HAL
open science

# Vers une méthodologie globale de conception et de génération semi-automatique des IHM pour les systèmes industriels

Faouzi Moussa

## To cite this version:

Faouzi Moussa. Vers une méthodologie globale de conception et de génération semi-automatique des IHM pour les systèmes industriels. Informatique [cs]. Faculté des Sciences Mathématiques, Physiques et Naturelles de Tunis. Université de Tunis El Manar, 2005. tel-01954655

**HAL Id: tel-01954655**

**https://hal.science/tel-01954655v1**

Submitted on 13 Dec 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE DE TUNIS EL MANAR
FACULTE DES SCIENCES DE TUNIS

ECOLE DOCTORALE EN INFORMATIQUE
Dossier présenté en vue de l'obtention de

# L'Habilitation Universitaire en Informatique

par

Faouzi MOUSSA

Docteur en Informatique
Maître-Assistant à la Faculté des Sciences de Tunis

Thème des travaux

# Vers une méthodologie globale de conception et de génération semi-automatique des IHM pour les systèmes industriels

## Rapport de synthèse

Soutenu le 13 juin 2005, devant le jury d'examen

| | | |
|---|---|---|
| MM. | Ahmed MAHJOUB | Président |
| | Bertrand DAVID | Rapporteur |
| | Patrick GIRARD | Rapporteur |
| | Mohamed MOALLA | Examinateur |
| | Riadh ROBBANA | Examinateur |
| | Christophe KOLSKI | Invité |

**INTRODUCTION GENERALE** ..... **4**
**CHAPITRE 1 : DE LA CONCEPTION DES INTERFACES HOMME-MACHINE** **7**
**1.1 Système Homme-Machine et Interface Homme-Machine** ..... **7**
1.1.1 Le système Homme-Machine (SHM) ..... 7
1.1.2 L'Interface Homme-Machine (IHM) ..... 8
**1.2 Conception, vérification et génération semi-automatique des IHM** ..... **9**
1.2.1 La psychologie cognitive dans la conception des IHM ..... 9
1.2.1.1 La modélisation de l'opérateur humain ..... 9
1.2.1.2 L'analyse de la tâche humaine dans la conception des IHM ..... 14
1.2.2 L'ergonomie cognitive dans la conception des IHM ..... 17
1.2.3 Les cycles de vie Génie Logiciel « enrichis » dans la conception des IHM ..... 18
1.2.3.1 Modèle de Long ..... 19
1.2.3.2 Modèle en étoile ..... 19
1.2.3.3 Modèle ∇ (Nabla) ..... 19
1.2.4 Approches et modèles spécifiques de la conception des IHM ..... 20
1.2.4.1 Travaux de Palanque et Bastide : le modèle ICO ..... 20
1.2.4.2 Travaux de De Rosis : le formalisme XDM ..... 21
1.2.4.3 Travaux de Mahfoudhi et Tabary : la méthode TOOD ..... 21
1.2.4.4 Travaux de Ait Ameur : la méthode B ..... 22
1.2.4.5 Travaux de Rodriguez : le projet ALACIE ..... 22
1.2.4.6 Travaux de Bodart : le projet TRIDENT ..... 23
1.2.4.7 Travaux de Moussa : le système Ergo-Conceptor ..... 24
1.2.4.8 Synthèse ..... 24
**1.3 Conclusions** ..... **25**

**CHAPITRE 2 : CONTRIBUTIONS AUX METHODOLOGIES DE CONCEPTION DES IHM** ..... **28**
**2.1 Démarche globale en U dédiée aux systèmes industriels complexes** ..... **28**
**2.2 Contributions** ..... **30**

**CHAPITRE 3 : OUTILS FORMELS POUR LA SPECIFICATION DES IHM** ..... **38**
**3.1 Taxonomie de spécification des systèmes interactifs** ..... **38**
3.1.1 L'apport des sciences cognitives ..... 38
3.1.2 L'apport des approches algébriques ..... 38
3.1.2.1 L'approche basée sur les modèles ..... 38
3.1.2.2 L'approche algébrique ..... 40
3.1.3 L'apport des théories des langages ..... 40
3.1.3.1 Les grammaires hors contexte ..... 40
3.1.3.2 Les langages à événements ..... 41
3.1.3.3 Les langages synchrones ..... 41
3.1.3.4 Les techniques orientées objet ..... 43
3.1.4 L'apport de la théorie des graphes : ..... 43
3.1.4.1 Les automates à états finis ..... 43
3.1.4.2 Les Réseaux de Petri ..... 44
3.1.5 L'apport des logiques temporelles ..... 45
**3.2 Conclusions** ..... **46**

**CHAPITRE 4 : CONTRIBUTIONS AU DOMAINE DE LA SPECIFICATION FORMELLE DES SYSTEMES INTERACTIFS ..... 48**
**4.1 Principe de construction du modèle global de l'interaction homme-machine .... 48**
4.1.1 Composition séquentielle ..... 50
4.1.2 Composition parallèle ..... 50
4.1.3 Composition de fermeture ..... 52
4.1.4 Illustration ..... 52
**4.2 Spécification du dialogue homme-machine ..... 55**
**4.3 Vérification des propriétés de l'Interface ..... 58**
4.3.1 Les propriétés à vérifier ..... 58
4.3.2 Principe de vérification ..... 59
**4.4 Conclusions ..... 60**

**CHAPITRE 5 : CONTRIBUTIONS A LA GENERATION SEMI-AUTOMATIQUE DES IHM ..... 62**
**5.1 Spécification de la présentation assistée par un système à base de connaissances62**
**5.2 Génération de l'interface ..... 64**
5.2.1 La génération graphique ..... 65
5.2.2 La gestion de la dynamique des IHM ..... 65
5.2.3 Intégration et évaluation de l'IHM générée ..... 67
**5.3 Conclusions ..... 68**

**TRAVAUX EN COURS ET PERPECTIVES ..... 69**

**REFERENCES BIBLIOGRAPHIQUES ..... 70**

**LISTE DES FIGURES ..... 80**
**LISTE DES TABLEAUX ..... 81**

## INTRODUCTION GENERALE

Le développement de la microélectronique depuis le début des années 70 a induit une évolution sans cesse croissante de l'automatisation. Petit à petit, le rôle de l'Opérateur Humain (OH) a évolué depuis l'action physique de commande directe des machines ou des systèmes vers des activités plus « intellectuelles » de coordination puis de supervision. A cette fin, des Interfaces Homme-Machine (IHM) ont été élaborées.

Aussi, était-on tenté de penser, avec l'avènement des nouvelles générations de systèmes flexibles automatisés à outrance, que l'on allait pouvoir progressivement se passer totalement de l'OH, surtout si l'on parvenait à intégrer à la commande la détection des dysfonctionnements et les procédures de reprise (métro à pilotage automatique, pilote automatique d'avion, première génération d'ateliers d'assemblage au japon, etc.).

Mais on s'est vite révisé en prenant conscience que, dans les systèmes où les pertes matérielles et notamment humaines dues aux dysfonctionnements sont importantes, l'OH demeurait en fait indispensable. D'une part, l'intelligence humaine fait que celui-ci est à même d'envelopper une supervision intelligente et toujours enrichie par l'expérience. D'autre part, la présence de l'OH s'avère nécessaire pour réagir face à des situations de dysfonctionnement imprévues. Les Systèmes Homme-Machine (SHM) étaient nés.

Dès lors, la conception des IHM devait être axée sur deux rôles prépondérants :

- offrir à l'OH le moyen de superviser l'état de fonctionnement des principaux constituants du système,
- préparer les moyens qui permettent (voire assistent) l'intervention de l'OH en cas de dysfonctionnement.

L'utilisation de l'interface graphique (exclusivement ou partiellement) pour constituer le support de l'IHM s'est avérée d'un grand intérêt. En effet, les interfaces graphiques permettent des affichages diversifiés et évolutifs dans le temps. Elles permettent aussi et surtout une interaction homme-machine facile et performante qui aide l'OH dans ses réactions face aux événements imprévisibles.

La conception des IHM est devenue depuis une spécialité à part entière. Cependant, jusqu'à un passé relativement récent, elle est restée axée fondamentalement sur un rôle informatif : Quel affichage en fonction de l'état du fonctionnement du système ? ; Autrement dit, le QUOI en information, le QUAND en instant et le COMMENT en forme d'affichage, alors que l'assistance possible de l'OH était reléguée au second plan.

Néanmoins, l'opérateur ciblé par l'IHM est devenu au centre de la conception de celle-ci, dans la mesure où l'efficacité de l'IHM est très liée à la capacité de l'OH de capter et d'assimiler l'affichage et de réagir à travers cette interface pour piloter le système.

Cette conception centrée autour de l'utilisateur (voir à ce sujet le livre intitulé User Centered System Design (UCSD) [Norman, 86]) se doit d'intégrer les enseignements et les résultats de plusieurs disciplines :

- l'automatique et ses apports, entre autres, dans la modélisation du procédé, la surveillance, la coopération entre l'homme et la machine, etc.,
- la psychologie cognitive permettant, entre autres, la modélisation de l'opérateur humain, l'analyse et la modélisation de la « tâche » et de « l'activité » de l'opérateur humain ainsi que la détermination des besoins en information de celui-ci,
- les langages de spécification et d'analyse permettant, selon la rigueur de leurs formalismes, de modéliser les tâches opérateurs, les objets de l'interface et leurs comportements, etc. Les

possibilités de vérification des propriétés des modèles établis sont ici des atouts appréciables,

- l'ergonomie cognitive, venant principalement en complément à la psychologie cognitive, pour la résolution des problèmes autour de la présentation de l'information à l'écran, la prise en charge des problèmes sur le plan du confort opérateur, etc.,
- l'informatique dans la dimension la plus large du Génie logiciel et dans certaines spécialités sous-jacentes comme les Systèmes à Base de Connaissances (SBC), les Systèmes Multi-Agents (SMA), les systèmes de diagnostic et d'assistance, etc.

Toutes ces disciplines concourent à apporter des réponses à la problématique de conception des Systèmes Homme-Machine (SHM) d'une manière générale et des Interfaces Homme-Machine (IHM) en particulier.

Or, force est de constater qu'à ce jour, et en dépit de l'intérêt grandissant, aucune des approches publiées dans la littérature ne couvre le processus de conception/réalisation d'une manière complète et intégrée. Chacune se focalise sur une partie de la problématique. Un groupe de chercheurs s'est concentré sur l'étude des aspects ergonomiques. Un autre s'est focalisé sur les aspects d'analyse de la tâche humaine. D'autres chercheurs ont étudié l'intégration des outils formels dans le processus de spécification. En outre, peu de travaux se sont intéressés aux spécificités des systèmes industriels complexes.

Les travaux présentés ici ont été motivés, précisément, par l'objectif de contribuer à l'élaboration d'une approche globale de conception et de génération semi-automatique des Interfaces Homme-Machine (IHM). Une approche dont le souci majeur est d'offrir au concepteur une démarche complète et intégrée, supportée par des outils informatisés et prenant en considération, d'une manière explicite, les facteurs d'interaction homme-machine.

Une telle approche doit s'attacher à soulever et résoudre les points principaux suivants :

- Proposer une modélisation, orientée IHM, du SHM en question. Dans une approche du type MBD (Model Based Design), cette modélisation doit constituer la base de départ aux différentes étapes devant aboutir à la génération semi-automatique de l'interface. Les tâches opérateurs devraient s'en dégager clairement.
- Identifier les tâches opérateurs et les analyser afin de déduire les Besoins Informationnels des Opérateurs (BIO). Il y a lieu de distinguer les tâches de surveillance et celles de diagnostic. Au cours d'une tâche de diagnostic, l'OH est soumis à une forte charge de travail et à un niveau de stress important. Une assistance s'avère nécessaire pour l'aider dans l'accomplissement de sa tâche. Cette assistance passe par des BIO adéquats au contexte, présentés au bon moment et sous la forme appropriée.
- Déduire les objets qui composent l'interface en fonction des BIO obtenus.
- Utiliser des techniques formelles pour spécifier l'interface, ses objets graphiques et leurs comportements.
- Valider théoriquement les spécifications avant de procéder à la génération effective de l'interface.
- Utiliser des outils informatisés basés sur les recommandations ergonomiques pour spécifier graphiquement les objets déduits de l'interface. Des critères spécifiques définis au niveau de ces recommandations doivent garantir une ergonomie de présentation et de dialogue.
- D'une façon générale, assister le concepteur de l'interface tout au long des différentes étapes de l'approche avec des outils informatiques. Le passage délicat de la conception à la réalisation est crucial. En effet, les méthodes issues de la conception n'indiquent pas toujours comment implémenter les interfaces.

Ce mémoire d'habilitation se situe dans cette mouvance pluridisciplinaire. J'y synthétise mes contributions au domaine de l'interaction Homme-Machine. C'est le résultat d'un travail d'équipe mené depuis 1995 au sein des laboratoires LIP2, de la faculté des sciences de Tunis, et le LAMIH de l'université de Valenciennes et du Hainaut-Cambrésis,

Le premier chapitre s'intitule « De la conception des IHM ». Il aborde les questions de la conception et génération semi-automatique des IHM dans les systèmes industriels complexes. En fait, mes investigations sur cette thématique ont démarré bien avant l'année 1995, précisément à partir de 1988 quand j'ai commencé la préparation de ma thèse de doctorat de l'université de Valenciennes et du Hainaut-Cambrésis soutenu en 1992. Entre 1997 et 2004, j'ai co-encadré, sur cette thématique, les travaux de Mme Meriem Riahi entrant dans le cadre de la préparation de son mémoire de DEA intitulé « Spécification formelle des interfaces homme-machine : application au cas du contrôle des procédés industriels » (soutenu en novembre 1997), puis dans le cadre de la préparation de sa thèse de doctorat intitulée « Contribution à l'élaboration d'une méthodologie de spécification, de vérification et de génération semi-automatique d'interfaces homme-machine : Application à l'outil Ergo-Conceptor+ » (soutenue en Septembre 2004).

Le second chapitre s'intitule « Contributions aux méthodologies de conception des IHM ». J'y synthétise mes propositions en situant leurs apports nouveaux relativement aux travaux et résultats obtenus dans ce domaine de recherche.

Le troisième chapitre s'intitule « Taxonomie de spécification des systèmes interactifs ». L'accent est mis, ici, sur les apports des méthodes semi-formelles et formelles dans la spécification des systèmes interactifs d'une manière générale et des IHM en particulier. De 1999 à 2001, j'ai co-encadré, sur cette thématique, les travaux de Mme Nawel Khelil entrant dans le cadre de la préparation de son mémoire de DEA intitulé « Modélisation sûre des tâches interactives dans les interfaces homme-machine » (soutenu en octobre 2001).

Le quatrième chapitre s'intitule « Contributions au domaine de la spécification formelle des systèmes interactifs ». J'y développe une technique de modélisation, au moyen des RdP Synchronisés, appliquée à la spécification de l'interaction homme-machine, à la spécification du dialogue homme-machine et enfin à la vérification des propriétés de l'interface.

L'utilisation, dans la conception des IHM, d'outils mettant en oeuvre des guides de style (Tools For Working With Guidelines) fait l'objet du cinquième chapitre intitulé « Contributions à la spécification et à la génération semi-automatique des IHM ». J'y aborde, en particulier, l'intérêt de la mise en œuvre, à ces fins, de l'approche des Systèmes à Bases de Connaissance (SBC).

Le dernier chapitre, intitulé « Travaux en cours et perspectives », tire les conclusions générales des travaux effectués et suggère un certain nombre de prolongements utiles. J'y expose, notamment, les orientations qui suscitent mon intérêt dans mon projet de recherche actuel, comme la prise en compte du temps, la conception des IHM plastiques et l'expérimentation de l'approche Systèmes Multi-Agents (SMA) dans la spécification et la génération en temps réel des IHM. Sur cette dernière problématique, j'ai déjà encadré les travaux du mémoire de stage de DEA de Mme Sonia El Béji, intitulé « Apports des Systèmes Multi-Agents dans la spécification et la génération des interfaces graphiques », soutenu en septembre 2003. Des travaux de thèse portant sur la conception des IHM plastiques viennent de démarrer.

# CHAPITRE 1 : DE LA CONCEPTION DES INTERFACES HOMME-MACHINE

*Ce chapitre est dédié à un survol de l'état de l'art actuel en matière de conception des IHM.*

*La première section rappelle les deux concepts clés du domaine, à savoir le Système Homme-Machine (SHM) et l'Interface Homme-Machine (IHM).*

*La deuxième section développe la question du processus de conception de vérification et de génération semi-automatique des IHM. A cet effet, un recueil des disciplines impliquées dans le processus de conception des IHM est proposé, suivi d'une synthèse des approches et modèles considérés comme spécifiques de la conception des IHM dont la démarche globale proposée par P. Millot pour la conception des système homme-machine.*

*La troisième et dernière section présente nos conclusions et contributions à ce sujet.*

## 1.1 Système Homme-Machine et Interface Homme-Machine

### 1.1.1 Le système Homme-Machine (SHM)

Il nous semble, ici, opportun de rappeler ce qu'est qu'un SHM. Cet éclaircissement est indispensable puisqu'il nous permettra de mettre le doigt sur l'ensemble des moyens à mettre en œuvre dans la construction d'un tel système et par la même nous renseigner sur les disciplines sous jacentes à maîtriser dans un processus de conception d'IHM dans les systèmes industriels complexes (automatique, automatique humaine, ingénierie, intelligence artificielle, ergonomie, psychologie cognitive, informatique, etc.).

Un SHM consiste en une synthèse fonctionnelle entre un homme et un système technologique. Il est composé au minimum d'un opérateur humain, d'une machine et d'une interface permettant la communication entre les deux [Johansen, 82].

L'architecture évoluée d'un SHM peut être celle illustrée par la figure 1.1. Les trois couches "commande locale", "coordination" et "pilotage automatisé" constituent la commande temps réel. Souvent, on y adjoint un système de surveillance automatisé. L'IHM gère les interactions de l'opérateur aussi bien avec le système de pilotage (informations sur l'avancement de l'activité du système) qu'avec le système de surveillance automatisé (informations sur l'état de fonctionnement des composants du système). Elle peut être soutenue par un système d'assistance de l'opérateur (spécifications de références, procédures de diagnostic, etc.).

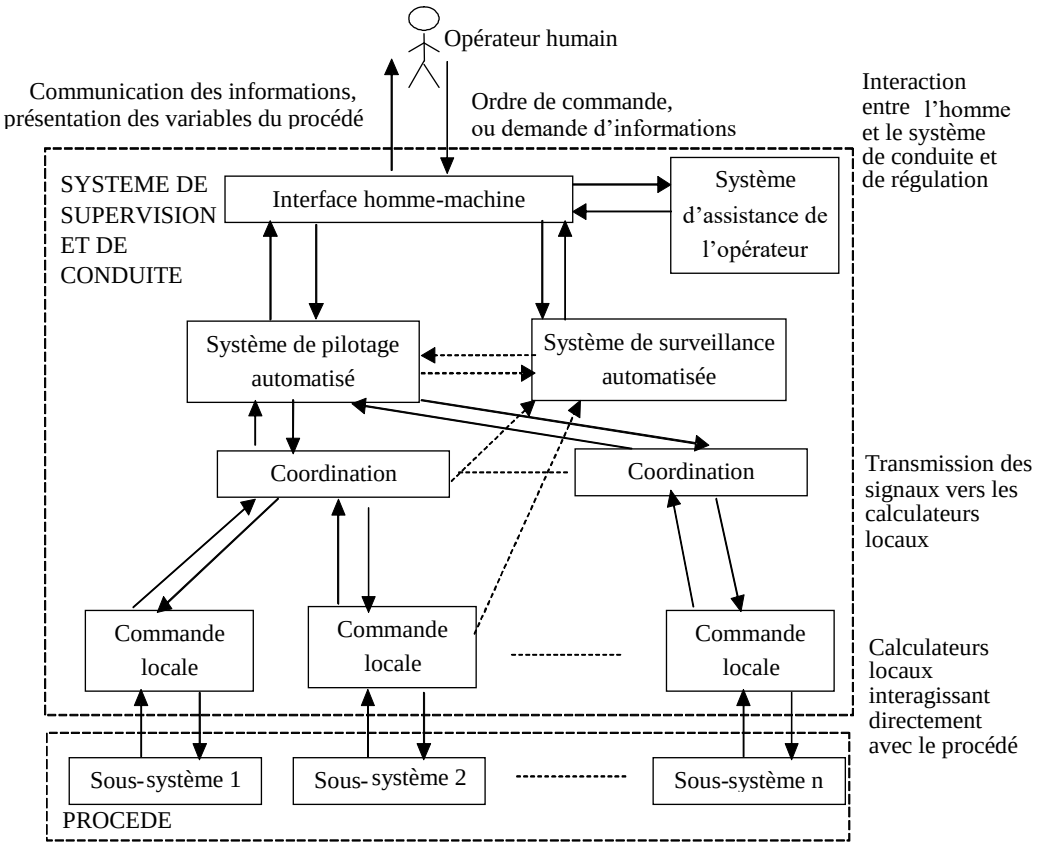



Figure 1.1: Architecture d'un Système Homme-Machine [Riahi, 04]
inspirée de [Sheridan, 85]

### 1.1.2 L'Interface Homme-Machine (IHM)

Il ressort de la définition du SHM, que l'interface constitue la charnière entre le système technologique (y compris son sous-système de supervision et de conduite) et l'OH qui, guidé par des objectifs précis et sous des contraintes particulières, doit pouvoir exécuter ses tâches en déroulant un plan d'action établi.

L'IHM représente le moyen privilégié, pour l'OH, pour superviser le système et le corriger en cas de besoin. L'IHM assure une communication à double sens :

1. informer l'OH du bon fonctionnement des composants du système et de l'avancement des activités entreprises. Dans ce sens, les informations internes du système traversent l'interface de sortie du procédé en subissant des transformations pour aboutir à une présentation perceptible par l'opérateur grâce à l'IHM,
2. offrir à l'OH le moyen de passer de ses intentions à l'exécution des actions pour modifier l'état du système. Dans ce sens, l'IHM assure les dispositifs physiques mis à la disposition de l'OH pour accomplir ses tâches.

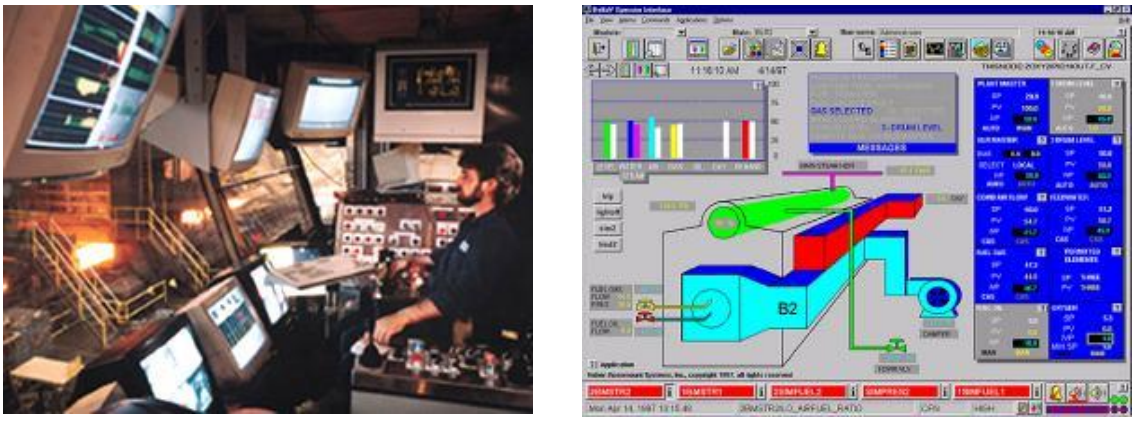



Figure 1.2 : (a) poste de supervision (b) exemple d'Interface Homme-Machine

Ainsi, l'OH ne possède pas de vue directe sur le procédé et sur les résultats de ses actions, et doit donc se forger une représentation mentale du fonctionnement du système et des différentes variables qui le composent. Il travaille alors à travers la représentation mentale qu'il se fait à chaque instant du système, représentation qu'il obtient grâce notamment à l'IHM [Norman, 86], [Rasmussen, 86] [Richard et al., 90], [kolski, 95].

Parmi les différents types d'IHM, nous considérons, dans notre travail de recherche, l'interface graphique (notée dans la suite de ce document IHM). Celle-ci est proposée sur un ou plusieurs écrans de visualisation de tailles variables. Elle renferme un ensemble de vues, parfois plusieurs dizaines voire centaines selon la complexité du procédé.

En raison du nombre élevé de variables à considérer et de la complexité des tâches à réaliser, il s'avère que l'utilisation des IHM peut entraîner des erreurs humaines. Ces erreurs proviennent souvent de lacunes lors de l'analyse des Besoins Informationnels des Opérateurs (BIO) pour les différentes tâches à accomplir, de connaissances insuffisantes des limites des OH, d'erreurs dans la présentation et la structuration de l'information sur les écrans, etc. De telles erreurs peuvent être évitées en prenant le problème des IHM à la base c'est-à-dire au niveau de la conception.

## 1.2 Conception, vérification et génération semi-automatique des IHM

Le processus de conception des IHM implique le recours à plusieurs disciplines et approches différentes mais complémentaires. On y propose ci-dessous celles qui nous semblent les plus pertinentes, à savoir la psychologie cognitive, l'ergonomie cognitive, les cycles de vie génie logiciel enrichis et, enfin, les approches et modèles considérés comme spécifiques pour la conception des IHM.

### 1.2.1 La psychologie cognitive dans la conception des IHM

Nous proposons ci-dessous deux concepts essentiels de la psychologie cognitive : la modélisation de l'opérateur humain et l'analyse de la tâche humaine.

#### 1.2.1.1 La modélisation de l'opérateur humain

Avant d'entamer la présentation de plusieurs théories, représentations et modélisation de l'opérateur humain, il est important d'ouvrir ici une parenthèse pour souligner la coexistence, dans le domaine de l'interaction homme-machine, de deux approches de modélisation fondamentalement distinctes :

- *L'approche fiabiliste :* elle a été initiée par Swain en 1963 [Swain, 63]. Son objectif est d'appliquer à l'opérateur les mêmes techniques d'évaluation de la fiabilité des matériels pour aboutir par la suite à des évaluations de la fiabilité globale du SHM. Comme exemple de cette approche, nous pouvons citer THERP (Technique for Human Error

Rate Prediction) [Swain, 64] qui quantifie la fiabilité humaine en passant par la probabilité d'occurrence des erreurs des opérateurs au même titre que les dysfonctionnements des systèmes [Swain et al., 81]. Plus tard, SHERPA, une variante de THERP, est venue ajouter à l'évaluation quantitative une évaluation qualitative dans l'espoir de rendre cette approche plus « humaine » [Embrey, 86]. Cette approche a beaucoup servi le département américain de la défense, mais n'a pas eu beaucoup de succès auprès de la communauté des chercheurs en interaction Homme-Machine.

- *L'approche cognitiviste* : elle vise à comprendre la genèse des erreurs de l'OH à partir des concepts tirés de la psychologie cognitive et s'appuie, pour cela, sur une description des mécanismes décisionnels humains mis en jeu pour accomplir les tâches. Elle facilite ainsi la déduction des besoins informationnels et des besoins d'assistance, utiles pour la spécification de l'interface. Contrairement, donc, à l'approche fiabiliste, l'approche cognitiviste met davantage l'accent sur la construction plutôt que sur l'évaluation. C'est pour cette raison que nous adopterons, dans nos travaux, cette approche.

Comme on vient de l'exposer dans le paragraphe précédent, en psychologie cognitive, la prise en compte de l'utilisateur se fait à travers des modèles permettant de rapprocher le fonctionnement de celui-ci. Les psychologues, communauté très active dans ce champ de recherche, tentent de proposer des modèles servant de cadre général de réflexion pour les chercheurs en interaction homme-machine. Sans souci d'exhaustivité, on propose, dans ce qui suit, le modèle du processeur humain, la théorie décisionnelle de Rasmussen et la théorie de l'action de Norman.

#### 1.2.1.1.1 Le modèle du processeur humain

Le modèle du processeur humain, proposé par Card [Card et al., 1983], peut être considéré comme le premier modèle, de la psychologie cognitive, étudié dans le cadre des systèmes interactifs. Il consiste en la combinaison de trois sous-systèmes interdépendants (i) le système sensoriel, (ii) le système moteur (iii) et le système cognitif ; qui possèdent, chacun, une mémoire et un processeur (figure 1.2).

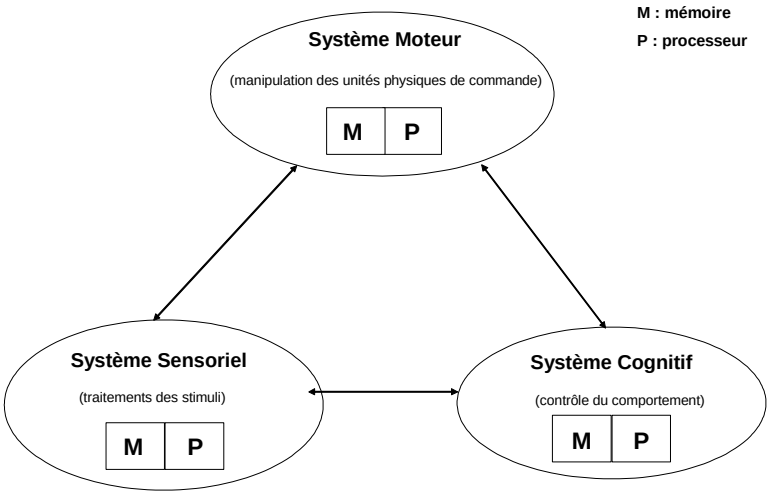



Figure 1.3 : Modèle du processeur humain

Cette structure à base de processeurs communicants, a permis à Card de procéder à des expérimentations dans le but de déterminer les caractéristiques de ces processeurs. Par exemple, un des trois processeurs de ce modèle est le processeur moteur qui est responsable des mouvements physiques de l'utilisateur. Il a été évalué empiriquement que le temps moyen pour effectuer un mouvement élémentaire est inclus dans l'intervalle [30,100] millisecondes. Ces résultats bien qu'empiriques sont facilement intégrables dans une spécification IHM.

#### 1.2.1.1.2 La théorie décisionnelle de Rasmussen

En 1980, Rasmussen fut le premier à proposer un cadre, appelé « échelle de décision » (décision ladder) [Rasmussen, 80], illustrant la démarche générale de résolution de problème susceptible d'être suivie par des utilisateurs d'interfaces graphiques en salle de supervision de système industriel. L'échelle de décision de Rasmussen comprend quatre étages séquentiels de traitement d'informations, figure 1.3 :

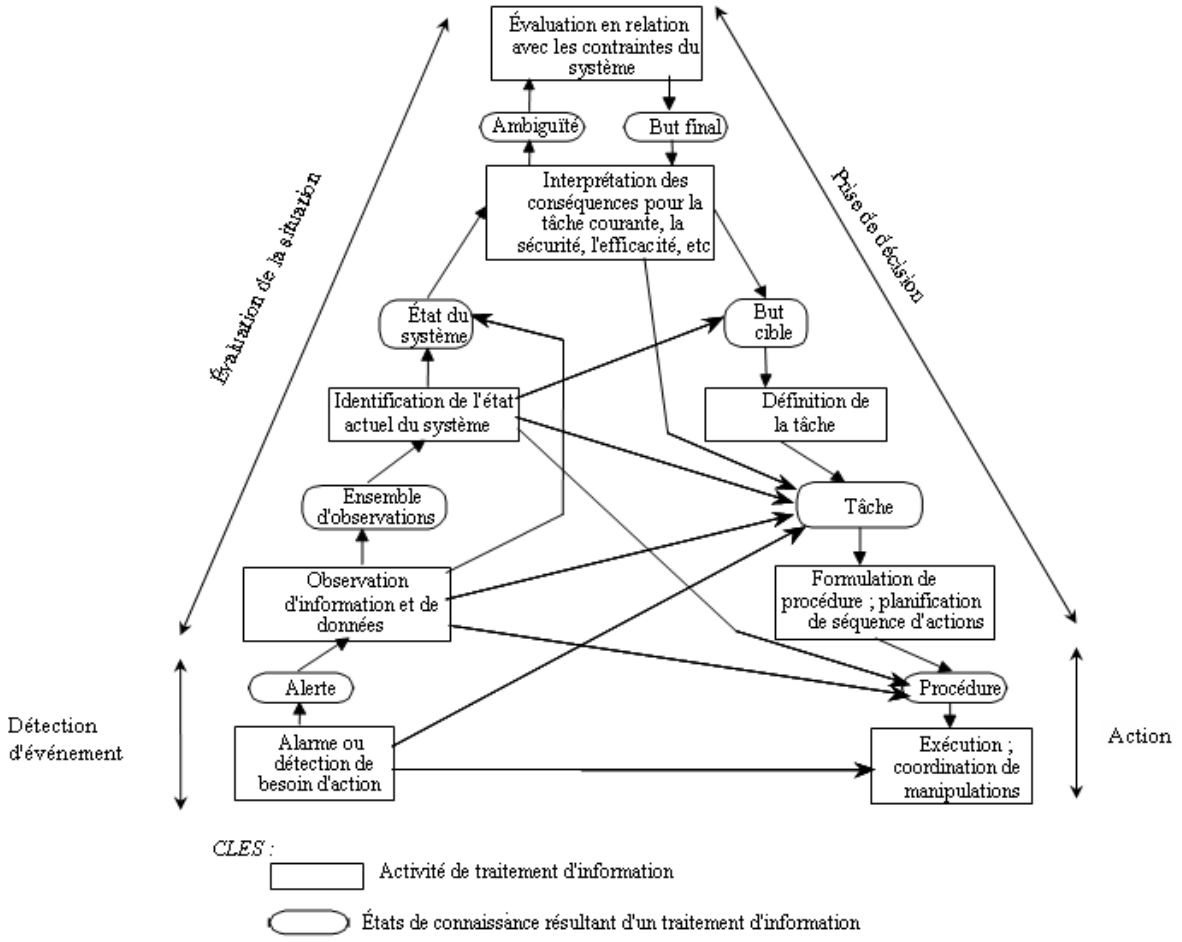



Figure 1.4 : L'échelle de décision de Rasmussen

- détection d'un événement anormal mettant l'opérateur en état d'alerte (détection par une alarme ou par une observation de l'évolution anormale d'une ou de plusieurs variables),
- évaluation de la situation en observant l'ensemble des informations utiles de façon à identifier l'état du procédé,
- définition d'une stratégie générale de correction, compte tenu de l'état identifié et des objectifs de conduite qui lui sont assignés, puis décomposition en tâches et procédures d'actions,
- exécution des actions.

Bien entendu, le raisonnement de l'opérateur ne se déroule pas toujours séquentiellement en suivant toutes les étapes visibles dans ce modèle (raccourcis proposés sur la figure 1.3) notamment lorsqu'il est face à une situation connue. Ces différences de comportement sont très importantes, pour les concepteurs d'interfaces, puisqu'elles entraînent des variations sur les

besoins informationnels de l'opérateur dont il faut tenir compte lors de la spécification de l'interface. Dans ce sens Rasmussen distingue trois types de comportement de l'OH :

- Comportement basé sur l'habilité (skill based behaviour) où l'opérateur exécute d'une façon quasi réflexe des actions correctrices en réponse à des informations perçues sous forme de signaux. Dans ce cas l'opérateur passe directement de l'étape de détection d'un état anormal à l'étape d'exécution d'une procédure.
- Comportement basé sur les règles (rule based behaviour). Ce comportement est adopté par l'opérateur quand celui-ci se trouve confronté à une situation connue. Dans ce cas l'opérateur applique une procédure prédéfinie et mémorisée dans sa mémoire à long terme et/ou disponible sous une forme logicielle ou papier.
- Et comportement basé sur la connaissance (knowledge based behaviour) pouvant être mis en évidence lors de situations inhabituelles ou inconnues obligeant l'opérateur à mener des raisonnements réellement intelligents pour résoudre le problème.

L'échelle de décision de Rasmussen offre un bon cadre pour comprendre et analyser le comportement cognitif de l'opérateur et proposer en conséquence une IHM adaptée à chaque type de comportement suivi. Notons aussi que ce modèle a fait l'objet d'extensions par Hoc et Amalberti sous l'angle de la dynamique des situations [Hoc et al., 95].

#### 1.2.1.1.4 La théorie de l'action de Norman

Cette théorie introduit les différentes étapes cognitives nécessaires à la réalisation d'une tâche exécutée par l'opérateur humain à l'aide d'un système informatique [Norman, 86].

La représentation mentale de la connaissance en terme de variables psychologiques représente le modèle conceptuel. Ainsi, à chaque concept, unité de connaissance ou sujet d'intérêt, correspondra une variable psychologique (valeur d'une variable du procédé, état d'un sous-système, état but à atteindre, etc.).

Cette représentation mentale dépend de la connaissance déjà acquise et de la compréhension de la situation présente. Ce modèle mental est donc incomplet, imprécis et il évolue avec l'expérience et l'apprentissage.

Selon Norman, la réalisation d'une tâche met en jeu au moins plusieurs activités (figure 1.4) :

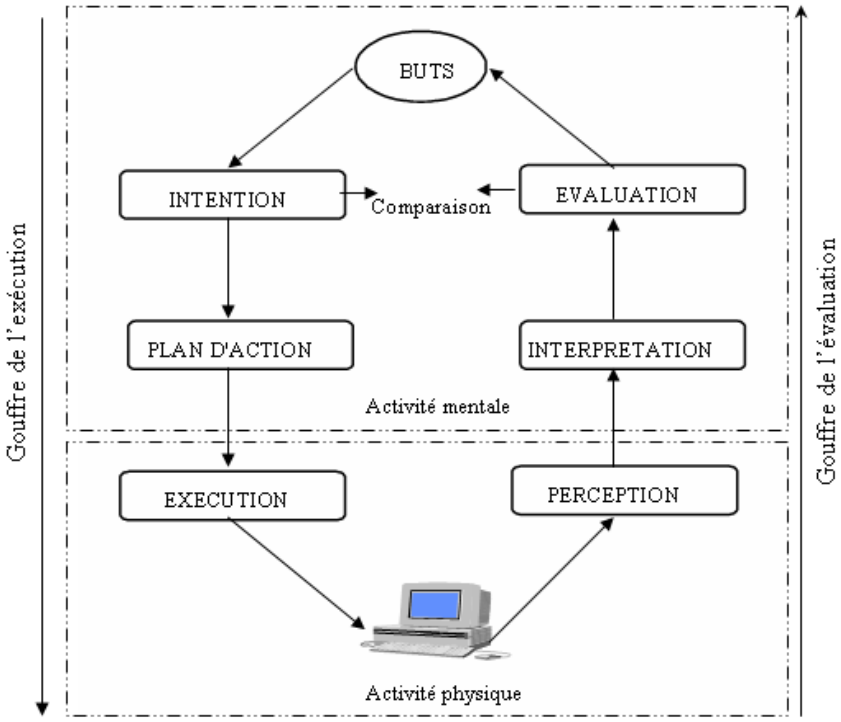



Figure 1.5 : La théorie de l'action de Norman

**1-** L'établissement d'un but. Un but est une représentation mentale de l'état du système que l'utilisateur souhaite atteindre et ce, en agissant sur des dispositifs physiques mis à sa disposition au niveau de l'interface.

**2-** La formulation d'une intention est la décision d'agir de façon à atteindre le but pré-établi.

**3-** L'adoption par l'utilisateur d'un plan d'action : c'est la représentation psychologique de l'ensemble des actions et de leur ordonnancement que l'opérateur doit exécuter au moyen de dispositifs physiques de contrôle et ce, dans le but d'atteindre son objectif.

**4-** L'exécution du plan d'action : il s'agit dans cette étape de mettre en application le plan d'action en agissant, à partir des dispositifs de contrôle, sur le procédé piloté. L'état du système va donc changer.

**5-** La perception du nouvel état du système : dans cette étape, l'opérateur constate les changements survenus sur le procédé.

**6-** L'interprétation de la modifications des variables physiques en terme psychologique : elle aboutit à une représentation mentale du nouvel état du système.

**7-** La comparaison du nouvel état du système avec le but pré-établi et les intentions formulées : elle peut conduire à la poursuite du plan d'action ou à sa modification.

Enfin, cette théorie de l'action de Norman souligne que l'activité de l'opérateur nécessite le franchissement, de deux chemins, qu'il importe de réduire au maximum, traduit par [Nigay, 94] comme étant :

- Le gouffre de l'exécution qui désigne la distance mentale que l'opérateur doit parcourir pour passer de ses intentions à l'exécution des actions, en tenant compte des propriétés des dispositifs physiques mis à sa disposition, à travers l'IHM, pour modifier l'état du système.

- Le gouffre de l'évaluation qui définit le parcours inverse ; les informations internes du système traversant l'interface de sortie du procédé en subissant des transformations pour aboutir à une présentation perceptible par l'opérateur grâce à l'IHM.

Le dénominateur commun de ces modèles issus de la psychologie cognitive est l'analyse des tâches opérateurs. En effet, ces modèles mettent l'accent sur une description à haut niveau d'abstraction de la notion de « tâche ». Cependant, leur inconvénient majeur est le manque de support formel. Ils ne peuvent pas être appliqués d'une manière directe et systématique pour la conception des IHM. De nombreux chercheurs travaillent depuis des années sur la formalisation de ce concept clé de la psychologie cognitive : "la tâche opérateur". Une synthèse de certains travaux représentatifs est maintenant présentée.

#### 1.2.1.2 L'analyse de la tâche humaine dans la conception des IHM

L'intégration d'un modèle de la tâche dans le processus de conception des IHM semble aujourd'hui faire l'unanimité au sein de la communauté des chercheurs en interaction homme-machine. En effet, l'analyse et la modélisation de la tâche apportent beaucoup de connaissances en terme de BIO ; besoins à transformer et intégrer au niveau des IHM.

Une majorité de chercheurs convergent aujourd'hui vers cette définition de la tâche : « c'est une activité dont l'accomplissement par l'utilisateur produit un changement d'état significatif d'un domaine d'activité donné, dans un contexte donné » [Tarby, 93], [Bodart et al., 95], [Palanque, 97].

Une précision importante au niveau du concept de tâche opérateur est à souligner. Elle renvoie à la distinction à faire entre :

- la tâche prescrite, prévue ou prédite par les concepteurs, résultant d'un processus d'identification, d'analyse et de modélisation,
- et la tâche réelle, effective ou accomplie, renvoyant aux séquences d'actions cognitives ou physiques produite par l'utilisateur et les ressources impliquées dans ces actions. La reconstitution de la tâche réelle est souvent obtenue par des processus d'observation et de recueil de données dans un SHM réel ou simulé.

La comparaison des modèles de la tâche prescrite et de la tâche réelle permet de dégager des erreurs de conception, des incohérences, des oublis, etc., et de procéder à des ajustements et des affinements au niveau des IHM et des outils d'assistance.

L'analyse de la tâche permet de déboucher sur un ensemble de données. Ces données peuvent être, globalement, exploitées de deux manières :

- en impliquant d'une manière empirique ces données « brutes » dans une approche de conception non supportée (ou partiellement supportée) par des outils informatiques,
- en modélisant ces données en vue d'intégrer le(s) modèle(s) obtenu(s) au sein d'approches informatisées [Robert et al., 97] [Sheridan, 97] [Terwilliger et al., 97][Shepherd, 98].

Nous résumons ici trois types de modélisation des tâches, issus des travaux de Abed, de Scapin et de ceux de Paterno, qui nous semble assez proches de notre problématique. De nombreux autres exemples peuvent être trouvés dans l'ouvrage de Diaper et Stanton [Diaper et al., 01].

##### 1.2.1.2.1 Les travaux de Abed

Abed propose une démarche de description de la tâche basée sur une combinaison de la méthode SADT (Structured Analysis and Design Technique) et des réseaux de Petri. Cette démarche est constituée de quatre étapes [Abed, 01] [Abed et al., 01] :

- Utiliser la méthode SADT pour décomposer hiérarchiquement le comportement du SHM en tâches.
- Identifier les tâches élémentaires au dernier niveau de la décomposition et les répartir entre la machine et l'opérateur humain.
- Analyser les tâches humaines et distinguer entre les tâches interactives, qui définissent des routines d'action, et les tâches non interactives de simple surveillance.
- Décrire le fonctionnement et la dynamique des tâches élémentaires interactives en utilisant les réseaux de Petri Synchronisés (RdPS).

Ce principe de décomposition SADT/Petri (figure 1.5) permet de décrire d'une manière rigoureuse, aussi bien l'aspect statique que l'aspect dynamique du SHM, et conduit à identifier les besoins informationnels des opérateurs humains qui sont en fait les entrées et les contraintes imposées sous forme de données de contrôle des boîtes actigrammes du modèle SADT.

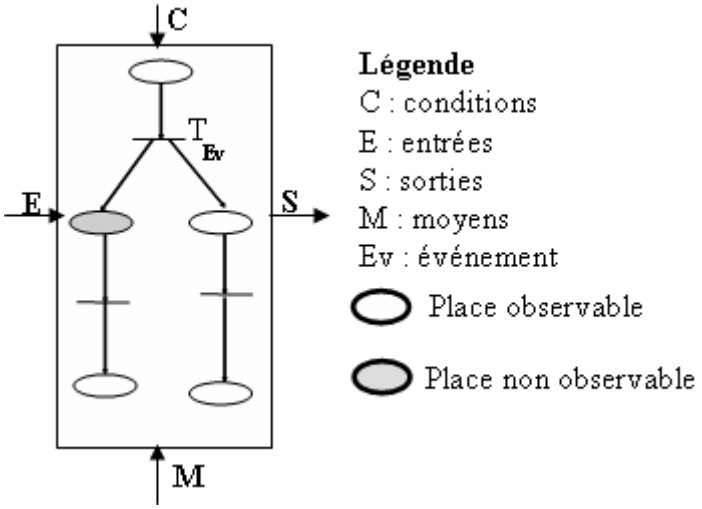



Figure 1.6 : Une boîte SADT relative à une tâche humaine interactive dont le fonctionnement est décrit par un RdPS

Mahfoudhi [Mahfoudhi, 97] puis [Tabary, 01] proposeront par la suite une extension des travaux de Abed, en y apportant, principalement, l'informatisation au moyen de la technologie objet de ce modèle. On y reviendra dans le paragraphe 2.4.3.

**1.2.1.2.2 Les travaux de Paterno**

Paterno propose une démarche et un outil de modélisation des tâches nommé CTT (Concur Task Trees) [Paterno, 01] [Mori et al., 02]. L'objectif principal de CTT, est de fournir une notation facile de modélisation des tâches, supportant la conception des applications interactives. Cette notation distingue quatre types de tâches :

- les tâches utilisateur : relatives à des activités cognitives de réflexion, sélection, choix, etc. ;
- les tâches système (ou tâches application) : relatives à des traitements informatiques et la génération de résultats ;
- les tâches interactives : relatives à des actions de l'opérateur et des "feedbacks" du système ;
- les tâches abstraites (ou tâches non élémentaires) : relatives à une composition de tâches de différents types.

CTT propose une structuration hiérarchique des tâches utilisateurs (figure 1.6). Il se base sur la notation UAN (User Action Notation) développée au début des années 80, par Antonio Siochi, H. Rex Hartson et Deborah Hix [Hix et Hartson, 93].

UAN offre sur une notation textuelle modélisant les différentes tâches de l'interaction. Les tâches élémentaires sont décrites par des tables précisant : l'ensemble des actions opérateur, les réponses (feedback) du système et les éventuelles modifications d'états. Sa notation textuelle et l'absence d'un outil de support, ont fait que l'utilisation d'UAN est restée limitée dans la pratique. Cependant, elle a servi de base à la notation graphique CTT pour la modélisation des tâches concurrentes.

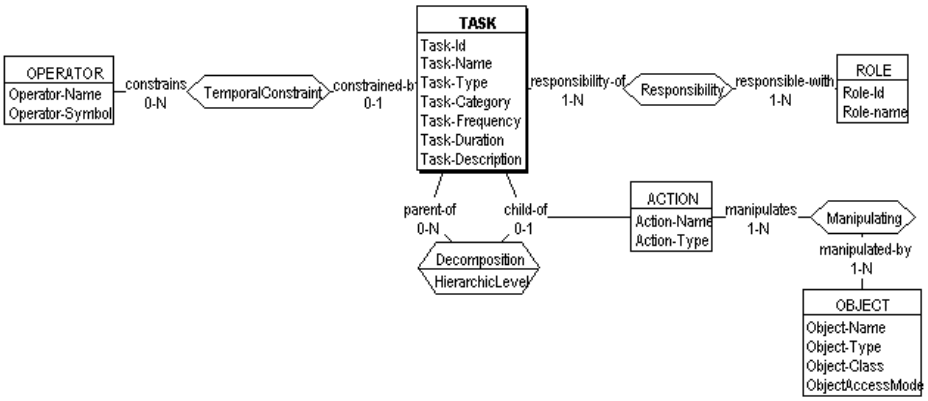



Figure 1.7 : MCD d'une modélisation CTT

#### 1.2.1.2.3 Les travaux de Scapin

MAD (Méthode Analytique de Description de Tâches) proposée par Scapin en 1989 est un formalisme de description de tâches. Des objets tâches sont conçus comportant chacun [Scapin, 89] [Scapin et al., 01] (figure 1.7) :

- *un état initial (I) :* sous-ensemble de l'état du monde de l'application, constitué de la liste des objets et arguments d'entrée de la tâche ;
- *un état final (F) :* sous-ensemble de l'état du monde de l'application, constitué de la liste des objets et arguments de sortie de la tâche ;
- *un but (B) :* sous-ensemble de l'état final indiquant explicitement le but recherché dans l'exécution de la tâche ;
- *des pré-conditions (C.N) :* ensemble de prédicats exprimant les contraintes qui doivent être satisfaites par l'état initial pour le déclenchement de l'exécution de la tâche ;
- *des post-conditions (P.C) :* ensemble de prédicats exprimant les contraintes qui doivent être satisfaites par l'état final, après l'exécution de la tâche.

MAD distingue deux types d'objets tâches : les tâches élémentaires indécomposables correspondant aux actions et les tâches composées. Pour les tâches composées, plusieurs opérateurs ont été définis au niveau de ce modèle tels que : SEQ pour décrire le séquencement, PAR pour exprimer le parallélisme, ALT pour spécifier l'exécution alternative, BOUCLE pour préciser l'itération etc.

Ce modèle fournit une représentation simple des tâches, compréhensible aussi bien du point de vue de l'ergonome que de celui de l'informaticien. Cependant, il décrit le travail indépendamment de la répartition des tâches entre l'homme et la machine, ce qui peut rendre son utilisation peut adaptée aux applications hautement interactives.

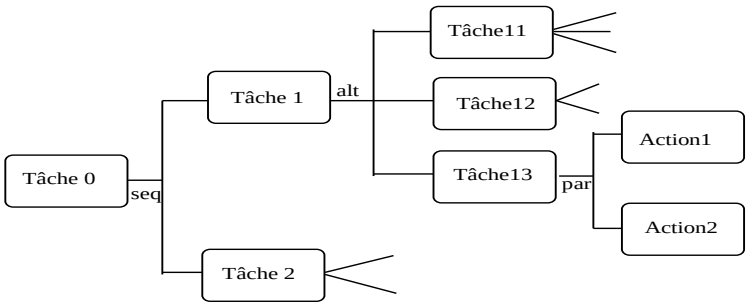



Figure 1.8 : Une arborescence de tâches MAD

Afin d'illustrer les apports de la psychologie cognitive, nous avons proposé deux concepts clés à savoir le concept de modèle de l'opérateur humain et celui d'analyse de la tâche humaine. Il importe, à présent, de parler de l'apport primordial de l'ergonomie cognitive dans la conception des IHM. [Amalberti et al, 98] [Jacko et al., 03].

### 1.2.2 L'ergonomie cognitive dans la conception des IHM

Avant d'aborder la question d'ergonomie cognitive (Human Factors), revenons d'abord sur l'origine du mot ergonomie. L'ergonomie est un mot qui vient du grec (*ergo* : travail ; *nomie* : règles, lois). C'est donc la science des règles de travail. Elle est définie comme étant le rassemblement des connaissances sur le fonctionnement de l'homme en activité afin de les appliquer à la conception des tâches, des machines, des outillages, des bâtiments et des systèmes de production. Pour ce qui est de la conception des SHM, il s'agit de concevoir les postes de travail, au sens large ; en fonction des caractéristiques physiologiques et psychologiques de l'OH et de son comportement en situation de travail, [Bastien, 96] [Scapin and Bastien, 97][Bastien et Scapin, 01]

L'ergonomie cognitive pour la conception et l'évaluation des IHM, peut être considérée comme une approche légère de conception venant en complément à celle de la psychologie cognitive. C'est une approche principalement expérimentale qui tente de s'opérationnaliser en se basant sur la transformation de l'expérience acquise dans divers domaines en règles ergonomiques applicables [Vanderdonckt, 99], [Smith, 86], [Nielsen, 93], [Shneiderman, 98], [Grammenos et al., 99]. Ces règles peuvent être groupées en huit classes de critères comme suit [Scapin, 89][Bastien, 96] [Reed et al., 99] :

**1-** *Le guidage* : c'est l'ensemble des moyens permettant de conseiller, orienter, informer et conduire l'utilisateur lors de ses interactions avec l'ordinateur (messages, alarmes, labels, etc.).

**2-** *La charge de travail* : c'est l'ensemble des éléments de l'interface qui ont pour rôle de réduire la charge perceptive ou mnésique des utilisateurs et d'augmenter l'efficacité du dialogue.

**3-** *Le contrôle explicite* : c'est la prise en compte par le système des actions explicites des utilisateurs tout en leur laissant le contrôle sur le traitement de leurs actions. Il intègre donc l'anticipation des actions utilisateur et l'offre des options appropriées.

**4-** *L'adaptabilité* : c'est la flexibilité du système et sa capacité à réagir selon le contexte, les besoins, les préférences et le niveau d'expérience des utilisateurs.

**5-** *La gestion des erreurs* : c'est l'ensemble des moyens permettant d'une part d'éviter ou de réduire les erreurs, et d'autre part de les corriger lorsqu'elles surviennent en intégrant des messages d'erreur brefs avec des termes aussi spécifiques que possible.

**6-** *L'homogénéité et la cohérence* : c'est la façon avec laquelle les choix de conception de l'interface sont conservés pour des contextes identiques, et sont différents pour des

contextes différents (présenter toujours la même information au même endroit, utiliser toujours le même format de champs d'entrée de données,…).

7- *La signifiance des codes et les dénominations* : c'est l'adéquation entre l'objet ou l'information affichée ou l'entrée et son référent pour faciliter le rappel et la reconnaissance.

8- *La compatibilité* : c'est l'accord pouvant exister entre les caractéristiques des utilisateurs (mémoire, perceptions, habitudes, compétences, attentes, âge, etc.) et des tâches, d'une part, et l'organisation des sorties, des entrées et du dialogue d'une application donnée, d'autre part. Ça concerne également le degré de similitude entre divers environnements ou applications.

Dans la pratique, l'exploitation des connaissances ergonomiques se fait selon trois approches principales : le recours à un expert humain, l'utilisation de guides ergonomiques et enfin le recours à des systèmes informatisés.

Ces systèmes informatisés mettent en œuvre des recommandations ergonomiques formalisées provenant d'experts humains et/ou de guides de style. L'objectif étant d'automatiser leurs applications dans le processus de conception des IHM. Cette approche consiste à faire appel à des techniques de l'intelligence artificielle pour mettre en place des systèmes à base de connaissances (SBC), prenant la forme de « Tools For Working With Guidelines » ( TFWWG) [Vanderdonckt, 00 ][Iwc, 99][Iwc,00]. Nous pouvons citer à ce sujet, comme exemple, le système SYNOP [Kolski et al., 91][Kolski et al., 96], ou encore le système à base de connaissances ERGO-CONCEPTOR [Moussa, 92] [Moussa, 00]. Nous reviendrons, sur ce point, plus en détails au niveau du chapitre 5.

Ayant présenté, jusqu'ici, les concepts primordiaux de l'interaction homme-machine à savoir : la modélisation de l'OH, la modélisation de la tâche humaine et la prise en considération des connaissances ergonomiques, il s'avère nécessaire d'introduire à ce niveau, des exemples représentatifs d'approches et de modèles spécifiques pour la conception des IHM. Mais avant cela, nous commençons par donner un bref aperçu de certains modèles du génie logiciel enrichis sous l'angle de l'IHM.

### 1.2.3 Les cycles de vie Génie Logiciel « enrichis » dans la conception des IHM

L'approche traditionnelle de génie logiciel distingue généralement six phases dans la vie d'un logiciel : la phase d'étude préalable, la phase de spécification, la phase de conception, la phase d'implémentation, la phase de test et la phase de maintenance. On retrouve ces phases dans les divers cycles de vie classiques proposés en génie logiciel tels que le modèle en cascade [Boehm, 88], le modèle en V [McDermid et al., 84], le modèle en spirale [Boehm, 88] ou encore le modèle par incréments.

L'ensemble de ces modèles classiques de génie logiciel, offre un cadre général très utile pour les concepteurs. Cependant, les IHM n'y sont pas citées même si la démarche les sous-entend. L'analyse et la modélisation des utilisateurs et des tâches humaines ne sont pas préconisées et sont laissées à l'appréciation des concepteurs. Afin de remédier à ces lacunes, d'autres modèles enrichis pour les IHM ont été mis au point pour le développement des applications hautement interactives.

L'idée d'enrichissement des cycles de vie pour les IHM, repose sur l'intégration, d'un point de vue méthodologique, des aspects fondamentaux de l'interaction homme-machine comme la modélisation des tâches humaines, la réalisation itérative des prototypes ou l'évaluation des systèmes interactifs. Nous avons choisi de présenter ici trois de ces modèles : le modèle de Long, le modèle en étoile, et le modèle ∇ (Nabla). D'autres modèles enrichis sont décrits dans [Kolski et al., 01].

### 1.2.3.1 Modèle de Long

Ce modèle [Long et al., 90] est proche du modèle en cascade du génie logiciel classique (figure 1.8). L'IHM est positionnée dans l'étape de conception tout en insistant sur l'importance de l'évaluation et des itérations lors du projet.

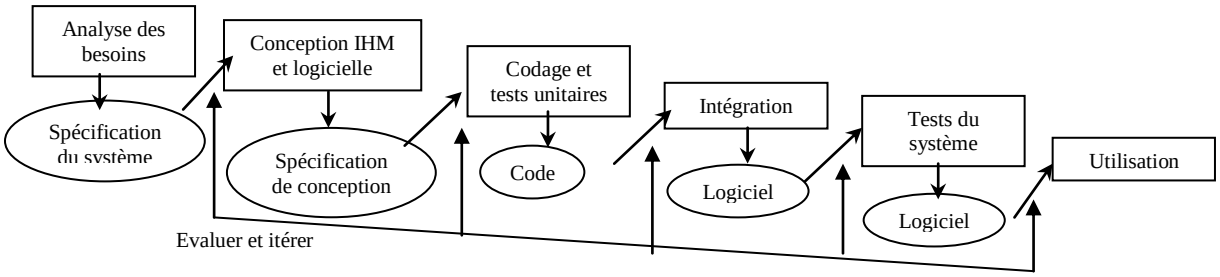



Figure 1.9 : Modèle proposé par Long et al.

Ce modèle, proche d'un modèle classique, est loin d'être parfait. Il a toutefois le mérite d'être incitateur sous l'angle des interactions homme-machine.

### 1.2.3.2 Modèle en étoile

Le modèle proposé par Hartson, [Hartson et al., 89], appelé modèle en étoile (figure 1.9) situe l'évaluation au centre même du cycle complet. Il préconise des interactions/itérations entre l'évaluation et chacune des autres étapes. L'étape d'évaluation est vue comme une étape intermédiaire permettant de protéger l'équipe de développement d'un rejet terminal.

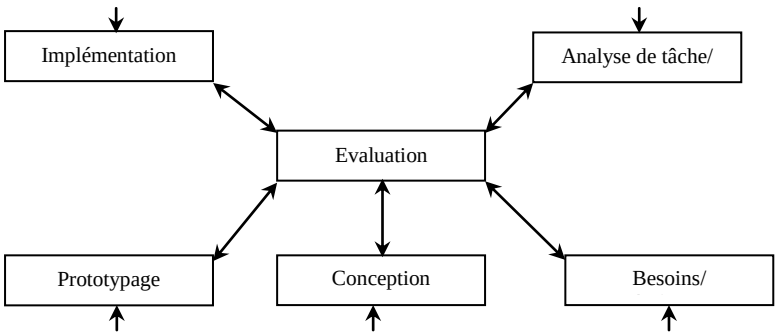



Figure 1.10 : Modèle en étoile

Le modèle en étoile n'impose pas d'ordre a priori dans l'accomplissement des étapes du processus, bien qu'en pratique, les activités de développement soient reportées en fin de cycle. Il sous-entend une conception participative visant la détection précoce de problèmes d'utilisabilité, requérant une forte adhésion de l'utilisateur par cette implication centrale.

### 1.2.3.3 Modèle ∇ (Nabla)

Le modèle ∇ [Kolski, 97] (figure 1.10) a pour objectif de situer les différentes étapes du génie logiciel nécessaires pour développer un système interactif, tout en différenciant l'interface proprement dite (partie gauche du modèle) des modules applicatifs ou d'aide éventuellement

accessibles à partir de ceux-ci (partie droite). Une des caractéristiques marquantes du modèle est de positionner des étapes, inexistantes dans les modèles classiques du génie logiciel, où les facteurs humains doivent être considérés par l'équipe de développement.

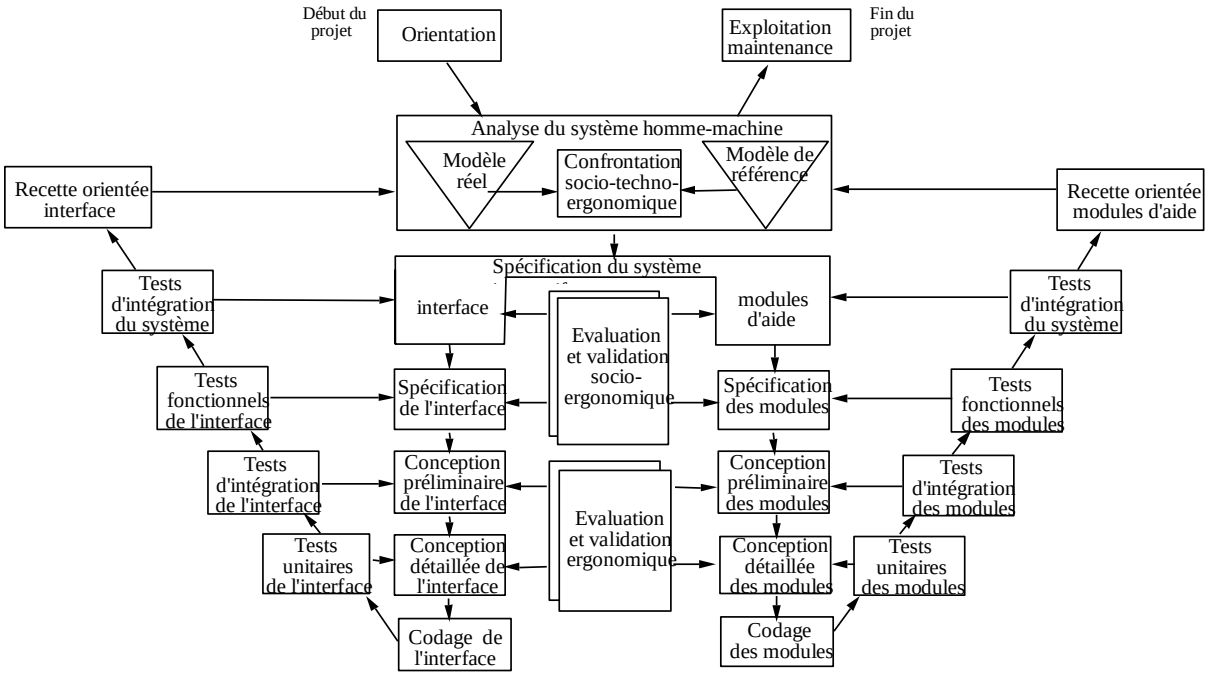



Figure 1.11 : Modèle ∇ (Nabla)

Comme nous l'avons vu jusqu'ici, la tâche de conception des IHM constitue un processus assez complexe, plus ou moins facilitée par les cycles de vie proposés, qu'ils soient classiques ou enrichis. Nous relevons dans la littérature plusieurs autres démarches, considérées comme spécifiques pour la conception des IHM, pouvant être suivies comme cycle de vie. Nous en proposons ci-dessous un recueil non exhaustif.

### 1.2.4 Approches et modèles spécifiques de la conception des IHM

L'objectif n'étant pas de lister d'une manière exhaustive et complète toutes les approches et méthodes disponibles dans la littérature, nous proposons, ci-dessous, un recueil de certaines approches et modèles spécifiques au domaine de la conception des IHM. Ceux-ci nous paraissent les plus proches de notre problématique et offrent, donc, un cadre de réflexion et de comparaison fort intéressant. Un tableau récapitulatif sera proposé à la fin de cette section.

#### 1.2.4.1 Travaux de Palanque et Bastide : le modèle ICO

Palanque et Bastide [Palanque, 97] [Palanque et al., 97] proposent de modéliser les interfaces des applications interactives en utilisant le formalisme ICO (Interactive Cooperative Objects).

Ce formalisme est basé sur une combinaison de techniques à objets et des réseaux de Petri à objet. Il permet une spécification semi-formelle de l'interface et de son comportement. Cette spécification facilite l'analyse fonctionnelle de l'interface pour procéder à sa validation avant son implémentation.

L'approche proposée hérite des avantages des approches objets tels que la réutilisabilité et la modularité. Cependant, la validation de modèles basés sur les réseaux de Petri à objets n'est pas évidente. En effet, afin de pouvoir valider un tel modèle, il est absolument nécessaire de préciser les règles et les conditions permettant la transition d'un réseau de Petri à objets vers un réseau de Petri ordinaire sur lequel la vérification des propriétés pourra être menée. Ce qui n'apparaît pas clairement dans cette approche [Palanque, 97]. De plus, la méthode n'explique pas le principe

d'identification des objets de l'interface. Elle laisse cette tâche au bon sens du concepteur de l'interface et ne lui fournit, à notre connaissance, aucun guide pratique.

#### 1.2.4.2 Travaux de De Rosis : le formalisme XDM

Le formalisme nommé XDM (pour conteXt sensitive Dialogue Modelling) a été proposé par De Rosis et Pizzutilo [De Rosis et al., 98] pour la description formelle et l'évaluation des IHM. Le formalisme est basé sur une extension des Réseaux de Petri. Le formalisme proposé permet : (1) la description des différents aspects statiques et dynamiques de l'interaction dans les différents contextes de fonctionnement du système, (2) et une évaluation pré-empirique de l'interface.

Le formalisme XDM proposé permet la modélisation du dialogue homme-machine en précisant les propriétés statiques et dynamiques de l'interface et la simulation de son comportement dans les différents contextes de fonctionnement possibles. Il permet, la vérification de quelques propriétés de l'interface telle que la complétude, l'absence d'ambiguïté, la consistance et la complexité de l'interface. Afin d'évaluer ce dernier critère, les auteurs proposent d'intégrer au formalisme la théorie des opérateurs KLM (Keystrokes-Level-Model) proposée par Card [Card et al., 83]. Le modèle KLM est un modèle donnant la prédiction de la durée d'une tâche à exécuter selon un scénario donné. Cependant, la manière d'identifier les besoins informationnels des opérateurs et la répartition de ceux-ci au niveau des places du réseau n'apparaît pas clairement dans l'approche. Le choix des objets graphiques pour la constitution de l'interface est laissé au concepteur sans fournir aucun guide particulier. En outre, la modélisation du comportement intrinsèque des objets graphiques de l'interface n'est pas prise en considération. On se demande, donc, si la validation, à un haut niveau d'abstraction, de l'interface sans prise en compte du comportement des objets graphiques peut être considérée comme irréfutable.

#### 1.2.4.3 Travaux de Mahfoudhi et Tabary : la méthode TOOD

La méthode TOOD (pour Task Object Oriented Description) propose une approche de conception des IHM basée sur une décomposition hiérarchique du SHM en tâches modélisées par réseaux de Petri à Objets (RdPO) [Mahfoudhi, 97] [Tabary et al., 98]] [Tabary, 01] [Abed, 01].

TOOD est fondée sur la représentation que l'utilisateur a de la tâche en dehors des considérations de traitement informatique [Tabary, 00]. Elle utilise l'approche objet et les réseaux de Petri objets pour décrire, d'une part les aspects fonctionnels et la dynamique des tâches utilisateur, et d'autre part les aspects comportementaux de l'IHM et de l'utilisateur pour spécifier comment s'effectuent les tâches. Son formalisme vise à couvrir la totalité du cycle de développement de l'analyse de l'existant jusqu'à la conception détaillée et l'implémentation (figure 1.11).

TOOD profite des avantages de l'approche orientée objet pour définir des objets tâches exploitant la notion d'événement. Ils utilise pour cela un format extension de SADT pour représenter l'objet tâche intégrant une structure de contrôle de la tâche appelée TCS (Task Control Structure) modélisée par les Réseaux de Petri à Objets.

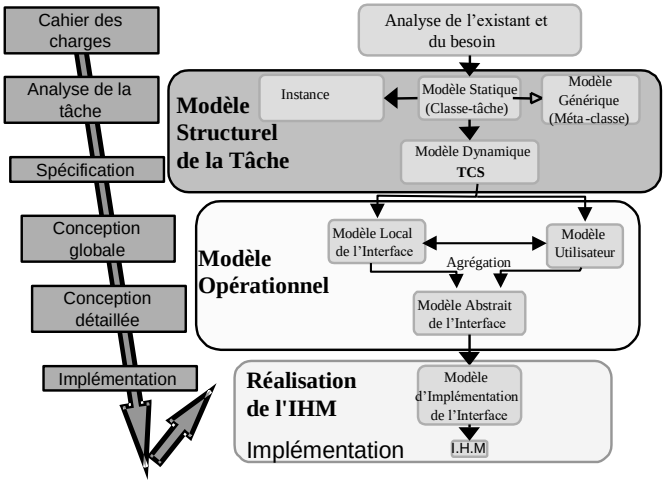



Figure 1.12 : Cycle de développement de TOOD.

Dans cette approche, certains problèmes pour la génération de l'interface restent encore non résolus : la méthode ne montre pas clairement comment procéder pour spécifier les vues graphiques ni comment elles sont générées. En outre, TOOD ne considère pas l'aspect des dysfonctionnements dans la partie analyse des SHM ce qui rend l'analyse de la tâche dissociée des différents contextes de fonctionnement du système. De même, similairement à ICO, la vérification des propriétés des réseaux de Petri à objets mérite de plus amples approfondissements.

#### 1.2.4.4 Travaux de Ait Ameur : la méthode B

Ait-Ameur, [Ait-Ameur et al., 98] vise à exploiter la méthode B pour la spécification des systèmes interactifs. La méthode B est basée sur une description formelle du modèle tout comme les méthodes Z [Spivey, 89] et VDM [Bjorner,87]. Ces méthodes consistent en effet à définir un modèle du système utilisant des attributs de variables caractérisant le système, des invariants que le système devrait satisfaire et les différentes opérations altérant ces variables. La méthode permet de supporter des spécifications à travers des machines abstraites aussi bien que les raffinements et l'implémentation. Avec cette méthode, il est possible d'assurer tout le développement dans un langage commun avec la même sémantique et les mêmes preuves techniques. Elle fournit ainsi une approche uniforme pour la description des programmes de développements.

Cette approche consiste, donc, à fournir une méthode « indépendante du modèle » qui pourra par la suite être adaptée (ou personnalisée) aux modèles spécifiques d'IHM. Le point fort de la méthode B est sa capacité de raffinement. En effet, la possibilité qu'offre cette méthode pour dériver des programmes concrets à partir des spécifications abstraites est très importante. Cependant, cette méthode n'explique pas la manière à suivre pour constituer les besoins informationnels de l'opérateur (BIO) ni comment décider des objets de l'interface. La génération effective de l'interface nécessite aussi plus de développement. Notons aussi que la principale critique apportée aux techniques telles que la méthode B, Lustre ou Lotos concerne la difficulté qu'on rencontre dans la lecture et la manipulation de leurs notations.

#### 1.2.4.5 Travaux de Rodriguez : le projet ALACIE

F.Gamboa-Rodriguez propose dans ses travaux de recherche un Atelier Logiciel d'Aide à la Conception d'Interfaces Ergonomiques (ALACIE) s'appuyant sur une méthodologie de conception cyclique fondée sur la modélisation de la tâche et le principe de prototypage des interfaces [Gamboa-Rodriguez, 98]. Le projet ALACIE se base sur deux modèles complémentaires :

- Un modèle MAD* (une extension du modèle MAD de Scapin [Gamboa-Rodriguez et al., 97], intégrant la notion de temps et la gestion des interruptions pour la modélisation de la tâche utilisateur ainsi que sa simulation,
- Un modèle SSI (Spécification Sémantique de l'Interface) permettant la spécification sémantique abstraite de l'interface et ce, en traduisant les éléments des tâches principales en éléments sémantiques de l'interface et en assignant une représentation physique à chacun de ces éléments.

F.Gamboa-Rodriguez a cherché principalement dans ses travaux à apporter des solutions à la problématique de l'intégration de l'analyse de la tâche dans les cycles de conception des interfaces. L'apport des ergonomes dans l'approche ALACIE représente une ouverture intéressante, quoique empirique, à la fluidité du passage de l'Analyse de la Tâche à la spécification des IHM. Néanmoins l'étape délicate de l'extraction des caractéristiques importantes et des BIO à partir des résultats de l'analyse de la tâche reste non formalisée.

#### 1.2.4.6 Travaux de Bodart : le projet TRIDENT

TRIDENT (Tools foR Interface Development EnvironmeNT), [Bodart et al., 95] apporte une méthodologie de développement des systèmes interactifs et fournit un environnement de génération automatique des interfaces. Il a été défini dans le cadre de la conception d'une méthodologie de développement d'applications interactives de gestion. Le but est de fournir une structure générale couvrant différents styles d'interaction (menus de sélection, langage de commande, manipulation directe, etc.) et différentes structures de dialogue (synchrone/ asynchrone, etc.). Cette structure générale sera, ensuite, spécialisée en structures affinées selon les classes de problèmes rencontrés (figure 1.12).

Dans TRIDENT, la spécification de l'interface utilisateur se base sur l'analyse de la tâche, à partir de laquelle on identifie les besoins informationnels et fonctionnels ainsi que le graphe d'enchaînement des activités pour pouvoir définir par la suite les objets de l'interface. Cependant, cette approche ne fourni pas de méthode permettant l'identification des BIO à partir de l'analyse de la tâche, ni le processus permettant l'identification des objets de l'interface à partir de ces BIO. Ce passage délicat de l'analyse de la tâche à la spécification de la présentation reste flou, aucune indication ou règle n'est donné à ce sujet.

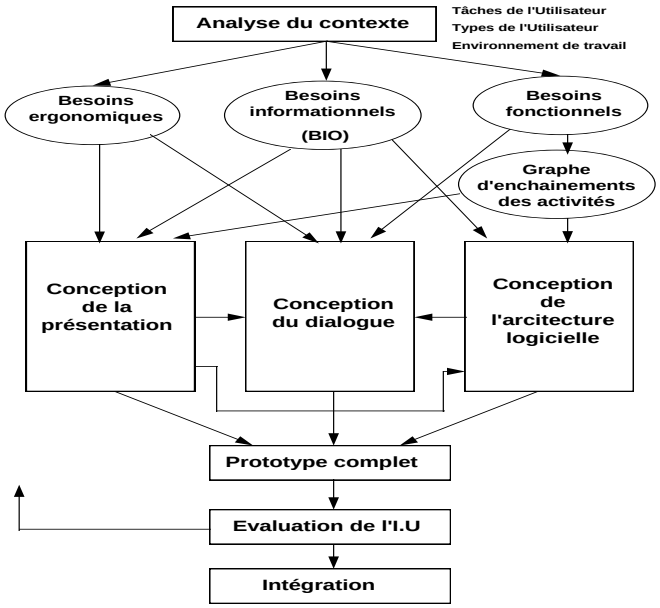



Figure 1.13 : Structure Méthodologique de TRIDENT

### 1.2.4.7 Travaux de Moussa : le système Ergo-Conceptor

Concernant le domaine particulier des applications industrielles, le système Ergo-conceptor présente une approche ergonomique de conception des IHM [Moussa, 92]. La démarche suivie par Ergo-Conceptor a pour objectif de fournir des spécifications utiles à la génération semi-automatique des vues graphiques. Ces spécifications sont déduites à partir : (voir figure 1.13):

- des données techniques du système complexe dont on veut construire les interfaces de supervision et de ses objectifs,
- ainsi que des résultats issus de l'analyse de la tâche et des facteurs ergonomiques, utiles à la génération des vues.

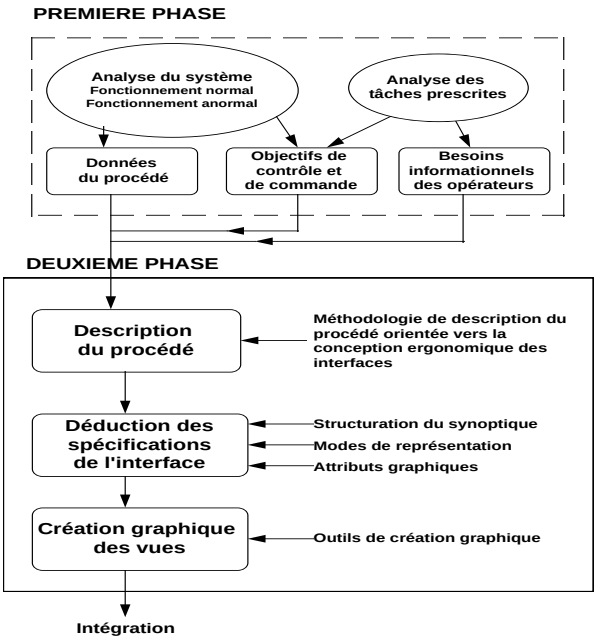



Figure 1.14 : Démarche d'Ergo-Conceptor

Bien que représentant une alternative assez originale dans le domaine de la conception des interfaces, ce système a aussi démontré certaines limites pour ce qui est de sa première version. (ayant fait l'objet de ma thèse). En effet, Ergo-Conceptor ne génère que des vues statiques, en ce sens que ces vues ne gèrent pas le dialogue homme-machine mais plutôt l'impact des variables de contrôle et de commande du système par rapport à l'interface. En outre, la méthodologie utilisée pour la description du procédé ne permet pas de gérer certains aspects fonctionnels importants tels que le parallélisme ou encore la synchronisation et ne permet pas non plus de valider mathématiquement le modèle de l'interface décrit vu son aspect assez informel. De tels constats ont été pris en considération dans le cadre de nos recherches pour étendre la démarche suivie par Ergo-Conceptor.

### 1.2.4.8 Synthèse

A la lumière de ce panorama de méthodes de conception des IHM, on relève que :

1- Aucune de ces approches ne considère l'aspect crucial de l'analyse des dysfonctionnements du système. De ce fait, les interfaces produites ne seront adaptées qu'aux situations de fonctionnement normal. Un dysfonctionnement pourrait, dans ce cas, ramener le système à une situation critique imprévisible.

2- La plupart des approches discutées ici, considèrent le concept de l'analyse de la tâche. Néanmoins, peu d'entre elles expliquent comment exploiter et intégrer les résultats de cette analyse de la tâche systématiquement dans leurs modèles.

3- Mis à part l'outil Ergo-Conceptor, aucune des approches proposées ne vise à automatiser l'utilisation des recommandations ergonomiques dans la spécification de l'interface.

Un tableau comparatif est proposé ci-dessous. Ce tableau donne une vision synthétique et rapide des caractéristiques de l'ensemble des méthodes citées.

| | **Palanque : ICO** | **Abed et al. : TOOD** | **De Rosis : XDM** | **Rodriguez : ALACIE** | **Ait-Ameur et al. : B** | **Moussa : Ergo-Conceptor** |
|---|---|---|---|---|---|---|
| **Analyse de la tâche** | Non | Oui | Non | Oui | Non | Non |
| **Analyse des dysfonctionnements** | Non | Non | Non | Non | Non | Non |
| **Expliquer comment identifier les BIO** | Non | Oui | Non | Non | Non | Non |
| **Considérer les BIO dans la conception des IHM** | Non précisé | Oui | Oui | Oui | Non | Oui |
| **Considérer les critères ergonomiques** | Non précisé | Non précisé | Non précisé | Oui | Non précisé | Oui |
| **Utiliser une technique formelle** | Oui : RdP | Oui : RdP | Oui : RdP | Oui : MAD | Oui : B | Non |
| **Fournir un outil informatique** | Oui | Oui | Oui | Oui | Oui | Oui |
| **Valider les spécifications** | Oui Partiellement | Oui Partiellement | Oui | Oui prototypage | Oui | Non |
| **Générer automatiquement l'interface** | Dialogue | Oui (partiellement) | Non | Non | Non | Vues graphiques |
| **Fournir une approche complète et intégrée** | Non | Perspective | Non | Non | non | Non |

Tableau 1.1 : Caractéristiques de quelques approches proposées pour la conception de l'interface

## 1.3 Conclusions

Comme nous l'avions souligné au niveau de l'introduction générale, ce premier chapitre a mis l'accent sur l'aspect pluridisciplinaire de la conception des IHM. Nous constatons, d'abord, qu'aucune approche n'a tenté de résoudre le problème de la conception des IHM d'une manière globale. Chacune se focalisant sur un aspect particulier de la conception. En outre, l'implication, dans ce processus de conception, d'acteurs apportant un savoir et un savoir faire pointus par discipline ne semble pas être rempli comme il se doit. Ceci implique, comme l'a montré clairement l'étude bibliographique, des lacunes touchant le processus de conception à différents niveaux et différentes spécialités. Ces lacunes peuvent être résumées comme suit :

Lacunes dans la modélisation du SHM. En effet, le recours indispensable à la modélisation du procédé, particulièrement dans les applications industrielles complexes, ne semble pas être complètement assimilé. Fort heureusement, nous constatons, ces derniers temps, l'engouement des chercheurs autour des approches de conception à base de modèle (Model Based Design). Cette classe d'approche implique forcément le recours aux modèles (procédé, tâche, interaction, etc.) dans le processus de conception. Malgré ceci, soulignons que dans la majorité des approches de conception des IHM étudiées, quand, dans les meilleurs des cas, le procédé est modélisé, celui-ci n'est pas représenté en état de dysfonctionnement. De ce fait, les interfaces produites ne seront adaptées qu'aux situations de fonctionnement normal. L'occurrence d'un dysfonctionnement ramènera le système à une situation critique imprévisible où l'absence d'informations adéquates au contexte anormal rendra la tâche de l'opérateur humain davantage difficile et son niveau de stress plus important. Le besoin de l'OH en assistance devient indispensable.

Lacunes au niveau de l'analyse de la tâche : La plupart des approches discutées ici, considèrent le concept de l'analyse de la tâche. Nous distinguons à ce niveau certains problèmes résumés ci-dessous en trois points :

- choix d'un formalisme pour la modélisation de la tâche : Comme on vient de monter, à travers certains exemples du § 1.2.1.2, selon le contexte d'application et d'utilisation, chaque approche de modélisation a choisi un ensemble d'aspects caractérisant son modèle de tâche. Il importe alors de décider de la technique de représentation à utiliser ou à mettre au point. Afin d'être facilement approuvée et adoptée par les concepteurs, cette technique doit être pédagogique et facile à utiliser. Ceci est possible en favorisant, entre autres, la simplicité et la caractéristique graphique de la modélisation.

  Adopter une notation « personnel » ou recourir à des formalismes existants tel est la question que l'on doit se poser à ce niveau de la conception. Adopter une notation personnelle conduit le plus souvent à un manque de rigueur dans la modélisation et à des problèmes de communication entre les acteurs participant au processus de conception. Aussi, des confusions peuvent apparaître du fait qu'il existe plusieurs interprétations possibles du modèle de la tâche obtenu. Ce dernier peut même s'écarter des renseignements avancés par les cogniticiens et ne plus être fidèle à l'esprit de la tâche opérateur originale.

  Adopter un formalisme existant est manifestement un choix plus judicieux. D'ailleurs, la majorité des approches que propose la littérature fait état de l'emploi de modèles plus ou moins formels. Le choix d'un formalisme adapté à la spécification de la tâche opérateur est capital. Sa rigueur, la richesse de ses opérateurs ou la facilité des ses notations sont parmi les critères de choix. Rappelons, dans ce sens, que la principale critique apportée à des techniques telles que la méthode B, Lustre ou Lotos concerne la difficulté qu'ont rencontré les utilisateurs de ces méthodes dans la lecture et la manipulation de leurs notations.

- Validation du modèle de la tâche : les travaux étudiés font état de certaines extensions apportés au formalisme RdP. Citons à titre d'exemple les RdP à Objet (RdPO), utilisés, entre autres, dans les ICO, ou encore dans TOOD. Si le pouvoir d'expression des RdP se trouve amélioré avec l'emploie des technologies objets, la validation des modèles RdPO obtenus ne devient plus évidente. En effet, afin de valider un tel modèle, il faut obligatoirement préciser les règles et les conditions permettant le passage d'un réseau de Petri à objets vers un réseau de Petri ordinaire sur lequel la vérification des propriétés pourra être menée. Chose qui n'apparaît explicitement dans la majorité des travaux.

- Passage du modèle de tâche vers les BIO et les objets de l'interface : un des avantages de l'utilisation d'une technique de spécification formelle, c'est de pouvoir générer automatiquement l'IHM et son dialogue. Rappelons que cet aspect entre dans le cadre d'une approche globale de conception, vérification et génération semi-automatique d'IHM. Une

approche supportée dans toutes ses phases par des outils informatiques emboîtés à la manière d'un lego®. La rupture à un niveau quelconque de ce processus remettra en cause l'objectif initial. Toutefois, force est de constater, que dans la littérature, la majorité des approches ne réussi pas le passage, certes délicat, de l'analyse de tâche vers la déduction des BIO et/ou du modèle spécifiant la tâche vers les objets de l'interface. Citons à titre d'exemple les travaux de Abed, Alacie, Trident, XDM et bien d'autres.

Il se dégage de cette critique un souci commun à toutes les approches, à savoir, apporter des informations adéquates répondant au mieux aux besoins des opérateurs en matière de supervision et de commande. Ceci passe par la préparation d'une assistance rigoureuse à l'opérateur basée sur une analyse complète du procédé à contrôler. Dans le chapitre suivant, nous tenterons d'apporter des réponses à ses lacunes à travers une brève synthèse de nos travaux.

# CHAPITRE 2 : CONTRIBUTIONS AUX METHODOLOGIES DE CONCEPTION DES IHM

*Notre objectif est de proposer une approche globale de conception vérification et génération semi-automatique des IHM pour le contrôle des procédés industriels permettant de surmonter les limites de la plupart des approches proposées et discutées au niveau du premier chapitre.*

*Dans ce domaine d'application, parmi les méthodologies de conception existantes, la méthodologie globale de conception et d'évaluation des SHM proposée par Millot et Abed [Millot, 90][Abed 01] [Lepreux et al., 03], constitue un cadre général très intéressant représentant une source d'inspiration dans notre travail de recherche. Il importe donc de, commencer par la présenter. Dans la deuxième section de ce chapitre, nous proposerons une synthèse des travaux et des résultats obtenus dans ce sens.*

## 2.1 Démarche globale en U dédiée aux systèmes industriels complexes

La démarche de conception et d'évaluation des SHM proposée par P. Millot et M. Abed soulève plusieurs problèmes fondamentaux liés respectivement à la modélisation du comportement décisionnel humain, à la coopération homme-machine dans les tâches décisionnelles et à la réalisation d'interfaces ergonomiques.

La démarche méthodologique de conception des SHM (démarche en U) comprend deux phases séquentielles :

- la première est une phase conceptuelle descendante de modélisation du SHM qui aboutit à la réalisation du système,

- la deuxième phase est ascendante ; elle vise à évaluer le système conçu et réalisé dans l'étape précédente. Cette étape ne sera pas proposée ici. Le lecteur intéressé peut, toutefois, consulter ses détails dans l'article de Patrick Millot paru en 1990 [Millot, 90].

La phase descendante relative à la modélisation du SHM commence par la modélisation du SHM, pour aboutir à une mise en oeuvre sur site ou à une simulation. La démarche consiste d'abord à analyser le procédé et son système de commande pour en déduire les modes de fonctionnement et de dysfonctionnements prévisibles (figure 2.1).

Ceci conduit à la définition des tâches humaines pour l'exécution desquelles il convient ensuite d'évaluer les capacités humaines à partir du modèle de l'opérateur. Cette évaluation des ressources humaines en regard des contraintes techniques à satisfaire conduit à la définition d'outils d'aide et à l'IHM [Millot, 99]. Cette phase descendante de conception se compose de cinq étapes :

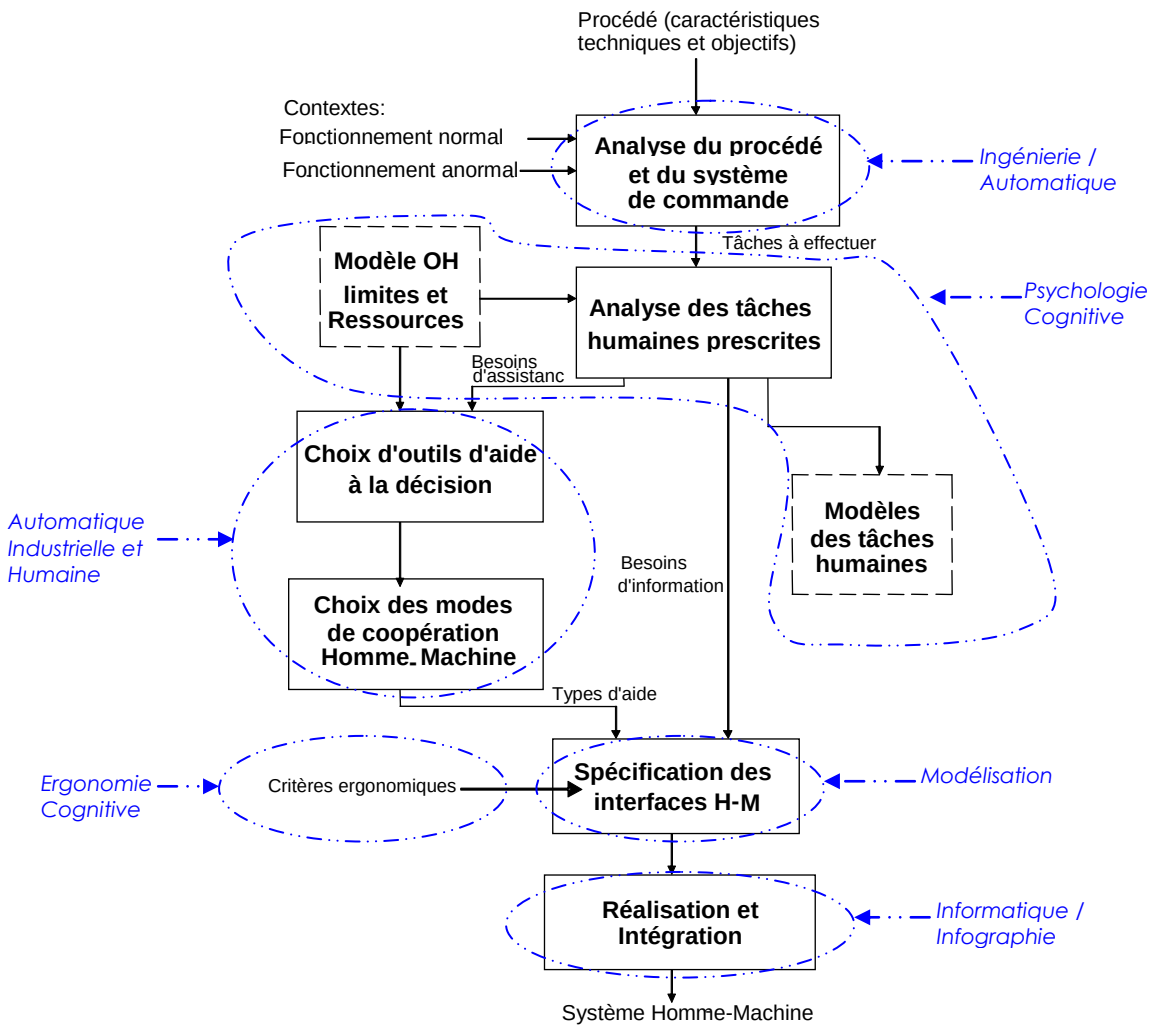



Figure 2.1 : Phase descendante de conception du système Homme-Machine inspirée de Millot.

Etape 1 : Analyse du procédé et du système de commande

Le renseignement le plus important à retenir ici, concerne l'analyse du procédé et de son système de commande. Cette analyse doit se faire, bien sûr, selon le contexte de fonctionnement prévisible normal, mais aussi et surtout en mode de dysfonctionnement, car c'est dans ces moments de perturbation que l'opérateur se sent le plus vulnérable et est le plus en besoin d'information. La définition des tâches à effectuer devient possible. Seuls les tâches manuelles et les tâches partageables (pouvant être affectées indifféremment aux automatismes ou à l'opérateur humain) seront considérés, dans la problématique de conception des IHM, dans la mesure où les tâches totalement automatisées ne concernent pas l'intervention de l'opérateur humain et par suite l'IHM.

Etape 2 : Analyse des tâches prescrites

L'analyse cognitive des tâches opérateurs doit être, à présent, réalisée. Cette analyse prend en considération l'OH. En effet, le concept d'analyse tâche dans la conception des IHM va de paire avec celui de l'Opérateur Humain. Modéliser, tant bien que mal, l'opérateur humain est essentiel, particulièrement dans les systèmes complexes. Le modèle de l'OH doit permettre, en effet, de rapprocher les ressources humaines disponibles ainsi que les limites intrinsèques des opérateurs sur les plans perceptuels (acquisition), cognitif (traitement des informations) et physiques (exécution des actions). Ce modèle sera mis à contribution lors de l'analyse des tâches. En effet,

une vérification de l'adéquation des capacités humaines aux contraintes engendrées par l'exécution des tâches, doit être faite.

Etape 3 : Spécification des outils d'assistance

Parmi les tâches humaines, certaines tâches peuvent se révéler extrêmement complexes. L'OH est alors assisté par des outils d'aide (synthèse d'information, diagnostic, aide à la reprise de défauts, etc., qu'il s'agit de spécifier.

Etape 4 : Spécification de l'IHM

La notion de Besoins Informationnels des Opérateurs (BIO) renvoie aux données déduites à partir de l'analyse de la tâche. C'est à partir de ces données que les informations à afficher sur les écrans de supervision seront choisies et composées. Un travail de structuration et d'habillage graphique ergonomique finira de composer les vues graphiques qui seront proposées à l'OH selon les contextes opérationnels du procédé. Soulignons ici, que de nombreuses expériences menées ces dernières années, arrivent, moyennant des systèmes issus de l'intelligence artificielle, de réaliser cette phase de spécification (et même de génération graphique) en temps réel et nom plus au préalable comme ce fut le cas à l'origine. Nous y reviendrons dans le chapitre 3.

Notons, enfin, l'existence d'un document de synthèse très intéressant sur la spécification des systèmes interactifs, dressant un bilan des méthodes disponibles, sans lien direct avec un domaine d'application particulier [Jambon et al., 01].

Etape 5 : Réalisation et intégration

Elle consiste en l'implémentation réelle de l'interface et son intégration avec les différents outils d'assistance et d'automatisation pour constituer ainsi le SHM. Le choix d'un outil de construction d'interface dépend fortement du domaine dans lequel on se situe (systèmes industriels, informatique de gestion, web, informatique mobile, etc.) [Fekete et al., 01]. Ces outils peuvent être classés comme suit : (i) boîte à outils, (ii) squelette d'application, Système de Gestion d'Interface Utilisateur (SGIU ou UIMS pour User Interface Management System) ou encore des outils de création graphique automatique utilisés principalement dans des approches à base de modèles (Model Based Design). Le système Ergo-Conceptor+ [Moussa et al., 2002] fait partie des approches à base de modèles.

## 2.2 Contributions

L'approche globale de conception, vérification et génération des IHM que nous proposons se compose de sept étapes (figure 2.2) :

1. Une première étape d'analyse préliminaire du procédé et de son système de commande est nécessaire. Cette analyse fournit un document rassemblant les données du procédé et ses différentes contraintes techniques et fonctionnelles. Cette étape est réalisée avec les responsables de l'installation industrielle et implique des spécialités du genre ingénierie des systèmes, génie automatique, etc.
2. La seconde étape consiste à analyser le SHM en termes de procédé, de système de commande et des tâches opérateur. Il s'agit ici d'identifier et de décrire les tâches opérateur en précisant l'enchaînement des actions pour l'accomplissement de ces tâches. Des techniques et méthodes d'analyse, telles que SADT, AMDE, AdD, sont proposées à cet effet. Cette étape voit la participation des concepteurs de l'installation industrielle, des OH futurs utilisateurs de l'interface, des psychologues, etc. Elle implique des disciplines telles que la psychologie cognitive, l'ergonomie cognitive, la fiabilité des systèmes industriels, etc.

3. La troisième étape consiste en la modélisation de l'interaction Homme-Machine. Cette modélisation doit exprimer l'interaction entre l'opérateur et l'interface. Une modélisation formelle de l'interaction homme-machine, moyennant les réseaux de Petri Interprétés, permet, par la suite, de situer en chaque point d'interaction, l'ensemble des BIO nécessaires.
4. La quatrième étape assure l'identification des BIO en termes d'objets graphiques de l'interface.
5. Une fois les objets de l'interface déduits, cette étape consiste à spécifier l'interface en termes de vues et d'objets graphiques. La prise en compte des critères ergonomiques de présentation sera assurée, à ce niveau avec l'intégration d'un outil à base de connaissances, du type TFWWG (Tools For Working With Guidelines selon la dénomination proposée par Jean Vanderdonckt au niveau international). Ce type d'outil permettra de décider du choix des objets adéquats de présentation et de la qualité ergonomique des différentes vues graphiques à générer.
6. Nous tirons bénéfice de la technique formelle utilisée dans la spécification de l'interface pour procéder à des vérifications théoriques formelles et une validation *a priori* des interfaces.
7. La dernière étape de cette approche est dédiée à la génération semi-automatique des interfaces.

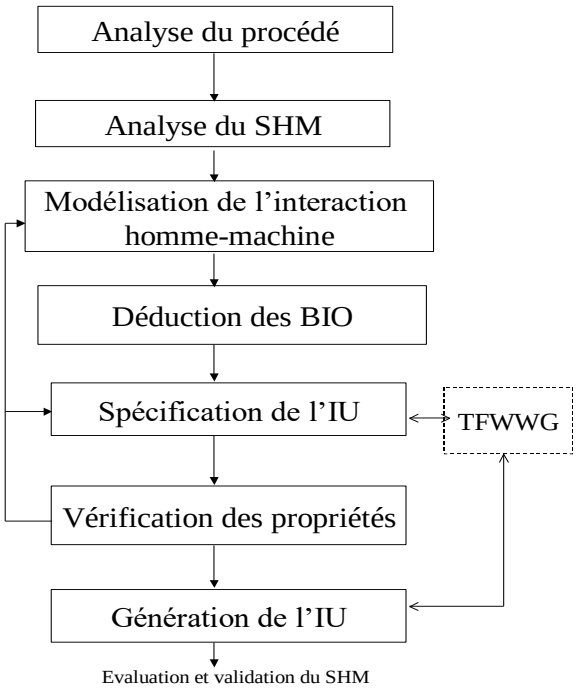



Figure 2.2 : Squelette méthodologique de l'approche proposée

Nous revenons, à présent, sur les étapes d'analyse du procédé et du SHM. Nous proposons ci-dessous un aperçu de nos contributions à ce niveau méthodologique.

Dans nos articles parus en 1995 [MOUSSA et al., 95], 1996 [MOUSSA et al., 96] et en 2000 [MOUSSA et al., 00], nous avons soulevé la question de la modélisation du procédé en vue de la génération de l'interface. Nous avons recommandé, à ce moment, la prise en charge du procédé à travers un formalisme de modélisation mis au point à cet effet. Dans ce formalisme, la description du procédé se fait selon trois axes complémentaires : une description en terme de hiérarchie d'abstraction « moyens/objectifs » (voir figure 2.3), une description structurelle et une description fonctionnelle (voir figure 2.4).

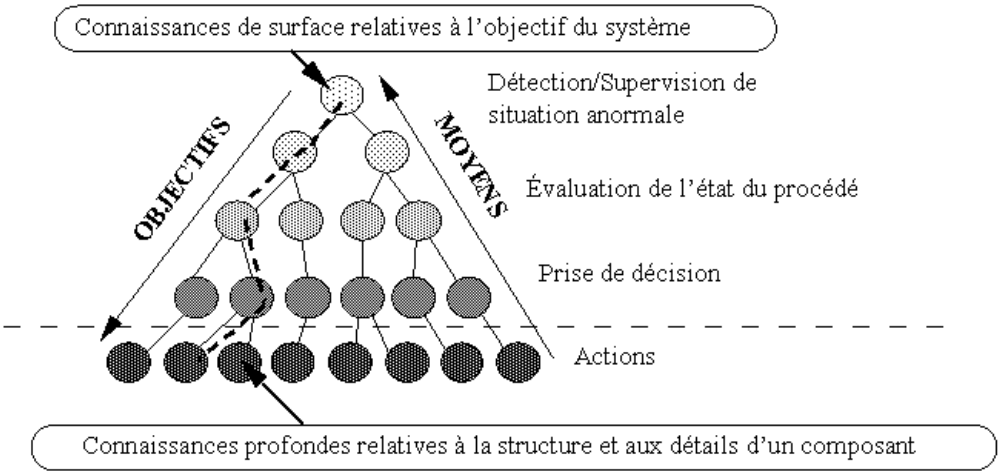
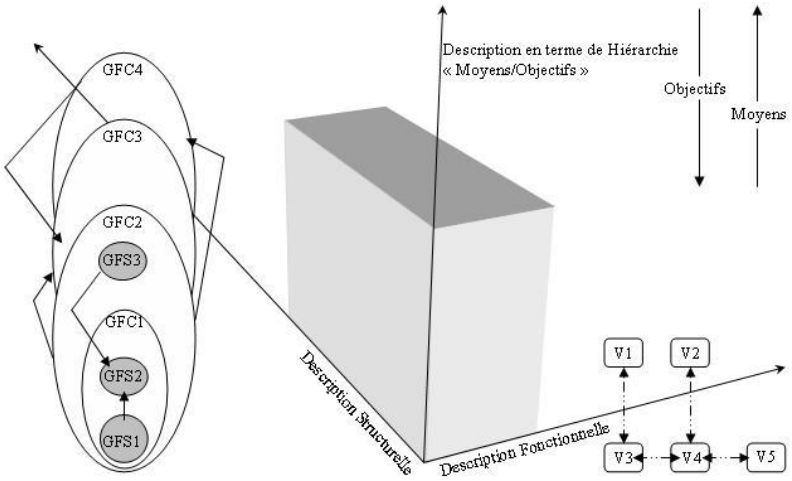



Figure 2.3 description du procédé en terme de hiérarchie d'abstraction « moyens/objectifs » [Moussa, 92]

Au niveau le plus bas, un sous-système est un élément indécomposable représenté par un Groupement Fonctionnel Simple (GFS), comprenant des composants reliés entre eux. Un Groupement Fonctionnel Composé (GFC) représente un regroupement de sous-systèmes formant un sous-système de plus haut niveau. La description fonctionnelle est réalisée grâce à la représentation des interactions entre les différents sous-systèmes qui le composent, par des liens d'influence regroupés sous forme d'un réseau de causalité. Le fonctionnement d'un sous-système est décrit par des variables interconnectées reflétant son état. La description est faite en terme d'abstraction hiérarchique moyen/objectif afin de permettre aux opérateurs humains de passer par les étapes identifiées par Rasmussen (voir § 1.2.1.1.2) lors de la résolution d'un problème. Dans cette modélisation, les informations expriment progressivement le passage d'une connaissance particulière et précise traduisant les moyens, à une connaissance synthétique relative au fonctionnement lié à l'objectif général du système.




Clés : lien d'influence : ←··→

Lien d'Entrée Sortie : ⟶

Variable : Vi

GFS : Groupement Fonctionnel Simple

GFC : Groupement Fonctionnel Composé

Figure 2.4 : Les trois axes de description d'un procédé [Moussa, 92]

Avec le recul, nous avons constaté que la modélisation du procédé au moyen d'une approche et d'un formalisme « faits maison » a vite montré ses limites. Nous avons alors proposé dans les articles parus en 1998, [Riahi et al., 98], [Riahi et al., 98a] et en 2002 [Moussa et al., 02] une nouvelle vision de la modélisation du SHM. Notre souci étant de couvrir, à travers cette modélisation, les aspects de fonctionnement normal mais aussi et surtout de dysfonctionnement dans l'objectif de mieux cerner les BIO à proposer, selon les différents contextes opérationnels. Notre attention ne se porte plus, uniquement, sur le procédé, mais sur le procédé et sur l'homme ; plus généralement sur le SHM.

L'analyse du SHM consiste à identifier les sous-systèmes significatifs et importants à contrôler et à analyser leur comportement. L'objectif recherché de cette analyse est d'identifier les différentes vues graphiques appropriées pour le contrôle et la commande de ces sous-systèmes. Le découpage du système à contrôler est donc fortement influencé par l'IHM qui va le supporter. L'objectif est de fournir à tout instant, une vue graphique par sous-système, reflétant son état de fonctionnement et regroupant l'ensemble des BIO à cet instant là. Trois étapes sont nécessaires (figure 2.5).

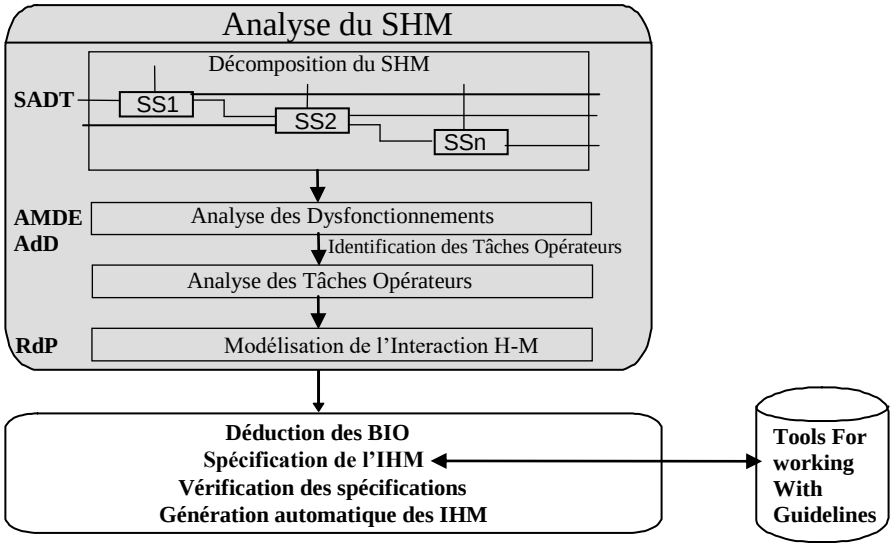



Figure 2.5: Analyse du Système Homme-Machine

**<u>Etape 1 : Procéder à une décomposition hiérarchique du SHM</u>** *:* Nous préconisons de commencer l'analyse par une décomposition hiérarchique du SHM identifiant un ensemble de sous-systèmes élémentaires plus clairs, faciles à étudier et à modéliser.

De nombreuses méthodes d'analyse et de modélisation du système sont disponibles [Larvet, 94] [Calvez, 93].

Trois principales classes de méthodes sont distinguées pour modéliser le système dans son état de fonctionnement normal :

- Une première classe procédant à une découpe cartésienne du système : ces méthodes consistent en un découpage fonctionnel et hiérarchique du système telles que SA, SADT, etc.
- Une deuxième classe relevant de l'approche systémique : cette classe de méthodes procède à deux modélisations ; l'une pour les données et l'autre pour les traitements telles que MERISE, AXIAL, etc.

- Une troisième classe de méthodes orientées objet : elles procèdent à une structuration du système en termes d'objets (encapsulant données et traitements) et relations entre ces objets comme dans Booch, OOSE, OMT, ou encore UML considérée désormais comme étant la méthode la plus complète de cette classe.

La problématique est donc de savoir choisir les meilleures méthodes d'analyse et outils de modélisation permettant de satisfaire ces besoins. Pour cela, nous avons établi un certain nombre de critères d'évaluation des différentes méthodes et outils existants, permettant de sélectionner les plus adéquats pour notre cas d'étude. Des critères sont définis ci-dessous :

- le degré du recouvrement de l'aspect structurel de l'analyse,
- le degré du recouvrement de l'aspect fonctionnel de l'analyse,
- le degré du recouvrement de l'aspect dynamique de l'analyse,
- le taux d'utilisation de la méthode dans la pratique,
- la capacité du découpage hiérarchique du système,
- le degré de visibilité de la circulation du flux d'information.
- le degré de simplicité impliquant la facilité d'apprentissage de la méthode ou de l'outil,
- l'aspect graphique de la méthode ou de l'outil de modélisation,
- le degré d'applicabilité pour le domaine concerné par notre étude à savoir les applications industrielles,
- l'existence d'un outil informatique supportant la méthode ou le langage de modélisation.

Le tableau proposé ci-dessous récapitule les résultats de l'évaluation des méthodes et outils selon cet ensemble de critères (plus de détails concernant ces méthodes sont disponibles dans [Fadier , 90], [Riahi, 04]. Nous avons choisi d'assigner, d'une manière empirique, pour chaque critère une note entre 0 et 2 montrant le degré de satisfiabilité de la méthode ou de l'outil à ce critère :

- 0 : ne satisfait pas le critère
- 1 : satisfait moyennement le critère
- 2 : satisfait bien le critère

Selon nos objectifs d'analyse, cette évaluation informelle montre que la méthode SADT est la plus adéquate. En effet, la méthode SADT, bien que développée en fin des années 70, reste parmi les méthodes d'analyse fonctionnelle les plus utilisées pour les systèmes industriels. Elle est particulièrement adaptée à l'analyse et la conception des systèmes. L'analyse d'un système par la méthode SADT suit, en effet, une démarche descendante, modulaire, hiérarchique et structurée Elle est proposée, ici, spécialement pour avoir une vue structurée sur le système global, qui est généralement complexe. La décomposition avec SADT nous permettra par la suite :

- d'étudier les sous-systèmes considérés comme « élémentaires »,
- de préciser l'ensemble des variables pertinentes d'entrée et de sorties de chaque sous-système ainsi que l'ensemble des contraintes et moyens adéquats,
- et d'appliquer, aux systèmes élémentaires dégagés, les méthodes d'analyse de dysfonctionnements.

| | Méthodes d'analyse des systèmes | | | | | |
|---|---|---|---|---|---|---|
| Méthodes et Outils<br>Critères d'évaluation | SA | SADT | SART | Graphe de fluence | MERISE | UML |
| Recouvrement de l'aspect structurel | 0 | 2 | 1 | 0 | 1 | 1 |
| Recouvrement de l'aspect fonctionnel | 2 | 2 | 2 | 2 | 2 | 2 |
| Recouvrement de l'aspect dynamique | 0 | 0 | 2 | 1 | 1 | 2 |
| Taux d'utilisation en pratique | 1 | 2 | 2 | 1 | 2 | 2 |
| Découpage hiérarchique du sys. | 1 | 2 | 1 | 0 | 1 | 0 |
| Visibilité de la circulation du flux d'info. | 2 | 2 | 2 | 1 | 2 | 1 |
| Simplicité | 2 | 2 | 1 | 2 | 2 | 1 |
| Aspect graphique | 2 | 2 | 2 | 2 | 2 | 2 |
| Domaine d'application industrielle | 2 | 2 | 2 | 2 | 0 | 1 |
| Outil informatique | 2 | 2 | 2 | 0 | 2 | 2 |
| **TOTAL** | **14** | **18** | **17** | **11** | **15** | **14** |

Tableau 2.1 : Evaluation informelle des méthodes d'analyse des systèmes

**Etape 2 : Analyser les dysfonctionnements de chaque sous-système** *:* une fois les sous-systèmes élémentaires identifiés, l'analyse des dysfonctionnements devient possible.

Rappelons ici que notre objectif de mener une analyse des dysfonctionnements des systèmes, doit déboucher principalement sur le fait de pouvoir dégager l'ensemble des variables pertinentes et significatives pour chaque cas de fonctionnement du système ainsi que l'ensemble des causes possibles d'un dysfonctionnement donné, afin de pouvoir aider l'opérateur à localiser la cause de la panne et à rétablir une situation de fonctionnement normal.

Trois principales classes de méthodes sont distinguées pour modéliser le système en mode dégradé ou de dysfonctionnement. (Pour une revue quasi-exhaustive, voir l'article « Fiabilité Humaine: Méthodes d'analyse et domaine d'application [Fadier, 90] ou le livre « Sûreté de fonctionnement des systèmes industriels ; fiabilté, facteurs humains, informatisation » [Villemeur, 88].

Une première classe d'analyse ayant pour objectif de procéder à l'identification préliminaire des dysfonctionnements, de leurs causes et des dangers encourus. D'évaluer la gravité de ces dysfonctionnements et de déduire les moyens à mettre en œuvre pour les maîtriser. Citons ici la méthode d'Analyse Préliminaire des Dangers (APD) qui représente au mieux cette classe.

- Une deuxième classe de méthode qui consiste à analyser et regrouper les pannes. Citons ici :
  - la méthode AMDE (Analyse des Modes de Défaillances et de leurs Effets)
  - la méthode MCPR (Méthode de Combinaisons des Pannes Résumées)
  - la méthode MTV (Méthode de la Table de Vérité)

- et une troisième classe de méthodes proposant une analyse détaillée causes/effets d'une panne
  - La méthode MDS (Méthode de Diagrammes de Succès
  - - La méthode AdD (Arbre des Défaillances)
  - - La méthode MACQ (Méthode de l'Arbre des Conséquences)
  - la méthode MDCC (Méthode du diagramme Causes-Conséquences)

L'AMDEC (Analyse des Modes de Défaillances et de leurs Effets et Criticités) [Suhner et al., 92], permet d'établir la liste des dysfonctionnements possibles d'un système donné et l'AdD (Arbre des Défaillances) [Lambert, 73] la complète en analysant finement ces défaillances et précisant leurs causes possibles. Une combinaison de ces méthodes paraît donc la plus adéquate pour notre cas d'étude. Pour plus de détails concernant le choix des méthodes d'analyse des systèmes en cas de dysfonctionnement [Riahi, 04].

**Etape 3 : Identifier pour chaque sous-système et pour chaque contexte de fonctionnement, les tâches nécessaires de l'opérateur** *:* l'analyse des différents contextes de fonctionnement du système étudié étant faite, il importe, à présent, d'identifier pour chacun des contextes préalablement identifiés (contexte de fonctionnement normal et contextes de fonctionnement anormaux) l'ensemble des tâches opérateur nécessaires pour le suivi et le contrôle. L'analyse de ces tâches permet de définir les actions à entreprendre par les opérateurs. Les paramètres de ces actions constitueront avec l'ensemble des variables d'entrée/sorties et les variables de contrôle (identifiées au moyen de SADT, AMDE et AdD) l'ensemble des besoins informationnels de l'opérateur (BIO).

Ainsi, cette analyse du SHM a pour principal objectif d'identifier les différents contextes de fonctionnement du système ainsi que les interventions de l'OH relatives à ces situations. L'étape qui suit consiste à modéliser le processus de l'interaction homme-machine à travers cet ensemble de vues. Les réseaux de Petri interprétés sont proposés à cet effet. Ce passage depuis les trois étapes déjà exposées vers cette quatrième étape, implique un changement fort intéressant qui mériterait un rapide éclaircissement. En effet, nous franchissons ici une frontière virtuelle séparant :

- une première phase où il est fortement question d'un savoir et d'un savoir faire typiquement méthodologiques impliquant des connaissances, et sur le procédé (analyse systémique, fiabilité, etc.) et sur l'homme (identification et analyse des tâches, définition des activité, BIO, etc.).
- et une deuxième grande phase où il est question d'un savoir plus concret, plus technique faisant appel, le plus souvent, à des techniques informatiques de représentation et d'exploitation de données issues de la première phase (spécification de la tâche, .spécification de l'interface graphique, vérification des propriétés de l'IHM, génération de l'IHM, etc.)

La méconnaissance de l'existence de cette frontière virtuelle et le manque de rigueur dans l'application des recommandations de cette approche peuvent être à l'origine de confusion sur le rôle que doit remplir chaque acteur, selon sa spécialité, dans le processus complet de conception des IHM. A titre d'exemple, l'informaticien ne doit pas être tenté de se substituer au cogniticien dans l'exercice d'analyse de la tâche, quand son rôle doit se limiter à sa spécification informatique et sa génération.

Nous passons à présent à la dernière étape de la modélisation du système homme-machine (voir figure 2.5). L'objectif, ici, est de proposer une approche de modélisation de l'interaction homme-machine lors de l'accomplissement de l'opérateur de ses tâches de supervision et de commande [Riahi et al. 00] [Moussa et al., 02]. Cette approche doit fournir un modèle vérifiable sur lequel on

pourra valider certaines propriétés importantes de l'interface. Nous avons, donc, choisi la technique formelle des réseaux de Petri. En effet, on considère que cette technique est assez efficace pour l'expression d'aspects présents en interaction homme-machine comme le synchronisme, la concurrence ou le parallélisme. Les RdP proposés ici sont les RdPI (Réseaux de Petri Interprétés) [Moalla, 85]. Ce type de réseaux introduit les notions d'événements et de conditions ainsi que la notion d'actions.

Cette technique est largement employée dans le domaine des IHM. Au début, les RdP n'ont été utilisés qu'à des fins de description des tâches à informatiser. Mais par la suite, et surtout avec l'apparition des Réseaux de Petri de Haut Niveau, les chercheurs ont profité de leur puissance pour modéliser le dialogue Homme-Machine. Willem et Biljon ont été les premiers à avoir pensé à adapter les RdP à la modélisation du dialogue Homme-Machine [Willem et al., 88]. Beaucoup d'autres travaux ont suivi, citons ceux de Palanque avec le formalisme ICO; ceux de Mahfoudhi et Tabary avec TOOD ou encore ceux de De Rosis avec XDM. Toutefois, la puissance offerte par les capacités d'extension des RdP ordinaires a son revers. Il se situe principalement au niveau de la perte, en partie, de la validation des propriétés fortes des modèles obtenus.

Dans la suite, nous présentons (i) une taxonomie des techniques de spécification des systèmes interactifs, (ii) notre principe de construction du modèle de l'interaction homme-machine au moyen des RdP Interprétés garantissant le maintient de la validation de propriétés telles que l'absence de blocage, la disponibilité de service, l'absence de conflit et la stabilité et (iii) nous apporterons, également, des éléments de réponses aux étapes de déduction des BIO de la spécification du dialogue homme-machinet et de la vérification des spécifications obtenues.

# CHAPITRE 3 : OUTILS FORMELS POUR LA SPECIFICATION DES IHM

*Dans ce chapitre nous proposons, d'abord, une taxonomie des techniques de spécification. Nous passerons en revue l'apport des sciences cognitives, des approches algébriques, des théories des langages, des théories des graphes et enfin des logiques temporelles.*

## 3.1 Taxonomie de spécification des systèmes interactifs

Dans le processus global de conception des IHM, l'activité de spécification (procédé, tâche interactive de supervision, etc.) fait appel aux notations et formalismes pour supporter l'activité de conception dont l'objet est la modélisation du système à concevoir. Le formalisme est, donc, une convention de représentation des concepts du modèle [Barthet, 88].

Nous proposons dans ce chapitre une taxonomie des techniques de spécification faisant ressortir le champ d'application et l'apport des différentes familles d'approches de modélisation (figure 3.1). En voici un recueil non exhaustif :

Rappelons, à ce sujet, la publication d'une des rédactions les plus complètes, parue en 2001, dans le volume 1 du livre « Analyse et Conception de l'IHM, Interaction Homme-Machine pour les Systèmes d'Information » [Jambon et al., 01].

### 3.1.1 L'apport des sciences cognitives

Soulignons, ici, que les sciences cognitives (voir § 1.2.1, 1.2.2) ont donné lieu à des modèles théoriques et expérimentaux. Les modèles théoriques comme la théorie décisionnelle de Rasmussen [Rasmussen, 83] ou la théorie de l'action de Norman [Norman, 86] décrivent les processus cognitifs qui régissent le comportement de l'utilisateur tandis que les modèles expérimentaux décrivent le comportement lui-même. Citons, ici à titre d'exemples (voir § 1.2.4), le formalisme nommé XDM [De Rosis et al., 98] intégrant lui même la théorie des opérateurs KLM (Keystrokes-Level-Model) [Card et al., 83] ; UAN (User Action Notation) [Hix et Hartson, 93], etc. Comme nous l'avions déjà mentionné aux chapitres un et deux, les renseignements tirés de cette classe d'approches (qu'elles soient théoriques ou expérimentales) ont enrichi considérablement notre contribution à la mise en place d'une démarche globale de conception et de génération semi-automatique des IHM pour les systèmes industriels complexes.

### 3.1.2 L'apport des approches algébriques

Dans cette catégorie de formalismes on distingue deux approches issues de l'algèbre [Jambon et al., 01] : une approche basée sur la théorie des modèles et une seconde approche basée sur la théorie des catégories et des types abstraits appelée approche algébrique.

#### 3.1.2.1 L'approche basée sur les modèles

Appelée aussi approche constructive, opérationnelle ou basée sur l'état, l'approche basée sur les modèles consiste à définir un modèle du système à l'aide d'opérations qui ne sont pas liées au type de donnée que l'on veut étudier. Une opération est spécifiée à l'aide de pré- et post-conditions et s'applique à la manipulation des ensembles et à la consultation de l'état du système. Les opérations dont on se sert forment le modèle sous-jacent sur lequel la définition du type est construite. Ce modèle sous-jacent est en général celui de la théorie des ensembles et des relations muni des opérations d'unions, différence, intersection, etc.

Les représentants les plus connus de cette famille de langages sont VDM [Jones, 80], apparu dès les années 60, et Z [Spivey, 89] apparu avec les années 70. Une autre variante de ces langages a été proposée par la suite, avec la méthode B [Abrial, 96].

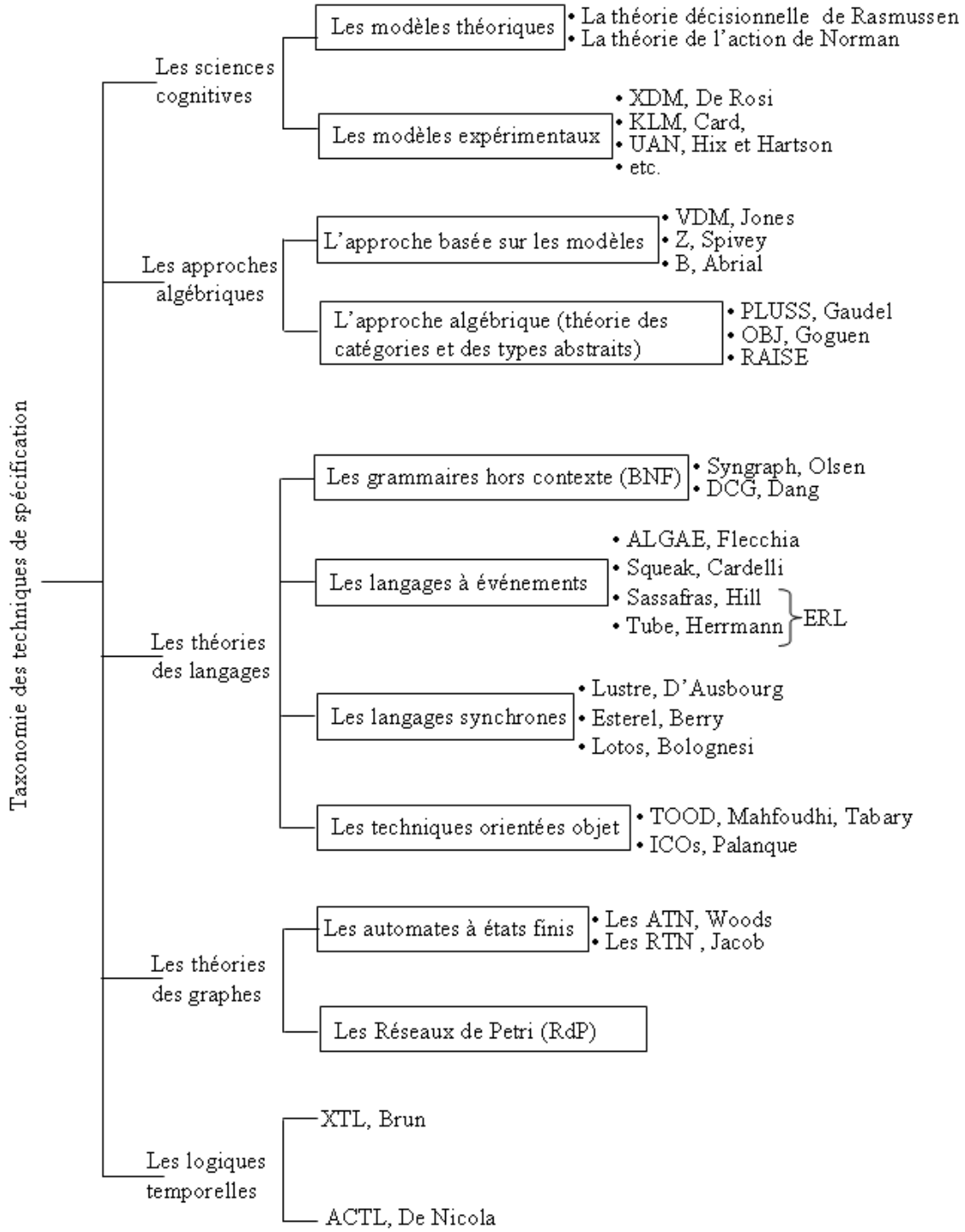



Figure 3.1 : Taxonomie des techniques de spécification (inspirée de [Jambon et al., 01])

Ces techniques se différencient suivant la manière d'utiliser les notations ensemblistes, la logique employée ou la facilité avec laquelle elles permettent de passer de la spécification à la programmation.

Avec Z, la notion d'ensemble est utilisée comme moyen d'expression universel, ce qui a l'avantage de l'uniformité : l'espace des états du système modélisé est un ensemble, les types sont des ensembles, même les opérations sont des ensembles. Elles sont modélisées par des relations, c'est-à-dire des sous-ensembles du carré cartésien de l'ensemble des états.

Le système de notations possède des symboles pour les différentes sortes de relation (fonctions, injections, etc.) et différents opérateurs permettant de construire des relations.

Les différentes structures et opérations sont déclarées au moyen de tables appelées schémas. Un schéma comporte deux parties :

- une partie déclarative définissant les variables (un peu dans le style du langage Pascal), à partir des types construits au moyen d'ensembles prédéfinis et des opérateurs ensemblistes usuels (union, produit cartésien,…), et
- une partie prédicat précisant les contraintes sur les valeurs au moyen d'assertions logiques.

La méthode B est née avec l'idée d'un processus conduisant d'une spécification au code exécutable. Elle utilise la théorie des ensembles dans le même esprit que Z et en reprend les notations. Elle diffère pour les mécanismes de structuration et la manière de spécifier les changements d'états. Elle regroupe :

- un langage de spécification (les machines abstraites),
- une technique de raffinage et de codage,
- des obligations de preuve associée à chaque étape,
- des mécanismes de structuration visant à décomposer les machines abstraites,
- des outils permettant d'assister et de contrôler les tâches.

Le langage Z a été proposé pour la modélisation de l'interaction homme-machine dès les débuts des années 90 [Harrison et al., 90]. Récemment, la méthode B a été adoptée pour la spécification des IHM. Cette première expérience est présentée dans les travaux de Ait-Ameur et al. (voir chapitre 1).

#### 3.1.2.2 L'approche algébrique

L'objectif, de cette classe d'approches, est de décrire un système par un ensemble d'équations pour en saisir le comportement. Les approches les plus connues sont PLUSS [Gaudel, 84], OBJ [Goguen, 96] et RAISE [RAISE, 92]. Ces approches sont bien adaptées à la spécification des systèmes séquentiels, en revanche les notions de concurrence ou de comportement temporel ne sont pas explicitement pris en compte. Chose préjudiciable pour les systèmes industriels complexes qui requièrent, entre autres, l'usage de formalismes modélisant naturellement les notions de concurrences.

### 3.1.3 L'apport des théories des langages

#### 3.1.3.1 Les grammaires hors contexte

Les grammaires permettent de définir la syntaxe du dialogue, c'est-à-dire l'ensemble des actions autorisées pour l'utilisateur. Les spécifications sont souvent issues du formalisme BNF (Backus Naur Form). Les grammaires hors contexte [Payne, 86] se sont avérées particulièrement efficaces pour des interfaces du type textuel, mais beaucoup moins adaptées pour les interfaces graphiques à haute interactivité. La gestion des erreurs est souvent très complexe à mettre en œuvre [Karsenty, 90].

Citons à titre d'exemples Syngraph (SYNtax direct GRAPHics) [Olsen et al., 83] qui génère un programme sur la base d'une description écrite à l'aide d'une grammaire étendue du formalisme BNF ou encore DCG (Definite Clause Grammar) [Dang, 88] qui définit une règle au moyen de trois éléments dans sa partie droite : (i) les terminaux sont les actions en entrée des utilisateurs, (ii) les non terminaux sont spécifiques au type d'interface que l'on considère (lecture au clavier, choix

d'un item dans un menu, etc.) et (iii) les commandes correspondant à des appels de procédure de l'application.

L'aspect textuel de la notation, la difficulté à gérer les erreurs et les événements imprévus représentent un handicap majeur dans l'application de ce type d'un formalisme à grammaire hors contexte dans les IHM graphiques.

#### 3.1.3.2 Les langages à événements

Le modèle à événements résulte des travaux autour de Smalltalk [Goldberg et al., 83]. Le modèle à événement peut être vu comme étant une application interactive composée de plusieurs modules dirigés par des événements. Ces événements peuvent être produits soit par des périphériques, soit par l'interface elle-même, soit par l'application.

Un événement est caractérisé par son origine et un ensemble d'attributs. Le traitement d'un événement déclenche une action qui peut provoquer l'exécution de fonctionnalités à l'intérieur de l'interface ou dans l'application, générer de nouveaux événements, créer ou détruire des gestionnaires d'événements [Green, 86].

Le modèle à événements est utilisé dans bon nombre de systèmes comme ALGAE [Flecchia et al., 87], Squeak [Cardelli et al., 85]. Dans ces systèmes, les gestionnaires d'événements sont programmés à l'aide d'un langage spécialisé classique. Au contraire Sassafras [Hill, 86] et Tube [Herrmann, 89] utilisent ERL un langage basé sur le formalisme des règles de production.

De tels formalismes présentent l'avantage de pouvoir traiter des processus multiples, ce que grammaires et automates ne permettent pas. Les inconvénients résident, toutefois, dans la complexité de contrôle des flux d'entrées/sorties ainsi que dans la difficulté de compréhension et de mise à jour du code quand la taille de l'application devient conséquente.

#### 3.1.3.3 Les langages synchrones

Les langages synchrones ont été introduits spécialement pour la programmation des systèmes réactifs. Dans les langages synchrones, un programme peut être considéré comme un réseau d'opérateurs, et l'hypothèse de synchronisme consiste à supposer que chaque opérateur de ce réseau répond instantanément à ses entrées [Halbwachs et al., 91]. Nous citons, ci-dessous, à titre d'exemple les langages synchrones Lustre et Esterel et nous terminons avec Lotos qui assure la modélisation de communication synchrone et asynchrone.

##### 3.1.3.3.1 LUSTRE

Le langage Lustre est un langage synchrone fondé sur le modèle flot de données. L'idée de base est d'exprimer, avec Lustre, les propriétés requises d'un programme, profitant de sa nature déclarative et d'utiliser, par la suite, l'automate produit par le compilateur pour vérifier ses propriétés. Lustre s'appuie sur trois concepts fondamentaux :

- le flot de données : c'est un couple formé d'une suite, éventuellement infinie, de valeurs d'un type donné, et d'une horloge représentant une suite d'instants.
- Un opérateur de retard «pre»
- Un opérateur d'initialisation (→)

Un programme lustre est composé d'un système d'équations et d'assertions. Les assertions sont une généralisation des équations. Elles consistent en des expressions booléennes qui sont supposées toujours vraies, pour optimiser le code.

Exemple : assert not(x and y) : pour exprimer le fait que deux événements en entrée représentés par les flots booléens x et y, ne surviennent jamais simultanément.

Ce langage a été proposé avec les travaux de D'Ausbourg et al. [D'Ausbourg et al., 96] [D'Ausbourg et al., 97], pour la vérification automatique des propriétés des IHM.

#### 3.1.3.3.2 ESTEREL

Citons, également, le langage synchrone Esterel [Berry et al., 88] comme exemple. Esterel est destiné à décrire des systèmes temps réels dits réactifs. Il est considéré comme étant un modèle à événements et est composé d'un ensemble de modules communiquant entre eux par envoi d'événements. Le rôle d'Esterel est de décrire le comportement d'objets interactifs [Clement, 88] en réponse à un stimulus, et ce, indépendamment de leur représentation finale (figure 3.2).

```
<être à l'intérieur ou à l'extérieur d'un bouton>
<contrôler l'inversion du bouton>
<attendre le relachement du click souris et agir en conséquence>
LOOP
<traitement de l'apparence d'un bouton>
call XOR () ( ?BUTTON) ;        <inversion vidéo>
await BUTTON_OUT                <attendre le signal de sortie>
call XOR () ( ?BUTTON)          <retour apparence normal>
        await BUTTON_IN         <attendre le signal d'entrée>
END
```

Figure 3.2 : Extrait d'un programme Esterel.

#### 3.1.3.3.3 LOTOS

Lotos constitue l'une des techniques de description formelle basée sur le concept des systèmes à états et transitions pour la spécification du comportement. Cette technique permet la modélisation du comportement séquentiel, du choix et du non déterminisme d'une manière non ambiguë. Elle assure la modélisation de communication synchrone et asynchrone. C'est une technique proposée pour des types de données abstraits (ADT : Abstract Data Types) [Turner, 93].

Avec Lotos [Bolognesi, 89] on définit quatre parties :

- la syntaxe,
- la sémantique statique,
- la sémantique algébrique des types de données, et
- la sémantique dynamique du comportement.

Lotos se base sur trois concepts de base :

- les processus (le système et ses composants) : un processus est représenté par une boite noire avec des points de connexion pour représenter ses attachements physiques ou logiques avec le reste du système,
- les événements, et
- les arbres pour exprimer le comportement (figure 3.3)

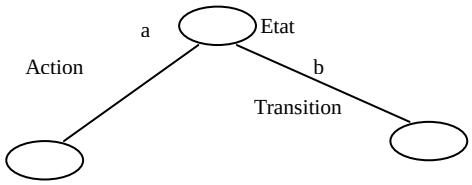



Figure 3.3 : Arbre en Lotos

Cette technique a servi de base pour la description formelle du concept d'interacteur dans la spécification des IHM. Nous pouvons citer à ce sujet principalement les travaux réalisés à Pise : FullLotos [Paterno et al., 92] et TLIM (Task Lotos Interactors modelling Model) [Paterno et al., 99].

A l'image des Réseaux de Petri, les langages synchrones permettent de régler le problème du parallélisme ainsi que celui du synchronisme des tâches utilisateur. Toutefois, ils présentent des difficultés pour le contrôle des flux d'entrées/sorties et pour la compréhension et la mise en œuvre du code vu le bas niveau de l'interaction.

#### 3.1.3.4 Les techniques orientées objet

Les techniques orientées objet, que nous ne présentons pas ici vu l'abondance de la littérature à ce sujet, se sont révélées efficaces pour l'implémentation des systèmes interactifs [Bernstein, 95] [Young, 92]. Bien que les langages à objets soient maintenant très répandus, il n'existe toujours pas de formalisme orienté objet qui soit largement accepté. On peut néanmoins citer les formalismes TOOD [Mahfoudhi, 97], [Tabary, 00a] (Description Orientée-Objet des Tâches utilisateur) et les ICOs [Palanque et al., 95] (Interactive Cooperative Objects, i.e. réseaux de Petri orientés objet) qui ont été utilisés avec succès pour la spécification des systèmes interactifs. Signalons, toutefois, qu'il reste beaucoup à faire sur le plan de la validation des modèles obtenus sur la base de cette technique (voir § 1.2.4.1 et 1.2.4.3).

### 3.1.4 L'apport de la théorie des graphes :

La théorie des graphes a engendré de très nombreux formalismes comme les automates ou les réseaux de Petri.

#### 3.1.4.1 Les automates à états finis

Les automates d'état offrent l'avantage de visualiser les enchaînements d'actions autorisées. Les automates sont spécifiés statiquement par l'ensemble des états d'interaction et dynamiquement par un ensemble d'arcs étiquetés par des jetons d'entrée représentant les actions de l'utilisateur, caractérisés par un nom et un ensemble de données (figure 3.4).

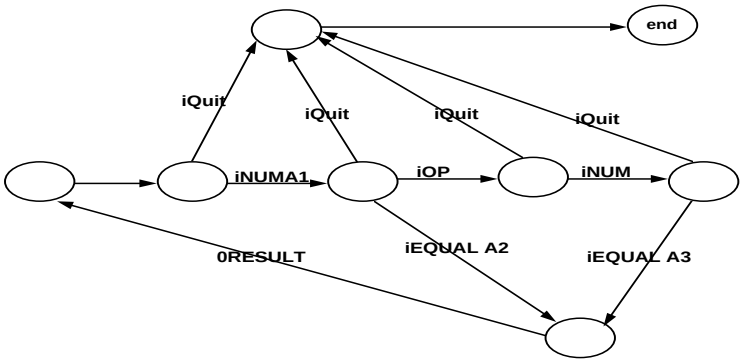



Figure 3.4 : Automate à états finis décrivant une opération de calcul dans une calculatrice

A l'aide de cette spécification, on devient capable de décrire les actions du système et leurs séquences vis-à-vis des entrées [Harel et al., 87] [Vallee, 92]. Mais en contrepartie, l'automate devient complexe et difficile à comprendre à cause de sa taille qui grandit rapidement en fonction des états possibles. Plusieurs extensions ont été proposées :

* *Les ATN* (Augmented Transition Network) : proposés par Woods, ils introduisent des extensions au schéma de base du diagramme de transition récursif en associant des variables simples, des conditions et des expressions de sorties textuelles aux noeuds et/ou arcs de l'automate.

* *Les RTN* (Recursive Transition Network) : les arcs y sont étiquetés par des appels récursifs à d'autres automates afin de limiter le problème majeur de la taille des diagrammes [Jacob, 85].

Les diagrammes états-transitions ont été proposés dès les années 60 pour la modélisation de l'enchaînement des écrans graphiques [Parnas, 69]. Jacob a proposé par la suite l'utilisation de ces diagrammes pour les interfaces à manipulation directe [Jacob, 85]. Actuellement, ils font l'objet d'extensions dédiées à la modélisation de la navigation dans les applications web (StateWebCharts) [Winckler et al., 03].

#### 3.1.4.2 Les Réseaux de Petri

Les réseaux de Petri (RdP) ont été proposés par Carl Adam Petri en 1962. Ils représentent un formalisme mathématiquement fondé dédié à la modélisation du parallélisme et du synchronisme dans les systèmes discrets, prenant en considération les aspects événementiels qui ne peuvent pas être traités par les automates [Petri, 62] [David et al., 92] [Berthomieu et al., 01] [Girault et al., 02].

Un RdP est un graphe biparti fondé sur deux types de noeuds : " O " pour représenter les places et " —" pour représenter les transitions avec un ensemble d'arcs reliants les différentes places aux différents transitions (figure 3.5). Il est défini par un quadruplet R = <P, T, Pré, Post > avec :

- **P** : l'ensemble des places du réseau.
- **T** : l'ensemble des transitions du réseau.
- **Pré : P x T → |N**, une fonction d'incidence avant, définissant le nombre de jetons nécessaires dans une place Pi entrée d'une transition Tj permettant la franchissabilité de cette transition.
- **Post : P x T → |N**, une fonction d'incidence arrière, définissant le nombre de jetons à déposer dans une place Pi sortie d'une transition Tj après son franchissement.

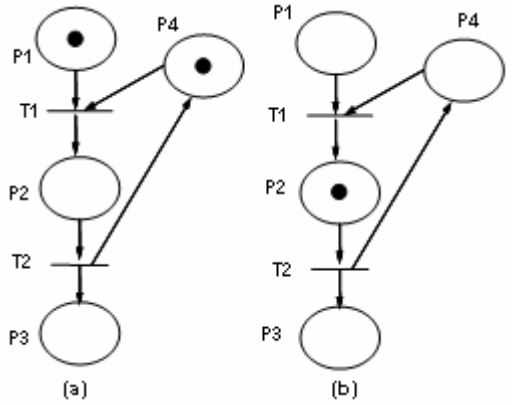



Figure 3.5 : Exemple de l'état d'un RdP avant et après le franchissement d'une transition

Les RdP sont en expansion continue et représentent un outil adéquat pour la modélisation de l'Interface Homme-Machine. Au début, ils n'ont été utilisés qu'à des fins de description des tâches à informatiser. Mais par la suite, et surtout avec l'apparition des Réseaux de Petri de Haut Niveau, on a commencé à profiter de leur puissance pour modéliser le dialogue Homme-Machine. Nous citons, principalement, les travaux de Willem et Biljon qui ont été les premiers à avoir pensé à adapter les RdP à la modélisation du dialogue Homme-Machine [Willem et al., 88] ; ceux de Palanque et al. avec le formalisme ICO ; ceux de Tabary et al. avec TOOD (faisant suite aux travaux de [Abed, 90] [Abed et al., 98]) et ceux de De Rosis et al. avec XDM (voir chapitre 1).

Toutefois, la vérification des propriétés du système est souvent présentée comme étant « un problème difficile à résoudre » du fait de l'explosion combinatoire des états possibles. En effet, L'analyse d'un RdP nécessite souvent le développement du graphe des marquages accessibles dont la taille rend généralement son étude laborieuse. Nous développons, dans le chapitre suivant, une technique de modélisation au moyen des RdP Synchronisés, (appliquée à la spécification de l'interaction homme-machine, à la spécification du dialogue homme-machine et enfin à la vérification des propriétés de l'interface), permettant, à notre sens, de contourner ce problème d'explosion combinatoire constaté dans les RdP de taille importante.

### 3.1.5 L'apport des logiques temporelles

Les logiques temporelles commencent à être utilisées depuis plusieurs années pour la spécification des systèmes interactifs aux stades de l'analyse des besoins et des spécifications principalement [Brun, 98], [Johnson, 95]

Des modèles d'interaction abstraits basés sur la logique du premier ordre ont été proposés pour fournir une structure de spécification de l'Interface pour les systèmes complexes. Dans ces modèles, l'interaction est décrite par les quatre ensembles suivants [Johnson et al., 92] :

P : qui exprime l'ensemble des séquences d'entrée,

C : qui définit l'ensemble des commandes possibles,

S : qui définit l'ensemble des états possibles du système et

D : qui décrit l'ensemble des écrans possibles.

Quelques exemples de spécification utilisant cette logique sont donnés ci-dessous :

* Exemple 1 : $\forall\ p \in P, \forall\ s \in S,\ \exists!\ c \in C$: interpret (p, s, c)

Ce qui veut exprimer que pour toute entrée p, il existe une unique commande c, telle que l'interprétation de p dans l'état s est la commande c.

* Exemple 2 : $\forall\ c \in C, \forall\ s \in S,\ \exists!\ s' \in S$: effect (c, s, s')

Cette exemple veut dire que l'effet de toutes les commandes c est de transformer l'état du système vers l'unique état s'.

* Exemple 3 : $\forall\ s \in S,\ \exists\ d \in D$: view (s, d)

Cela veut dire que tous les états s peuvent être visualisés à travers un écran d.

A l'aide de cette logique abstraite et déclarative, le concepteur n'aura pas besoin de spécifier les périphériques à utiliser dès le départ. Les détails d'entrées, de commandes et de comportement pourront être ajoutés au fur et à mesure que la conception progresse. Mais la principale limite de ce formalisme est qu'il ignore l'information temporelle et donc le séquencement des commandes ne peut pas être modélisé. Comme solution à ce problème, Johnson et Harrison ont proposé la logique temporelle qui est une extension de la puissance déclarative statique de la logique du premier ordre. Le but de cette logique était d'introduire l'information temporelle dans la

spécification des systèmes interactifs. Il a été alors nécessaire de concevoir de nouveaux opérateurs tels que [Johnson et al., 92] :

✧ : pour exprimer l'éventualité,

O : pour exprimer la notion de suivant,

□ : pour exprimer la notion de toujours,

U : pour exprimer la notion jusqu'à (Until).

Bien que ces extensions ont réussi à dépasser quelques limites de la logique du premier ordre, elles ne sont pas arrivées à résoudre complètement tous les problèmes de la spécification. On peut remarquer par exemple que la propriété d'éventualité n'est pas toujours appropriée. Elle peut risquer des erreurs graves et causer par la suite l'échec du système. D'autant plus que cette logique temporelle manque encore de moyens pour intégrer les facteurs humains et incorporer la notion du temps réel.

ACTL (Action Based Temporal Logic) [De Nicola, 91] est une extension de la logique temporelle permettant de décrire les propriétés spécifiques d'une application interactive [Duce et Paterno, 93].

### 3.2 Conclusions

Rappelons d'abord que notre travail de recherche se situe dans le domaine des procédés industriels complexes et critiques pour la sûreté. Rappelons, également, que notre objectif est de proposer une démarche méthodologique de conception mais aussi de génération semi-automatique des IHM à partir des spécifications obtenues. Ce type d'approche impose la définition d'un langage de spécification doté d'une sémantique non-ambiguë pouvant être interprétée par un outil automatique. Donc d'embler le choix d'une technique de spécification formelle s'impose et est totalement justifié.

Parmi les modèles et formalismes disponibles, notre choix, comme nous l'avions déjà souligné au chapitre 2, s'est porté sur les Réseaux de Petri. Le choix des RdP a été opéré suite à une étude bibliographique ayant mis en évidence l'aptitude de cette technique à répondre à des critères qui sont importants et applicables pour la spécification des IHM pour les procédés industriels complexes. En effet, les RdP disposent d'une définition formelle, ils offrent un grand pouvoir d'expression des aspects tels que le synchronisme, le parallélisme, ils sont exécutables, ils disposent de nombreuses techniques de vérification automatique des propriétés (borné, vivant, réinitialisable, etc.), il offrent une représentation graphique peu contraignante, etc. ; autant de critères adaptés aux contraintes de spécification et de génération automatique des interfaces graphiques pour les applications industriels complexes et critiques.

Les RdP ont, d'ailleurs, toujours été efficaces dans le domaine de l'automatique et ils deviennent de plus en plus utilisés dans celui de la conception d'interfaces homme-machine [Abed, 01]. Nous proposons, ci-dessous, certains critères de comparaison entre les techniques formelles étudiées et qui ont conduit à l'adoption des RdP pour nos travaux (plus de détails sont disponible dans [Riahi, 2004] et [Jambon et al., 2001]) :

- la puissance de l'outil en terme de vérifiabilité des propriétés : la vérification sur modèle permet d'assurer, de manière certaine, les propriétés. Cette méthode exige que les propriétés soient au préalable formellement définies. Les réseaux de Petri permettent de s'assurer, en explorant le graphe des marquages, que l'utilisateur a bien accès à toutes les commandes du système [Palanque et al., 95] ou que celui-ci peut achever une action entreprise.

- le degré de la prise en considération de l'aspect parallélisme : la concurrence dans la modélisation des tâches opérateur est un critère fort important. On peut modéliser la concurrence en utilisant le parallélisme vrai ou l'entrelacement. Les RdP comme la logique temporelle XTL ou la notation UAN sont des exemples qui remplissent cette propriété.
- le degré de la prise en considération de l'aspect synchronisme : implémenter la synchronisation des actions et des processus est indispensable dans la modélisation du dialogue homme-machine. Les RdP, Lotos et Lustre assurent ce genre de critère.
- Le séquencement des actions et les contraintes temporelles : ce critère regroupe cinq sous-critères [Balbo, 94] permettant de préciser les relations entre les actions : la séquence entre deux tâches (A puis B) ; l'alternative (ou choix) entre deux tâches (A ou B) ; la composition de deux tâches dans un ordre quelconque (A & B) ; l'itération (A un certain nombre de fois) ; l'expression des délais entre les tâches. Nous présentons dans la deuxième section de ce chapitre l'application naturelle de ces constructeurs au moyen d'une modélisation par RdP Synchronisés. Des principes de construction, que nous avons mis au point, sont également donnés. Ils permettent, s'ils sont bien appliqués, de sauvegarder la notion de vérifiabilité des propriétés du modèle RdP obtenu.
- Les capacités génératrices : selon les objectifs méthodologiques que l'on s'attache, une approche de conception peut aboutir sur des spécifications informelles inexploitable d'une manière automatisée, comme elle peut aboutir sur la génération automatique ou semi-automatique des IHM à partir des spécifications obtenues. Bien que souvent on se contente de l'utilisation d'une notation si elle est suffisamment expressive pour nos besoins, les capacités génératrices restent la spécificité des formalismes. Par exemple, les réseaux de Petri, dont le graphe des marquages fournit l'état courant du système, son évolution possible, ainsi que les étapes précédentes, permettent de générer de l'aide contextuelle en expliquant à l'utilisateur comment il est parvenu dans l'état courant, comment il peut continuer sa tâche et même lui donner le chemin optimal de réalisation comprenant le moins d'étapes possibles [Palanque, 97].

Nous proposons dans le chapitre suivant nos contributions au domaine de la spécification formelle des systèmes interactifs.

# CHAPITRE 4 : CONTRIBUTIONS AU DOMAINE DE LA SPECIFICATION FORMELLE DES SYSTEMES INTERACTIFS

*Nous proposons, dans ce chapitre, notre technique de modélisation, au moyen des RdP, appliquée aux étapes de modélisation de l'interaction homme-machine, de déduction des BIO depuis le modèle obtenu, de spécification du dialogue homme-machine et enfin de vérification des propriétés de l'interface (figure 4.1) [Moussa et al., 02] [Riahi et al. 00].*

## 4.1 Principe de construction du modèle global de l'interaction homme-machine

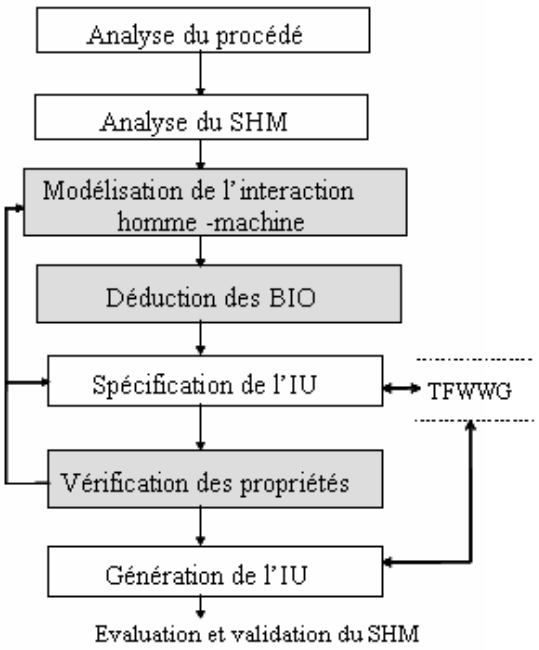



Figure 4.1: Rappel du squelette méthodologique de l'approche proposée

Les RdP proposés ici sont les RdPI (Réseaux de Petri interprétés) [Moalla, 85]. Ce type de réseaux introduit les notions d'événements et de conditions ainsi que la notion d'actions. En effet, une condition de passage (Cj), un événement de déclenchement (Evj) et une action éventuelle (Aj) sont associés à chaque transition Tj d'un RdPI (figure 4.2).

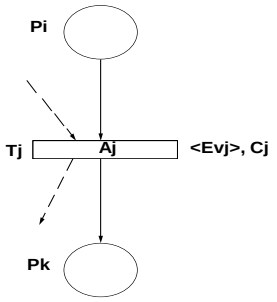



Figure 4.2 : Réseau de Petri Interprété (RdPI)

Pour modéliser l'interaction homme-machine moyennant les RdPI, nous avons convenu d'utiliser les places pour représenter « l'état du comportement de l'opérateur » vis-à-vis de l'évolution du système [Khelil, 01].

Nous considérons qu'une tâche opérateur est composée d'un ensemble organisé d'actions élémentaires. La structure modélisant une action élémentaire de l'opérateur (par exemple une action de régulation de température) est donnée ci-dessous (figure 4.3).

La validation de la condition i (transition T1) modélise le fait que l'opérateur va commencer l'exécution de l'action relative à cette condition. L'apparition, par la suite, de l'événement "fin

action" (transition T2) exprime le fait que l'action de l'opérateur a été exécutée et a pris fin. La place P2 exprime un état d'attente de la fin de l'exécution de l'action, alors que les places P1 et P3 modélisent l'état de l'opérateur avant et après l'exécution de son action. Par exemple, P1 modélise l'intention mentale de l'opérateur pour agir. La place P3 exprime son état à la fin de l'accomplissement de l'action.

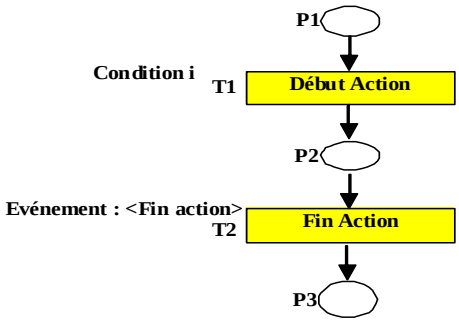



Figure 4.3 : Structure modélisant l'état d'un opérateur face à une action élémentaire

La construction du modèle global d'une tâche utilisateur s'effectue à partir des structures de base modélisant les différentes actions élémentaires de l'opérateur. Elle se base sur l'application des différentes opérations de composition explicitées ci-dessous et son principe réside dans le fait que la construction du modèle de la tâche n'utilise que des structures et des règles de composition définies. Cela est important pour assurer par avance de bonnes propriétés au modèle obtenu.

Toutes les actions de l'opérateur (élémentaires ou non) sont ordonnées selon des compositions typiques : séquentielle, parallèle, alternative, de choix, itérative ou de fermeture [Khelil, 01]. Nous nous limiterons ici à la présentation de trois compositions (séquentielle, parallèle et de fermeture). Nous présentons également la structure globale du modèle de l'interaction homme-machine ainsi construit.

Remarquons ici, que normalement, toute tâche doit être « réinitialisable » pour que l'on puisse la réexécuter de nouveau. La construction du modèle de la tâche doit donc se terminer obligatoirement par une composition de fermeture que nous proposons plus loin.

Par ailleurs, l'application des règles de composition peut conduire à des réseaux dans lesquels apparaissent des simplifications évidentes qui peuvent être effectuées pour réduire la taille du modèle tout en conservant ses bonnes propriétés. Ces simplifications concernent, en fait, des états instables. C'est le cas par exemple de la composition séquentielle de tâches élémentaires non soumises à des conditions explicites de déclenchement.

Notons, enfin, que chaque changement au niveau de l'état du système doit absolument affecter l'interface de contrôle afin d'informer l'opérateur de la situation et lui faciliter sa tâche de commande. Cela implique un changement au niveau des vues graphiques affectant les paramètres des objets graphiques (couleurs, formes, etc.) voire même un changement au niveau du contenu de la vue (apparition et/ou disparition de certains objets, etc.).

Cette technique de modélisation nous permet de déduire les BIO en fonction des changements de contexte de fonctionnement du système industriel.

### 4.1.1 Composition séquentielle

La composition séquentielle de n actions traduit le séquencement de leur exécution.

*Règle de composition :*

La composition séquentielle de n actions est assurée en fusionnant la place *fin* du réseau modélisant l'action i, avec la place *début* de l'action i+1 (figure 4.4).

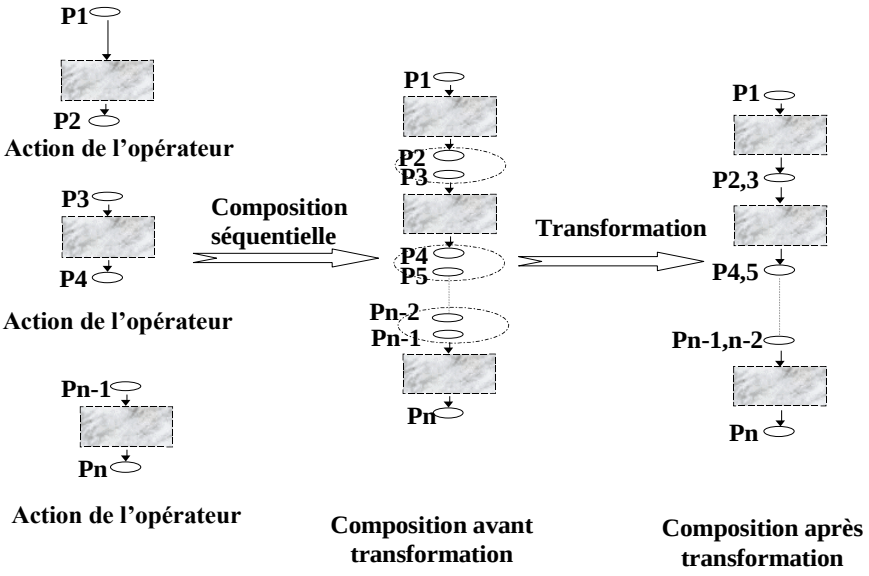



Figure 4.4 : Composition séquentielle

*Précaution à prendre :*

Dans le contexte considéré de l'exécution de la tâche de l'opérateur humain, le séquencement des actions décrites doit être possible.

Comme exemple de composition séquentielle d'actions de l'opérateur, nous pouvons imaginer une situation pour laquelle l'analyse de la tâche a révélé nécessaire que l'opérateur humain accomplisse trois actions l'une à la suite de l'autre : commencer par fermer un robinet, puis commander l'ouverture d'une première vanne de sécurité V1 et commander toute de suite après une deuxième vanne de sécurité V2. La modélisation d'une telle interaction sera modélisée par une composition séquentielle de trois structures élémentaires (modélisant les trois actions élémentaires) comme précisé dans la figure 4.4.

### 4.1.2 Composition parallèle

La composition parallèle exprime la possibilité d'exécution simultanée de plusieurs actions. Le parallélisme est assuré par le fait qu'une place de synchronisation d'entrée, active en même temps toutes les places d'initialisation des actions parallèles à exécuter. Cependant, le parallélisme effectif ne peut s'effectuer que si les actions à exécuter ne sollicitent pas les mêmes ressources. Sinon, un séquencement partiel ou total serait nécessaire.

*Règle de composition :*

La composition parallèle de n réseaux relatifs à n actions fait intervenir deux structures de composition PAR1 et PAR2.

- La structure PAR1 modélise une synchronisation de départ de n réseaux (figure 4.5).
- La structure PAR2 modélise une synchronisation à l'arrivée des n réseaux (figure 4.6).

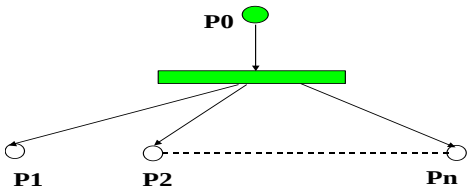



Figure 4.5 : Structure PAR1

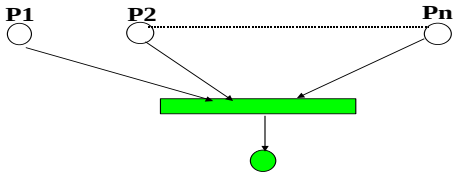



Figure 4.6 : Structure PAR2

Evidemment, le nombre de places Pn dans les deux structures PAR1 et PAR2 devrait être égal au nombre des actions parallèles Ai. Ainsi, pour assurer la composition parallèle des actions, il est nécessaire de synchroniser les places d'entrée et celles de sortie de ces actions (figure 4.7).

*Précaution à prendre :*

Dans le cas où une ou plusieurs ressources peuvent être sollicitées par l'exécution des branches (actions) parallèles, une analyse fine doit juger de l'opportunité de la parallélisation. A signaler que, d'une façon générale, la gestion des ressources partagées s'effectue d'une façon tout à fait indépendante des tâches.

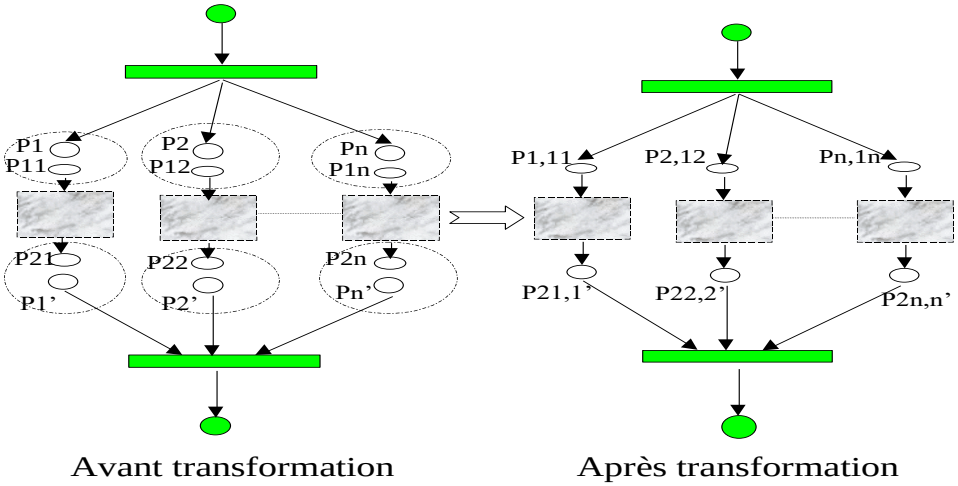



Figure 4.7 : Composition parallèle

Considérons, par exemple, une situation dans laquelle il est demandé à l'opérateur d'ajuster les températures de deux produits en parallèle. La modélisation de cette interaction pourrait être assurée par la composition en parallèle de deux branches relatives aux actions de régulation des températures.

### 4.1.3 Composition de fermeture

La composition de fermeture d'un réseau traduit la mise en boucle de ce réseau.

*Règle de composition :*

La composition de fermeture d'un réseau est réalisée en englobant le réseau dans une structure F de fermeture (figures 4.8 et 4.9).

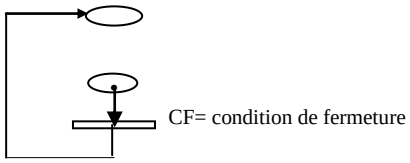



Figure 4.8 : Structure de fermeture

*Précautions à prendre :*

La condition de fermeture peut être utilisée comme déclencheur de la tâche globale. Tout contexte qui la valide doit, en principe, garantir l'évolution jusqu'au bout de l'exécution de la tâche.

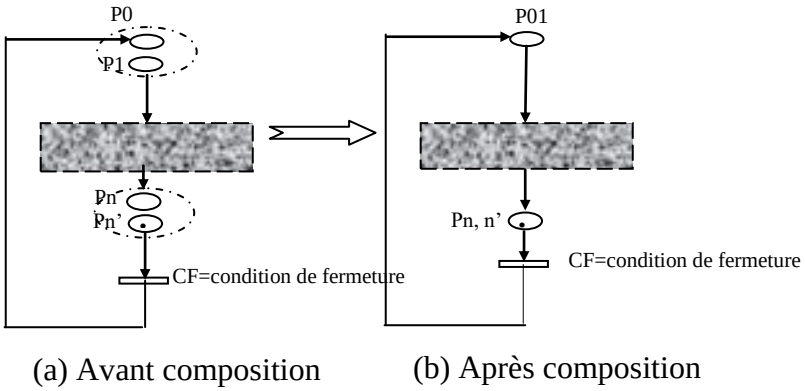



Figure 4.9 : Composition de fermeture

La composition de fermeture sera utilisée pour construire le modèle global de l'interaction homme-machine et d'exprimer la boucle de surveillance et de contrôle permettant de revenir à la situation initiale de surveillance d'un système donnée.

### 4.1.4 Illustration

Dans la figure 4.10, nous illustrons le processus de modélisation de l'interaction homme-machine moyennant les RdPI et se basant sur ce principe de composition. Nous supposons, dans cet exemple, que la tâche de l'opérateur revient à accomplir une première action A1, qui consiste pour ce cas à ouvrir une vanne de sécurité, en séquence avec deux autres actions en parallèle A2 et A3, permettant de régler le débit d'une solution mélange en agissant simultanément sur les débits d1 et d2 des deux produits en entrée.

Chacune de ces actions est modélisée par la structure élémentaire présentée préalablement (figure 4.3), et le modèle global est construit par une composition de ces trois structures élémentaires. Les places P2, P5 et P8 modélisent des états d'attente de l'exécution des actions de l'opérateur, respectivement A1, A2 et A3.

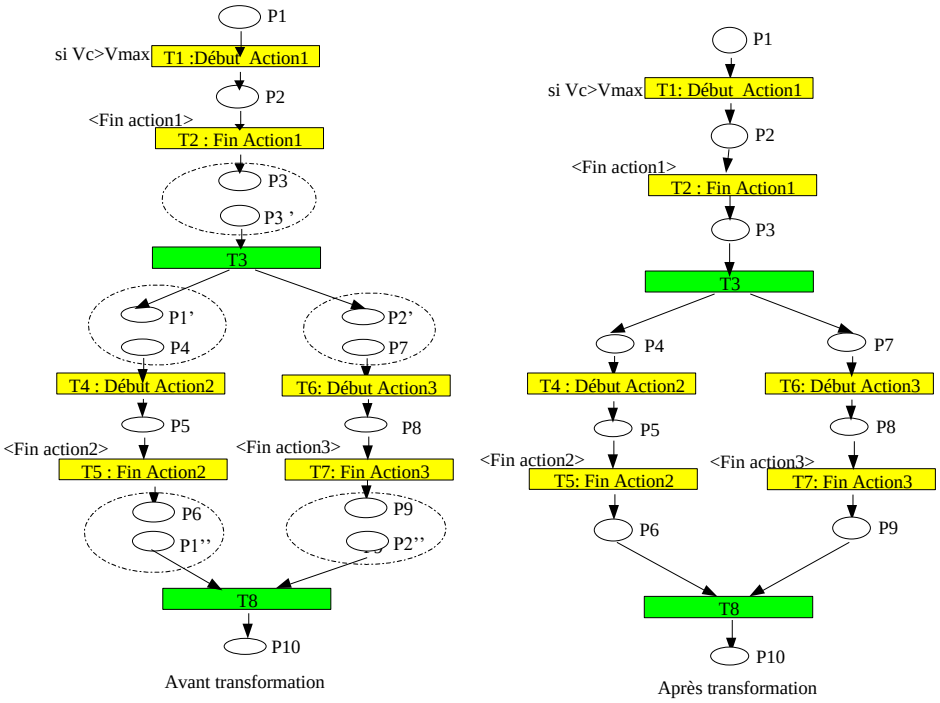



Figure 4.10 : Exemple d'un modèle d'interaction homme-machine

Le RdPI proposé pour la modélisation de l'interaction homme-machine est alors défini par l'ensemble : < P, T, E, OB, Pre, Post, μ, Precond, Action, Info-Transition > avec :

- P : ensemble des places = {P1, P2,..., Pn},
- T : ensemble des transitions = {T1, T2,..., Tm},
- E : ensemble des événements externes incluant l'événement "toujours présent" <e>,
- OB : ensemble des objets graphiques de l'interface,
- Pre : P x T → N définit les poids des arcs reliant les places pi de P aux transitions tk de T,
- Post : P x T → N définit les poids des arcs reliant les transitions tk de T aux places pi de P,
- μ : T → E associe à chaque transition l'événement de déclenchement approprié,
- Precond : T → Expression Booléenne définit la condition nécessaire de passage de chaque transition,
- Action : T → A définit l'éventuelle procédure d'action appropriée associée à chaque transition,
- Info-Transition : T → OB associe à chaque transition l'ensemble des objets graphiques appropriés. Cette association est expliquée ci-dessous.

Une fois le comportement de l'opérateur modélisé, les BIO peuvent être déduits.Nous proposons ici notre technique de déduction des BIO. Rappelons que ce passage délicat de la modélisation de l'interaction homme-machine à la déduction des BIO est une sorte de « talon d'Achille » de la

majorité des approches étudiées. Nous y attachons un intérêt particulier puisqu'on vise la mise en place d'une approche globale de conception intégrée et supportée par des outils informatiques.

L'opérateur a besoin d'identifier instantanément l'état de fonctionnement du procédé qu'il contrôle. Cette information lui est transmise à travers différents objets au niveau de l'interface (messages, valeurs, graphiques, alarmes, etc.). Ces objets sont, en fait, reliés à des variables d'état du système. Il suffit donc d'identifier l'ensemble des variables d'information appropriées à chaque état du système. Ces variables d'information découlent naturellement de l'analyse du SHM menée auparavant.

De plus, et dans le but d'accomplir ses tâches, l'opérateur aura à intervenir et commander quelques variables lors de l'apparition des dysfonctionnements. L'interface doit donc présenter dans de telles situations, l'ensemble des objets de commande qui permettent à l'opérateur d'exécuter ses actions et commander le processus.

L'ensemble de ces variables d'information et de commande constitue les BIO. Nous expliquons ici la manière de déduire ces BIO à travers les précédentes analyses et le modèle de l'interaction homme-machine établi au niveau de l'étape précédente de la démarche.

Nous associons aux transitions "*début action*" les variables d'information et de commande reflétant l'état du système et respectant les besoins des opérateurs en matière d'information (ses BIO) (figure 4.11).

Par exemple, si nous considérons l'action A1 du modèle présenté dans la figure 4.11, cette action consiste à vider un silo de stockage d'un produit C. Au niveau de l'état modélisé par la place P2, l'opérateur doit disposer des BIO nécessaires pour bien accomplir sa tâche. La variable d'information VC informe l'opérateur du niveau courant du produit C dans le silo de stockage et la variable de commande CVC lui permet d'agir sur ce niveau à travers la vanne de sécurité. Ces deux variables constituent donc les BIO relatifs à cet état.

Pour les actions A2 et A3 qui modélisent des actions de l'opérateur sur les débits des produits en entrée A et B, l'opérateur aura besoin de connaître pour chacun de ces produits le volume courant (VC) du mélange ainsi que le débit courant (DA respectivement DB). Pour agir, il a besoin d'une variable de commande (CDA respectivement CDB). Cet ensemble de variables constitue les BIO relatifs à cette situation.

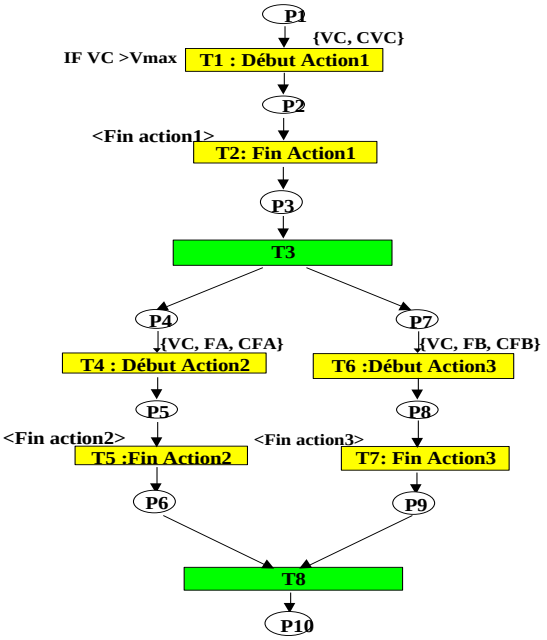



Figure 4.11 : Exemple d'un modèle de l'interaction homme-machine avec les BIO

Une fois les BIO identifiés (c'est-à-dire l'ensemble des variables d'information et de commande formé), l'étape suivante consiste à spécifier les objets graphiques relatifs à ces BIO :

- à chaque variable d'information, on associe un objet graphique d'information,
- à chaque variable de commande, on associe un objet graphique de commande.

Le principe d'association des objets graphiques et de leur gestion dynamique au niveau des différentes vues graphiques est expliqué dans la partie qui suit.

### 4.2 Spécification du dialogue homme-machine

Afin de modéliser le dialogue homme-machine, il est maintenant nécessaire de définir le comportement des différents objets graphiques de l'interface. Ce comportement dépend des caractéristiques graphiques des objets, des services qu'ils offrent, du domaine de l'application, de l'évolution du système, etc.

A chaque objet graphique, nous associons une structure de contrôle modélisée par un RdPI. Considérant les deux cas particuliers des objets graphiques identifiés (objet informationnel et objet de commande), nous proposons, à titre d'illustration, deux structures génériques de contrôle.

La première structure concerne les objets informationnels (figure 4.12.a). Ces objets passent principalement par deux états : affiché et masqué. La structure de contrôle d'un tel objet doit donc posséder deux transitions particulières dont les actions associées assurent l'affichage et le masquage de l'objet. Le second type de structure concerne les objets de commande (figure 4.12.b). Un objet de commande passe principalement par trois états : masqué, affiché en état désactivé (affiché/désactivé) et affiché en état activé (affiché/activé).

Le dialogue global de l'interface est ainsi décrit par l'ensemble des RdPI proposés ici, autrement dit, celui modélisant l'interaction homme-machine et ceux correspondants aux structures de contrôle des différents objets graphiques.

Se pose, alors, la question de la communication entre ces différents réseaux de Petri. La communication entre ces différents RdPI est assurée moyennant des variables logiques. Les valeurs de ces variables sont positionnées au niveau des transitions particulières "*début action*" et "*fin action*".

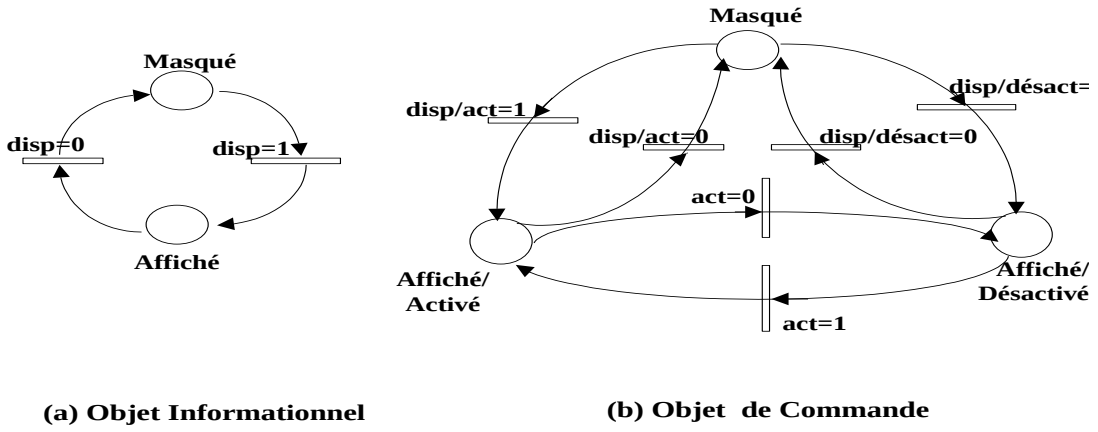



Figure 4.12 : Structures de contrôle des objets de l'interface

Le processus de communication permettant d'assurer la gestion dynamique du dialogue homme-machine est proposé à la figure 4.13. Nous retrouvons la variable d'information VC et la variable de commande CVC. Elle montre une partie du RdPI modélisant l'interaction homme-machine pour ce cas, ainsi que les structures de contrôle de deux objets graphiques :

- Objet informationnel : associé à la variable d'information VC (pouvant être spécifié en courbe, étoile barre graphe, etc.),
- Objet de commande : associé à la variable de commande CVC (pouvant être spécifier au moyen d'un bouton de commande, d'une vanne, etc.).

Considérons, par exemple, la transition T1 de ce modèle. Supposons que cette transition modélise le début d'une action élémentaire qui consiste à vider le silo de stockage d'un produit C suite à la détection d'un dysfonctionnement D1. L'action associée à cette transition consiste donc à afficher l'objet informationnel VC et activer l'objet de commande CVC. Cette action est modélisée au niveau de la transition par : A = {VC.disp:=1, CVC.disp/act:=1}.

En effet, quatre types de variables logiques sont utilisés pour assurer la communication entre ces différents réseaux de Petri :

- ***disp*** (pour disponible) : c'est une variable logique utilisée pour exprimer le besoin d'afficher ou masquer un objet graphique informationnel sur l'écran. Si disp est positionnée à 1, cela veut dire qu'on demande d'afficher l'objet informationnel. Si par contre, disp est positionnée à la valeur 0, c'est qu'on demande de masquer l'objet informationnel en question.
- ***disp/act*** (pour disponible et activé) : c'est une variable logique permettant d'exprimer les transitions d'un objet de commande entre les états « masqué » et « affiché en état activé ». Si elle est positionnée à 0, c'est qu'on demande de le masquer. Si on la positionne à 1, c'est qu'on demande de rendre disponible cet objet de commande, c'est-à-dire de l'afficher en état activé.
- ***disp /désact*** (pour disponible et désactivé) : c'est une variable logique, positionnée à 1, elle exprime le fait qu'on demande d'afficher un objet de commande en état désactivé. Si elle est positionnée à 0, c'est qu'on souhaite masquer l'objet.
- ***act*** (pour activé) : c'est une variable logique permettant d'exprimer les transitions d'un objet graphique de commande entre les états « affiché/ désactivé » et « affiché/activé ». Positionnée à 1, elle exprime le fait d'activer l'objet, et positionnée à 0, elle traduit une demande de désactivation.

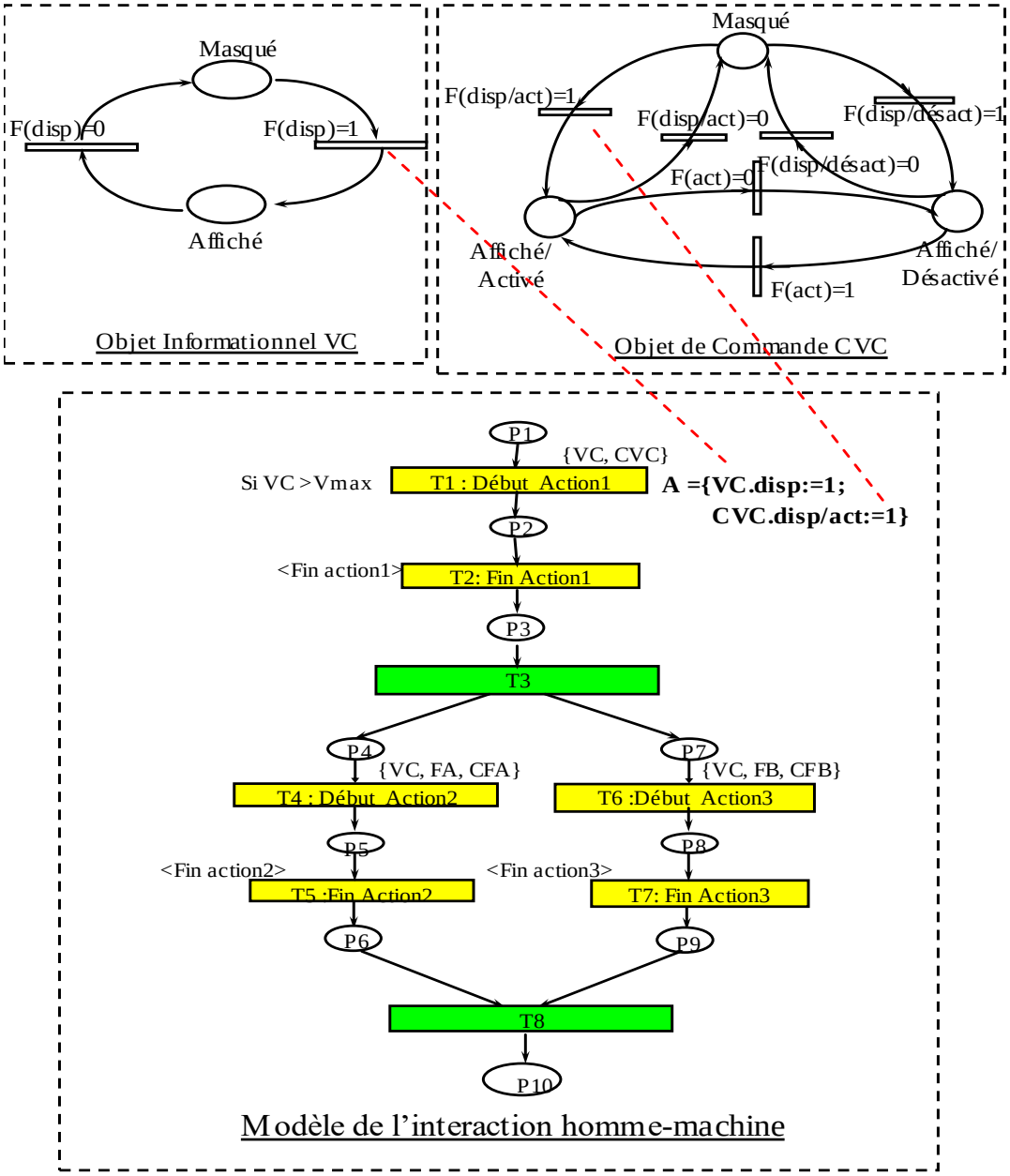



Figure 4.13 : Processus de communication entre les différents RdP

Ainsi, pour le cas de la transition T1, on demande d'afficher l'objet informationnel VC (VC.disp :=1) et d'afficher en état activé l'objet de commande CVC (CVC.disp/act :=1).

Les variables logiques sont évaluées au niveau des conditions associées aux transitions des structures de contrôle des différents objets graphiques en termes de Formule logique (F) (ex : F(disp=0), F(disp/act)=1, etc., (figure 4.13)). Ces formules logiques manipulent donc des variables logiques (attributs propres aux différents objets graphiques) positionnées au niveau du modèle de l'interaction homme-machine en fonction de l'évolution de l'état du système contrôlé.

D'une manière concrète, moyennant ce mécanisme et selon le contexte de fonctionnement les objets seront affichés, masqués, affichés en état actif ou affichés en état inactif (grisé par exemple).

### 4.3 Vérification des propriétés de l'Interface

Comme nous l'avons déjà mentionné (§ 3.2), la vérification sur modèle permet d'assurer, de manière certaine, un ensemble de propriétés. Cette méthode exige, toutefois, que les propriétés soient au préalable formellement définies. La validation d'un modèle basé sur les RdP dépend beaucoup de ses caractéristiques. En effet, les extensions apportées en général aux RdP causent des difficultés pour la vérification de plusieurs propriétés. Les RdPI, utilisés ici, sont non autonomes. En dépit de leur perte partielle de quelques résultats théoriques (pour la vérification des propriétés), ils permettent d'exprimer les concepts des systèmes temps réel et de modéliser de manière concise des applications complexes [Brams, 83]. Afin de pouvoir valider un tel modèle, il est donc nécessaire d'associer certaines contraintes et axiomes pour sa construction.

Comme nous l'avons expliqué auparavant (voir § 4.1), nous considérons, principalement, que les événements relatifs à des actions de l'opérateur sont toujours présents et qu'une action de l'opérateur doit toujours arriver à sa fin (grâce à l'intervention de l'opérateur humain ou à l'écoulement d'un certain temps d'attente fixé par un chien de garde associé à ces transitions). Ainsi, nous pouvons garantir que les événements externes ne causent jamais d'interblocage (deadlock) du modèle RdP et nous pouvons donc transiter vers un RdP ordinaire en supprimant tous les événements externes associés au modèle (car ils sont tous relatifs à des actions de l'opérateur humain qui vont certainement se produire). De ce fait, la vérification des propriétés devient possible. En outre, comme la construction du modèle global a été basée sur une technique de fusion des places de structures élémentaires, la composition va donc conserver les propriétés vérifiées sur les structures élémentaires.

En d'autres termes, nous pouvons dire que, en composant des structures élémentaires validées, le modèle global est lui même validé. La structure élémentaire sur laquelle est basée le modèle de l'interaction homme-machine vérifie un certain nombre de 'bonnes' propriétés telles que la bornitude, le non blocage, la vivacité et la persistance. Ces propriétés sont donc garanties, par le processus de construction suivi, pour le modèle global de l'interaction homme-machine [Khelil, 01]. Cette vérification de propriétés structurelles des RdP permet, de facto, de garantir certaines bonnes propriétés de l'interface spécifiée.

Nous commençons par exposer, ci-après, quelles sont ces bonnes propriétés à vérifier pour assurer le bon fonctionnement d'une interface donnée. Nous expliquons, par la suite, le processus suivi pour la vérification de ces propriétés sur le modèle préconisé de l'interaction homme-machine.

#### 4.3.1 Les propriétés à vérifier

Le bon fonctionnement d'une interface homme-machine donnée peut être assuré principalement par les trois propriétés suivantes :

- *L'absence de blocage* : l'interface proposée ne doit jamais se bloquer sous l'effet de n'importe quelle action de l'utilisateur.
- *La disponibilité de services* : à un moment donné, l'interface doit garantir la disponibilité de certains services relatifs à la situation en cours.
- *L'absence de conflit et la stabilité* : l'interface ne doit jamais présenter de situation de conflit effectif et elle doit assurer une certaine stabilité des différentes vues graphiques qu'elle présente. Il faudrait donc assurer l'absence de tout état d'évolution imprévisible de l'interface et il ne faut pas qu'il y ait un conflit de vues dans une situation donnée de fonctionnement.

Ces propriétés de bon fonctionnement sont garanties par des propriétés structurelles des RdP. En effet :

- La propriété de *bornitude* garantit un nombre fini d'états possibles du système ce qui se traduit par une stabilité des vues graphiques au niveau de l'interface pour ces différents états.
- La propriété de *vivacité* interprète la potentialité du système à atteindre tous les services possibles modélisés et garantit l'absence de blocage partiel ou total ; ce qui se traduit par l'absence de blocage de l'interface réalisée et l'accessibilité aux différents services qu'elle offre.
- la propriété de *persistance* garantit l'absence de conflit au niveau du réseau et par la suite la stabilité de l'interface et l'absence de confusion ou ambiguïté avec les différents états du système.

Nous proposons ci-dessous le principe de vérification de ces propriétés structurelles sur le modèle de l'interaction homme-machine adopté.

### 4.3.2 Principe de vérification

Le processus de construction par composition, étudié dans cette approche, a pour but d'établir les règles qui permettent de systématiser cette construction mais surtout de garantir, par avance, la vérification des propriétés caractéristiques de la bornitude, de la vivacité et de la persistance. Nous montrons ci-dessous de quelle manière la vérification des dites propriétés est effectivement garantie par ce processus de construction.

La propriété de persistance est vérifiée car les structures élémentaires et les compositions utilisées pour la construction du modèle de la tâche utilisateur n'admettent aucun conflit structurel qui peut conduire à un conflit effectif (voir § 4.1).

Rappelons que le processus de construction par de structures élémentaires modélisant les actions opérateurs (SEAO) et construit, par compositions successives, des réseaux que nous allons intentionnellement désigner par « réseaux correctement construits (RCC) ». La dernière composition de fermeture transforme le dernier RCC obtenu en un modèle de description de tâche utilisateur.

Un RCC peut être défini comme suit :

RCC = SEAO \ RCC obtenu par application d'une composition unique à un ensemble de RCC.

Un RCC a toujours une seule place d'entrée *début* et une seule place de sortie *fin*. Il a les propriétés suivantes :

Pr1 : Toute marque déposée dans la place *début* du RCC finit, tôt ou tard, par évoluer vers la place de sortie *fin*.

Pr2 : Dans son évolution de la place *début* à la place *fin*, et pour toute transition choisie dans le RCC on peut toujours trouver une configuration de valeurs des variables (internes et/ou opérateur) telle que l'évolution de la marque conduit à franchir cette transition.

Pr3 : Une fois arrivée à la place *fin*, toutes les places du RCC depuis la place *début* à part la place *fin*, auront un marquage vide.

En conséquence, si tout RCC vérifie ces trois propriétés, la composition finale de fermeture de tout RCC conduit à un modèle de tâche initialisé et ayant la propriété de bornitude et de vivacité.

La vérification de ces trois propriétés est montrée ci-après pour la structure élémentaire SEAO et pour une composition séquentielle de deux ou de plusieurs structures élémentaires.

- **Vérification des propriétés pour une structure élémentaire :**

Rappelons qu'une structure élémentaire SEAO se compose d'une place *début* P1, d'une place *action* P2 et d'une place *fin* P3 (figure 4.3). Le déclenchement de la transition T1 est conditionné par la condition (condition i). Le franchissement de la transition T2 se fera systématiquement sur occurrence de l'événement de fin d'exécution de l'action associée à la place P2. Moyennant la précaution d'interprétation du § 4.1.2, toute marque placée dans P1 évoluera tôt ou tard vers P2 par la vérification de la condition (condition i) et passera à la place P3 dès la fin de cette action.

***Proposition 1 :*** si un réseau R vérifie les propriétés Pr1, Pr2 et Pr3 alors la réduction de R par suppression d'une SEAO non initialisée (figure 4.14) préserve la réciprocité concernant la vérification des propriétés Pr1, Pr2 et Pr3.

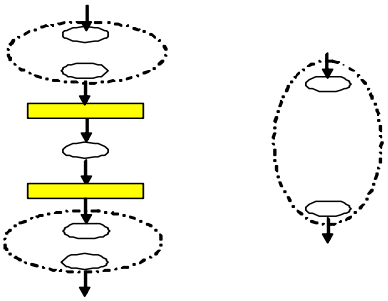


Figure 4.14 : Suppression d'une SEAO

***Preuve :*** de par la structure d'une SEAO, du fait qu'elle vérifie les propriétés Pr1, pr2 et Pr3 et tenant compte des précautions d'interprétations, la proposition se justifie directement par les approches classiques d'analyse par réduction.

- **Vérification des propriétés pour une composition séquentielle :**

D'après la proposition 1, la vérification des propriétés Pr1, Pr2 et Pr3 pour une composition séquentielle de deux ou plusieurs SEAO est triviale. En effet, une telle composition est réalisée en fusionnant la place *début* d'une SEAO et la place *fin* d'une autre SEAO à composer en séquentiel. En appliquant des réductions successives de SEAO, on aboutit à une seule SEAO qui vérifie, par hypothèse les trois propriétés.

Pour les autres compositions, la vérification des propriétés est disponible dans [Khelil, 01].

### 4.4 Conclusions

Nous avons proposé dans ce chapitre notre contribution au domaine de la spécification formelle des systèmes interactifs.
Nous avons d'abord présenté le principe de construction du modèle global d'interaction homme-machine. L'approche que nous proposons bénéficie, bien sûr, de la représentation graphique attrayante des RdP, qui accroît la lisibilité et facilite la compréhension des modèles. Nous avons tenté d'apporter un plus méthodologique à travers l'emploi de règles de compositions (parallèle, séquentielle, etc.) pour pallier à la carence principale des RdP à savoir leurs manques de structuration qui conduit à des modèles dont la taille croît rapidement avec la complexité du système, et qui deviennent rapidement difficile à comprendre et à gérer. Le grand pouvoir d'expression dont dispose les RdP nous autorise à décrire des systèmes aux comportements complexes, réactifs ou concurrents. Ceci nous réconforte dans notre choix, puisque, rappelons le, nous nous situant dans le domaine des IHM pour les procédés industriels complexes et critiques. Les RdP produisent des modèles qui possèdent une sémantique sans ambiguïté. La déduction des BIO se fait, alors, en toute assurance et sans souci de retombées négatives dues à la technique de

spécification employée. Nous avons, enfin, abordé la question cruciale de vérification des spécifications obtenues. Les RdP se distinguent à ce niveau puisqu'il est possible de rechercher des propriétés génériques telles que le caractère borné, vivant ou réinitialisable, ou des propriétés spécifiques telles que l'existence d'invariants.

Nous avons ainsi terminé la partie « formelle » en rapport avec la modélisation de l'interaction homme-machine, la spécification du dialogue homme-machine et la vérification des modèles obtenus. Nous présentons dans le chapitre suivant les étapes de la spécification et de la génération semi-automatique de l'interface.

# CHAPITRE 5 : CONTRIBUTIONS A LA GENERATION SEMI-AUTOMATIQUE DES IHM

*Nous allons relater ici notre expérience en matière de spécification et génération semi-automatique des IHM. Il s'agit de spécifier les deux principales composantes de l'interface (en plus de la composante relative à l'application) conformément au modèle de Seeheim [Pfaff, 85] ou au modèle Arch [Bass et al., 91] ; à savoir : la composante présentation et la composante dialogue. Le principe de la définition de chaque composante est également présenté (figure 5.1).*

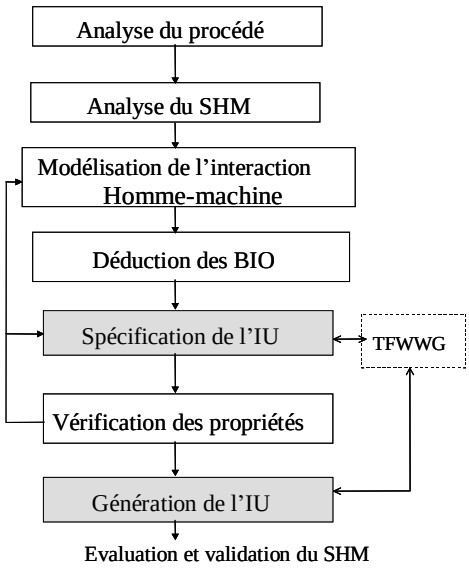



Figure 5.1 : Rappel du squelette méthodologique de l'approche proposée

## 5.1 Spécification de la présentation assistée par un système à base de connaissances

La principale problématique au niveau de la spécification de la présentation est la prise en compte des critères et recommandations ergonomiques disponibles à ce sujet (voir chapitre 1 §I.3).

En effet, de nombreuses règles ergonomiques sont proposées dans la littérature pour la spécification des interfaces. Quelques tentatives d'inventaire des règles ergonomiques pour la conception et l'évaluation des applications interactives de contrôle de procédés industriels sont à signaler :

- *Gilmore et al.* [Gilmore et al., 89] ont beaucoup contribué à ce sujet et leurs travaux ont abouti aux premières versions du standard NUREG (dédié aux applications nucléaires). Ils ont proposé un guide ergonomique pour la conception des interfaces de contrôle des procédés industriels dans lequel une centaine de recommandations ergonomiques, adaptées à ce domaine particulier sont listées. Les règles présentées dans ce manuel sont organisées selon principalement quatre catégories : (1) les vues graphiques, (2) les périphériques de contrôle et d'entrée/sortie, (3) l'intégration vues/contrôle, (4) la disposition de l'espace de travail et des facteurs de l'environnement. Chaque recommandation est présentée selon un format structuré.
- *O'Hara et al.* [O'Hara et al., 96][O'Hara et al., 99] ont contribué aux dernières versions du standard NUREG et ont proposé la première version d'un système *hypermédia* décrivant les recommandations ergonomiques applicables au domaine de contrôle des procédés. Bien que leurs premiers résultats soient prometteurs, les auteurs soulignent quelques problèmes méthodologiques, conceptuels et pratiques liés aux spécificités de ce domaine d'application particulier.

Soulignons aussi que les recommandations ergonomiques ne sont pas bien connues ni utilisées dans la pratique et ce en dépit d'un besoin réel en industrie. En effet, les recommandations sont bien plus fréquentes pour le bas niveau d'abstraction touchant les objets de l'interface, mais beaucoup moins disponibles pour le haut niveau d'abstraction de l'interface [Moussa, 92] [Moussa et al., 00a]. Ainsi, le fait de composer une interface à partir d'objets graphiques élémentaires intégrant des critères ergonomiques (bouton, vanne, cadran, etc.), ne garantit pas pour autant une interface ergonomique.

Dans notre démarche, pour la présentation des interfaces, nous proposons de bénéficier des travaux menés à ce sujet depuis les années 80 par un certain nombre de chercheurs [Alty et al., 85][Johannsen, 95][Moussa et al., 90][Kolski et al., 91][Kolski et al., 96] concernant des approches à bases de connaissances pour la conception et/ou l'évaluation automatiques des interfaces dédiées au contrôle des procédés industriels. En particulier, la première maquette de l'outil Ergo-Conceptor, proposé dans nos recherches antérieures, vise à décider « globalement » de la présentation appropriée à donner aux vues graphiques associées à chaque sous-système à contrôler. Il considère, à cet effet, d'une part les caractéristiques de chaque sous-système à contrôler (l'ensemble de ses différents états de fonctionnement, les BIO associés à chaque état, etc.) et d'autre part, un premier niveau de recommandations ergonomiques spécifiques formalisées et stockées dans sa base de connaissances.

Les étapes précédentes de notre approche, comme nous l'avons déjà expliqué, aboutissent à l'identification de plusieurs sous-systèmes dont chacun dispose des variables informationnelles et/ou de commande jugées comme adéquates (figure 5.2) [Moussa et al. 99][Moussa et al., 00].

La question qui se pose ici est comment décider :

- du nombre de vues graphiques à associer à chaque sous-système : en fonction du volume des BIO,
- des types de vues graphiques : selon le niveau d'abstraction du sous-système et son rôle dans la description du SHM (ex : informationnelle, ou de commande), et
- des types de représentation : selon le type de la vue associée (ex : supervision ou historique pour une vue informationnelle).

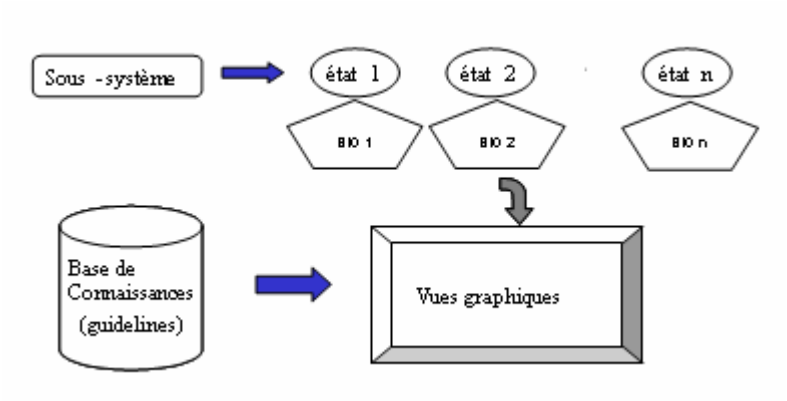



Figure 5.2 : Processus de spécification de la présentation de l'interface

Afin d'assurer la meilleure qualité ergonomique possible, quelques recommandations ergonomiques sont formalisées sous forme clausale et stockées dans la base de connaissances du système Ergo-Conceptor+ l'outil informatique supportant la méthodologie que nous proposons (figures 5.3 et 5.4). Ces règles constituent des exemples qui montrent comment on peut décider (1) des types des vues et des représentations associées selon le niveau d'abstraction du sous-système en question, et (2) du choix des objets graphiques à associer aux variables du système.

```
SI niveau_abstraction (sous-sys) =  "bas"
        ALORS générer_vue_commande (sous-système)
        ET générer_vue_informationnelle (sous-système)
SI niveau_abstraction (sous-sys) =  "haut"
        ALORS générer_vue_informationnelle (sous-système)
SI quantite_BIO (sous-sys) >= MAX
        ALORS  générer_multiple_vues (sous-système)
```

Figure 5.3 : Exemples de recommandations ergonomiques pour la génération des vues graphiques

```
SI type_variable (V) = "informationnelle"

        ET type_vue (sous-système) = "informationnel"

        ET type_representation (sous-système) = "historique"

        ALORS  select_courbe (V)

SI type  variable (V) = "informationnelle"
```

Figure 5.4 : Exemples de recommandations ergonomiques pour la sélection des objets de l'interface

Ainsi, pour un niveau d'abstraction bas du sous-système à contrôler (figure 5.3), on préconise la génération de deux types de vues : une vue informationnelle de supervision et une vue de commande. Si par contre, le niveau d'abstraction est élevé, une seule vue peut être générée ; c'est la vue informationnelle de supervision. En effet, à un haut niveau d'abstraction on ne peut pas autoriser une commande du système. Il faudrait passer à un niveau plus bas mettant en évidence les différentes variables de commande du sous-système en question (voir à ce sujet [Moussa, 92] [Moussa et al, 96]).

Par ailleurs, si pour un sous-système et pour une situation de fonctionnement donné, on se retrouve avec une quantité de BIO qui dépasse un nombre maximal recommandé (figure 5.4), il faudrait penser à générer plusieurs vues. Chaque vue doit proposer un niveau informationnel raisonnable et gérable par l'opérateur [Card et al., 83].

Aussi, pour le choix de la représentation graphique d'un objet donnée, certaines règles ergonomiques sont proposées. Par exemple, pour le cas de la représentation graphique d'une variable d'information dans une vue informationnelle avec un type de représentation « historique », on recommande l'utilisation d'une courbe. Si par contre, le type de représentation associé à cette vue est « supervision », on conseille d'utiliser les barregraphes. Pour le cas d'une vue de commande, s'il s'agit d'une variable de commande, des modes de représentation « bouton de commande » sont prévus.

Une fois les modes graphiques de représentation de l'interface décidés, la deuxième étape de la spécification de l'interface consiste à préciser et définir le dialogue approprié. Le dialogue de chaque objet graphique est modélisé par les RdPI. Ceci est expliqué ci-dessous.

## 5.2 Génération de l'interface

L'approche adoptée pour la spécification de la présentation de l'interface est celle des systèmes à base de connaissances permettant de décider des vues et objets graphiques adéquats en fonction des BIO identifiés et des recommandations ergonomiques proposées dans le domaine de

contrôle des procédés industriels complexes. Cette approche de spécification et génération graphique est supportée dans nos travaux par l'outil informatique Ergo-Conceptor+. Cet outil intègre à cet effet :

(i) un système à base de connaissances pour la spécification et génération des vues et objets graphiques, et

(ii) un joueur de RdPI permettant de gérer l'évolution du dialogue homme-machine et la dynamique de l'affichage des objets graphiques.

Ces deux principaux modules de l'outil Ergo-Conceptor+ sont succinctement présentés ici. Ils sont détaillés dans [Riahi, 04].

### 5.2.1 La génération graphique

Le système à base de connaissances intégré dans l'outil Ergo-Conceptor+ assure la prise en considération de recommandations ergonomiques dans le processus de spécification et de génération semi-automatique des vues graphiques.

En effet, ce système doit regrouper un ensemble de recommandations ergonomiques du domaine rassemblées, organisées et formalisées sous forme de règles de production (figure 5.5). Ces règles sont exploitées par la suite, moyennant un moteur d'inférence capable de décider des vues et objets graphiques appropriés à un instant donné, en fonction des BIO fournis et d'une bibliothèque d'objets graphiques prédéfinis.

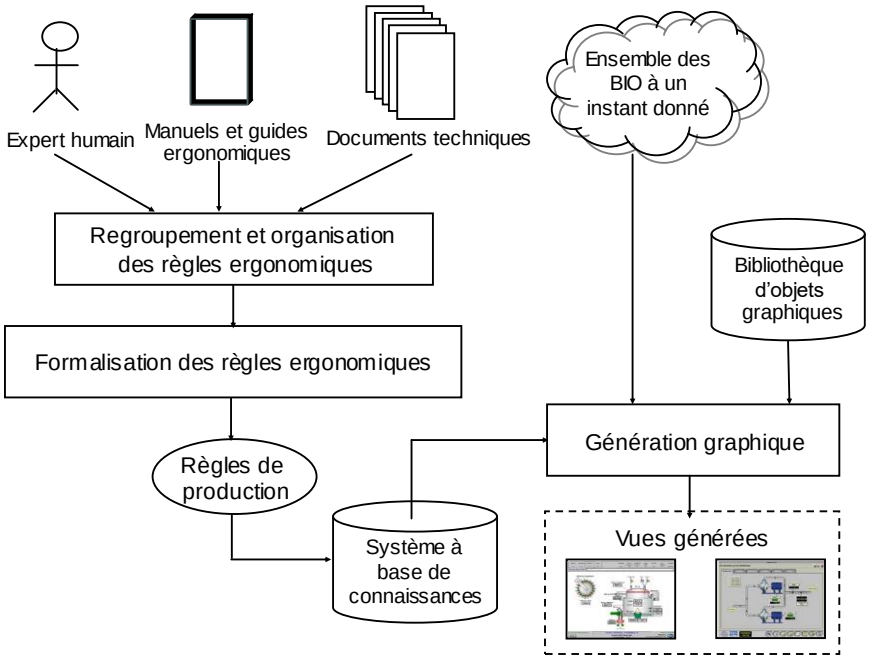



Figure 5.5 : Système à base de connaissances de l'outil Ergo-Conceptor+

### 5.2.2 La gestion de la dynamique des IHM

Une fois les vues générées, la gestion de leur dynamique est assurée par le module joueur des RdPI (appelé aussi interpréteur).

L'outil informatique supportant la méthodologie globale de conception et de génération des IHM, Ergo-Conceptor+ [Moussa et al., 02], prend en charge la gestion de la dynamique des vues (voir figure 5.5). Son principe consiste, en effet, à interpréter l'évolution des marquages des différents RdPI du système modélisant le dialogue homme-machine, en fonction de l'évolution de l'état du SHM et des différents événements qui se produisent sur le procédé.

Le joueur des RdP est basé principalement sur sept composants (figure 5.6) :

1- la base de données du procédé contenant les différentes variables du système,

2- la base de données des objets de l'Interface contenant l'ensemble des instances créées,

3- les matrices "Pre" des différents réseaux de Petri, indiquant les conditions de franchissement des transitions,

4- les matrices "Post" des différents réseaux de Petri, exprimant ce qu'on doit faire lors du franchissement des transitions,

5- l'ensemble des marquages courants des différents réseaux de Petri,

6- le code des différents programmes des traitements relatifs aux actions associées aux transitions des réseaux, et

7- une liste des événements produits pour simuler l'évolution de l'état du système et par suite l'évolution des différents réseaux de Petri.

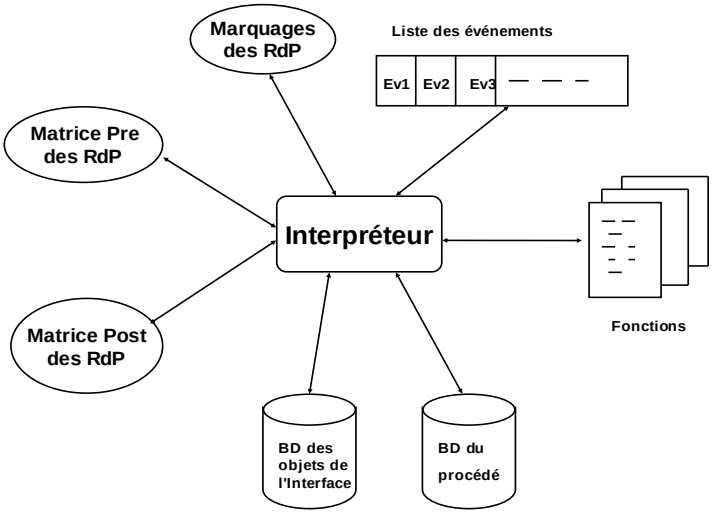



Figure 5.6 : Joueur (interpréteur) des RdPI

Afin de refléter les spécificités des RdPI employés dans notre approche de modélisation de l'interaction homme-machine, nous présentons succinctement les matrices « Pre » et « Post », permettant de définir les règles de franchissement des RdP, par trois vecteurs (figure 5.7) :

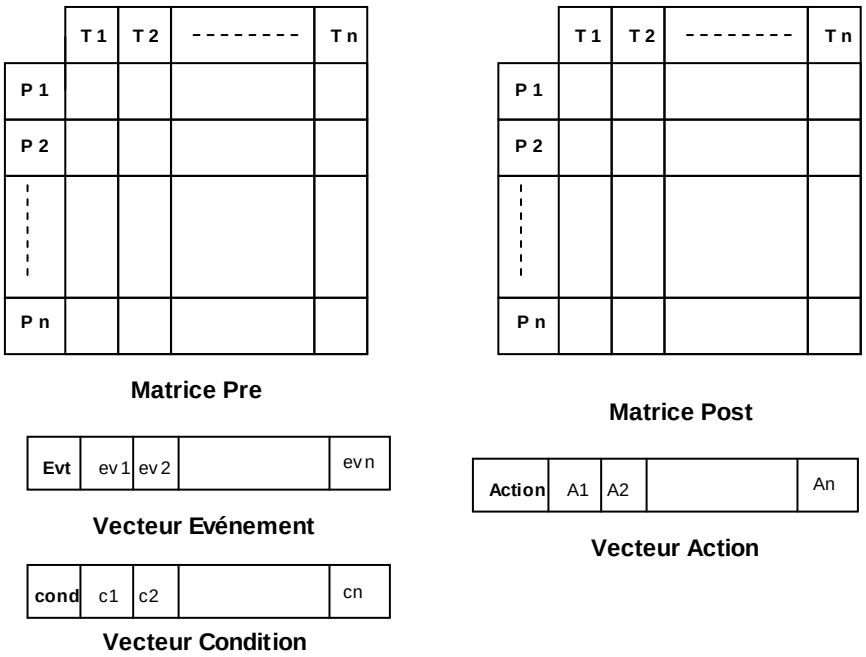



Figure 5.7 : Structures des matrices « Pre » et « Post »

- un vecteur *Evénement* associant pour chaque transition l'événement nécessaire pour la rendre franchissable,
- un vecteur *Condition* précisant les conditions nécessaires au franchissement de chaque transition, et
- un vecteur *Action* associant les actions respectives à accomplir au franchissement d'une transition donnée.

### 5.2.3 Intégration et évaluation de l'IHM générée

Une fois l'interface générée, l'étape qui suit ( en se référant à un cycle de vie classique des systèmes ), consiste à l'intégrer dans le système global de contrôle du procédé en question et procéder à son évaluation.

L'évaluation d'un système homme-machine (SHM) concerne généralement deux types de critères [Grislin, 95] [Grislin et al., 97][Kolski, 97][Nielsen, 93][Bastien et Scapin, 01] :

- les critères de performances du SHM, calculés en termes d'écart entre les objectifs de l'entreprise et les résultats obtenus, et
- les critères ergonomiques, liés à l'utilisation du système interactif vis-à-vis de la réalisation de ses tâches dans différents contextes de fonctionnement (normaux et anormaux).

Au niveau de cette étape, c'est d'évaluation *a posteriori* dont il s'agit ici. L'évaluation *a priori* est, en effet, assurée avec (1) l'intégration d'un outil du type TFWWG permettant la prise en considération des critères ergonomiques de présentation dès les premières étapes de la conception de l'interface, et (2) l'utilisation d'une technique formelle pour la modélisation de l'interaction homme-machine garantissant les bonnes propriétés de l'interface, qui sont d'ailleurs vérifiées avant la génération proprement dite de l'interface.

L'évaluation *a posteriori* se fait, dans une première étape, dans un contexte simulé de fonctionnement pour ne pas perturber le fonctionnement réel du procédé et par la suite sa production. Puis, une fois cette étape validée, on passe à une évaluation sur site réel. Différentes

techniques et méthodes existent à cet effet. Leur choix dépend de leurs conditions d'application et de l'aménagement de la salle de contrôle [Taborin, 89]. Ce travail d'intégration et d'évaluation de l'IHM n'a pas fait particulièrement l'objet de recherches de notre part durant ces dernières années.

## 5.3 Conclusions

Ce chapitre a été consacré à la présentation de nos contributions en matière de spécification et de génération semi-automatique d'interfaces homme-machine. Nous avons tenu à ce que ces étapes s'intègrent à l'approche méthodologique globale visant à satisfaire le cahier des charges identifié au niveau du chapitre 1. Cette approche couvre toutes les étapes nécessaires depuis l'analyse du SHM jusqu'à la génération semi-automatique de l'interface, en passant par la modélisation de l'interaction homme-machine, la déduction des BIO et la spécification des différentes vues et objets graphiques. L'application de cette démarche sur un exemple pédagogique concerne le cas d'un procédé industriel de fabrication de godets métalliques remplis par une solution chimique préparée préalablement. Le déroulement complet de notre démarche globale sur cet exemple est détaillé dans [Riahi, 04].

Par ailleurs, de nombreux travaux sont en cours. Ils sont de nature méthodologique, théorique et pratique. Nous les exposons dans la dernière section de ce mémoire : travaux en cours et perspectives.

## TRAVAUX EN COURS ET PERPECTIVES

- **Partenariat industriel** : Nous cherchons vivement à monter un partenariat industriel qui nous permettra d'appliquer notre méthodologie sur un exemple industriel réel. Ceci nous permettra de dégager, sans équivoque, les éventuels carences, oublis et ambiguïtés de cette démarche et de l'adapter aux exigences des applications industrielles.

- **Prise en compte du temps** : Dans sa définition actuelle, le formalisme des RdPS permet de décrire le comportement dynamique d'un système sans référence quantitative au temps réel qui s'écoule lors des invocations. Nous souhaitons aujourd'hui explorer la modélisation du temps réel qui est considéré comme une caractéristique forte des systèmes industriels. Une étude bibliographique des moyens permettant d'atteindre cet objectif est nécessaire. Nous envisageons d'étudier, en particulier, l'intégration à notre approche des techniques issues des réseaux temporisés dans le but de pouvoir décrire ces phénomènes.

- **Les Systèmes Multi-agents pour la spécification et la génération des IHM :** Durant ces dernières années, nous avons travaillé, entre autres, sur la définition d'une approche globale de conception et de génération des IHM. Nous nous sommes intéressés à la spécification des IHM au moyen de la technique des Systèmes à Base de Connaissances et ce à l'image de la tendance générale. Récemment, nous avons démarré l'exploration de la voie des Systèmes Multi-agents pour la spécification et la génération des IHM [Riahi et al., 01] [El Béji, 01]. En effet, ces systèmes semblent représenter une voie prometteuse et alternative aux SBC. L'objectif c'est de procéder à des comparaisons qualitatives et quantitatives entre ces deux techniques afin de répondre à des questions du genre « Qui des SBD et des SMA offre » :
    - une meilleure organisation des connaissances ergonomiques et règles de style ?
    - une meilleure flexibilité et facilité dans la gestion et la maintenance des connaissances manipulées ?
    - une plus grande facilité dans l'écriture et la maintenance du code informatique (règle de production, agent) exploitant les connaissances ergonomiques

- **Interfaces Homme-Machine Plastiques** : Nous avons engagé également des travaux de recherches qui s'inscrivent dans le domaine de l'Interaction Homme-Machine et répondent à un nouveau requis de l'informatique diffuse (ubiquitous computing, pervasive computing, etc.). Dans ce travail de recherche, récemment engagé, nous cherchons à proposer un cadre de référence qui structure le processus de conception et de développement d'IHM plastiques. Il s'agit de définir clairement, pour ce processus de conception, les étapes et les modèles sous-jacents. Un générateur d'IHM plastiques, qui suppose la spécification par le concepteur, de modèles donnés (modèles de tâche, modèle des plates-formes cible, etc.) est également à mettre au point.

## REFERENCES BIBLIOGRAPHIQUES

**[Abed, 90]** M. Abed. Contribution à la modélisation de la tâche par outils de spécification exploitant les mouvements oculaires : application à la conception et l'évaluation des interfaces homme-machine. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis, Septembre 1990.

**[Abed et al., 98]** M. Abed, H. Ezzedine. Vers une démarche intégrée de conception-évaluation des systèmes Homme-Machine. Journal of Decision Systems, vol. 7, pp. 147-175, 1998.

**[Abed et al., 01]** M. Abed. Modélisation des tâches dans la conception et l'évaluation des systèmes interactifs : la méthode SADT/Petri. In Kolski C. (Ed.), Analyse et Conception de l'IHM. Interaction Homme Machine pour les SI, Volume 1, pp. 145-174. Paris: Éditions Hermes, 2001.

**[Abed, 01]** M. Abed. Méthodes et modèles formels et semi-formels pour la conception et l'évaluation des systèmes homme-machine. Habilitation à diriger des recherches. Université de Valenciennes et du Hainaut-Cambrésis, France, mai 2001.

**[Abrial, 96]** J. R. Abrial. The B Book : Assigning Programs to Meanings. Cambridge University Press, 1996.

**[Ait-Ameur et al., 98]** Y. Ait-Ameur, P. Girard and F. Jambon. A uniform approach for specification and design of interactive systems: the method B. Proceedings DSV-IS'98, Coesner's House, Abingdon, UK, June 3-5 1998, pp. 333-352.

**[Ajmone et al., 95]** M. Ajmone, M. G. Balbo, C. Conte, Susanna Donatelli, and G. Franceschinis. ModellingWith Generalized Stochastic Petri Nets. Wiley (1995).

**[Alty et al., 85]** J.L. Alty, P. Elzer, O. Holst, G. Johannsen, S. Savory. Literature and user survey of issues to man-machine interfaces for supervision and control systems. Report of Esprit project P600, Copenhagen : C.R.I., 1985.

**[Amalberti et al., 98]** R. Amalberti, J.M. Hoc. Analyse des activités cognitives en situation dynamique : pour quels buts? Comment ? Le travail humain vol 61, n°3, 1998, pp. 209-234.

**[Balbo, 94]** S. Balbo. Un pas vers l'évaluation automatique des interfaces homme-machine. Thèse en informatique, Université Joseph Fourier (Grenoble 1), septembre 1994.

**[Barthet 88]** M.-F. Barthet. Logiciels Interactifs et Ergonomie – Modèles et méthodes de conception. Dunod Informatique, 1988.

**[Bass et al., 91]** L. Bass, R. Little, R. Pellegrino, S. Reed, S. Seacord, S. Sheppard, M. Szesur. The Arch model : Seeheim revisited. Proceedings of User Interface Developpers' Workshop, at CHI'91, pp. 2-26, avril 1991.

**[Bastien et al., 01]** J.M.C. Bastien, D.L. Scapin. Évaluation des systèmes d'information et Critères Ergonomiques. In C. Kolski (Ed.), Systèmes d'information et interactions homme-machine. Environnement évolués et évaluation de l'IHM. Interaction homme-machine pour les SI (Vol. 2, pp. 53-79). Paris: Hermès, 2001.

**[Bastien, 96]** J. M. C. Bastien. Les critères ergonomiques : un pas vers une aide méthodologique à l'évaluation des systèmes interactifs. Thèse de doctorat d'ergonomie cognitive, Paris, France, 1996.

**[Bernstein, 95]** D. J. Bernstein. Using Motif with C++, SIGS Books, 1995.

**[Berry et al., 88]** G. Berry, G. Gonthier. The Esterel Synchronous Programming Language : Design, Semantics, Implementation. Technical report 842, INRIA, 1988.

**[Berthomieu et al., 01]** B. Berthomieu, M. Boyer, M. Diaz, C. Girault, S. Haddad, P. Moreaux, J.F. Pradat-Peyre, P. Sénac, F. Vernadat. Les Réseaux de Petri, Modèles Fondamentaux, Ed. Hermes 2001, ISBN 2-7462-0250-6.

**[Bjorner, 87]** D. Bjorner. VDM : a Formal Method at Work. In Proc. VDM Europe Symposium'87, Springer-Verlag, 1987.

**[Bodart et al., 95]** F. Bodart, A. Hennebert, J.M. LeHeureux, I. Provot, J. Vanderdonckt et G. Zuchinetti. Key activities for a development Methodology of Interactive Applications. Chapter 4 in Critical Issues in User Interface Systems Engineering. D. Benyon and Ph. Palanque (Eds), Springer-Verlag, 1995.

**[Boehm, 88]** B.W. Boehm. A spiral model of software development and enhancement. Computer may 1988.

**[Bolognesi, 89]** T. Bolognesi et E. Brinksma. The formal description technique LOTOS, Introduction to the ISO specification language LOTOS. Elsevier Science Publishers, 1989.

**[Brams, 83]** G.W. Brams. Réseaux de Petri : Théorie et Pratique, Tome 2 : modélisation et application. Editions Masson, Paris, 1983.

**[Brun, 98]** P. Brun. XTL : une logique temporelle pour la spécification formelle des systèmes interactifs. Thèse en informatique, Université Paris XI – Orsay, septembre 1998.

**[Calvez, 93]** J.P Calvez. Spécification et conception des systèmes : une méthodologie. pp. 93-157, Edition Masson, 1993.

**[Card et al., 83]** S.K. Card, T.P. Moran and A. Newell. The psychology of human-computer interaction. Hillsdale, NJ. Erlbaum, 1983.

**[Cardelli et al., 85]** L. Cardelli, R. Pike. Squeak : A language for communicating with mice. Computer graphics, vol. 19. July, 1985.

**[Clement, 88]** D. Clement. Specifying the behaviour of graphical objects using Esterel. Technical report 836. INRIA, 1988.

**[D'Ausbourg et al., 96]** B. D'Ausbourg, G. Durrieu and P. Rocher. Deriving a formal model of interactive system from its UIL description in order to verify and to test its behaviour. In Proceedings of DSV-IS'96, springer verlag, pp. 104-122.

**[D'Ausbourg et al., 97]** B. D'Ausbourg, G. Durrieu and P. Rocher. Vers un environnement de validation automatique des IHM. IHM'97, 9ème journées sur l'ingénierie de l'interaction homme-machine, 10-12 septembre 1997, France.

**[Dang, 88]** W. Dang. Formal specification of interactive languages using Definite Clause Grammers. Réunion internationale sur l'implantation de langage de programmation et la programmation logique, Orléans, Mars 1988.

**[David et al., 92]** R. David and H. Alla. Du GRAFCET aux Réseaux de Petri, Editions HERMES, Paris, 1992.

**[De Nicola, 91]** R. De Nicola. Action and State-based Logics for Process Algebras. CONCUR '91, 2nd International Conference on Concurrency Theory, Amsterdam, The Netherlands, August 26-29, 1991, Proceedings. Lecture Notes in Computer Science 527 Springer 1991, ISBN 3-540-54430-5.

**[De Rosis et al., 98]** P. De Rosis. Formal Description and Evaluation of User Adapted Interfaces. Int. Journal of Human-Computer Studies, vol. 49, 1998, pp. 95-120.

**[Diaper et al., 01]** D. Diaper et N. Stanton (Eds). The handbook of task analysis for human-computer interaction. Lawrence Erlbaum Associates, London 2001.

**[Duce et Paterno., 93]** D. A. Duce, F. Paternò. A Formal Specification of a Graphics System in the Framework of the Computer Graphics Reference Model. Comput. Graph. Forum 12(1): 3-20 (1993).

**[El Béji, 01]** S. El Béji. Apports des Systèmes Multi-Agents dans la spécification et la génération des interfaces graphiques. Mémoire de DEA en Informatique. Institut Supérieur de Gestion de Tunis. Février 2004.

**[Embrey, 86]** D. Embrey. A systematic human error reduction and prediction approach, in proceeding of international topical meeting on advances in human factors in nuclear power systems. Knoxville Tenessee, USA, 1986.

**[Fadier , 90]** E. Fadier. Fiabilité Humaine : Méthodes d'analyse et domaine d'application. In : J. Leplat, G. De Terssac, (Eds.) : Les facteurs humains de la fiabilité dans les systèmes complexes. Editions Octarès, Marseille (1990) 47-80.

**[Fekete et al., 01]** J. D. Fekete, P. Girard. Environnements de développement des systèmes interactifs. Chapitre 1, dans "Environnements évolués et évaluation de l'IHM : interaction homme-machine pour les SI 2", C. Kolski (Eds.), editions Hermes, 2001, pp. 23-52.

**[Flecchia et al., 87**] M. Flecchia, R. D. Bergeron. Specifying complex dialogues in ALGEA. Proceeding CHI and graphics interface, ACM, New York.

**[Gamboa-Rodriguez et al., 97]** F. Gamboa-Rodriguez et D. L. Scapin. Editing MAD* task descriptions for specifying user interfaces, at both semantic and presentation levels. In Proceedings DSV-IS '97, 4th International Eurographics Workshop on Design, Specification, and Verification of Interactive Systems (Granada, Spain, 4-6 June 1997).

**[Gamboa-Rodriguez, 98**] F.Gamboa Rodriguez. Spécification et implémentation d'ALACIE : atelier logiciel d'aide à la conception d'interfaces ergonomiques, thèse en informatique, Université Paris sud XI, octobre 1998.

**[Gaudel, 84]** M.-C. Gaudel, A first introduction to Pluss, Alvey Workshop on Formal Specifications. Swindon (GB), October 1984.

**[Gilmore et al., 89]** W.E. Gilmore. D.I. Gertman and H.S. Blackman. User Computer Interface in process Control : A Human factor Engineering Handbook. Academic Press, 1989.

**[Girault et al., 02]** C. Girault, R. Valk. Petri Nets for Systems Engineering : A Guide to Modeling, Verification, and Applications. Springer-Verlag, 2002, ISBN: 3-540-41217-4.

**[Grislin, 95]** M. Grislin. Définition d'un cadre pour l'évaluation a priori des Interfaces Homme-Machine dans les systèmes industriels de supervision. Thèse de doctorat de l'Université de Valenciennes et du Hainaut-Cambrésis, Novembre 1995.

**[Grislin et al, 97]** M. Grislin, C. Kolski. Proposition d'une démarche d'évaluation a priori des IHM de supervision : application au étapes de spécification fonctionnelle et de conception préliminaires. RAIRO-APII-JESA, vol 31, n°7, 1997, pp. 1111-1154.

**[Goguen, 96]** J.A. Goguen et G. Malcolm. Algebraic Semantics of Imperative Programs. The MIT Press, 1996.

**[Goldberg et al., 83]** A. J. Goldberg, D. Robson Smalltalk-80 : The Language and Its Implementation. Reading, MA, Addison-Welsey, 1983.

**[Grammenos et al., 99]** D. Grammenos, D. Akoumianakis, C. Stephanidi. Integrated support for working with guidelines : the guideline management system. Interacting with Computers, 12, 2, 1999, pp. 281–311.

**[Green, 86]** M. Green. A survey of three dialog models. ACM Transaction on Graphics, vol. 5, n°3, p. 244-275, 1986.

**[Halbwachs et al., 91]** N. Halbwachs, P.Caspi, P. Raymond, D. Pilaud. Programmation et vérification des systèmes réactifs : le langage Lustre. Techniques et Sciences Informatiques, vol 10, n°2, 1991, pp. 139-158.

**[Harel et al., 87]** D. Harel, Statecharts : A visual formalism for complex systems. In Science of Computer Programming, volume 8, 1987.

**[Harrison et al., 90]** M. Harrison and M.Thimbelbey. In formal methods in human computer interaction. Harrison & Thimbelby (Eds), Cambridge university Press, 1990.

**[Hartson et al., 89]** H.R. Hartson, D. Hix. Human computer interface development : concepts and systems for its management. ACM : computer surveys, vol 21, number1. pp. 5-92, 1989.

**[Hill, 86]** R.D. Hill. Supporting Concurency. Communication and Synchronisation in Human-Computer Interaction. ACM transaction on Graphics, vol. 5, n°3, p. 179-210, July 1986.

**[Hix et Hartson, 93]** D. Hix, H. R. Hartson. Developing user interface : Ensuring usability through Product process. John Wiley Sons, New York, 1993.

**[Hoc and Amalberti, 95]** J.M. Hoc, R. Amalberti. Diagnosis : Some theoretical questions raised by applied research. Current Psychology of Cognition, 14 (1), pp. 73-101, 1995.

**[IWC, 99]** Interacting with Computers. Special issue, Volume 12, 2, 1999.

**[IWC, 00]** Interacting with Computers. Special issue, volume 12, 3, 2000, 2000.

**[Jacko et al., 03]** J. A. Jacko et A. Sears (Eds.) : The human-computer interaction handbook : fundamentals, evolving technologies and emerging applications (human factors and ergonomics). Lawrence Erlbaum Associates, 2003.

**[Jacob, 85]** R.J.K. Jacob. A state transition diagram language for visual programming. IEEE Computer, vol. 18 (8), August 1985, pp. 51-59.

**[Jambon et al., 01]** F. Jambon, Ph. Brun, Y. Aït-Ameur. Spécification des systèmes interactifs. Dans Kolski C. (Ed.), Analyse et Conception de l'IHM. Interaction Homme-Machine pour les SI, Volume 1, pp. 175-206. Paris: Éditions Hermes, 2001.

**[Johannsen , 82]** G. Johannsen. Man-Machine systems introduction and background. Proceedings IFAC International Conference on Analysis Design and Evaluation of Man-Machine Systems, Baden-Baden, Federal Republic of Germany.

**[Johannsen, 95]** G. Johannsen. Conceptual design of multi-human machine interfaces. Proceedings 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, MIT, Cambridge, USA, June 27-29, 1995.

**[Johnson et al, 93]** P. Johnson, S. Wilson. ADEPT, Advanced Environment for Prototyping with Task Model. Proceedings of ACM INTERCHI'93, Conference on Human Factors in Computing Systems, pp. 56-57. Amsterdam, the Netherlands, 1993.

**[Johnson et al., 92]** C.W. Johnson et M.D Harrison. Using Temporal logic to support the specification and prototyping of interactive control systems, Int. J. Man-Machine Studies, vol. 37, pp. 357-385, 1992.

**[Johnson, 95]** C. Johnson. Time and the web : representing and reasonning about temporal properties of interaction with distributed systems. People and Computers. HCI'95, Cambridge University Press, p 39-50, 1995.

**[Jones, 80]** C.B. Jones. Software Development – A Rigourous Approach. Prentice Hall, 1980.

**[Karsenty, 90]** S. Karsenty. La spécification d'interfaces. Bulletin de liaison de la recherche en informatique et en automatique, INRIA, 127. 1990.

**[Khelil, 01]** N. Khelil. Modélisation sûre des tâches interactives dans les interfaces homme-machine. Mémoire de DEA, faculté des sciences de Tunis, octobre, 2001.

**[Kolski et al., 01]** C. Kolski, H. Ezzedine, M. Abed. Développement du logiciel : des cycles classiques aux cycles enrichis sous l'angle de l'IHM. In C. Kolski (Ed.), Analyse et conception de l'IHM : Interaction Homme-Machine pour les SI 1, Editions Hermes, Paris, 2001.

**[Kolski et al., 91]** C. Kolski and P. Millot. A rule-based approach for the ergonomic evaluation of man-machine graphic interface. International Journal of Man-Machine Studies, 35, pp. 657-674, 1991.

**[Kolski et al., 96]** C. Kolski and F. Moussa. Two examples of tools for working with guidelines for process control field : Synop and ERGO-CONCEPTOR. Third Annual Meeting of the International Special Interest Group on "Tools for working with guidelines", J. Vanderdonckt (Ed.), Namur, Belgium, 4 June 1996.

**[Kolski, 95]** C. Kolski. Méthodes et modèles de conception et d'évaluation des interfaces homme-machine. Habilitation à Diriger des Recherches, Université de Valenciennes et du Hainaut-Cambrésis, Janvier, 1995.

**[Kolski, 97]** C. Kolski. Interfaces homme-machine, application aux systèmes industriels complexes. Editions Hermes, Paris, 1997.

**[Lepreux et al., 03]** S. Lepreux, M. Abed, C. Kolski. A human-centred Methodology Applied to Decision Support System Design and Evaluation in Railway Network Context. Cognition Technology and Work, 5, pp. 248-271, 2003.

**[Lambert, 73]** H.E. Lambert. System Safety Analysis and Fault Tree Analysis. UCID-16238, 31,9 Mai 1973.

**[Larvet, 94]** P. Larvet. Analyse des systèmes : de l'approche fonctionnelle à l'approche objet. Inter Editions, 1994.

**[Lonczewski et Schreiber, 96]** F. Lonczewski, S. Schreiber. The Fuse-system : an integrated User Interface Environment. In Vanderdonkt J. (Ed.), Computer-Aided Design of User Interfaces, Proceedings of the second International Workshop on Computer-Aided Design of User Interfaces, CADUI'96, pp. 141-144. Namur, June, 1996.

**[Long et al., 90]** J.B. Long, I. Denley. Evaluation for practice. Tutorial, Ergonomics society annual conference, 1990.

**[Mahfoudhi, 97]** A. Mahfoudhi. TOOD : Une méthodologie de description orientée objet des tâches utilisateur pour la spécification et la conception des IHM : Application au contrôle du trafic aérien. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis, France, 1997.

**[McDermid et al., 84]** J. McDermid, K. Ripkin. Life cycle support in the ADA environment, University Press Cambridge, 1984.

**[Millot, 90]** P. Millot. Coopération homme-machine dans les procédés industriels. Dans les facteurs humains de la fiabilité dans les systèmes complexes. J. Leplat et G. De Terssac (Eds.), Eition Octares/ entreprises, Marseille, 1990.

**[Millot, 99]** P. Millot. La supervision et la coopération Homme-Machine dans les grands systèmes industriels ou de transports. In J-g Ganascia (Eds.), Sécurité et Cognition, Hermès, Paris, pp. 124-145.

**[Moalla, 85]** M. Moalla. Réseaux de Petri interprétés et Grafcet. Revue TSI de l'AFCET vol. 4 (1), 1985.

**[Mori et al., 02]** G. Mori, F. Paterno, C. Santoro. CTTE : Support for developing and analyzing task models for interactive system design, IEEE transactions on software engineering, vol. 28, n° 9, septembre 2002.

**[Moussa et al., 90]** F. Moussa, C. Kolski and P. Millot. Artificial intelligence approach for the creation and the ergonomic design of man-machine interfaces in control room. Ninth European Annual Conference on Human decision making and manual control. Varese, Italy, September 10-12, 1990.

**[Moussa et al., 99]** F. Moussa, M. Riahi, M. Moalla and C. Kolski. A petri net method for specification and automatic generation of user interface. In Human-Computer Interaction : Ergonomics and user interfaces. volume 1, Proceedings HCI'99, 8th International Conference on Human-Computer Interaction. Munich, Germany, August, H.J. Bullinger, J. Ziegler (Eds.), Lawrence Erlbaum Associates, 1999, pp. 988-992.

**[Moussa, 00]** F. Moussa. Exemple de développement d'un outil d'ingénierie pour la conception d'interface : ERGO-CONCEPTOR, dans Ingénierie des interfaces homme-machine : conception et évaluation. C. Kolski. Hermes, 2ième édition 2000. Chapitre 13, pp. 289-335.

**[Moussa et al., 00]** F. Moussa, C. Kolski, M. Riahi. A Model Based Approach to Semi-Automated User Interface Generation for Process Control Interactive Applications. Revue Internationale : Interacting with Computers. Edition Elsevier. 12, 3 (2000) 245–279.

**[Moussa et al., 00a]** Moussa F., M. Riahi. Use of formalized guidelines for semi-automated generation of GUI : the Ergo-Conceptor+ tool. In J. Vanderdonckt, C. Farenc (Ed.), Tools for Working With Guidelines TFWWG'2000, Biarritz, October, (pp. 237-246), London: Springer, 2000.

**[Moussa et al., 2002]** F. Moussa, M. Riahi, C. Kolski, M. Moalla. Interpreted Petri Nets used for Human-Machine Dialogue Specification. International journal : Integrated Computer-Aided Engineering (ICAE). Volume 9, N° 1, 2002, (pp. 87-98). Edition Iopress. ISSN: 1069-2509.

**[Moussa et al., 95]** F. Moussa, M. Moalla. Toward a user centered system design methodology : application to the graphical interfaces design. AMSE periodicals "Advanced in modeling and analysis", vol. 35, n°1, pp. 1-10, 1995.

**[Moussa et al., 96]** F. Moussa, M. Riahi. Génération automatique d'Interface " Intelligente " pour la conduite de procédé. Forum de la recherche en informatique. Tunis, 16-18 juillet 1996.

**[Moussa, 92]** F. Moussa. Contribution à la conception ergonomique des interfaces de supervision dans les procédés industriels, application au système ERGO-CONCEPTOR. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis, 1992.

**[Nigay, 94]** L. Nigay. Conception et modélisation logicielles des systèmes interactifs : application aux interfaces multimodales. Thèse de l'Université de Joseph Fourier. Grenoble I. Mars, 1994.

**[Nielsen, 93]** J. Nielsen. Usability engineering. Academic Press, 1993.

**[Norman, 86]** D.A. Norman. Cognitive engineering, In D.A. Norman & S.W. Draper (Eds), User centred system design : new perspectives on human computer interaction. Hillsdale N.J., Elbraum, 1986.

**[O'Hara et al., 96]** J. O'Hara, W. Brown, W. Stubler, J. Wachtel and J. Persensky. Human-Machine Interface Design Review Guideline (NUREG-0700, rev. 1). U.S. Nuclear Regulatory Commission, Washington, 1996.

**[O'Hara et al., 99]** J. O'Hara and W. Brown. Software Tool for the Use of Human Factors Engineering Guidelines to Conduct Control Room Evaluations. In : Büllinger, H.J., Ziegler, J. (Eds.) : Human-Computer Interaction : communication, cooperation and application design. Lawrence Erlbaum Associates, London, pp. 973–977, 1999.

**[Olsen et al., 83]** D. R. Olsen, E. P. Dempsey. Syngraph : A graphical user interface generator. ACM Transaction on Graphics, July 1983.

**[Palanque et al., 95]** P. Palanque, R. Bastide. Spécifications formelles pour l'ingénierie des interfaces homme-machine. Technique et Science Informatique, vol. 14, n° 4, éditions Hermès, p. 473-500, 1995.

**[Palanque, 97]** P. Palanque. Spécifications formelles et systèmes interactifs : vers des systèmes fiables et utilisables. Habilitation à diriger des recherches, Université de Toulouse I, 1997.

**[Palanque et al., 97]** P. Palanque and F. Paterno (Eds.). Formal Methods in Human-Computer Interaction, Springer Verlag, 1997.

**[Palanque, 95]** P. Palanque et R. Bastide. Spécifications formelles pour l'ingénierie des interfaces homme-machine. Technique et Science Informatique, vol. 14, n° 4, éditions Hermès, p. 473-500, 1995.

**[Parnas, 69]** D.L. Parnas. On the use of transition diagrams in the design of user interface for an interactive computer system, 24 th ACM Conference, 1969, pp. 379-385.

**[Paterno, 01]** F. Paterno. Towards a UML for interactive systems. Poc. Eng. Human-Computer Interaction conference, HCI'01, pp. 175-185, 2001.

**[Paterno et al., 92]** F. Paterno and G. Faconti. On the use of Lotos to describe graphical interaction. In proceedings of people and computer VII, HCI'92 conference, cambridge university press, 1992, pp. 155-174.

**[Paterno et al., 99]** F. Paterno, C. Mancini. Designing usuable hypermedia, empirical software engineering, 4(1), pp. 11-42, 1999.

**[Payne, 86]** S.J. Payne et M. Green. The structure of command languages : an experiment on task-action grammar. Int. Journal of Man-Machine Studies, vol. 30, Academic Press, p. 213-234, 1989.

**[Petri, 62]** C.A. Petri. Kommunication mit automaten, PhD Dissertion. University of Bonn, 1962.

**[Pfaff, 85]** G.E. Pfaff. User Interface Management Systems, Edition Eurographics Seminars, Springer-Verlag, 1985.

**[RAISE, 92]** The RAISE Language Group. The RAISE Specification Language, Prentice Hall, 1992.

**[Rasmussen, 80]** J. Rasmussen. The human as a system component. In H.T. Smith and T.R.G. Green (Eds.), Human Interaction With Computer, London Academic Press. 1980.

**[Rasmussen, 86]** J. Rasmussen. Intelligent Decision Support in Process Environments. A framework for cognitive Task Analysis in System Design. In : Hollnagel, E., Mancini, G., Woods, D.D. (Eds.). NATO ASI series. Vol. F21. Springer-Verlag, Berlin (1986).

**[Reed et al., 99]** P. Reed, K. Holdaway, S. Isensee, E. Buie, J. Fox, J. Williams and A. Lund. User interface guidelines and standards : progress, issues, and prospects, Interacting With Computers, 12, 2, 1999, pp. 119–142.

**[Riahi et al., 98]** M. Riahi, F. Moussa, M. Moalla et C. Kolski. Vers une spécification formelle des interfaces homme-machine basée sur l'utilisation des réseaux de Petri. Dans M.F. Barthet (Ed.), Actes du 6ème Colloque ERGO IA'98 Ergonomie et Informatique Avancée. (pp. 196-205). Bayonne: Edt : ESTIA/ILS, ISBN :2-9503134-9-3, 1998.

**[Riahi et al., 00]** M. Riahi, F. Moussa, C. Kolski and M. Moalla. Use of interpreted petri nets for human-machine dialogue specification in process control. Proceedings ACIDCA'2000 International Conference on Artificial and Computational Intelligence for Decision, Control and Automation in Engineering and Industrial Applications. 22-24 March, Monastir, Tunisia.

**[Riahi et al., 98a]** M. Riahi, F. Moussa, M. Moalla et C. Kolski. Approche de spécification formelle des interfaces homme-machine basée sur une analyse du système homme-machine, application au cas du contrôle des procédés industriels. Sixième colloque Maghrébin sur les Modèles Numériques de l'Ingénieur, Tunis, Tunisie, novembre 1998.

**[Riahi et al., 01]** M. Riahi, F. Moussa. Contribution of the Petri Nets and the multi Agent system in HCI Specification. In M.J. Smith, G. Salvendy, D. Harris, R. Koubek (Ed.), Usability evaluation and Interface design: Cognitive Engineering, Intelligent Agents and Virtual Reality, volume 1. (pp. 76-80). London: Lawrence Erlbaum Associate Publishers, 2001. (Actes de HCII2001, 9th International Conference on Human-computer Interaction, August New orleans, LA, USA.).

**[Riahi, 04]** M. Riahi. Contribution à l'élaboration d'une méthodologie de spécification, de vérification et de génération semi-automatique d'interfaces homme-machine : Application à l'outil Ergo-Conceptor+. Thèse de Doctorat, Université de Valenciennes et du Hainaut-Cambrésis. France. Septembre 2004.

**[Richard et al., 90]** J.F. Richard, C. Bonnet, R. Guiglione (Fds.). Traité de Psychologie Cognitive, le traitement de l'information symbolique. Paris, 1990. Dunod.

**[Robert et al., 97]** J. M. Robert, F Aubin, S. Jagannath. Des tâches génériques aux interfaces génériques. IHM'97, 9ème journées sur l'ingénierie de l'interaction homme-machine, 10-12 septembre 1997, France, pp. 39-45.

**[Rodriguez, 98]** F.G. Rodriguez. Spécification et implémentation d'ALACIE : atelier logiciel d'aide à la conception d'interfaces ergonomiques. Thèse de doctorat, Université Paris sud, Paris XI, France, octobre, 1998.

**[Scapin et al., 97]** D.L. Scapin, J.M.C. Bastien. Ergonomic criteria for evaluating the ergonomic quality of interactive systems. Behaviour and Information Technology, 16, p. 220-231, 1997.

**[Scapin et al., 01]** D.L. Scapin, J.M.C. Bastien. Analyse des tâches et aide ergonomique à la conception : l'approche MAD*. In Kolski C. (Ed.), Analyse et Conception de l'IHM. Interaction Homme Machine pour les SI, Volume 1, pp. 85-116. Paris: Éditions Hermes, 2001.

**[Scapin et al., 89]** D.L, Scapin et C. Pierret-Golbreich. MAD : une Méthode Analytique de Description des tâches, Actes du Colloque sur l'Ingénierie des Interfaces Homme-Machine, Sophia-Antipolis, 24-26 Mai, 1989.

**[Shepherd, 98]** A. Shepherd. HTA as a framework for task analysis. Ergonomics, vol 41, n°11, 1998, pp. 1537-1552.

**[Sheridan, 97]** T.B. Sheridan. Task analysis, task allocation and supervisory control. In Handbook of Human-Computer Interaction, Helander M.G., Landauer T.K., Prabhu P. (Eds), North Holland, Amsterdam, pp. 87-105, 1997.

**[Sheridan, 85]** T.B. Sheridan. Forty Five years of man machine system : history and trends. 2nd IFAC congress on analysis, design and evaluation of man-machine systems. Varese, Italy, September, 1985.

**[Shneiderman, 98]** B. S. Shneiderman. Designing the user interfaces : strategies for effective human-computer interaction. Third edition, Reading MA, Addison Wesley, 1998.

**[Smith, 86]** S.L. Smith and J.N. Mosier. Guidelines for designing user interface software. Report EDS-TR-86-278, The MITRE Corporation, Bedford, MA, 1986.

**[Spivey, 89**] J. M. Spivey. The Z notation : a reference manual, Prentice-Hall, Inc., Upper Saddle River, NJ, 1989.

**[Suhner et al., 92]** M. C. Suhner, M. Gabriel, L. Claude. Utilisation de l'AMDEC pour générer la base de connaissances d'un système expert d'aide au diagnostic : le cas des systèmes complexes automobiles, Revue diagnostic et sûreté de fonctionnement, vol 2, n°2, 1992, pp. 133-153.

**[Swain et al., 81]** A.D. Swain, H.G. Guttman. Handbook of human reliability analysis with emphasis on nuclear power plant applications. US Nuclear regulatory commission technical report NUREG/CR 1278, 1981.

**[Swain, 63]** A.D. Swain. A method for performing a human factors reliability analysis. Monograph SCR, 685, Sandia corporation, 1963.

**[Swain, 64]** A.D. Swain. THERP, Sandia lab, Albuquerque, W, Mex Rep SCR 64-1338, 1964.

**[Tabary et al., 98]** D. Tabary and M. Abed. TOOD : An object-oriented methodology for describing user task in interface design and specification - An application to air traffic control. La Lettre de l'Intelligence Artificielle, vol 134-135-136, pp. 107-114, 1998.

**[Tabary, 00]** D. Tarby. L'approche orientée objet et les réseaux de Petri pour la spécification formelle des tâches. RJCIHM, 2000.

**[Tabary, 00a]** D. Tabary, M. Abed, C. Kolski. Object-oriented modelling of manual, automatic, interactive tasks in mono or multi-user contexts using the TOOD method. Proceedings MCPL'2000 Second IFAC/IFIP/IEEE Conference on Management and Control of Production and Logistics, Grenoble, 5-8 July 2000.

**[Tabary, 01]** D. Tabary. Contribution à TOOD, une méthode à base de modèles pour la spécification et la conception des systèmes interactifs. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis, France, décembre, 2001.

**[Taborin, 89]** V. Taborin. Communication homme-machine au sein des systèmes de supervision de procédé continu, réalisation des interfaces entre opérateur et système d'aide à la décision. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis, France, Mars 1989.

**[Tarby et al., 01]** J.C. Tarby, M.F. Barthet. Analyse et modélisation des tâches dans la conception des systèmes d'informations : mise en œuvre avec la méthode Diane+. Chapitre 4. C. Kolski (Ed.), Analyse et conception de l'IHM, Interaction Homme-Machine pour les systèmes d'information, vol. 1. Editions Hermès, Paris, 2001.

**[Tarby, 93]** C. Tarby. Gestion automatique du dialogue homme-machine à partir de spécifications conceptuelles. Thèse de doctorat en Informatique. Université de Toulouse I. Septembre 1993.

**[Terwilliger et al., 97]** R. B. Terwilliger, P. G. Polson. Relationships between usersband interface task representations, CHI'97, 22-27 march 1997, pp. 99-106.

**[Turner, 93]** K.J. Turner. Using formal description techniques: An introduction to Estelle, Lotos and SDL, Edition John, 1993.

**[Vallee, 92]** F. Vallee. Statemate : une méthode et un outil pour la spécifications des systèmes réactifs, RGE, n° 4/92, pp. 46-51, Avril 1992.

**[Vanderdonckt, 00]** J. Vanderdonckt. A small Kowledge-Based System for selecting interaction styles, Proceedings of international workshop on Tools for Working with Guidelines, TFWWG-2000, Biarritz, 7-8 octobre 2000, Springer-verlag, Londres, pp. 247-262, 2000.

**[Vanderdonckt, 99]** J. Vanderdonckt. Development Milestones toward a Tool For Working With Guidelines. Interacting with Computers 12, 2 (1999) 81–118.

**[Villemeur, 88]** A. Villemeur. Sûreté de fonctionnement des systèmes industriels ; fiabilté, facteurs humains, informatisation, Edition Eyrolles, paris 1988.

**[Wallace, 93]** C. Wallace. The semantics of the C++ programming language. In Lecture Notes in Computer Science, n°702, Springer-Verlag, 1993.

**[Willem et al., 88]** R. Williem and V. Biljon. Extending Petri Nets for specifying Man-Machine dialogues, International Journal of Man-Machine Studies, vol. 28, 1988, pp. 437-45.

**[Wilson et al, 95]** S. Wilson, P. Johnson. Empowering users in a Task-Based approach to design. Proceedings DIS'95, pp. 25-31. Michigan, USA, 1995.

**[Winckler et al., 03]** M. Winckler et P. Palanque. StateWebCharts : a formal description technique dedicated to navigation modeling of web applications. Proceedings DSVIS'2003, Funchal, Portugal, June 2003.

**[Young, 92]** D.A. Young. Object-Oriented Programming with C++ and OSF/Motif, Prentice Hall, 1992.

**LISTE DES FIGURES**

Figure 1.1 : Architecture d'un Système Homme-Machine 7
Figure 1.2 : (a) poste de supervision (b) exemple d'Interface Homme-Machine 8
Figure 1.3 : Modèle du processeur humain 9
Figure 1.4 : L'échelle de décision de Rasmussen 10
Figure 1.5 : La théorie de l'action de Norman 12
Figure 1.6 : Une boîte SADT relative à une tâche humaine interactive dont le fonctionnement est décrit par un RdPS 15
Figure 1.7 : MCD d'une modélisation CTT 14
Figure 1.8 : Une arborescence de tâches MAD 16
Figure 1.9 : Modèle proposé par Long et al. 18
Figure 1.10 : Modèle en étoile 18
Figure 1.11 : Modèle ∇ (Nabla) 19
Figure 1.12 : Cycle de développement de TOOD 21
Figure 1.13 : Structure Méthodologique de TRIDENT 22
Figure 1.14 : Démarche d'Ergo-Conceptor 23
Figure 2.1 : Phase descendante de conception du système Homme-Machine inspirée de Millot 28
Figure 2.2 : Squelette méthodologique de l'approche proposée 30
Figure 2.3 description du procédé en terme de hiérarchie d'abstraction « moyens/objectifs » 31
Figure 2.4 : Les trois axes de description d'un procédé [Moussa, 92] 31
Figure 2.5: Analyse du Système Homme-Machine 32
Figure 3.1 : Taxonomie des techniques de spécification 38
Figure 3.2 : Extrait d'un programme Esterel 41
Figure 3.3 : Arbre en Lotos 42
Figure 3.4 : Automate à états finis décrivant une opération de calcul dans une calculatrice 42
Figure 3.5 : Exemple de l'état d'un RdP avant et après le franchissement d'une transition 43
Figure 4.1: Squelette méthodologique de l'approche proposée 47
Figure 4.2 : Réseau de Petri Interprété (RdPI) 47
Figure 4.3 : Structure modélisant l'état d'un opérateur face à une action élémentaire 48
Figure 4.4 : Composition séquentielle 49

Figure 4.5 : Structure PAR1 ..... 50
Figure 4.6 : Structure PAR2 ..... 50
Figure 4.7 : Composition parallèle ..... 50
Figure 4.8 : Structure de fermeture ..... 51
Figure 4.9 : Composition de fermeture ..... 51
Figure 4.10 : Exemple d'un modèle d'interaction homme-machine ..... 52
Figure 4.11 : Exemple d'un modèle de l'interaction homme-machine avec les BIO ..... 53
Figure 4.12 : Structures de contrôle des objets de l'interface ..... 54
Figure 4.13 : Processus de communication entre les différents RdP ..... 56
Figure 4.14 : Suppression d'une SEAO ..... 59
Figure 5.1 : rappel du Squelette méthodologique de l'approche proposée ..... 61
Figure 5.2 : Processus de spécification de la présentation de l'interface ..... 62
Figure 5.3 : Exemples de recommandations ergonomiques pour la génération des vues graphiques ..... 63
Figure 5.4 : Exemples de recommandations ergonomiques pour la sélection des objets de l'interface ..... 61
Figure 5.5 : Système à base de connaissances de l'outil Ergo-Conceptor+ ..... 64
Figure 5.6 : Joueur ou interpréteur des RdPI ..... 65
Figure 5.7 : Structures des matrices « Pre » et « Post » ..... 66

## LISTE DES TABLEAUX

Tableau 1.1 : Evaluation de quelques approches proposées pour la conception de l'interface ..... 24
Tableau 2.1 : Evaluation informelle des méthodes d'analyse des systèmes ..... 34