

Cette opération d'établissement des diagrammes de décision relatifs aux événements est à réitérer pour chaque fonction, selon une démarche ascendante par rapport à la décomposition hiérarchique. Finalement, on obtient alors un diagramme de décision pour la fonction principale du système qui pour l'occurrence d'un mode de défaillance donnée, fournit selon les différentes combinaisons des états possibles des composants, les événements résultants affectant le système.

De ce diagramme, une quantification probabiliste des événements résultants est alors facilement réalisable et utilise les données de disponibilité de chacun des composants et la probabilité d'occurrence du mode de défaillance considéré.

Enfin, en réitérant cette étude pour chaque mode de défaillance de chaque composant et en en faisant la synthèse, on obtient globalement la probabilité d'occurrence de chaque événement. Ces résultats peuvent alors être utilisés dans l'évaluation des critères relatifs à la fiabilité et être transposés en un coût.

3.3 Méthode d'optimisation

La méthodologie d'évaluation associée à l'allocation des fonctions élémentaires permet une optimisation du système d'automatisation à concevoir. Ainsi, il s'agit d'envisager plusieurs solutions d'allocation (pour lesquelles les contraintes sont respectées), d'évaluer pour chacune d'elles les critères et enfin de ne retenir que celles ayant la meilleure notation. Dès que le système à concevoir atteint une taille suffisamment importante, le nombre de solutions est tel que l'emploi d'un algorithme de recherche est nécessaire.

Une première forme d'algorithme de recherche consiste à explorer l'ensemble de l'espace des solutions. Cela revient à envisager pour les fonctions élémentaires toutes les allocations possibles. Chaque allocation détermine une architecture matérielle et organique. Si

les contraintes sont respectées, l'évaluation des critères peut être entreprise afin de n'en retenir que la meilleure. Cette technique est utilisée pour la première application présentée dans le prochain chapitre.

L'algorithme suivant effectue cette recherche, pour un système composé de n fonctions élémentaires où a_i est l'allocation de la fonction i et A_i l'ensemble des allocations possibles pour cette fonction.

Etudier tout l'espace des solutions n'est réalisable que quand celui-ci n'est pas trop grand. Quand le nombre de combinaisons devient trop important, la recherche devient trop longue et des méthodes stochastiques doivent être envisagées. Ces méthodes partent d'une ou plusieurs solutions initiales, y appliquent des transformations élémentaires en ne retenant que celles améliorant les solutions initiales. En réitérant de nombreuses fois ces transformations, les solutions retenues tendent alors dans le meilleur des cas vers la solution ayant la meilleure notation, mais le plus souvent vers des maximums locaux. Ainsi l'emploi des ces algorithmes n'assure pas d'obtenir la solution optimale, mais peut fournir au concepteur des solutions acceptables vis-à-vis des objectifs qu'il s'est imposé.

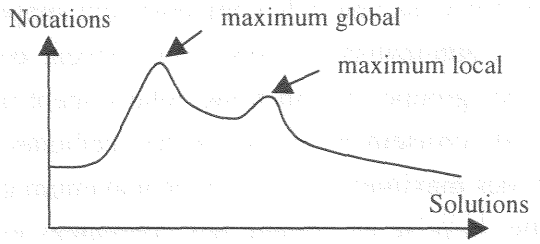


Figure IV.10 : Maximum global et local

Les méthodes actuellement les plus utilisées pour les problèmes d'allocation sont :

- Le gradient descendant : Cette méthode itérative débute à partir d'une solution initiale. Cette solution dite "courante" est évaluée et comparée à un ensemble de solutions obtenues en appliquant des transformations élémentaires. Dans le cas de la répartition des fonctions, ces transformations peuvent correspondre au changement d'affectation d'une des fonctions. Si une solution jugée meilleure est trouvée, celle-ci est conservée et un nouveau cycle de transformations est effectué. Un exemple d'application à des arbres de défaillances est fourni par THOMAS [THO 98]. L'inconvénient de cette méthode est que la recherche s'arrête dès qu'un maximum local (et pas forcément global) est atteint. Les méthodes suivantes proposent des solutions pour tenter d'échapper aux maximums locaux.
- Le recuit simulé [KIR 83] : Cette méthode inspirée de la mécanique statistique, utilise le même principe que celui du gradient descendant. Cependant, il autorise ponctuellement et pour quelques itérations de retenir des solutions ayant une moins bonne notation. Cet additif peut permettre à la solution courante de sortir d'un éventuel maximum local dans lequel elle pourrait se trouver.

- La recherche Tabou [GLO 89] [GLO 90] : Cette méthode ajoute à celle du gradient descendant, un mécanisme de mémorisation des solutions trouvées afin de sortir des maximums locaux et d'éviter d'y revenir.
- Les algorithmes génétiques [GOL 87] : Cette famille de méthodes s'inspire des mécanismes de sélection naturelle. Leur principe de fonctionnement s'appuie sur la représentation des solutions envisageables sous la forme d'un ensemble de gènes. A partir d'une population d'individus (c'est-à-dire des solutions d'architecture), des opérations de croisements associées parfois à des mutations permettent de générer de nouveaux individus. Après évaluation, seuls les meilleurs individus sont conservés pour former une nouvelle population sur laquelle un nouveau cycle d'opérations pourra être entrepris. Au fur et à mesure des itérations, la population finale regroupe les meilleures solutions parmi lesquelles l'optimum peut se trouver.

Cette dernière méthode semble être une des plus prometteuses pour traiter les problèmes d'allocation. En effet, tout d'abord, il existe une représentation simple des solutions d'architectures exprimable sous la forme d'un vecteur où chaque "gène" représente l'affectation d'une fonction et sa valeur le composant qui supporte la fonction correspondante. De plus, les systèmes d'automatisation sont très souvent composés de fonctions ou sous-systèmes (c'est-à-dire des groupes de fonctions) relativement indépendants entre eux. Dans un tel cas, le mécanisme de croisement utilisé par les méthodes d'inspiration génétique facilite leur convergence vers des maximums locaux ou le maximum global.

En contrepartie, la prise en compte des contraintes est plus délicate. Elle peut ainsi devenir pénalisante lorsqu'une forte part des individus obtenus lors du croisement ne vérifie pas les contraintes. Cela peut entraîner pour l'algorithme un grand nombre d'opérations inutiles et en conséquence un temps de traitement trop important. Cette difficulté peut être compensée en associant les contraintes au choix des gènes lors du croisement. Cette technique est utilisée dans la deuxième application présentée dans le chapitre suivant.

3.4 Optimisation du choix des fonctions

Une extension de la méthode d'allocation présentée permet une participation aux choix des fonctions à implanter. En effet, une partie des fonctions élémentaires recensées n'est pas indispensable à l'exécution de la mission du système ; il s'agit principalement des redondances ou des mécanismes de sécurité. Elles ont été retenues par le concepteur comme devant a priori améliorer la disponibilité ou/et la fiabilité sans être trop onéreuses. Cependant, décidés très tôt lors de la démarche de conception, ces choix n'ont pas pu être évalués auparavant et ne peuvent être complètement validés que pendant la phase d'évaluation des solutions envisagées.

Cet aspect peut être intégré à la méthode d'optimisation. Il s'agit alors d'ajouter à l'ensemble des allocations possibles pour les fonctions concernées, une possibilité de non-affectation à un composant. Dans la phase d'évaluation, les fonctions élémentaires ayant ce statut de "non-affecté" sont considérées comme indisponibles de manière permanente.

Indirectement, on optimise ainsi le choix des fonctions à implanter et ce conformément aux critères choisis pour juger le système à concevoir.

Cette technique est appliquée dans la première application du chapitre suivant et détermine les mécanismes de sécurité à utiliser.

4 Conclusion

La méthodologie présentée propose pour concevoir un système d'automatisation de s'appuyer sur la décomposition fonctionnelle. Celle-ci permet de définir un ensemble de fonctions élémentaires dont chacune est susceptible d'être accomplie par un composant unique. Le problème de la conception revient alors à déterminer un ensemble organisé de composants chargé d'accueillir ces fonctions.

Pour y parvenir, la démarche proposée nécessite de définir une architecture préliminaire volontairement surdimensionnée. De celle-ci, seule une partie de ses composants est retenue par l'allocation des fonctions élémentaires et compose l'architecture finale. Cette allocation est soumise aux respects de plusieurs contraintes relatives aux capacités des composants, aux contraintes temporelles associées aux fonctions et à leur besoin de communication.

De nombreuses solutions sont alors possibles et l'emploi de critères permet de les évaluer en vue de n'en retenir que la ou les meilleures. Outre le coût des composants, ces critères sont relatifs à la sûreté de fonctionnement par l'évaluation a priori de la disponibilité du système et de ses probabilités d'occurrence de défaillances aux conséquences quantifiables. Ces évaluations sont réalisées grâce à l'enrichissement de la décomposition fonctionnelle avec des informations décrivant le comportement du système en présence de défaillances.

Enfin, dû au nombre important de solutions envisageables, l'emploi d'un algorithme de recherche combinatoire est nécessaire. Ce chapitre en présente plusieurs parmi lesquels les méthodes d'inspiration génétique sont préconisées.

Le chapitre suivant présente l'application de cette méthodologie à deux applications de nature très différentes.

5 Bibiographie

[CON 97] B. CONRARD, M. ROBERT, "A method to evaluate the dependability of DIBAS", 3rd IFAC Symposium on Intelligent Components and Instruments for Control Applications, SICICA'97, Annecy (France), p 181-192, 9-11 Juin 1997.

[CON 99] B. CONRARD, M. ROBERT, J.M. THIRIET, Prise en compte de la Sûreté de Fonctionnement dans la conception des SAID, 3ème Congrès International Pluridisciplinaire sur la Qualité et la Sûreté de Fonctionnement, QUALITA'99, p231-241, Paris, Mars 99.

[GLO 89] F. GLOVER, Tabu search : Part I, ORSA Journal on Computing, volume 1, pages 190 à 206, 1989.

[GLO 90] F. GLOVER, Tabu search : Part II, ORSA Journal on Computing, volume 2, pages 4 à 32, 1990.

[GOL 87] D.E. GOLDBERG, Genetic algorithms in search, optimization, and machine learning, Edition Addison-Wesley, Reading Mass., 1987.

[KIR 83] S. KIRKPATRICK, C.D. GELATT, M.P. VECCHI, Optimization by simulated annealing, Science, volume 220, n°4598, pages 671 à 680, 1983.

[THO 98] P. THOMAS et al., Sherloc : Outil d'allocation d'indisponibilité pour les arbres de défaillance, acte de λμ11, Arcachon, pages 278 à 283, 1998.

Chapitre V

Applications de la méthode de conception

Chapitre V

Applications de la méthode de conception

Ce chapitre a pour objectif d'illustrer la méthodologie proposée par sa mise en œuvre sur deux applications. La première est relative à une "cuve chauffante" pour laquelle l'architecture matérielle optimale du système d'automatisation est recherchée à partir d'une architecture matérielle préliminaire surdimensionnée et de la décomposition fonctionnelle de ce système. La seconde concerne le système de commande (sous-ensemble du système d'automatisation) d'un processus thermique. Une partie de l'architecture matérielle est fixée par le concepteur. L'objectif est alors de déterminer les capteurs, les actionneurs et les traitements à associer à des automates connectés à un réseau de terrain.

1 Première application

1.1 Le contexte

La méthode est appliquée au processus suivant. Il s'agit d'une cuve destinée à fournir un fluide à une température voulue. Le processus comporte une pompe permettant le remplissage de la cuve et un corps de chauffe (une résistance) afin de maintenir ce fluide en température. En fonctionnement, le système d'automatisation est donc chargé de maintenir un niveau suffisant de fluide dans la cuve et de lui fournir l'énergie nécessaire pour obtenir une température constante, et ceci malgré les variations de la demande en fluide.

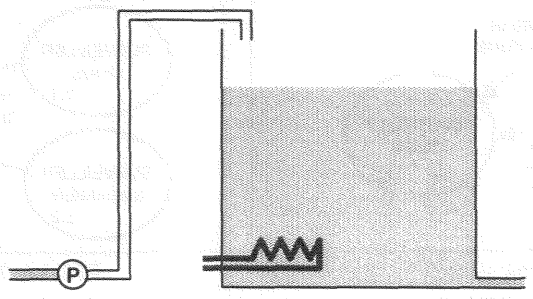


Figure V.1 : Schéma du processus

On fait comme hypothèse que la pompe soutire le fluide d'un réservoir que l'on suppose toujours plein.

On suppose que l'emploi d'une cuve vise à améliorer la régulation face aux variations, particulièrement pour des demandes ponctuelles de grande quantité.

1.2 Evénements redoutés

Pour un tel procédé, les seuls événements redoutés relatifs à l'exécution de la mission sont par ordre croissant d'importance des conséquences :

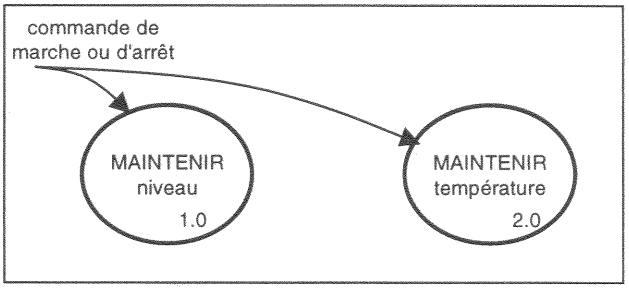
- l'arrêt intempestif de l'installation,
- la fourniture du fluide à une mauvaise température,
- le débordement ou la destruction du corps de chauffe en cas d'alimentation de celui-ci en l'absence de fluide.

1.3 Décomposition fonctionnelle

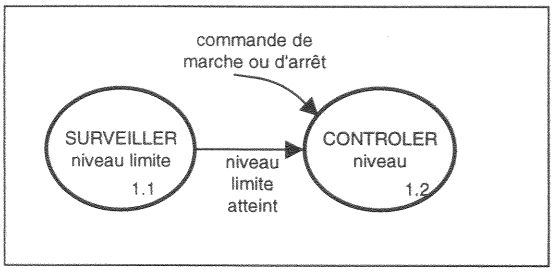
La première étape de la démarche consiste à décrire les fonctions et services que doit accomplir le système d'automatisation. Cette démarche s'appuie sur la décomposition itérative des fonctions en sous-fonctions. La fonction principale, sommet de la décomposition, est "maintenir le niveau et la température du fluide dans la cuve".

La représentation employée pour cette application s'inspire des diagrammes de flots de données (DFD) auxquels quelques adaptations ont été apportées. Ainsi, une seule forme de flot de données est utilisée pour représenter les flux entre processus (dénomination des fonctions pour les DFD). Il n'y a pas de processus de stockage. Enfin, le schéma de contexte n'est pas représenté dans la mesure où les interactions avec le processus physique sont effectuées par des processus des diagrammes.

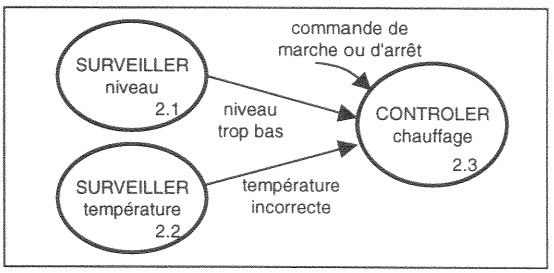
Seule une partie de la décomposition (complète en annexe) est représentée ci-après :



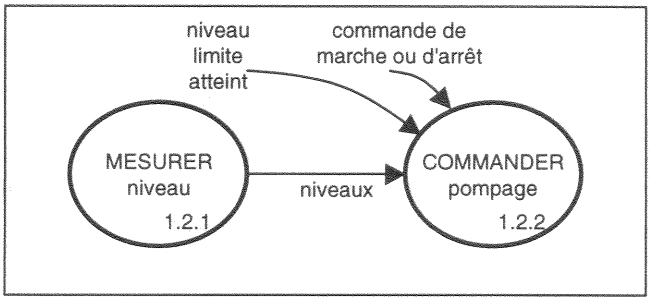
DFD 0 : Maintenir le niveau et la température d'un fluide dans une cuve



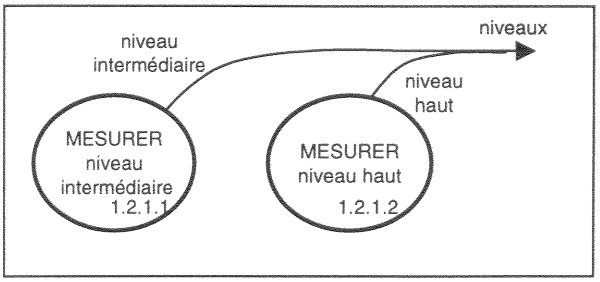
DFD 1.0 : Maintenir niveau



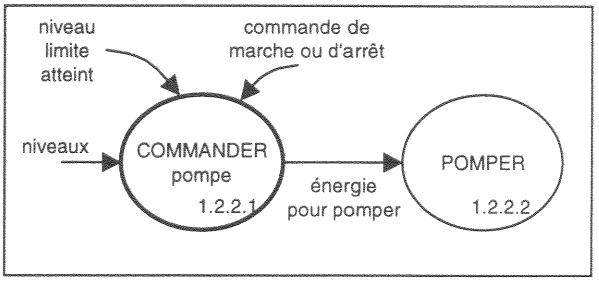
DFD 2.0 : Maintenir température



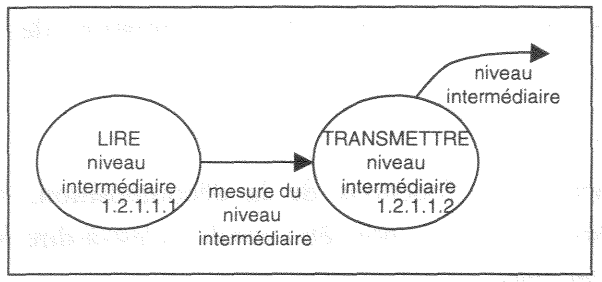
DFD 1.2 : Contrôler niveau



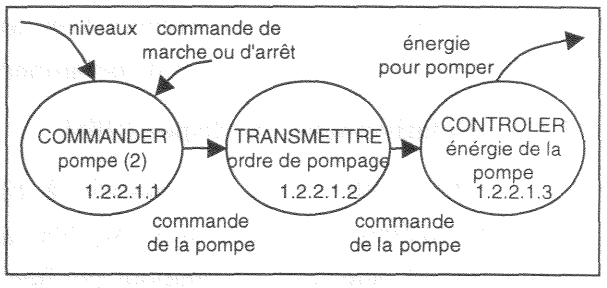
DFD 1.2.1 : Mesurer niveau



DFD 1.2.2 : Commander pompage



DFD 1.2.1.1 : Mesurer niveau intermédiaire



DFD 1.2.2.1 : Commander pompe

Figure V.2 : Extrait des diagrammes flots de données du système

Cette représentation des fonctions permet d'identifier d'une part les traitements élémentaires à répartir sur les équipements (représentés par des cercles fins), et d'autre part les flux interconnectant ces traitements.

On distingue différents types de traitements élémentaires. Ce peut être la simple transmission d'une information entre équipements (ex : TRANSMETTRE niveau intermédiaire, 1.2.1.1.2), l'acquisition d'une mesure (ex : LIRE niveau intermédiaire, 1.2.1.1.1), l'action sur le processus par l'intermédiaire d'un actionneur (ex : POMPER, 1.2.2.2) ou enfin l'exécution d'un algorithme (ex : COMMANDE pompage, 1.2.2.1.1, qui fournit les ordres à pompe suivant le niveau du fluide dans la cuve).

L'ensemble de la décomposition hiérarchique utilisée pour cette application, est présentée sur la page suivante de manière plus concise.

Dans cette décomposition, certaines fonctions sont annotées de la mention "optionnel". Il s'agit de fonctions de surveillance dont la mission est l'arrêt du système dès qu'une situation dangereuse est détectée. Ce peut être un niveau de fluide trop haut dans la cuve, une température trop élevée ou un niveau trop bas pour autoriser le chauffage.

L'absence de ces fonctions ne compromet pas la mission du système ; il est toujours capable de maintenir le fluide à un niveau et à une température voulus. Cependant, lors de l'occurrence de défaillances susceptibles d'entraîner un mauvais comportement du système dans l'accomplissement de cette mission, l'action de ces fonctions peut limiter les effets et conséquences néfastes de ce dysfonctionnement. En revanche, les occurrences possibles de défaillances sur ses fonctions de sécurité peuvent entraîner des arrêts intempestifs et diminuer la disponibilité globale du système.

Les indiquer comme "optionnel", permet d'envisager différentes stratégies de répartition incluant ou non des équipements associés. L'évaluation des critères (et indirectement des rapports coûts des équipements, disponibilité du système, probabilités d'occurrence des effets néfastes) associée à la recherche de l'architecture à la meilleure rentabilité estimée, effectue une sélection des mécanismes de sécurité à utiliser.

1.4 Spécification des aspects liés à la sûreté de fonctionnement

La décomposition fonctionnelle permet de recenser les traitements (ou fonctions) élémentaires et les flux qui les interconnectent. En vue d'étudier la sûreté de fonctionnement,

il s'agit d'enrichir cette décomposition avec des données permettant l'estimation de la disponibilité et des occurrences de défaillance.

1.4.1 Relativement à la disponibilité

Dans cette étape, on associe à chacune des fonctions de la décomposition, un ensemble d'états décrivant si le service associé peut ou non être rendu (c'est-à-dire sa disponibilité), ainsi que les relations qui lient ces états.

Pour l'application "cuve chauffante" concernée, les états possibles des fonctions élémentaires ont été limités à "disponible" et "indisponible". Il en est de même pour les fonctions hiérarchiquement supérieures, sauf pour "Maintenir température" qui comporte trois états. Cet état supplémentaire est "disponible insécurisé" qui précise que le service de maintien de la température est valide, mais qu'en cas de vidange de la cuve, ce service ne détectera pas la disparition du fluide d'où un risque de poursuite du chauffage et d'endommagement du corps de chauffe.

Décomposition fonctionnelle et hiérarchique
du maintien du niveau et de la température
d'un fluide dans une cuve.

- MAINTENIR le niveau et la température de la cuve
  - MAINTENIR niveau
    - CONTROLER niveau
      - MESURER niveau
        - MESURER niveau haut
          - *LIRE niveau haut*
          - *TRANSMETTRE lecture niveau haut*
        - MESURER niveau intermédiaire
          - *LIRE niveau intermédiaire*
          - *TRANSMETTRE lecture niveau intermédiaire*
      - COMMANDER pompage
        - *POMPER*
        - COMMANDER pompe
          - *COMMANDER pompe suivant niveaux*
          - *TRANSMETTRE ordre de pompage*
          - *CONTROLER énergie électrique pour la pompe*
    - SURVEILLER niveau limite atteint *(optionnel)*
      - *MESURER niveau limite*
      - *TRANSMETTRE ordre d'arrêt*
      - *ARRETER pompe*
  - MAINTENIR température
    - CONTROLER la température
      - *MESURER température*
        - *LIRE mesure de température*
        - *TRANSMETTRE mesure de température*
      - COMMANDER chauffage
        - REGULER chauffage
          - *CALCULER commande de chauffage*
          - *TRANSMETTRE commande de chauffage*
          - *CONTROLER énergie électrique pour la résistance*
        - *CHAUFFER*
    - SURVEILLER le niveau *(optionnel)*
      - *MESURER niveau trop bas*

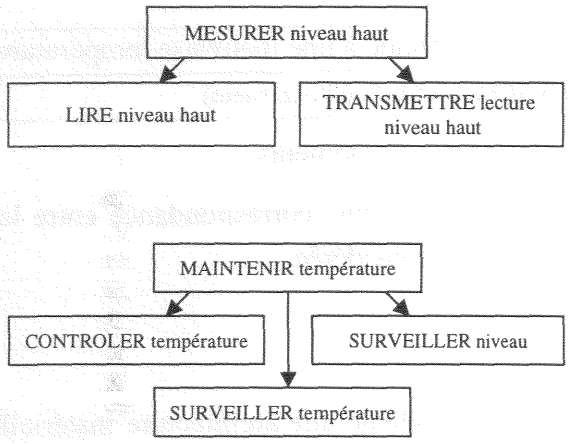
- *TRANSMETTRE mesure de niveau bas*
- *ARRETER le chauffage*

- SURVEILLER la température *(optionnel)*
  - *MESURER température trop élevée*
  - *TRANSMETTRE mesure de niveau bas*
  - *ARRETER le chauffage*

Tableau V.1 : Décomposition hiérarchique du système

Les relations entre états sont définies à l'aide de tables de vérité. La plupart des fonctions ne sont dans un état disponible que si toutes leurs sous-fonctions le sont également. Les exceptions concernent les sous-fonctions correspondant à des mécanismes de surveillance où leur indisponibilité n'influe pas sur la réalisation du service (même si son absence peut en cas de défaillance, avoir des conséquences dangereuses)

A titre d'illustration, les tables pour les fonctions "MAINTENIR température" et "MESURER niveau haut" sont les suivantes :



MESURE niveau haut		TRANSMETTRE lecture niveau haut	
		disponible	indisponible
LIRE niveau haut	disponible	disponible	indisponible
	indisponible	indisponible	indisponible

MAINTENIR température		CONTROLER température	
SURVEILLER niveau	SURVEILLER température	disponible	indisponible
disponible	disponible	disponible	indisponible
	indisponible	disponible	indisponible
indisponible	disponible	dispo. inséc.	indisponible
	indisponible	dispo. inséc.	indisponible

dispo. inséc. ≡disponible insécurisé

Figure V.3 : décomposition et tables de vérité pour deux fonctions

1.4.2 Relativement à la fiabilité

Le second aspect, relatif à l'étude de la sûreté de fonctionnement dans la méthodologie proposée, est l'évaluation de la probabilité de défaillance du système. Il s'agit ainsi de définir la propagation des défaillances (ou événements) à travers la décomposition hiérarchique.

Pour l'application "cuve chauffante", les événements considérés sont entre autres, "Arrêt" du service (pour les traitements de données ou la transmission), "Blocage actif" ou "Blocage inactif" (pour les fonctions de mesure associées à des détecteurs), "Arrêt intempestif" ou "Démarrage intempestif" (pour les fonctions d'actionnement, pompe, corps de chauffe, relais...). Concernant la mission du système, "MAINTENIR le niveau et la température dans une cuve", les événements associés sont "Arrêt intempestif", "Fourniture du fluide à une mauvaise température", "Accident (débordement, destruction du corps de chauffe, bouillonnement)".

Pour les fonctions élémentaires, ces événements sont directement déduits de la nature des fonctions individuellement considérées. Ils reflètent indirectement les modes de défaillance des composants qui les supportent.

Pour les fonctions hiérarchiquement supérieures, il s'agit de définir la propagation des événements et ceci selon l'état des ressources. Cette correspondance entre événements est la suivante pour la mission du système :

Sous-fonctions	Evénements sources	Evénements correspondants	
MAINTENIR niveau	Arrêt du service	Arrêt du service	
	Débordement	Risque d'accident (débordement)	
	Vidange intempestive		Etat de MAINTENIR niveau
		Arrêt du service	disponible
		Risque d'accident (destruction du corps de chauffe)	disponible insécurisé
MAINTENIR température	Arrêt du service	Arrêt du service	
	Température incorrecte	Fourniture du fluide à une mauvaise température	
	Chauffage permanent	Risque d'accident (bouillonnement)	

Tableau V. 2 : Table de propagation des événements

De la même manière, on définit pour chaque fonction, une correspondance entre les événements des sous-fonctions et ceux relatifs à la fonction considérée.

1.5 Architecture matérielle

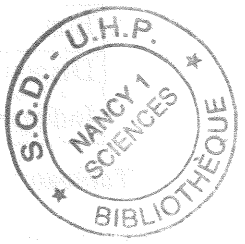
La méthodologie proposée requiert de définir au préalable une architecture matérielle préliminaire. La table suivante décrit l'ensemble des équipements susceptibles d'être employés dans le système.

Type d'équipement	Caractéristiques / connexions	Coût équip.	Informations liées à la disponibilité	Mode et nombre de défaillances
Pompe	-1 entrée, alimentation de puissance	500	Disponible : 98 % Indisponible : 2 %	Arrêt intempestif : 3
	-1 entrée, alimentation de puissance et relais -1 entrée marche, 1 entrée arrêt	600	Disponible : 97 % Indisponible : 3 %	Arrêt intempestif : 3 Démarrage intempestif : 2
	-1 entrée, alimentation de puissance -1 interface réseau de terrain (RdT)	700	Disponible : 97 % Indisponible : 3 %	Arrêt intempestif : 3 Démarrage intempestif : 1
Détecteur de niveau	-1 sortie ouvert/fermée*	20	Disponible : 99 % Indisponible ouvert : 1 %	Blocage ouvert : 1 Blocage fermée : 1
	-1 interface RdT*	50	Disponible : 99 % Indisponible 1 %	Blocage ouvert : 1 Blocage fermée : 1 Arrêt : 1

Type d'équipement	Caractéristiques / connexions	Coût équip.	Informations liées à la disponibilité	Mode et nombre de défaillances
Capteur de température	-1 sortie 4-20mA	30	Disponible : 99 % Indisponible: 1 %	Mesure faussée : 3
	-1 interface RdT	50	Disponible : 99 % Indisponible 1 %	Mesure faussée : 3 Arrêt de la mesure : 1
	Capteur à seuil*	20	Disponible : 99,5 % Indisponible: 0,5 %	Arrêt : 1
	Capteur à seuil avec interface RdT*	40	Disponible : 99 % Indisponible: 1 %	Arrêt : 2
Résistance de chauffage	-1 entrée, alimentation de puissance	200	Disponible : 99 % Indisponible : 1 %	Arrêt intempestif : 2
	-1 entrée 4-20mA Régulateur intégré	300	Disponible : 99 % Indisponible 1 %	Arrêt intempestif : 2 Chauffage intempestif : 0,5
	-1 entrée, alimentation de puissance -1 interface réseau de terrain (RdT)	300	Disponible : 99 % Indisponible 1 %	Arrêt intempestif : 2 Chauffage intempestif : 0,5
Relais	-relais de puissance*	20	Disponible : 99 % Indisponible 1 %	Blocage ouvert : 1 Blocage fermée : 1
	-relais de puissance avec interface RdT*	30	Disponible : 99 % Indisponible 1 %	Blocage ouvert : 1 Blocage fermée : 1
Système de communication	connexion point-à-point*	5	Disponible : 99,5 % Indisponible : 0,5 %	Rupture : 1 Court-circuit : 1
	réseau de terrain	50	Disponible : 99 % Indisponible : 1 %	Arrêt communication : 5
Système de commande	Automate avec entrées sorties tout ou rien et analogique	200	Disponible : 99 % Indisponible : 1 %	Arrêt : 2 Fonctionnement incohérent 0,5
	Automate avec interface RdT	200	Disponible : 99 % Indisponible : 1 %	Arrêt : 2 Fonctionnement incohérent 0,5

Tableau V. 3 : Données de sûreté de fonctionnement des composants

Pour les équipements notés par * et utilisés uniquement dans des fonctions de sécurité (ex : éviter le débordement), on fait l'hypothèse que leur taux d'indisponibilité est triplé. En effet, certaines de leurs défaillances n'ont pas d'effet immédiat sur le procédé et sont donc masquées. Seuls des tests périodiques liés à une activité de maintenance permettent de les détecter. Leur disponibilité en est donc directement affectée.



L'architecture matérielle préliminaire est la suivante :

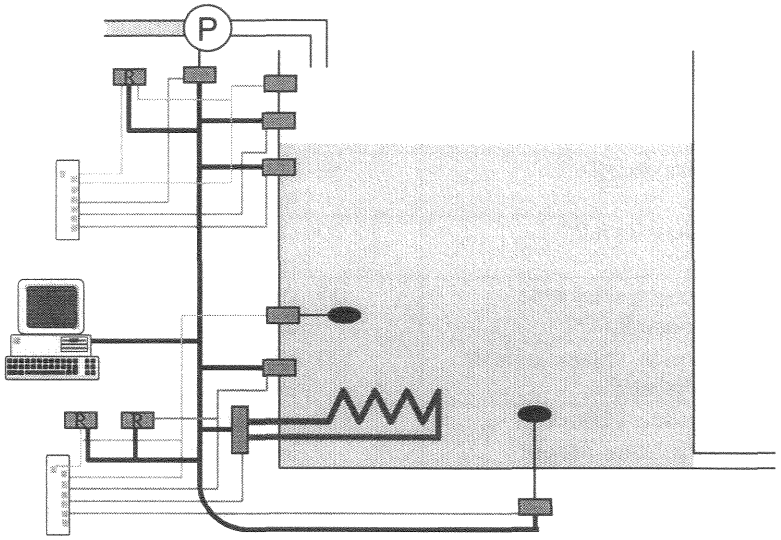


Figure V.4 : Architecture matérielle préliminaire du système

L'architecture matérielle choisie offre de nombreuses formes possibles d'implantation. Ainsi les transmission de données peuvent être effectuées à travers un réseau de terrain ou par des lignes point-à-point (ex : 4-20mA). Il est possible d'utiliser des automates pour la commande des actionneurs simples ou des actionneurs intelligents capables de dialoguer directement avec les capteurs (par le réseau de terrain ou les connexions point-à-point). Des relais permettent aussi la mise en repli des fonctions de pompage ou de chauffage et pourront être utilisés dans les fonctions de sécurité. Tous ces composants et leurs caractéristiques relatives à la sûreté de fonctionnement sont définis dans le tableau précédent.

Après avoir défini pour chaque composant quelles fonctions il est susceptible de supporter, l'étape de quantification et de sélection de la meilleure architecture peut débuter.

1.6 Quantification et optimisation

1.6.1 Correspondance des coûts

Afin de pouvoir comparer les différentes solutions, il est au préalable nécessaire d'exprimer la disponibilité et les probabilités d'occurrence d'événements en des coûts compatibles.

La disponibilité du système est évaluée selon deux taux pour lesquels on choisit d'appliquer la correspondance suivante :

Etat	Profits ou pertes correspondants
Disponible	1 % → 20
Indisponible	1 % → -5

Tableau V. 4 : Correspondance de la disponibilité en coût

Ces valeurs sont établies en fonction de la durée de vie du système. Ainsi, lorsque l'installation accomplit sa mission (ou est état de le faire) pendant un pour-cent de sa vie, les profits espérés sont de 20 unités monétaires, alors que son indisponibilité est assimilée à des pertes (5 unités monétaires) dues par exemple aux frais fixes de fonctionnement.

De même, trois événements sont associés à la défaillance de la mission du système. On fait correspondre pour chaque occurrence de l'un d'eux, une perte représentative des conséquences.

Evénement	Pertes engendrées par une occurrence
Arrêt intempestif	1 occurrence → -25
Fourniture du fluide à une mauvaise température	1 occurrence → -50
Accident (débordement, destruction du corps de chauffe, bouillonnement)	1 occurrence → -500

Tableau V.5 : Correspondance des occurrences des événements en coût

Ainsi on estime qu'un arrêt intempestif fait perdre 25 unités monétaires, alors que le non respect de la consigne de température en coûte le double et un accident vingt fois plus.

1.6.2 Recherche de la meilleure solution

La recherche de la meilleure architecture consiste à envisager chacune des configurations possibles (matérielles et répartition des traitements, au nombre de 2184 combinaisons possibles). Elle s'appuie sur une projection des fonctions élémentaires sur l'architecture matérielle préliminaire définie auparavant. Pour chaque projection, une vérification du respect des contraintes matérielles est effectuée ; elle contrôle que les équipements supportant deux fonctions liées par un flux de données sont physiquement "interconnectables". Le coût des composants utilisés puis les taux de disponibilité et le nombre a priori d'occurrences des événements pendant la durée de vie du système sont alors évalués pour aboutir à un coût final caractérisant la rentabilité du système.

Un outil informatique dédié à cette application a été développé pour effectuer cette recherche d'architecture.

Avec les valeurs choisies (coût des composants, probabilité de défaillance, transformation des taux en coût...) une seule architecture maximisant le coût a été trouvée. Les différents paramètres de cette solution sont les suivants :

		Taux et proba.	Coût
Coût des composants			-1010,0
Disponibilité	Taux de l'état "disponible"	94,12%	1882,4
	Taux de l'état "indisponible"	5,88%	-29,4
Fiabilité (nombre probable d'occurrences)	Arrêt intempestif	12,78	-319,5
	Température incorrecte	2,74	-137,0
	Accident	0,78	-392,3
		Total :	-5,8

Tableau V. 6 : Valeurs des critères de la meilleure architecture

Cette solution optimale est l'emploi d'une pompe "intelligente", d'une unité de chauffage "intelligente". La pompe intelligente (possédant sa propre unité de commande) est directement connectée aux détecteurs de niveau (haut, intermédiaire et "trop haut"), l'unité de chauffage intelligente est directement connectée au capteur de température. L'architecture préconisée est donc l'une des plus simples en nombre d'équipements employés. En effet, seul le détecteur de niveau "trop haut" (chargé d'éviter un débordement) pourrait être retiré sans que la mission du système en soit affectée (mais dès lors les probabilité d'occurrence de l'événement "Accident" atteindraient 4,22, rendant cette solution moins intéressante).

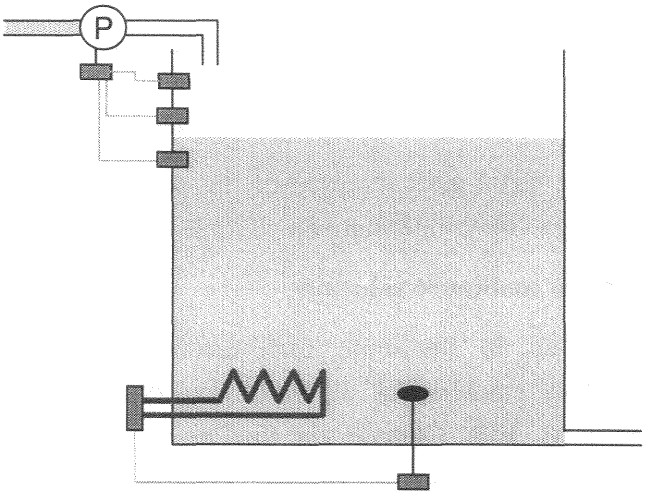


Figure V.5 : Schéma de l'architecture matérielle de la solution trouvée

Ainsi le coût des différents composants entraîne que l'emploi de lignes point-à-point apparaît comme plus rentable que celui d'un réseau de terrain, vis-à-vis du profit que l'on pourrait espérer de l'installation.

1.6.3 Les autres architectures

En approfondissant l'étude par le changement des pondérations affectées aux taux de disponibilité et à l'occurrence d'événements, d'autres architectures deviennent meilleures.

Les courbes suivantes définissent les domaines correspondant aux différentes architectures selon les pondérations appliquées à la transformations des critères en coût.

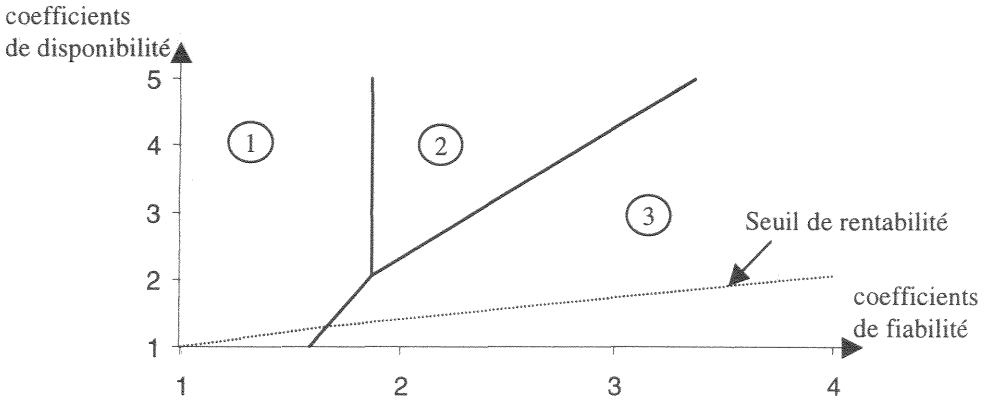
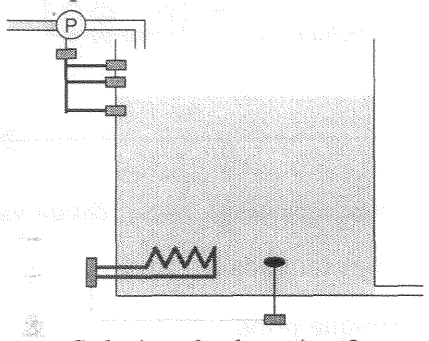


Figure V.6 : Domaines des différentes solutions

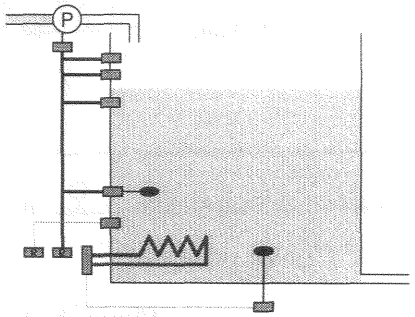
Le premier domaine correspond à l'architecture précédente. Les solutions associées aux domaines 2 et 3 utilisent un réseau de terrain.

Ces deux solutions se distinguent par l'emploi d'une pompe intelligente directement connectée au réseau de terrain. Le coût plus élevé de l'utilisation de ce réseau (de 225 avec les valeurs choisies) est compensé par un nombre probable de débordement plus réduit (0,62 au lieu de 0,78) qui s'explique par le fait que les défaillances des détecteurs de niveau ou du réseau sont plus facilement détectables par leur silence.

Enfin, la solution du domaine 3 intègre un plus grand nombre de fonctions de sécurité. Ainsi, un détecteur de niveau "trop bas" est connecté à un relais et un capteur de température à seuil commande un relais par l'intermédiaire du réseau de terrain. Ces nouveaux composants augmentent d'une part le coût d'installation du système mais, d'autre part, réduisent la probabilité d'occurrence de l'événement "débordement" (de 0,78 à 0,20) au détriment de celle "arrêt intempestif" (avec 14,05 au lieu de 12,78).



Solution du domaine 2



Solution du domaine 3

Figure V. 7 : Architecture matérielle des autres solutions

Le récapitulatif des résultats obtenus est le suivant :

		Solution 1	Solution 2	Solution 3
Coût des composants		-1010	-1235	-1350
Disponibilité	Taux de l'état "disponible"	94,12%	94,12%	91,32%
	Taux de l'état "indisponible"	5,88%	5,88%	8,68%
Fiabilité (nombre probable d'occurrences)	Arrêt intempestif	12,78	11,21	14,05
	Température incorrecte	2,74	2,74	2,74
	Accident	0,78	0,62	0,20

Tableau V.7 : Synthèse des différentes solutions

Ces résultats expriment ce que l'on connaît intuitivement. Plus le nombre de fonctions de sécurité est important, moindre est le risque d'incidents graves. Mais la disponibilité en est réduite et le nombre de mises en repli du système (suite à la défaillance d'un composant) augmente.

2 Seconde application

La seconde application de la méthode s'applique au processus thermique pilote bi-échangeurs du CRAN-ESSTIN. De précédents travaux [HER 97] s'y sont déjà intéressés pour la répartition des traitements mais la sûreté de fonctionnement y était traitée de manière relativement réduite.

2.1 Contexte

2.1.1 Le processus et sa mission

La figure suivante représente ce processus :

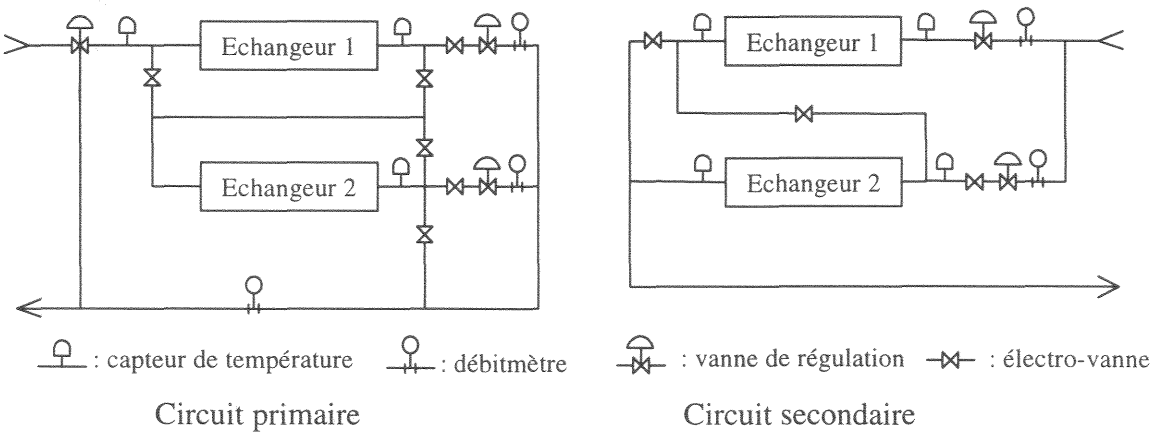


Figure V. 8 : Le processus thermique pilote

Deux circuits ouverts parcourus par de l'eau constituent le processus. Le premier est alimenté par un fluide chaud dont l'énergie thermique est transférée à un circuit secondaire grâce à deux échangeurs.

La fonction ou mission de ce processus est donc de contrôler la quantité de fluide traversant le circuit secondaire ainsi que sa température de sortie.

Si la régulation en débit du fluide dans le secondaire est réalisée grâce aux vannes qui le composent, celle de sa température de sortie nécessite de contrôler les flux thermiques dans les échangeurs. Ceci est accompli par la régulation en débit du fluide "chaud" dans le circuit primaire à l'aide des vannes qui le composent.

Le choix des capteurs et actionneurs et leur position est ainsi prédéfini. L'objectif de notre application est de déterminer le système d'automatisation contrôlant le transfert d'énergie.

Remarque : ce processus n'est qu'un sous-ensemble du véritable processus thermique pilote du CRAN-ESSTIN. Il intègre ainsi, une chaudière pour l'alimentation du primaire en fluide "chaud" et un radiateur pour refroidir le fluide du circuit secondaire.

2.1.2 Configuration des circuits

L'architecture matérielle de ce processus offre de nombreuses configurations. Il est ainsi possible de n'utiliser qu'un échangeur, ou bien les deux. Dans ce dernier cas, le parcours du fluide peut être série ou parallèle. Six configurations sont ainsi réalisables pour remplir la mission régulation en débit et en température du fluide de sortie du secondaire :

Configuration des circuits		Rendement du bilan thermique des échangeurs
Circuit primaire	Circuit secondaire	
parallèle	série	99 %
uniquement l'échangeur 2		97 %
parallèle	parallèle	95 %
uniquement l'échangeur 1		89 %
série	série	88 %
série	parallèle	66 %

Tableau V. 8 : table des rendements des bilans thermiques

Les caractéristiques des échangeurs étant différentes, un rendement thermique caractérise chacune de ces configurations.

Arbitrairement, on suppose que l'objectif est de profiter du meilleur bilan thermique. Ainsi, selon les possibilités du système (fonction des capteurs/actionneurs non-défaillants), la configuration retenue est celle qui donne le rendement le plus élevé parmi toutes celles disponibles.

2.1.3 Architecture matérielle du système d'automatisation

Le processus et son instrumentation existant déjà, l'objectif de l'application est donc de déterminer l'architecture opérationnelle du système de commande (sous système du système d'automatisation ne comportant pas les capteurs et actionneurs).

Arbitrairement, et en accord avec les composants actuellement disponibles, l'architecture matérielle choisie est composée de trois automates interconnectés par un réseau de terrain. Chaque automate est raccordé à une partie des capteurs et actionneurs de l'installation qu'il contrôle. Enfin, un pupitre de commande est connecté au réseau et permet aux opérateurs de fournir les consignes de fonctionnement et de surveiller le comportement du système.

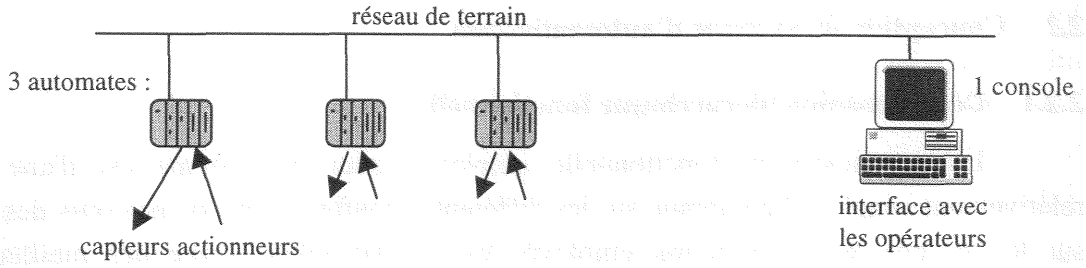


Figure V. 9 : Architecture matérielle du système de commande

L'objectif de la conception consiste donc à déterminer pour chacun des automates quels capteurs et actionneurs leur sont dédiés et quels traitements ils doivent supporter.

Chaque automate comporte un nombre limité d'entrées/sorties. Arbitrairement, les valeurs suivantes ont été choisies :

	Nombre maxima de connexions pour un automate	Nombre total de composants utilisés sur le processus
Actionneurs analogiques	4	5
Actionneurs tout ou rien	4	12
Capteurs analogiques	4	9

Tableau V. 9 : Nombre d'entrées/sorties par automate
et nombre de capteurs et d'actionneurs utilisés sur le processus

Concernant leur paramètres de sûreté de fonctionnement, les valeurs suivantes ont été choisies :

- disponibilité : 95 %
- nombre d'occurrences de la défaillance "arrêt et mise en repli" : 5 fois
- nombre d'occurrences de la défaillance "fonctionnement incohérent" : 1 fois

Le nombre d'occurrences est relatif à la durée de vie du système. Deux modes de défaillance ont été considérés. Le premier correspond à un arrêt de l'automate pour lequel ses sorties prennent une valeur par défaut. Cette valeur dans le cas du processus thermique entraîne la fermeture des vannes commandées qu'elles soient analogiques ou "tout ou rien". A titre d'exemple, ce mode de défaillance peut être consécutif à la défaillance de l'alimentation de l'automate conduisant à son arrêt. Le second mode de défaillance correspond à un comportement anormal de l'automate. Faisant par exemple suite à une défaillance du processeur de l'automate, ce mode se caractérise par des valeurs des sorties ne correspondant pas à celles spécifiées et par une mauvaise lecture des entrées. Pour notre système, cela revient à des ouvertures ou fermetures intempestives des vannes et par une mauvaise retransmission sur le réseau des mesures fournies par les capteurs.

2.2 Conception du système d'automatisation

2.2.1 Décomposition hiérarchique fonctionnelle

La décomposition fonctionnelle employée pour ce système est d'une structure relativement simple. Elle s'appuie sur les différentes configurations de parcours des fluides et sur le fait que la configuration employée est choisie selon l'ordre des meilleurs bilans thermiques (cf. : Tableau V. 9). Ainsi, la mission (ou fonction) principale "réguler en température et en débit le fluide traversant le circuit secondaire" est décomposée en 6 fonctions "réguler en utilisant la configuration numéro n". Ces fonctions sont elle-même décomposées en trois sous-fonctions. La première correspond à "mesurer et actionner les composants actifs à la configuration n", la deuxième à "mettre au repos les actionneurs inactifs à la configuration n", la troisième "surveiller le processus". La première sous-fonction regroupe l'ensemble des fonctions élémentaires "actionner la vanne x" et "effectuer la mesure y" participant soit à la régulation pour les capteurs et les vannes analogiques soit à la

configuration du parcours des fluides avec l'ouverture de certaines vannes "tout ou rien". La seconde regroupe les fonctions élémentaires "actionner la vanne x" dont le rôle "tout ou rien" est de participer à la configuration du parcours des fluides par la fermeture de ces vannes. Enfin la dernière sous-fonction regroupe plusieurs sous-fonctions de surveillance, chargées chacune de surveiller le bon fonctionnement de certains actionneurs ou capteurs et est décomposée en plusieurs fonctions élémentaires "effectuer la mesure z".

2.2.2 Aspects relatifs à la disponibilité

En accord avec la mission du système (fournir un fluide régulé en débit et en température), trois états ont été retenus. Le premier qualifié par "disponible nominal" permet au système de remplir cette mission avec un bilan thermique supérieur à 90%. Cette valeur peut être atteinte dans trois configurations présentées précédemment. Le deuxième état qualifié de "disponible dégradé" permet aussi de remplir la mission mais avec un bilan thermique moindre. Enfin, le dernier état qualifié "d'indisponible" ne permet pas au système d'accomplir sa mission suite à un trop grand nombre de composants en état de panne.

La correspondance choisie entre la probabilité de présence dans ces états et les coûts est la suivante :

Etat	Correspondance en coût
Disponible nominal	1 % → 10
Disponible dégradé	1 % → 8
Indisponible	1 % → -1

Tableau V. 10 : Table de correspondance entre disponibilité et coût

Concernant la décomposition fonctionnelle, les relations décrivant indirectement la disponibilité du système selon celle des fonctions élémentaires sont définies de la manière suivante. Le système n'est disponible nominal que si au moins une des 3 fonctions "réguler en utilisant la configuration numéro n" (celles dont le rendement thermique est supérieure à 90 %) est disponible. Dans le cas contraire, il est disponible dégradé que si au moins une des 3 autres fonctions est disponible. Pour une configuration, ces sous-fonctions ne sont disponibles que si leur sous-fonction "mesurer et actionner les composants actifs à la configuration n" est disponible. Celle-ci n'est disponible que si toutes les fonctions élémentaires qu'elle regroupe le sont aussi. Les autres sous-fonctions n'interviennent pas pour les aspects relatifs à la disponibilité. En effet, l'indisponibilité des fonctions de surveillance n'empêche pas un fonctionnement correct du système. De même, l'indisponibilité de certaines fonctions élémentaires appartenant à "mettre au repos les actionneurs inactifs à la configuration n" ne rendent pas le système indisponible, car on considère que leurs indisponibilités entraînent pour les vannes concernées une position par défaut, en l'occurrence leur fermeture. Ces deux dernières sous-fonctions ont cependant une influence dans la propagation des défaillances comme il est montré ci-après.

2.2.3 Aspects relatifs à la fiabilité

En accord avec les modes de défaillance des automates (seuls composants dont on étudie les défaillances), trois types d'événements peuvent influer sur la mission du système. Le premier correspond à un "micro-arrêt" ; il est consécutif à une défaillance détectée d'un des automates et pour laquelle les autres automates peuvent retrouver une nouvelle configuration du parcours des fluides assurant la mission. Le deuxième événement correspond à un "arrêt" ; il est lui aussi consécutif à la défaillance détectée d'un des automates mais pour laquelle il n'existe pas de reconfiguration permettant une reprise de la fonction principale. Enfin, le dernier événement est "fonctionnement anormal" du système ; suite à la défaillance d'un automate, le système n'est pas en mesure de détecter son mauvais comportement et peut donc continuer de fonctionner sans assurer sa mission de régulation et sans en avertir les opérateurs.

L'association de ces événements à des coûts est la suivante et rend compte de la gravité de l'occurrence :

Evénement	coût engendré par une occurrence
Micro-arrêt	1 occurrence → -1
Arrêt	1 occurrence → -10
Fonctionnement anormal	1 occurrence → -100

Tableau V. 11 : Association d'un coût aux occurrences d'événements

Les relations entre événements au sein de la décomposition fonctionnelle suivent les règles suivantes. L'arrêt d'un automate provoque l'arrêt de toutes les fonctions élémentaires qu'il supporte. Si au moins une de ces fonctions élémentaires participe à une fonction "mesurer et actionner les composants actifs à la configuration n", cette dernière génère un événement "arrêt". Concernant la fonction principale, l'arrêt de la configuration courante entraîne soit un micro-arrêt si une autre configuration peut prendre le relais, soit un arrêt si aucune n'est disponible. Le comportement anormal d'un automate (second mode de défaillance traité) provoque l'événement "mauvais comportement" pour toutes les fonctions élémentaires qu'il supporte. Si au moins une de ces fonctions participe à "mesurer et actionner les composants actifs à la configuration n" ou à "mettre au repos les actionneurs inactifs à la configuration n", la fonction "réguler en utilisant la configuration numéro n" en est influencée. S'il s'agit de la configuration courante, l'événement généré est soit "arrêt" dans la mesure où une fonction de surveillance est disponible et capable de détecter la défaillance, soit "mauvais comportement" si aucune n'a pu déceler la défaillance. Les conséquences sont pour la fonction principale soit "fonctionnement anormal" pour un "mauvais comportement" de la configuration courante, soit "arrêt" ou "micro-arrêt" si une autre configuration est disponible.

2.3 Recherche de la meilleure répartition

2.3.1 Algorithme de recherche

Un algorithme d'inspiration génétique est employé pour traiter ce problème d'optimisation. Ces principales caractéristiques sont les suivantes :

- Il travaille avec une population constante de 100 individus. Ainsi, après chaque génération d'un nouvel individu, la notation de celui-ci est comparée à celles des plus faibles de la population. S'il est considéré meilleur, il remplace un des individus dont la notation est la plus basse.
- Les contraintes sont traitées lors du croisement. Ainsi, lorsqu'un nouvel individu est généré par croisement, ses gènes sont choisis un par un. A chaque fois, on vérifie que les fonctions élémentaires affectées à chaque automate ne dépassent pas ses capacités. Si ces contraintes sont respectées, le gène suivant est traité. Dans le cas contraire, une autre valeur correspondant au gène traité est choisie.

Due au faible nombre de composants, l'évaluation des solutions envisagées est effectuée en étudiant les différents états de panne. Ainsi pour 3 composants dont les états sont soit disponible soit indisponible, il y a 8 états possibles pour le système et leurs 2 modes de défaillance ("arrêt et mise en repli" et "fonctionnement incohérent") conduit à traiter 24 événements (8×2×3/2).

2.3.2 Résultats obtenus

La meilleure notation trouvée est de 949,58. Cependant l'emploi d'un algorithme stochastique ne garantit pas qu'il s'agisse de l'optimum. De nombreux essais semblent indiquer que cette valeur ne peut être dépassée. Parmi les 20 architectures recensées (ou 120 si l'on considère les combinaisons entre les 6 automates) ayant obtenu cette valeur, une seule est présentée ci-après. Le schéma la représente en donnant pour chaque instrument le numéro de l'automate auquel il est connecté.

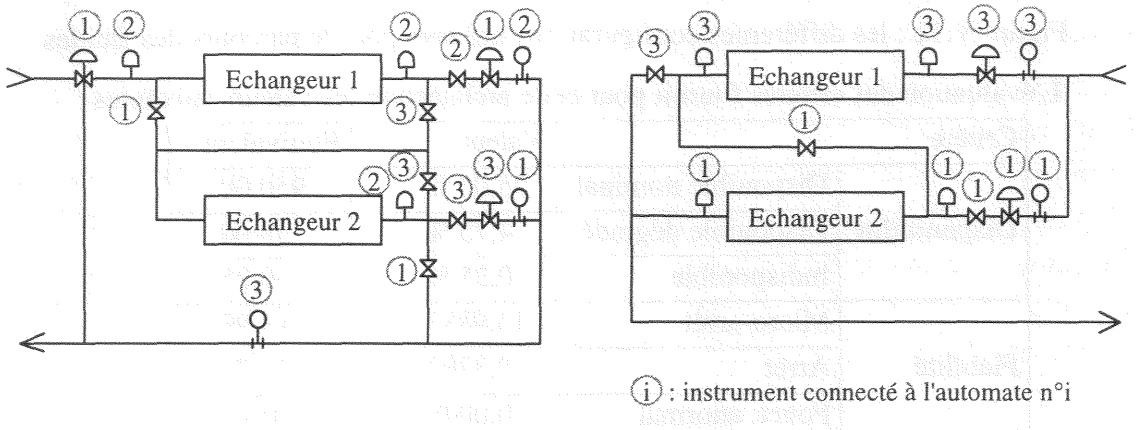


Figure V.10 : Architecture finale du raccordement des instruments

Selon les pannes et défaillances, le comportement de ce système est le suivant :

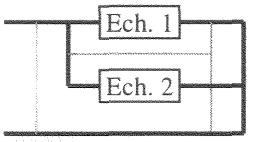
Etat des automates			Conséquence de l'arrêt de l'automate			Conséquence du fonctionnement incohérent de l'automate			Configuration
1	2	3	1	2	3	1	2	3	
dispo.	dispo.	dispo.	µA.	µA.	µA.	Ar.	µA.	Ar.	I
dispo.	dispo.	indispo.	Ar.	∅		Ar.	µA.		II
dispo.	indispo.	dispo.	µA.		∅	Ar.		Ar.	II
dispo.	indispo.	indispo.	Ar.			FA			II
indispo.	dispo.	dispo.		∅	Ar.		∅	Ar.	III
indispo.	indispo.	dispo.			Ar.			FA	III

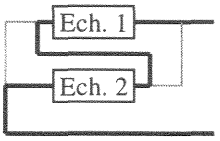
Légende : ∅ : aucune conséquence, µA. : micro-arrêt, Ar. : arrêt, FA : fonctionnement anormal.

Tableau V.12 : Comportement du système de commande aux défaillances

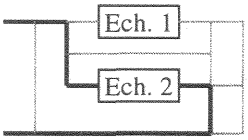
Les différentes configurations du parcours des fluides sont les suivantes :

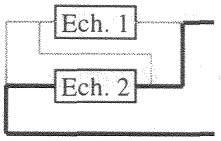
Configuration I (bilan : 99%):



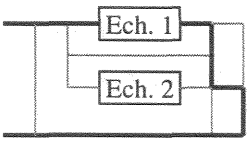


Configuration II (bilan : 97%):





Configuration III (bilan : 89%):



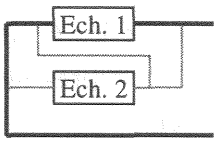


Figure V.11 : les différentes configurations utilisées pour le parcours des fluides

L'évaluation des critères fournit pour cette architecture les valeurs suivantes :

Critère		Valeur	Equivalent
Disponibilité	Disponible nominal	95,00 %	950,00
	Disponible dégradé	4,75 %	38,00
	Indisponible	0,25 %	-0,25
Fiabilité	Micro-arrêt	13,9887	-13,99
	Arrêt	2,3702	-23,70
	Fonct. anormal	0,0048	-0,48
		Total :	949,58

Tableau V.13 : Valeurs des critères et notation pour la solution proposée

3 Conclusion

Les deux applications présentées essaient de mettre en valeur les intérêts et les différentes possibilités de la méthodologie proposée. Applicable à un système complet d'automatisation (application 1) ou seulement à une sous-partie (application 2), elle permet la recherche de la meilleure architecture aussi bien matérielle (organisation de composants) qu'opérationnelle (organisation des fonctions élémentaires au sein de cette architecture matérielle). Elle requiert l'évaluation de critères permettant la comparaison des solutions envisagées (aspect relatif), mais fournissant aussi au concepteur une image de la robustesse du système conçu face aux défaillances possibles (aspect absolu).

En revanche, l'importance des informations à adjoindre à la décomposition fonctionnelle nécessaires pour permettre l'évaluation de la sûreté de fonctionnement, peut rendre cette méthodologie d'un emploi relativement lourd. Par ailleurs, elle impose l'utilisation d'un outil informatique pour traiter correctement le nombre important de combinaisons à envisager et d'informations à traiter. Malgré cela, cette méthodologie nous semble offrir une aide appréciable dans la conception de systèmes d'automatisation en vue de leur optimisation.

Bibliographie

[HER 97] F. HERMANN, B. CONRARD, Conception d'architecture distribuée avec allocations de tâches : application à un processus thermique équipé d'un réseau de terrain, 2ème Symposium International sur "les capteurs et leurs applications industrielles", Gabès (Tunisie), mai 1997.

Conclusion

Le travail présenté propose une démarche pour la conception des systèmes d'automatisation. Cette démarche procède de la façon suivante :

- une modélisation de l'architecture fonctionnelle traduit les spécifications attendues et détermine une organisation de traitements élémentaires que devront réaliser les composants du système final,
- un enrichissement de cette architecture fonctionnelle par des informations relatives à la sûreté de fonctionnement décrit le comportement du système lors ou en présence de défaillances des traitements élémentaires,
- une architecture matérielle préliminaire établie par le concepteur définit un ensemble d'organisation de composants susceptibles d'accueillir les traitements élémentaires identifiés précédemment,
- la projection des traitements élémentaires sur cette architecture matérielle consiste en une optimisation combinatoire où l'emploi d'algorithmes d'allocation est nécessaire. L'évaluation des solutions envisagées est réalisée par l'association de critères en une métrique unique, ceux-ci rendant compte du coût des équipements utilisés, de la capacité du système à accomplir sa mission et de l'estimation du nombre d'incidents susceptibles de survenir. Le mécanisme d'allocation a ici un double objectif ; outre décider du choix de la répartition des traitements, il détermine indirectement l'architecture matérielle convenant le mieux.

L'élaboration de cette méthodologie repose sur une agrégation de différentes techniques et outils. Ainsi l'architecture fonctionnelle peut être établie à l'aide de modélisation SADT ou SART. L'emploi de diagrammes de décision binaire peut permettre une évaluation globale de la sûreté de fonctionnement a priori par la quantification de différents critères. Enfin des algorithmes d'optimisation tels que ceux d'inspiration génétique offrent de bons outils pour déterminer la meilleure architecture ou tout au moins une architecture acceptable aux yeux des concepteurs.

Nous pensons qu'une telle démarche peut offrir une aide conséquente aux concepteurs. Ainsi, même si seulement une partie de tous les critères et toutes les contraintes utiles à l'évaluation de la qualité de solutions de systèmes d'automatisation est prise en compte, les critères choisis fournissent une image suffisamment complète pour permettre d'effectuer des pré-études correctes et aptes à fournir un premier schéma directeur dans la réalisation de systèmes d'automatisation.

L'aspect modulaire de la démarche permet d'approfondir ou de compléter l'évaluation de critères ou la prise en compte de contraintes non ou peu traitées.

- Ainsi concernant l'allocation des traitements, la vérification du respect des contraintes temporelles est relativement restreinte. Ainsi, les aspects temporels ne sont traités que comme une charge de traitement pour laquelle il convient de vérifier qu'elle ne dépasse pas une valeur limite sur chacun des composants (instruments et réseaux). Améliorer la prise en compte de ces contraintes pourrait alors permettre de spécifier des mécanismes de priorité ou de synchronisation susceptibles de profiter au mieux des capacités des composants à coopérer.

- La modélisation du comportement des systèmes d'automatisation est une "caricature" du réel. Ainsi, l'étude de la sûreté de fonctionnement est restreinte à une évaluation de probabilités d'états ou d'occurrence événements. Les aspects temporels pouvant intervenir dans des mécanismes complexes de sécurité (ex : temporisation) mériteraient d'être plus explicitement définis et pris en compte, aussi bien dans la modélisation que dans la quantification.

- Le rôle des opérateurs pourrait être développé. En effet, leur action ne se limite pas à simplement fournir des consignes de commande et à remettre en état les équipements défectueux. Certainement plus difficilement "modélisable", ils participent à la vie du système grâce à des activités de surveillance, mais aussi éventuellement par de fausses manœuvres en étant alors causes de défaillance.

- Enfin les différentes politiques possibles de maintenance ne sont prises en compte qu'au niveau des paramètres de sûreté de fonctionnement de chaque composant (disponibilité, fiabilité...). Il pourrait être intéressant de mieux exprimer les actions possibles (ex. : contrôle périodique pour la détection de faute cachée) afin de les intégrer dans l'évaluation des critères de sûreté de fonctionnement et éventuellement que les paramètres associés soient déterminés lors de la phase d'optimisation (ex. : période entre deux tests).

Tous les aspects décrits ici sont autant de pistes de réflexion pour le développement de travaux futurs.

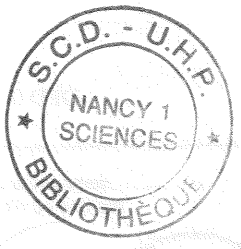
Pour la thèse de Blaise CONRARD,

Il n’y a pas de page 121.

Annexe

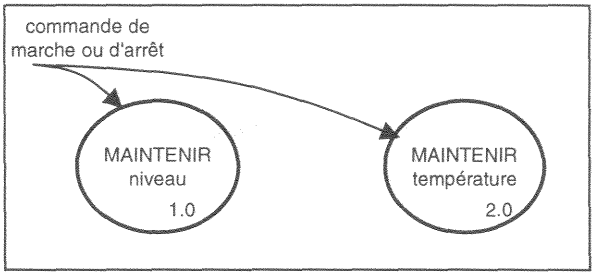
I- Décomposition fonctionnelle de l'application cuve chauffante

Décomposition fonctionnelle et hiérarchique du système de maintien du niveau et de la température d'un fluide dans une cuve.

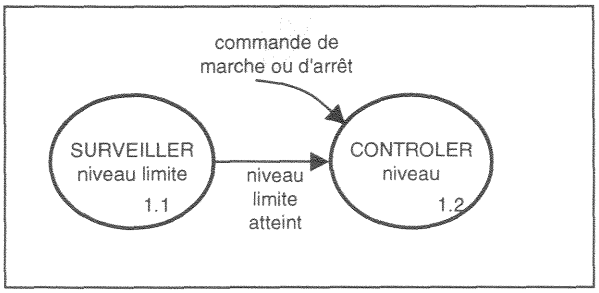


- MAINTENIR le niveau et la température de la cuve
  - MAINTENIR niveau
    - CONTROLER niveau
      - MESURER niveau
        - MESURER niveau haut
          - *LIRE niveau haut*
          - *TRANSMETTRE lecture niveau haut*
        - MESURER niveau intermédiaire
          - *LIRE niveau intermédiaire*
          - *TRANSMETTRE lecture niveau intermédiaire*
      - COMMANDER pompage
        - *POMPER*
        - COMMANDER pompe
          - *COMMANDER pompe suivant niveaux*
          - *TRANSMETTRE ordre de pompage*
          - *CONTROLER énergie électrique pour la pompe*
    - SURVEILLER niveau limite atteint *(optionnel)*
      - *MESURER niveau limite*
      - *TRANSMETTRE ordre d'arrêt*
      - *ARRETER pompe*
  - MAINTENIR température
    - CONTROLER la température
      - *MESURER température*
        - *LIRE mesure de température*
        - *TRANSMETTRE mesure de température*
      - COMMANDER chauffage
        - REGULER chauffage
          - *CALCULER commande de chauffage*
          - *TRANSMETTRE commande de chauffage*
          - *CONTROLER énergie électrique pour la résistance*
        - *CHAUFFER*
    - SURVEILLER le niveau *(optionnel)*
      - *MESURER niveau trop bas*
      - *TRANSMETTRE mesure de niveau bas*
      - *ARRETER le chauffage*
    - SURVEILLER la température *(optionnel)*
      - *MESURER température trop élevée*
      - *TRANSMETTRE mesure de niveau bas*
      - *ARRETER LE CHAUFFAGE*

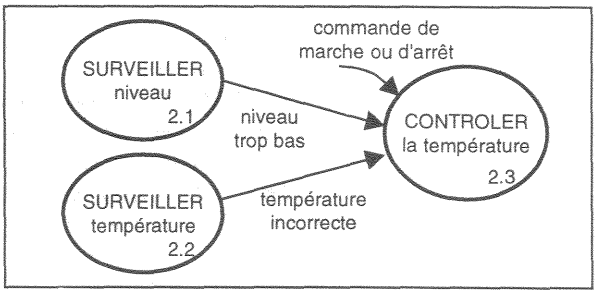
II- Décomposition fonctionnelle du système d'automatisation de contrôle du niveau et de la température d'une cuve



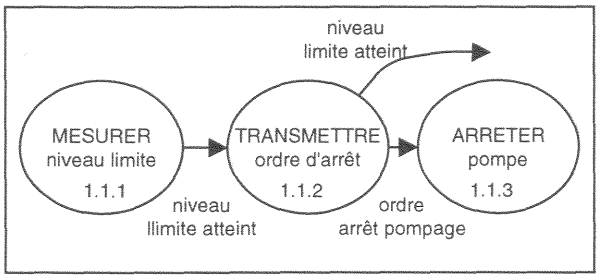
DFD 0 : Maintenir le niveau et la température d'un fluide dans une cuve



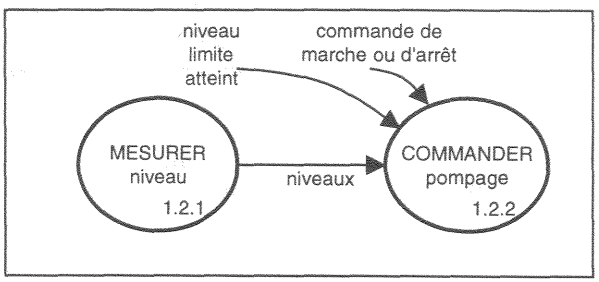
DFD 1.0 : Maintenir niveau



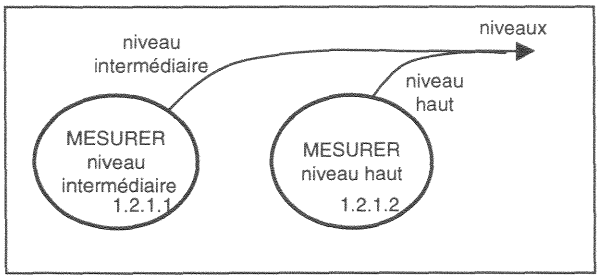
DFD 2.0 : Maintenir température



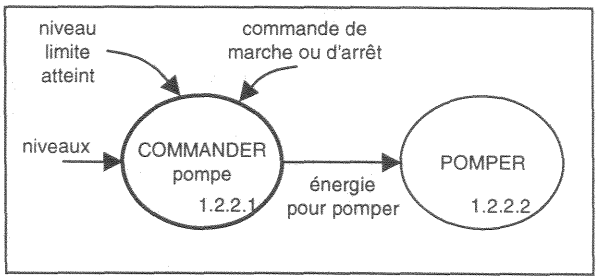
DFD 1.1 : Surveiller niveau limite



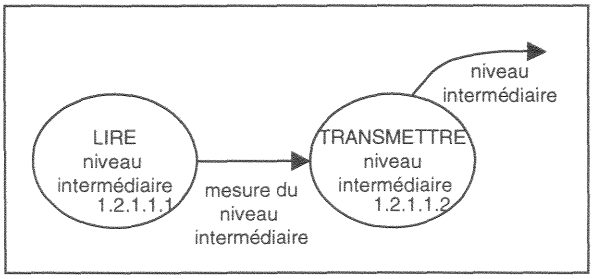
DFD 1.2 : Contrôler niveau



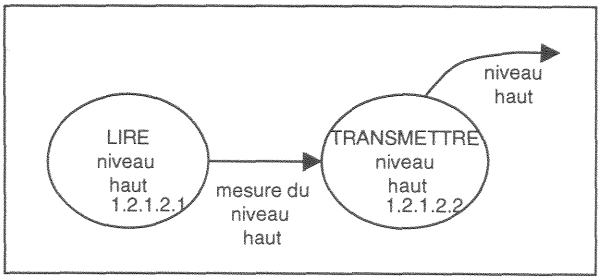
DFD 1.2.1 : Mesurer niveau



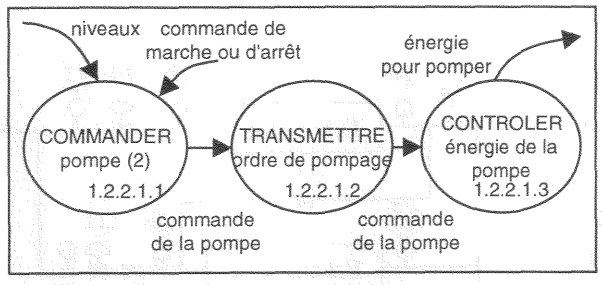
DFD 1.2.2 : Commander pompage



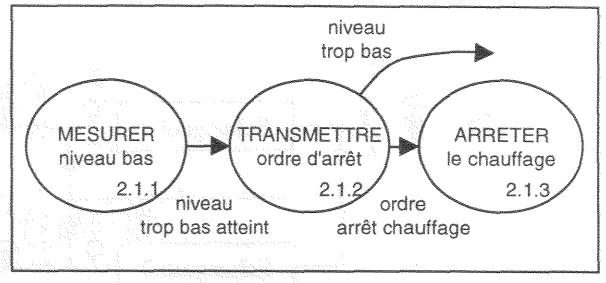
DFD 1.2.1.1 : Mesurer niveau intermédiaire



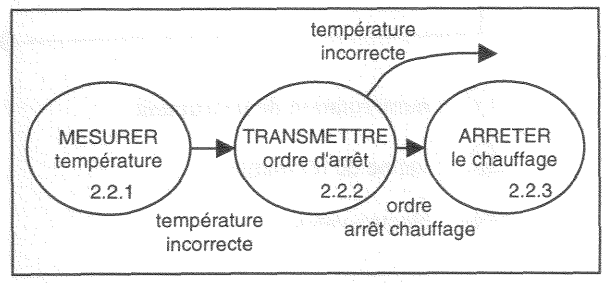
DFD 1.2.1.2 : Mesurer niveau haut



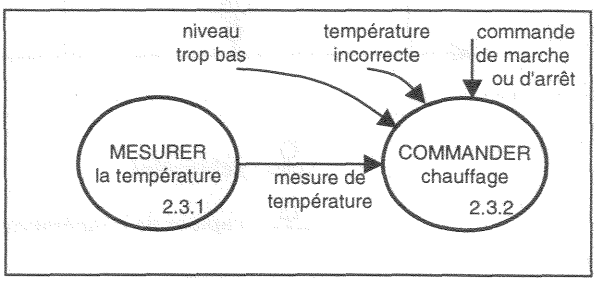
DFD 1.2.2.1 : Commander pompe



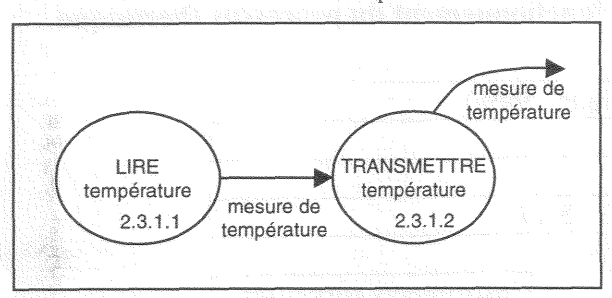
DFD 2.1 : Surveiller niveau



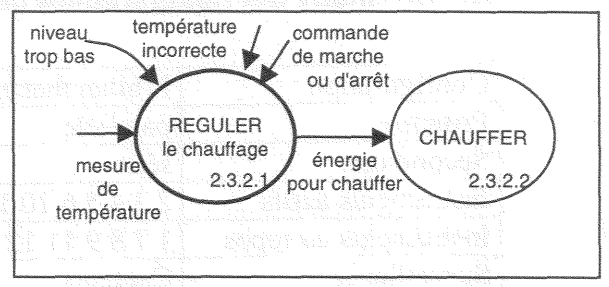
DFD 2.2 : Surveiller température



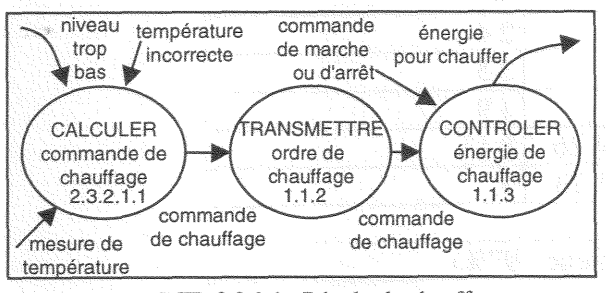
DFD 2.3 : Contrôler la température



DFD 2.3.1 : Mesurer niveau haut

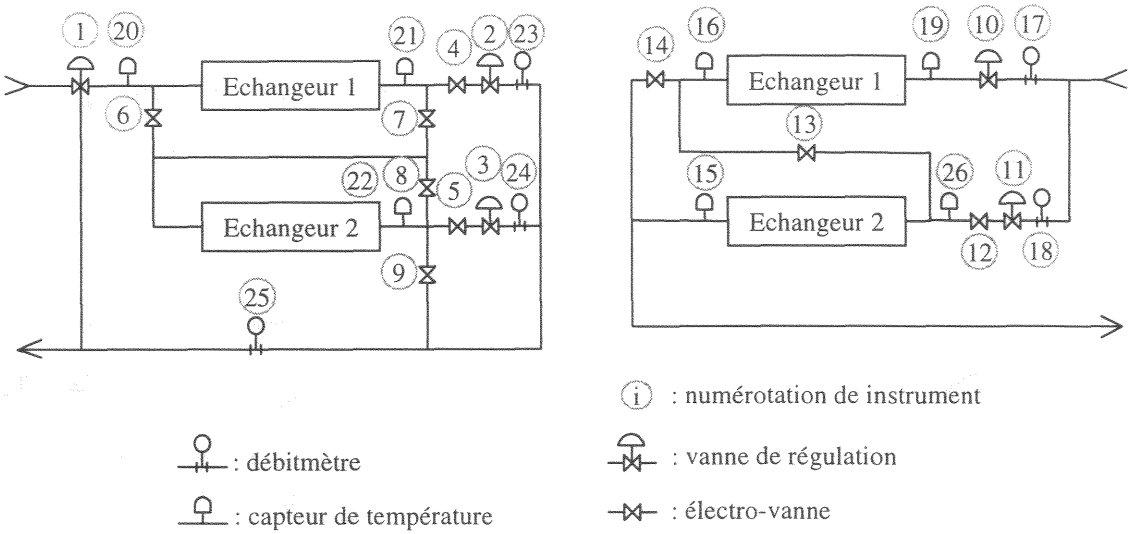


DFD 2.3.2 : Commander chauffage



DFD 2.3.2.1 : Réguler le chauffage

III- Instrumentation du processus thermique



III- Inventaire des configurations de fonctionnement du processus thermique

Configuration	I (bilan thermique : 99%)	
Primaire	parallèle	
Secondaire	série	
Instruments actifs	2 3 4 5 6 10 13 15 17	
Instruments au repos	1 7 8 9 11 12 14	
Surveillance	Capteurs	Instruments surveillés
	17	10
	23	2 4
	24	3 5
	23 24 25	1 9
	15 18 20 22 24 26	8 13
	16 17 19 20 21 23	7

Configuration	II (bilan thermique : 97%)	
Primaire	deuxième échangeur	
Secondaire	deuxième échangeur	
Instruments actifs	1 6 9 11 12 15 18	
Instruments au repos	2 4 3 5 7 8 10 13 14	
Surveillance	Capteurs	Instruments surveillés
	15	1 6
	17	10 13 14
	18	11
	23	2 4
	24	3 5
	25	1 9
	15 18 20 22 24 26	7 8

Configuration	III (bilan thermique : 97%)	
Primaire	deuxième échangeur	
Secondaire	deuxième échangeur	
Instruments actifs	3 5 6 11 12 15 18	
Instruments au repos	1 2 4 7 8 9 10 13 14	
Surveillance	Capteurs	Instruments surveillés
	17	10 13 14
	18	11 12
	23	2 4
	24	1 3 5
	25	1
	15 18 20 22 24 26	7 8
	24 25	9

Configuration	IV (bilan thermique : 95%)	
Primaire	parallèle	
Secondaire	parallèle	
Instruments actifs	2 3 4 5 6 10 11 12 14 15 16 17 18	
Instruments au repos	1 7 8 9 13	
Surveillance	Capteurs	Instruments surveillés
	17	10 13 14
	18	11 12
	23	2 4
	24	3 5 6
	23 24 25	1 9
	16 17 19 20 21 23	7
	15 18 20 22 24 26	13

Configuration	V (bilan thermique : 89%)	
Primaire	premier échangeur	
Secondaire	premier échangeur	
Instruments actifs	2 4 10 14 16 17	
Instruments au repos	1 3 5 6 7 9 11 12 13	
Surveillance	Capteurs	Instruments surveillés
	17	10 14
	18	11 12
	23	2 4
	24	3 5
	24 25	9
	16 17 19 20 21 23	6 7 9

Configuration	VI (bilan thermique : 89%)	
Primaire	premier échangeur	
Secondaire	premier échangeur	
Instruments actifs	3 5 7 8 10 14 16 17	
Instruments au repos	1 2 4 6 9 11 12 13	
Surveillance	Capteurs	Instruments surveillés
	17	10 14
	18	11 12
	23	2 4
	24	3 5 7
	24 25	2 4 9
	20 21 23 16 19 17	6

Configuration	VII (bilan thermique : 89%)	
Primaire	premier échangeur	
Secondaire	premier échangeur	
Instruments actifs	1 7 8 9 10 14 16 17	
Instruments au repos	2 3 4 5 6 11 12 13	
Surveillance	Capteurs	Instruments surveillés
	17	10 14
	18	11 12
	23	2 4
	24	3 5
	25	1 9 7
	16 17 19 20 21 23	6

Configuration	VIII (bilan thermique : 88%)	
Primaire	série	
Secondaire	série	
Instruments actifs	1 7 9 10 13 15 17	
Instruments au repos	2 3 4 5 6 8 11 12 14	
Surveillance	Capteurs	Instruments surveillés
	17	10 13
	18	11 12
	23	2 4
	24	3 5
	25	1 9 7
	16 17 19 20 21 23	6
	15 18 20 22 24 26	14 8

Configuration	IX (bilan thermique : 88%)	
Primaire	série	
Secondaire	série	
Instruments actifs	3 5 7 10 13 15 17	
Instruments au repos	1 2 4 6 8 9 11 12 14	
Surveillance	Capteurs	Instruments surveillés
	17	10 13
	18	11 12
	23	2 4
	24	3 5
	25	1 3 5 7
	20 21 23 16 19 17	6
	20 22 24 15 18 26	14 8
	24 25	2 4 9

Configuration	X (bilan thermique : 66 %)	
Primaire	série	
Secondaire	parallèle	
Instruments actifs	1 7 9 10 11 12 14 15 16 17 18	
Instruments au repos	2 4 3 5 6 8 13	
Surveillance	Capteurs	Instruments surveillés
	17	10 14
	18	11 12
	23	2 4
	24	3 5
	25	1 7 9
	16 17 19 20 21 23	6
	15 18 20 22 24 26	8 13

Configuration	XI (bilan thermique : 66 %)	
Primaire	série	
Secondaire	parallèle	
Instruments actifs	3 5 7 10 11 12 14 15 16 17 18	
Instruments au repos	1 2 4 6 8 9 13	
Surveillance	Capteurs	Instruments surveillés
	17	10 14
	18	11 12
	23	2 4
	24	3 5 9
	25	1 3 5 7
	16 17 19 20 21 23	6
	15 18 20 22 24 26	8 13
	24 25	2 4 9

Monsieur **Blaise CONRARD**

DOCTORAT DE L'UNIVERSITE HENRI POINCARE, NANCY-I

en AUTOMATIQUE

VU, AP PROUVÉ ET PERMIS D'IMPRIMER

Nancy, le 19 octobre 1999 n° 263

Le Président de l'Université

UNIVERSITÉ HENRI POINCARÉ, NANCY LE PRÉSIDENT

Claude BURLET

UNIVERSITÉ HENRI POINCARÉ · NANCY 1

3C-F9

24-30, rue Lionnois B.P. 3069 54013 Nancy cedex - Tél. 03 83 85 48 00 - Fax 03 83 85 48 48

Adresse électronique : _____ @uhp.u-nancy.fr

CONTRIBUTION A L'EVALUATION QUANTITATIVE DE LA SURETE DE FONCTIONNEMENT DES SYSTEMES D'AUTOMATISATION EN PHASE DE CONCEPTION

Résumé :

Les travaux exposés dans ce mémoire portent sur l'évaluation de la sûreté de fonctionnement explicitement utilisée comme un critère lors de la conception des systèmes d'automatisation. Ce document est organisé de la manière suivante :

Une première partie définit les différents termes relatifs aux systèmes d'automatisation et aux systèmes d'automatisation à intelligence distribuée puis la problématique générale de leur conception. Ensuite, la sûreté de fonctionnement et les principales notions s'y rattachant sont introduites.

Une deuxième partie présente les différentes représentations utilisées lors de la démarche de conception.

Une troisième partie aborde les principales méthodes employées pour quantifier les différents aspects de la sûreté de fonctionnement. Parmi celles-ci, celles relatives aux aspects disponibilité et fiabilité/sécurité sont plus particulièrement développées. Enfin, les arbres de décision binaire sont abordés comme outil pour la quantification de ces aspects.

Une quatrième partie présente la méthode proposée pour la conception des systèmes d'automatisation. Celle-ci s'appuie sur une recherche combinatoire afin d'optimiser l'architecture à concevoir prenant en compte ces critères relatifs à la sûreté de fonctionnement.

Une dernière partie illustre la démarche proposée en l'appliquant à deux exemples.

Mots clés : système d'automatisation, conception d'architectures, répartition de tâches, quantification de la sûreté de fonctionnement.

CONTRIBUTION TO THE QUANTITATIVE EVALUATION OF DEPENDABILITY DURING THE DESIGN PHASE OF AUTOMATION SYSTEMS

Abstract :

This work concerns the evaluation of dependability used as an explicit criterion for the design of automation systems. This document is organised in the following way:

A first part defines the concepts of automation systems and distributed intelligence base automation systems, and then lays down the general problem about their design. Dependability and the main notions about dependability are introduced.

A second part presents several representations which may be used when designing a system.

A third part deals with the usual methods used to quantify the different aspects of dependability. Among them, the methods which deal with availability and reliability/security are more developed. Finally, the binary decision diagrams are tackled as a tool to evaluate these aspects of dependability.

A fourth part presents the proposed method for the design of automation systems. It uses a combinatory research in order to optimise the designed architecture thanks to criteria about dependability.

A last part illustrates the proposed method on two examples.

Keywords : automation system, architecture design, tasks allocation, evaluation of dependability.