



**HAL**  
open science

# Problème du contrôle dans un système intégré de vision : utilisation d'un système multi-agents

Olivier Boissier

## ► To cite this version:

Olivier Boissier. Problème du contrôle dans un système intégré de vision : utilisation d'un système multi-agents. Interface homme-machine [cs.HC]. Institut National Polytechnique de Grenoble - INPG, 1993. Français. NNT: . tel-00005123

**HAL Id: tel-00005123**

**<https://theses.hal.science/tel-00005123>**

Submitted on 26 Feb 2004

**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.

THESE

*présentée par*

Olivier Boissier

*pour obtenir le grade de DOCTEUR*

de l'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

(Arrêté ministériel du 30 mars 1992)

spécialité: INFORMATIQUE

—

**PROBLEME DU CONTROLE  
DANS  
UN SYSTEME INTEGRE DE VISION  
Utilisation d'un système Multi-Agents**

—

Date de soutenance: 22 Janvier 1993

Composition du jury:

Président: M. Pierre Yves Cunin

rapporteurs: M. Jacques Ferber  
M. Gérard Masini

examineurs: Mme. Ruzena Bajcsy  
M. James L. Crowley  
M. Yves Demazeau

Thèse préparée au sein du laboratoire LIFIA-IMAG



## REMERCIEMENTS

Je tiens à remercier,

Monsieur Pierre Yves Cunin, professeur à l'Université Joseph Fourier, qui me fait l'honneur de présider ce jury,

Messieurs Jacques Ferber, professeur à l'Université Paris VI, et Gérard Masini, chargé de recherches CNRS au CRIN-INRIA Lorraine, pour avoir accepté de juger ce travail, pour le temps qu'ils y ont consacré et les remarques, toujours constructives, qu'ils ont bien voulu porter sur le manuscrit. Lors de ces échanges j'ai pu apprécier leurs qualités scientifiques et humaines.

Madame Ruzena Bajcsy, professeur à l'Université de Pensylvanie, qui me fait l'honneur de participer à ce jury,

Monsieur James L. Crowley, professeur à l'Institut National Polytechnique de Grenoble, qui en m'accueillant dans son équipe m'a permis de conduire cette thèse. Je lui suis tout aussi reconnaissant des conseils qu'il a su me donner et de la confiance qu'il m'a accordée.

Monsieur Yves Demazeau, chargé de recherche CNRS, qui m'a encadré ces deux dernières années. Je le remercie pour ses conseils, son accueil, sa disponibilité qui ne m'ont jamais fait défaut. Ses connaissances scientifiques, sa rigueur et son esprit d'analyse ainsi que ses qualités humaines m'ont permis de mener à terme ce travail de recherche. Je tiens à lui exprimer toute mon amitié.

La communauté Européenne, le projet BRA3038, l'Université Joseph Fourier ont rendu possible le financement de ces travaux.

Je tiens aussi à remercier :

Monsieur Philippe Jorrand, directeur du LIFIA, qui a accepté de m'accueillir dans ce laboratoire.

Les membres du LIFIA pour l'encadrement scientifique et chaleureux dont j'ai pu bénéficier.

Les membres de l'équipe PRIMA qui ont toujours su être là pour répondre à mes questions, ainsi que les stagiaires qui ont jalonné indirectement ou directement par leurs travaux et discussions l'avancement de la thèse.

Les membres du groupe PLEIAD dont les discussions ont constitué un élément important de réflexion sur les Systèmes Multi-Agents.

Les membres du projet VAP, en particulier Ata Etemadi, qui ont constitué le cadre de travail de cette thèse pendant un an.

Mme Chabre-Peccoud et M. J.P. Bertrandias, enseignants à l'Université Joseph Fourier qui ont su prendre en compte les contraintes d'un thésard en pleine rédaction. . .

Augustin Lux, Bruno Zoppis, Cécile Tron, Marie-hélène Stefanini, Philippe Bobet, Olivier Causse et Bernadette Dufour qui ont accepté avec gentillesse et courage de relire ce manuscrit.

Jaime Sichman sait combien je lui suis redevable de tout le temps qu'il a consacré à mes travaux, de la disponibilité dont il a toujours su faire preuve et de la pertinence de ses remarques. Son aide m'a été des plus précieuses et m'a permis de relativiser les moments difficiles.

Danielle Ziébelin, Philippe Genoud m'ont apporté une aide inestimable. Leur support moral et scientifique ne m'a jamais fait défaut.

A tous trois, je tiens à leur exprimer toute mon amitié.

Je voudrais enfin remercier :

Mes amis qui ont toujours su être là dans les moments difficiles et qui ont su se faire discrets et compréhensifs dans les moments de "bourre",

Mes parents et ma famille pour avoir su respecter et encourager mes objectifs,

Noémie qui a rempli ma vie de joies et de découvertes malgré le rythme de fous que je lui fais vivre depuis deux ans.

Marie qui . . . .

Je lui dédie ce manuscrit.





# Sommaire

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Contexte de notre étude . . . . .	1
1.1.1	Problème de contrôle . . . . .	2
1.1.2	Domaine d'étude . . . . .	3
1.2	Thèse défendue . . . . .	4
1.3	Démarche suivie . . . . .	5
1.4	Organisation du rapport . . . . .	7
<b>2</b>	<b>Contrôle</b>	<b>11</b>
2.1	Définitions de contrôle . . . . .	11
2.1.1	Automatique . . . . .	12
2.1.2	Théorie des systèmes hiérarchiques . . . . .	15
2.1.3	Intelligence Artificielle . . . . .	17
2.1.4	Intelligence Artificielle Distribuée . . . . .	21
2.1.5	Psychologie cognitive . . . . .	23
2.2	Nouvelle formulation du contrôle . . . . .	25
2.2.1	Préliminaires . . . . .	25
2.2.2	Problèmes de contrôle . . . . .	28
2.2.3	Couches de contrôle . . . . .	29
2.2.4	Définitions supplémentaires . . . . .	32
2.3	Conclusion . . . . .	33
<b>3</b>	<b>Contrôle dans les Systèmes Intégrés de Vision</b>	<b>35</b>
3.1	Vision par Ordinateur . . . . .	35
3.1.1	Définitions . . . . .	35
3.1.2	Système de Vision . . . . .	38

3.2	Systèmes Intégrés de Vision . . . . .	39
3.2.1	Définition . . . . .	39
3.2.2	Problématique d'un Système Intégré de Vision . . . . .	40
3.3	Exemples de Systèmes Intégrés de Vision . . . . .	44
3.3.1	Propriétés intrinsèques . . . . .	45
3.3.2	Procédures visuelles . . . . .	46
3.3.3	Stratégies perceptuelles . . . . .	49
3.3.4	Schéma . . . . .	50
3.3.5	Medusa . . . . .	53
3.3.6	Tea-1 . . . . .	54
3.3.7	Vision As Process . . . . .	55
3.4	Contrôle dans un Système Intégré de Vision . . . . .	58
3.4.1	Contrôleur et processus . . . . .	60
3.4.2	Couches de contrôle . . . . .	64
3.4.3	Connaissances de contrôle . . . . .	67
3.5	Conclusion . . . . .	70
<b>4</b>	<b>Propositions pour le contrôle d'un Système Intégré de Vision</b>	<b>75</b>
4.1	Expériences de contrôle dans le système VAP . . . . .	76
4.1.1	Contrôle au sein de la segmentation et de l'interprétation . . . . .	77
4.1.2	Contrôle dans le système . . . . .	84
4.1.3	Evaluation des expériences de contrôle dans VAP . . . . .	85
4.2	Le projet de recherches SATURNE . . . . .	86
4.3	Conclusions par rapport à VAP et SATURNE . . . . .	87
4.3.1	Architecture du Système . . . . .	88
4.3.2	Module général . . . . .	90
4.4	Intelligence Artificielle Distribuée . . . . .	90
4.4.1	Définition Agent/Société . . . . .	91
4.4.2	Description Agent/Société . . . . .	92
4.4.3	Problèmes abordés par l'Intelligence Artificielle Distribuée . . . . .	93
4.4.4	Recherches actuelles . . . . .	94
4.5	Principes généraux à la base de la structure de contrôle . . . . .	96
4.5.1	Organisations possibles du système . . . . .	97
4.5.2	Synthèse . . . . .	98
<b>5</b>	<b>Modèles pour l'étude du contrôle dans un Système Intégré de Vision</b>	<b>103</b>
5.1	Modèles . . . . .	104
5.1.1	Modèle d'Agent Social . . . . .	105
5.1.2	Modèle d'Agent Individuel . . . . .	119
5.1.3	Modèle de société . . . . .	120
5.2	Modèles de contrôle . . . . .	122
5.2.1	Modèle de contrôle individuel . . . . .	123
5.2.2	Modèle de contrôle social . . . . .	133

5.3	Modèles de Contrôle Individuel et Social d'un Système Intégré de Vision	146
5.3.1	Modèle de contrôle individuel d'un système intégré de Vision	149
5.3.2	Modèle de contrôle social d'un Système Intégré de Vision	159
5.4	Conclusion	162
<b>6</b>	<b>Réalisation</b>	<b>165</b>
6.1	Module Standard d'Agent	166
6.1.1	Séquenceur	169
6.1.2	Module Individu	171
6.1.3	Etat d'engagement social	184
6.1.4	Module-dialogue	185
6.1.5	Module-organisation	193
6.1.6	Bascule social	193
6.1.7	Bascule individu	194
6.2	Contrôle dans un Système Intégré de Vision : VAP	195
6.3	Contrôle dans l' <i>AGENT-DESCRIPTION-2D</i>	197
6.3.1	<i>AGENT-DESCRIPTION-2D</i>	197
6.3.2	Contrôle	203
6.3.3	Evaluation	206
6.4	Contrôle dans l' <i>AGENT-INTERPRETATION</i>	207
6.4.1	<i>AGENT-INTEPRETATION</i>	207
6.4.2	Contrôle	210
6.4.3	Evaluation	212
6.5	Contrôle Social dans un système intégré de vision : VAP	212
6.5.1	Capacités sociales	212
6.5.2	Exemple de contrôle social	216
6.5.3	Evaluation	221
6.6	Conclusion	222
<b>7</b>	<b>Conclusion</b>	<b>223</b>
7.1	Problèmes abordés	223
7.2	Perspectives	225
7.2.1	Intégration et Contrôle dans un système intégré de vision	225
7.2.2	Systèmes Multi-Agents	226
	<b>Bibliographie</b>	<b>229</b>



# Liste des Figures

1.1	Démarche suivie . . . . .	6
2.1	Contrôleur et Processus en Automatique . . . . .	13
2.2	Contrôle dans la théorie des systèmes hiérarchiques . . . . .	16
2.3	Formulation d'un système de production en terme de contrôleur et de processus . . . . .	26
2.4	Hiérarchie de contrôle . . . . .	30
3.1	Cycle de perception . . . . .	43
3.2	Cycle de contrôle du système <i>Vision As Process</i> . . . . .	43
3.3	Modèle des Procédures Visuelles . . . . .	47
3.4	SCHEMA : architecture du système . . . . .	51
3.5	Vision As Process : architecture du système . . . . .	56
3.6	Vision As Process : module standard . . . . .	56
3.7	Problèmes du contrôle dans un Système Intégré de Vision . . . . .	72
4.1	Cadre d'expérimentation pour l'étude du contrôle . . . . .	77
4.2	Image d'une scène . . . . .	79
4.3	Pyramide multi-résolutions . . . . .	79
4.4	Région d'intérêt . . . . .	79
4.5	Architecture du module de description 2D . . . . .	80
4.6	Architecture du module d'interprétation . . . . .	82
4.7	SATURNE : architecture du système . . . . .	87
4.8	Comparaison . . . . .	89
5.1	Architecture du modèle d'agent social . . . . .	107
5.2	Cycle de contrôle du modèle d'agent social . . . . .	108

5.3	Modèle de contrôle individuel . . . . .	124
5.4	Architecture d'une couche de contrôle du modèle de contrôle individuel	127
5.5	Répartition des états de décision et d'engagement sur les couches de contrôle . . . . .	129
5.6	Modèle de contrôle social . . . . .	134
5.7	Effets possibles d'une interaction . . . . .	136
5.8	Bascules sociales . . . . .	142
5.9	Modèle opératoire de contrôle social et individuel . . . . .	147
5.10	Modèle de contrôle individuel pour un système intégré de vision . . . . .	149
6.1	Architecture du Module Standard pour un Agent d'un Système Intégré de Vision . . . . .	167
6.2	Cycle de contrôle du Module Standard pour un Agent d'un Système Intégré de Vision . . . . .	168
6.3	Architecture du Module Individu pour un Agent d'un Système Intégré de Vision . . . . .	172
6.4	Liens entre les structures de l'architecture du Module Standard d'Agent	174
6.5	Architecture des capacités sociales du Module Standard pour un Agent d'un Système Intégré de Vision . . . . .	186
6.6	Système pour l'étude du contrôle individuel et social dans un Système Intégré de Vision . . . . .	196
6.7	Expériences de contrôle au sein de l'AGENT-DESCRIPTION-2D . . . . .	202
6.8	Protocole d'interaction descendant . . . . .	215
6.9	Exemple de contrôle social . . . . .	218
7.1	Perspective du modèle de contrôle social . . . . .	227

# Liste des Tables

3.1	Niveaux de représentation en Vision par Ordinateur . . . . .	41
3.2	Contrôle dans des Systèmes Intégrés de Vision . . . . .	59
3.3	Modes de Fonctionnement . . . . .	64
3.4	Types de fonctionnement . . . . .	68
3.5	Expression d'un centre d'intérêt privilégié dans l'organisation du système	70
5.1	Degrés d'une Interaction . . . . .	138
6.1	Valeurs permises pour la définition de la nature, de la force . . . . .	187
6.2	Valeurs permises pour le sujet . . . . .	214



# Liste des Exemples

6.1	Exemple d'HYPOTHESE . . . . .	175
6.2	Exemple de BUT . . . . .	176
6.3	Exemple de PLAN . . . . .	176
6.4	Exemple de CONFLIT . . . . .	177
6.5	Exemple d'INTENTION . . . . .	179
6.6	Exemple d'ENGAGEMENT . . . . .	180
6.7	Exemple de DONNEE . . . . .	189
6.8	Exemple d'ACCOINTANCE . . . . .	190
6.9	Exemple de rôle de résolution social . . . . .	216
6.10	Exemple de rôle de résolution individu . . . . .	216
6.11	Exemple de rôle de dialogue . . . . .	217

# Chapitre 1

## Introduction

### 1.1 Contexte de notre étude

Un système de vision *construit et maintient* une description interne de l'environnement à partir des informations fournies par ses capteurs. Dans le cadre de notre travail, nous nous intéressons plus particulièrement à la *perception visuelle* et donc à des capteurs de type caméra.

Imaginons un tel système dans une pièce où plusieurs objets sont présents, notamment une table sur laquelle sont placés des cubes. L'*objectif* de ce système est la surveillance des cubes placés sur cette table. Il décompose cette tâche en recherchant tout d'abord une table dans la pièce et en vérifiant ensuite la présence des cubes. Ce système dispose d'une description a priori des apparences possibles d'une table et de cubes dans une scène. Une vue d'ensemble de la pièce ne pouvant être obtenue, le système modifie sa position ainsi que celle de ses caméras en fonction des indications sur la position éventuelle de la table que lui fournit la *description* courante de la pièce. Pour toute nouvelle position, il exécute les traitements mettant en relation les *modèles* de table et de cubes avec les *données* perçues. Une fois la table trouvée, il réduit le champ de mouvement des caméras et les positionne sur le dessus de la table pour rechercher les cubes. Si ceux-ci ne sont pas trouvés, l'exploration de la pièce reprend. Dans l'autre cas, la position des caméras est stabilisée et le système met en place un fonctionnement de surveillance.

Dans cet exemple, le système de vision met en place un *fonctionnement* dans lequel

les *traitements* et la position des caméras sont contraints en vue de satisfaire l'objectif (trouver la table et les cubes, puis surveiller) étant donné l'évolution de la scène. Le système a ainsi mis en œuvre deux comportements : recherche de la table et des cubes, détection de modifications de la description de la scène une fois ceux-ci trouvés.

*Dans ce rapport, nous nous intéressons à l'ensemble des décisions conduisant à la définition de la position des capteurs, à la sélection et à la définition de l'enchaînement des traitements à mettre en œuvre, en fonction de l'évolution de l'environnement et des objectifs que le système doit satisfaire. Nous désignons par problème du contrôle l'ensemble des problèmes liés à la prise de ces décisions.*

### 1.1.1 Problème de contrôle

#### Contrôle

Plus généralement, nous appelons contrôle d'un système, l'instauration et la modification de son fonctionnement en appliquant des contraintes dépendant de l'objectif à poursuivre, de son état et de celui de l'environnement. Nous décomposons le problème de la définition du contrôle en quatre sous-problèmes : *décision*, *adaptation*, *commande* et *observation*.

- Le problème de *décision* concerne le choix d'un objectif en fonction de l'état de l'environnement, parmi l'ensemble des objectifs possibles du système.
- Le problème d'*adaptation* porte sur la définition des critères de sélection ou *loi de commande* à utiliser en fonction de l'objectif choisi.
- Etant donné un état de l'environnement, le problème de *commande* vise à déterminer l'action ou *commande* à exécuter dans l'environnement, à partir des critères définis.
- le problème d'*observation* concerne la sélection des éléments de l'environnement à prendre en compte pour la résolution des sous-problèmes précédents.

#### Structure de contrôle

Nous nous intéressons plus particulièrement à la *structure de contrôle* ou architecture du système, ensemble des mécanismes à la base de la résolution du problème de contrôle. Selon sa définition, la solution à chacun des sous-problèmes précédents est soit fixée lors de la conception du système, soit modifiable en fonction de l'évolution de l'environnement dans lequel le système opère.

La structure de contrôle peut ainsi être décomposée en trois *couches de contrôle* successives, ensemble de mécanismes, résolvant chacune respectivement le sous-problème de décision, d'adaptation et de commande. Une couche de contrôle définit ainsi les contraintes à appliquer au fonctionnement de la couche de contrôle inférieure. La résolution du problème d'observation se traduit par les échanges de données entre chacune de ces couches.

### 1.1.2 Domaine d'étude

#### Systeme intégré de vision

Dans cette thèse nous abordons le problème du contrôle dans le cadre de *Systèmes Intégrés de Vision par Ordinateur*. Un système intégré de vision combine en son sein plusieurs traitements dont leurs résultats sont inscrits sur des *niveaux de représentation*. Ceux-ci constituent une structure d'organisation de la description de la scène et des traitements. Ces derniers agissent d'un niveau à l'autre et au sein d'un même niveau pour construire la description. Ils sont nombreux et htrogues. Chacun est spécifique, par exemple, à un type de scène, à un type d'objet. Afin d'utiliser de tels systèmes dans des environnements différents, leur architecture doit être *ouverte* pour permettre l'ajout de nouveaux traitements.

Par ailleurs, afin de satisfaire les exigences actuelles par rapport à la vision par ordinateur, cette architecture doit être *flexible* afin de mettre en œuvre les fonctionnements suivants :

- *actif* en modifiant la position de ses capteurs, ou *passif*, dans ce cas les capteurs sont fixés,
- *dynamique* en traitant et acquérant continuellement de nouvelles informations, ou *ponctuel*,
- *réactif* en appliquant des réponses rapides, prédéfinies en réponse à un stimulus de l'environnement, ou *délibératif*, en mettant en œuvre des méthodes de recherche tenant compte de prédictions sur l'évolution de l'environnement et de l'historique des actions passées,
- *adaptatif* en modifiant ses paramètres de fonctionnement en fonction de l'évolution de l'environnement,
- *temps réel* en fournissant une réponse dans un temps borné,
- *téléonomique* c'est à dire fonctionnant selon un mode de *reconnaissance* ou selon un mode de *reconstruction*. Nous appelons *reconnaissance* le fonctionnement dans lequel les traitements du système sont orientés par la satisfaction d'un objectif précis. Le fonctionnement de *reconstruction* conduit à une description générale de la scène dans lequel le système de perception détermine lui-même ses propres objectifs à partir des informations qu'il perçoit dans la *scène*.

Les systèmes intégrés de vision actuels mettent en place une structure de contrôle dédiée à un mode de fonctionnement. Tout en étant indépendants et ouverts par rapport aux domaines d'application et par rapport aux connaissances, un seul type de fonctionnement est possible. Nous présenterons plus particulièrement un exemple d'un tel système: le système *Vision As Process* (VAP) <sup>1</sup>. Celui-ci aborde la vision selon les

---

<sup>1</sup>Ce système a été développé dans le cadre du projet Esprit BRA3038, BRA7108.

dimensions d'*intégration*, de *dynamisme* et la vision *active*. Le système *VAP* nous a servi de cadre d'étude du contrôle dans un système intégré de vision. Nous en avons dégagé les limitations et les caractéristiques à prendre en compte pour la définition d'une architecture de contrôle d'un système intégré de vision nous permettant d'étudier le contrôle d'un tel système. Notre approche est aussi influencée par le projet de recherche *SATURNE* dans lequel plusieurs méthodes d'inférence de forme interagissent au sein d'un même système pour reconnaître ou reconstruire la scène.

## Problématique du contrôle dans un système intégré de vision

Le problème de la définition du contrôle d'un système intégré de vision se formule par rapport aux niveaux de représentation. Il est lié à la recherche des conditions régissant le regroupement des traitements agissant sur un niveau en contrôleurs de traitements d'un niveau inférieur, aux règles spécifiant les interactions entre les traitements intra-niveaux et inter-niveaux. Par ailleurs la définition de la structure de contrôle d'un tel système concerne aussi la mise en place des mécanismes de décision et des commandes à utiliser pour contraindre le fonctionnement de chacun des traitements dans le système.

## 1.2 Thèse défendue

Nous proposons une *architecture* pour l'étude du contrôle dans le cadre d'un *système intégré de vision par ordinateur*.

Afin d'offrir une structure d'*intégration* de plusieurs traitements de vision et d'étude de l'expression de différents fonctionnements, l'architecture est basée sur la décomposition du système en *sous-systèmes*. Chacun d'eux est dédié à la construction de la description de la scène à un niveau de représentation donné, en privilégiant un centre d'intérêt. Ceux-ci se retrouvent d'un niveau de représentation à l'autre : région de la scène ou objet à interpréter se traduisant en représentations différentes selon le niveau considéré, méthode d'inférence de formes, par exemple.

Utilisant les modèles et techniques des systèmes multi-agents, chacun des sous-systèmes est exprimé en terme d'un *agent*, c.-à-d. un système capable de déterminer son propre fonctionnement et d'interagir avec les autres agents du système. Les trois couches de contrôle dédiées respectivement à la résolution des problèmes de *décision*, d'*adaptation* et de *commande* sont exprimées au sein de chacun d'eux afin d'offrir la flexibilité nécessaire à l'étude du contrôle. Chacun de ces agents interagit au travers des niveaux de représentation en utilisant les relations privilégiées inter-niveaux que définissent les centres d'intérêts. Les *protocoles d'interaction* définissent la gestion de ces interactions. Une *organisation* vient restreindre les interactions possibles et les traitements locaux entrepris par chacun des agents. L'organisation et les protocoles s'inscrivent dans un modèle de *société* dans lequel sont explicitées les trois couches de contrôle.

Ainsi, l'architecture proposée permet d'étudier au sein du même système, plus particulièrement, le contrôle nécessaire pour l'expression des modes de fonctionnement *délibératifs* et *réactifs* au sein de chacun des agents et, d'autre part, d'intégrer dynamiquement dans le système les modes de *reconstruction* et de *reconnaissance*.

### 1.3 Démarche suivie

La démarche suivie pour conduire notre étude du contrôle de la perception dans un système intégré de vision est illustrée par la figure 1.1. Les trois premières étapes (contrôle, contrôle en vision, propositions pour le contrôle d'un système intégré de vision) constituent la partie analytique de notre travail.

#### Définition des modèles

Prolongeant la décomposition d'un système de vision selon les niveaux de représentation, nous définissons des sous-systèmes par regroupement des traitements et des représentations situés à l'intersection des niveaux de représentation et d'une structuration transversale selon les centres d'intérêts apparaissant en cours de résolution ou lors de la conception. Cette structuration est le résultat d'une identité de focalisation des traitements d'un niveau de représentation à l'autre.

Nous nous tournons du côté de l'*Intelligence Artificielle Distribuée* (IAD) pour exprimer l'architecture de chacun des sous-systèmes et du système lui-même (étape de *définition des modèles* figure 1.1).

#### Modèle d'agent

Nous proposons ainsi un *modèle d'agent* pour la définition et la structuration des traitements au sein de chacune des entités.

Nous définissons un *agent* comme un système concret ou abstrait placé dans un environnement qu'il est capable de percevoir et sur lequel il peut entreprendre des actions. Par définition, un agent est *autonome* : il est capable de choisir ses propres objectifs.

#### Modèle de société

De manière similaire, nous définissons un *modèle de société* qui installe une structure générale ainsi que des règles d'interaction entre les agents au sein du système.

Nous définissons une *société* comme un ensemble d'agents pouvant entrer en interaction. La définition d'une société implique la définition d'un langage de communication, de liens et de protocoles d'interaction et d'une organisation.

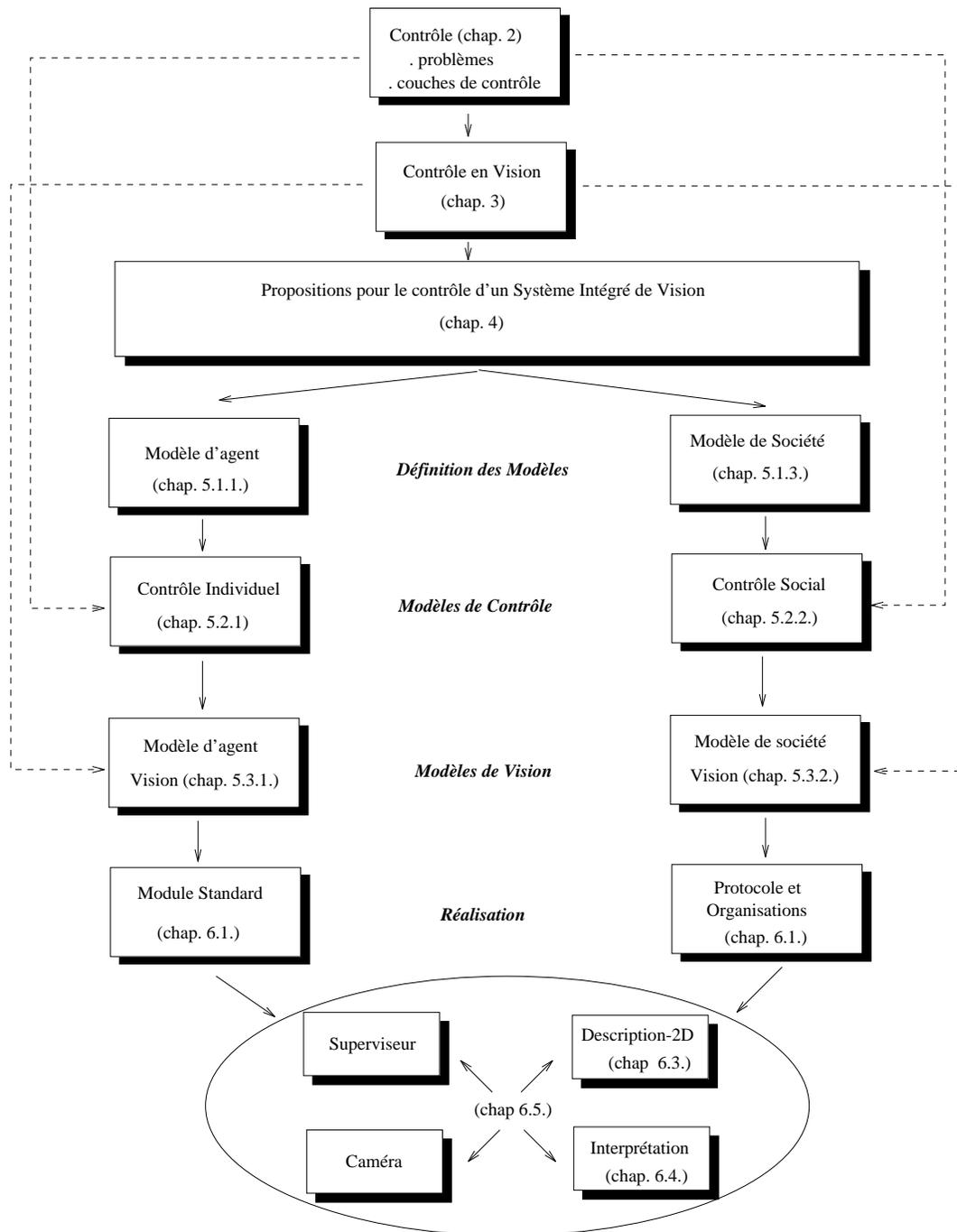


FIG. 1.1 - **Démarche suivie.** Les flèches pointillées montrent les interactions entre les chapitres. Les flèches en continu matérialisent l'organisation logique du rapport.

## Modèles de contrôle

Nous appuyant sur l'étude du chapitre 2, nous explicitons, dans chacun des modèles, les trois couches de contrôle et exprimons les mécanismes nécessaires à leur réalisation (*modèle de contrôle* figure 1.1).

Notre étude du contrôle est décomposée selon deux axes : *contrôle individuel* et *contrôle social*. Le contrôle individuel étudie les éléments permettant de modifier le comportement de l'agent. Le contrôle social étudie la sélection des interactions prenant place entre les agents ainsi que les relations entre les agents.

## Modèles de vision

Tirant parti de l'étude réalisée dans le chapitre 3, nous instancions ensuite ces modèles au cadre de notre étude : un système intégré de vision (*modèle de vision*, figure 1.1). Les modèles ainsi obtenus nous permettent d'étudier le contrôle dans un système intégré de vision.

## Réalisation

Ces modèles ont donné lieu à la réalisation : 1) d'un module standard d'agent dans lequel les traitements d'un système intégré de vision sont écrits, 2) d'un squelette générique d'intégration et d'interaction servant de base pour l'intégration de différents agents dans le système (*réalisation*, figure 1.1).

La validation de notre étude s'est déroulée en utilisant et en reformulant dans le module standard certains modules du système *VAP*.

# 1.4 Organisation du rapport

L'organisation du rapport suit la démarche que nous avons suivie, illustrée par la figure 1.1. Les flèches pointillées montrent les interactions entre les chapitres. Les flèches en continu matérialisent l'organisation logique du rapport. De nombreuses définitions sont introduites dans ce rapport. Un index en fin d'ouvrage permet au lecteur de retrouver plus facilement chacune des notions introduites.

## Chapitre 2 : Contrôle

Nous présentons dans un premier temps les définitions de contrôle relatives à différents domaines de recherche, proches de notre domaine d'application. Nous nous intéressons ainsi plus particulièrement aux domaines de l'automatique, de la théorie des systèmes hiérarchiques, à l'intelligence artificielle et à l'intelligence artificielle distribuée. De cette étude, nous proposons la *définition du contrôle* constituant l'armature de notre thèse. A chaque fois que nous utiliserons la notion de contrôle dans l'étude qui suit, nous nous rapporterons à cette définition.

Nous distinguons quatre sous-problèmes dont la résolution conduit à la définition du contrôle d'un système : problèmes de commande, d'adaptation, de décision et d'observation. Selon la structure de contrôle du système, nous pouvons mettre en évidence des couches de contrôle correspondant chacune à la résolution explicite d'un des problèmes de commande, d'adaptation ou de décision.

Nous complétons ce chapitre en définissant les termes concernant les mécanismes et les connaissances pour la réalisation du contrôle dans notre système.

### **Chapitre 3 : Contrôle dans les Systèmes Intégrés de Vision**

Dans ce chapitre, nous donnons une description du domaine de validation de notre étude sur le contrôle. Après diverses définitions, nous exposons les problèmes auxquels se trouve confrontés tout concepteur d'un système intégré de vision. Des exemples représentatifs de tels systèmes illustrent diverses solutions actuelles. A l'issue de cet état de l'art, nous mettons en avant les problèmes liés au contrôle de tels systèmes en utilisant la définition et la démarche d'analyse présentées dans le chapitre précédent.

### **Chapitre 4 : Propositions pour le contrôle d'un Système Intégré de Vision**

A partir de la problématique précédente, nous avons réalisé différentes expériences de contrôle dans le cadre du système *VAP*. Nous les reportons brièvement dans ce chapitre. Ces expériences nous amènent à élaborer un ensemble de propositions. Celles-ci sont alimentées aussi par le projet de recherche *SATURNE* auquel nous nous sommes intéressés. La problématique de ce système relève plus de l'intégration que du contrôle. Tirant les conclusions des limitations de ces deux systèmes nous formulons les principes de base de notre architecture d'intégration et d'étude de contrôle.

Ainsi en utilisant la structuration qu'offrent les niveaux de représentation, nous séparons les exigences de contrôle du système en deux points de vue considérant alternativement les entités de base ou *agents* constituant le système et le système complet, *Société* d'agents. Cette distinction décompose notre étude du contrôle en une étude locale : étude des aspects du contrôle au sein de chacun des agents (contrôle individuel), et en une étude globale : étude du contrôle des interactions au sein de la société entre les agents (contrôle social).

Nous introduisons l'*Intelligence Artificielle Distribuée*, domaine d'étude et de formalisation de ces notions d'agents et de société.

### **Chapitre 5 : Modèles pour l'étude du contrôle dans un Système Intégré de Vision**

Nous présentons dans ce chapitre les modèles utilisés pour notre étude du contrôle dans un système de vision.

Nous proposons un *modèle d'agent* pour la définition et la structuration des traitements au sein de chacun des agents. Ce modèle met clairement en évidence les facteurs

permettant de modifier le comportement de résolution de l'agent selon les trois couches de contrôle. Ces éléments nous permettent d'étudier le contrôle au sein de chacun des agents.

Notre problématique nous amène à nous intéresser aux interactions des différents agents au sein de la société. Nous nous posons ainsi la question de la répartition des différentes tâches de résolution, de la régulation des échanges. Nous définissons un *modèle de société* en explicitant le contrôle des interactions entre agents à partir des trois couches de contrôle et en proposant un ensemble d'éléments relatifs à l'étude de ces différents aspects : protocoles d'interaction et organisation.

Nous particularisons les deux modèles ainsi obtenus en tirant parti de notre problématique du contrôle dans un système intégré de vision.

## Chapitre 6 : Réalisation

Nous rapportons dans ce chapitre l'implantation de l'architecture du module standard définie à partir des modèles du chapitre précédent. Ce module est utilisé pour reformuler des traitements du système Vision As Process *VAP*. Les modules obtenus sont utilisés ensuite pour tester différents aspects du contrôle au sein de chacun des agents et pour tester différents protocoles et organisation permettant ainsi de montrer l'intérêt de l'architecture que nous avons construite pour l'étude du contrôle de la perception.



## Chapitre 2

# Contrôle

Ce chapitre a pour but de préciser la *définition du contrôle* que nous utiliserons tout au long de ce rapport.

Dans un premier temps, nous recherchons et comparons les notions de contrôle, satisfaisant la définition suivante : *Le contrôle établit et modifie le fonctionnement d'un système en appliquant des contraintes dépendant de son état, en vue de poursuivre un objectif donné.* Un *système* est un ensemble de traitements relié au monde extérieur uniquement par des entrées/sorties. Les *entrées* caractérisent les actions du monde sur le système, les *sorties* définissent les actions de celui-ci sur le monde ou environnement. Nous montrons que des approches similaires se dégagent dans les domaines de l'*Automatique*, de l'*Intelligence Artificielle* et de la *Psychologie cognitive*.

Dans la deuxième section de ce chapitre, nous utilisons ces points communs pour définir et préciser la problématique du contrôle que nous envisageons. Nous élaborons ainsi notre propre définition qui servira de cadre général à notre étude du contrôle dans un système de vision.

### 2.1 Définitions de contrôle

Dans cette étude bibliographique sur la notion de contrôle nous nous intéressons plus particulièrement aux domaines de l'*Automatique*, de l'*Intelligence Artificielle* et de la *Psychologie Cognitive*. Notre intérêt pour les deux premiers domaines est lié à leur

utilisation dans la construction de systèmes de Vision par Ordinateur.

Après la présentation du contrôle dans le domaine de l'Automatique où une formalisation de cette notion est disponible, nous étudions la *théorie des systèmes hiérarchiques*. Dans celle-ci, le contrôle entre plusieurs systèmes est abordé sous l'angle de la théorie précédente. Ces deux théories offrent des méthodes d'analyse et de synthèse des systèmes de contrôle, adaptées à des problèmes essentiellement modélisés en termes d'équations aux différences finies ou de systèmes d'équations différentielles.

Dans certains problèmes, cependant, une formalisation logique est plus adéquate pour représenter le système dynamique à contrôler. Nous abordons ainsi l'étude du contrôle par *planification* qui nous conduit directement au domaine de l'*Intelligence Artificielle*. Nous nous tournons ensuite vers l'*Intelligence Artificielle Distribuée*. Ces deux domaines abordent, avec des techniques différentes, la même problématique que la théorie de la commande et la théorie des systèmes hiérarchiques : le contrôle au sein d'un système et le contrôle entre plusieurs systèmes.

Des expériences de *Psychologie Cognitive* sont souvent citées comme sources d'inspiration des études sur le contrôle, notamment en Intelligence Artificielle et en Vision par Ordinateur. C'est la raison pour laquelle nous incluons également ce domaine dans notre tour d'horizon.

### 2.1.1 Automatique

En automatique, deux types de systèmes sont distingués : le *contrôleur* et le contrôlé appelé *processus*. Par exemple considérons un processus constitué d'un bassin alimenté par une valve d'alimentation à débit variable. Son évacuation est assurée par une valve d'écoulement à débit constant non modifiable. Le contrôleur de ce processus règle la valve d'alimentation pour que la hauteur d'eau dans le bassin soit constante (cf. figure 2.1).

#### a. Processus et contrôleur

##### Processus

Un état du processus est décrit sous la forme d'un *vecteur d'état*, ensemble de *variables* exprimant les grandeurs caractéristiques de l'action du processus sur l'environnement. Ces variables peuvent avoir des échelles temporelles différentes. Dans le cadre de notre exemple, le vecteur d'état est constitué de la position  $\theta(t)$  de la valve d'alimentation et de la hauteur du fluide  $h(t)$  à l'intérieur du bassin. Les *perturbations* sont les entrées du processus qui ne sont pas maîtrisables. C'est, par exemple, la pluie tombée dans le bassin. Elles s'opposent aux *actions* ou *commandes*, correspondant aux entrées définies par le contrôleur. La commande dans notre exemple, est le changement de position de la valve d'alimentation : ouvrir ou fermer en tournant la valve d'un angle donné.

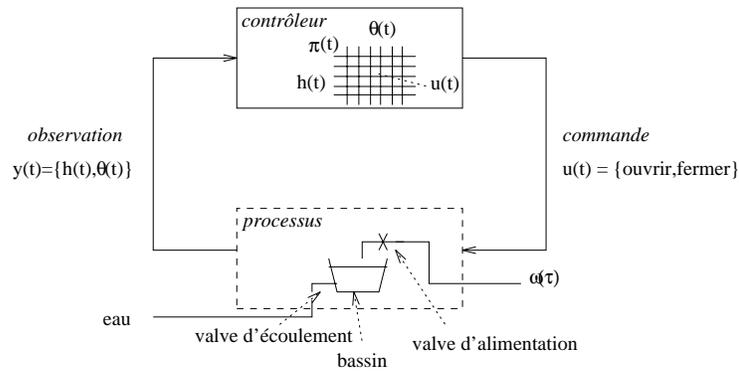


FIG. 2.1- **Contrôleur et Processus en Automatique.** Un contrôleur définit la modification à apporter à la valve d'alimentation d'un bassin afin que la hauteur d'eau soit constante. Les observations sont définies par la hauteur d'eau mesurée  $h(t)$  et la position de la valve  $\theta(t)$ . La loi de commande  $\pi(t)$  est exprimée par une table de décision faisant correspondre une valeur de commande pour les observations.

## Contrôleur

Ce système engendre les *commandes* appliquées en entrée du processus à partir d'un sous-ensemble du vecteur d'état, appelé *observations* et d'un *modèle* du fonctionnement du processus. L'utilisation des observations fait référence à un type de contrôle appelé *contrôle par rétroaction* (feedback control). Les commandes sont sélectionnées par rapport à un *objectif*. Celui-ci est exprimé par des contraintes que doivent *satisfaire* les variables du vecteur d'état ou en fonction d'*évaluations* exprimant des critères d'optimisation ou de stabilité. Dans le cadre du bassin, un modèle du processus est donné par les équations reliant la variation de hauteur avec le flux de sortie et le flux d'entrée. L'objectif peut être de garder  $h(t)$  constante. Les observations sont la hauteur d'eau du bassin  $h(t)$  et la position de la valve d'alimentation  $\theta(t)$ .

### b. Problème du contrôle

Nous pouvons ainsi formuler le problème du contrôle dans cette théorie comme :

◇ **Définition :** *Le problème du contrôle en Automatique.*

En Automatique, le *problème du contrôle* est relatif à la *définition de la commande* à appliquer au processus, à partir de son *modèle* et d'*observations* sur son état. Cette commande vise à obtenir un état du processus qui satisfait ou optimise des *objectifs* relatifs à son fonctionnement.

Le problème du contrôle est résolu par la formulation d'une *loi de commande*  $\pi$ . Celle-ci [Koh 91] est définie comme une fonction de l'ensemble des observations  $Y$ , des

objectifs  $O$ , des commandes admissibles<sup>1</sup>  $U$ . Une loi de commande engendre une commande  $u$  appliquée au processus. Elle traduit les objectifs du système en un ensemble de commandes.

$$u = \pi(O, Y, U)$$

La modélisation du processus joue un rôle important dans la formulation de cette loi. Son champ d'action est ainsi restreint aux classes de systèmes satisfaisant cette modélisation. Dans le cadre du contrôleur de bassin (cf. figure 2.1), la loi de commande peut consister en une table de correspondances entre  $h(t)$  observée et la position de la valve d'alimentation à l'instant  $t$ ,  $\theta(t)$ , résultat de la dernière commande. Cette table propose une commande *ouvrir* ou *fermer*. Elle est obtenue par essais exhaustifs préalables en se fondant sur la prédiction de l'évolution de  $h(t)$  donnée par le modèle du processus.

### c. Mécanismes de base

Souvent le mécanisme du contrôleur est décomposé en deux fonctions relatives respectivement à l'*estimation* de l'état du processus à partir des observations et à la *régulation* correspondant à la génération des commandes proprement dites. Ces mécanismes ne sont pas toujours explicitement séparés [Dea 91].

Le mécanisme de régulation étant réalisé, souvent, par une modélisation mathématique, une représentation explicite des commandes potentiellement applicables étant donné l'état du processus est absente dans ce mécanisme. La commande est le résultat d'un *calcul*.

### d. Conclusion

- En Automatique, deux systèmes sont considérés : le contrôleur et le processus. Le contrôleur agit sur un processus qui lui est complètement extérieur.
- Le problème du contrôle concerne la définition de la *commande* à appliquer au processus étant données les observations. Il est résolu par la définition d'une *loi de commande* qui intègre dans sa formulation des objectifs (fonctions à optimiser ou contraintes à satisfaire).
- Aucun mécanisme de base générique ne se dégage au sein du contrôleur. L'automatique fournit des méthodes pour l'analyse et la synthèse de systèmes de contrôle essentiellement utilisés pour des problèmes modélisés par des outils mathématiques.

Malgré quelques variations au niveau de la modélisation du processus<sup>2</sup>, la complexité des modèles et/ou des commandes rendent les solutions de la théorie classique

<sup>1</sup>Ensemble des commandes satisfaisant les contraintes physiques imposées au contrôleur.

<sup>2</sup>La théorie conventionnelle du contrôle est fondée sur des modèles mathématiques décrivant le comportement dynamique des systèmes contrôlés. Ceux-ci sont modélisés par un ensemble d'équations différentielles linéaires ou non, dérivées selon différentes approximations et simplifications. Depuis

du contrôle difficilement adaptables [Sho 91a]<sup>3</sup>. Le contrôleur possède des *objectifs et des lois de commande fixés* [Sho 91b]. Ces approches sont mises en défaut dans le cas de systèmes *autonomes* qui, afin de répondre aux évolutions de l'environnement, doivent définir eux-mêmes leurs propres objectifs en fonction de leurs observations, de leur expérience et aussi, par exemple, d'objectifs imposés par d'autres systèmes.

Ainsi une nouvelle approche, le *contrôle intelligent*, est apparue récemment. Dans celle-ci, les contrôleurs utilisent des connaissances et des mécanismes de raisonnement pour régir l'état courant du processus et pour détecter les changements intervenus dans l'environnement [Shi 91]. Ils peuvent utiliser des modèles quantitatifs ou qualitatifs des comportements des processus. Des capacités de perception leur permettent de les affiner. Nous présenterons cette approche dans le cadre de l'*Intelligence Artificielle*.

### 2.1.2 Théorie des systèmes hiérarchiques

La théorie des systèmes hiérarchiques [Mes 70] introduit la décomposition conceptuelle du contrôleur en sous-systèmes qu'elle organise selon une *hiérarchie de contrôle* : chaque sous-système contraint le fonctionnement des sous-systèmes de la couche inférieure, qui, en retour, lui renvoient des informations qu'il traduit et communique au niveau supérieur. Pour chaque paire de sous-systèmes situés sur des couches successives, on retrouve le problème du contrôle exprimé dans la théorie précédente : le contrôleur est le sous-système situé sur la couche supérieure, le processus est celui situé sur la couche inférieure. Les sous-systèmes d'une couche engendrent des *interventions* sur ceux du niveau inférieur. Les sous-systèmes sur la couche la plus basse de la hiérarchie engendrent les *commandes* du processus. Opposé à ce flux descendant d'informations, existe un flux ascendant : les *observations* (cf. figure 2.2). A priori, il n'existe aucune relation transversale, c.-à-d. pour une même couche de la hiérarchie. Ces échanges transitent par le sous-système de la couche supérieure.

#### a. Coordination

Le contrôleur étant constitué de plusieurs sous-systèmes, le problème du contrôle se reformule en termes de *problème de la coordination*, extension du problème du contrôle de la théorie précédente. L'intervention d'un sous-système sur un autre peut être un échange de commandes mais aussi d'objectifs ou de lois de commande. Ces types d'échanges correspondent à la *coordination* proprement dite. Le *mode de coordination* définit les interventions possibles entre les sous-systèmes et les problèmes abordés par chacun d'eux. La notion de *cohérence* vient contraindre la résolution du problème de

---

quelques années la théorie du contrôle délaisse ces systèmes ayant des lois d'évolution temporelle continues et se tourne vers les systèmes dont la dynamique est régie par l'apparition d'événements (*Discrete Event Dynamic Systems* (DEDS) se reporter à [IEE 89] pour en avoir un aperçu).

<sup>3</sup>Certains systèmes, *adaptive systems*, mettent cependant en œuvre une adaptation du modèle du processus en fonction des observations. Les commandes émises sont ainsi adaptées à l'évolution du système.

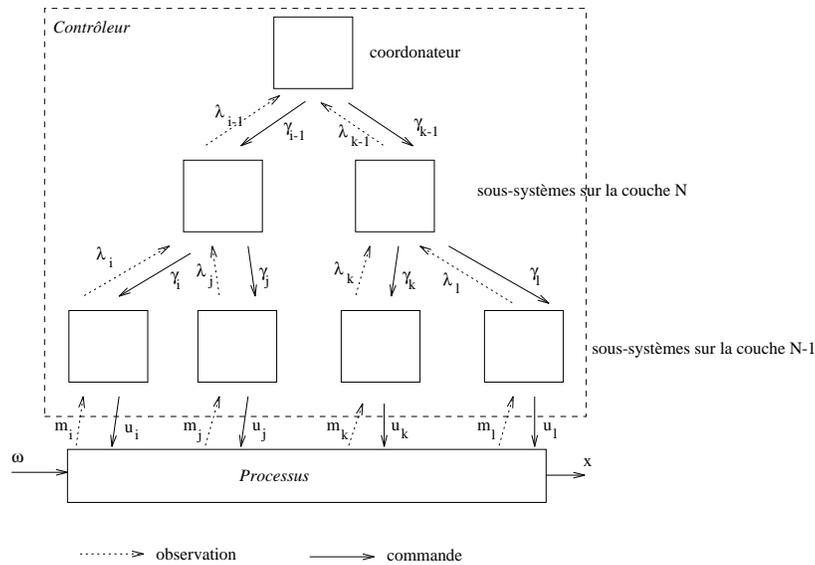


FIG. 2.2 - Contrôle dans la théorie des systèmes hiérarchiques. Un exemple d'un tel type de contrôle est celui de l'organisation régissant une entreprise. Sur la figure, les lettres  $u$  indiquent les commandes du contrôleur sur le processus. Les interventions entre les sous-systèmes sont représentées par la lettre  $\gamma$ . Commandes et interventions correspondent aux flèches en trait plein. Les observations en provenance d'un sous-système ( $\lambda$ ) et en provenance du processus ( $m$ ) sont représentées par les flèches en pointillés.  $\omega$  est une perturbation entrant dans le processus.  $x$  est l'état du processus.

coordination en imposant que la définition des objectifs, des lois de commande et des commandes s'inscrive dans une direction commune au niveau du système.

◇ **Définition :** *Le problème du contrôle dans la théorie des systèmes hiérarchiques.*

Dans la théorie des systèmes hiérarchiques le problème du contrôle consiste, d'une part à décomposer le contrôleur d'un processus en une hiérarchie de sous-systèmes et, d'autre part, à définir le *mode de coordination* et les échanges de *coordination* entre ces sous-systèmes qui débouchent sur la génération de commandes sur le processus. Le *mode de coordination* définit l'ensemble des interactions possibles entre les sous-systèmes. Ainsi, selon le mode, une *interaction* est un échange d'objectif, de loi de commande ou de commande entre sous-systèmes (échange de coordination ou intervention). La contrainte de *cohérence*, complétant l'objectif du contrôleur, s'ajoute aux contraintes appliquées sur la définition des interactions dans le but de faire progresser le contrôle du processus dans une direction commune à l'ensemble des sous-systèmes.

Plusieurs *modes de coordination* sont possibles. La coordination par *prévision* est un cas extrême. Dans celle-ci, les sous-systèmes d'une couche sont complètement sous la dépendance du contrôleur situé sur la couche supérieure. La coordination par *coalition* est le cas opposé. Dans celle-ci, les échanges entre les sous-systèmes d'une même couche sont autorisés et l'influence du contrôleur est moindre.

## b. Conclusion

- La notion de contrôle est élargie en affinant la définition du contrôleur en une hiérarchie de sous-systèmes.
- De cette manière plusieurs couches de contrôle sont définies sur lesquelles un contrôleur définit l'intervention à appliquer sur le sous-système de la couche inférieure en fonction des observations reçues. Sur la couche la plus basse, l'intervention est la commande du processus.
- La palette d'influences du sous-système d'une couche de contrôle sur l'autre s'élargit. En plus d'échanges de commandes, elle inclut la possibilité d'échange de lois de commandes et d'objectifs.
- Le problème du contrôle est celui de la coordination. Il consiste à définir les interactions possibles (mode de coordination) entre les sous-systèmes en termes de commandes, de lois de commandes et d'objectifs en vue de satisfaire le critère de cohérence.

### 2.1.3 Intelligence Artificielle

#### a. Contrôle par planification

En restant dans la même problématique du contrôle que celle de l'Automatique, l'Intelligence Artificielle a introduit une modélisation logique du processus qui lui per-

met d'aborder d'autres types de problèmes que ceux de cette théorie. En se fondant sur cette modélisation du processus, la commande, appelée *action*, est obtenue par un raisonnement symbolique appelé *planification* d'actions. La spécification des actions que doit exécuter le processus, en fonction d'un objectif ou *but*, est appelée : *plan*. Par analogie avec la théorie du contrôle, le plan correspond à une loi de commande, sauf que des informations symboliques plutôt que numériques sont manipulées.

Dans l'exemple classique du monde des blocs, le processus à contrôler est la main du robot déplaçant et empilant des cubes  $A, B, C$  sur une table. Le contrôleur est le planificateur. Le plan consiste en une suite d'actions du type :

$$(\textit{poser } A \textit{ } B)(\textit{prendre } C) \dots$$

La modélisation du processus passe par la définition des actions qu'il peut prendre dans l'environnement<sup>4</sup>, et par la modélisation de son état : par exemple un ensemble de listes exprimant des relations entre les cubes du type  $(SUR \ A \ B)$ , l'état de la main  $(LIBRE \ main)$ , etc. L'objectif exprime, en rapport avec cette modélisation, une description de ce que devra satisfaire l'environnement :  $(SUR \ A \ C)$ , c.-à-d. poser le cube A sur le cube C.

Selon l'expression du plan, l'enchaînement des actions peut être *séquentiel*, *conditionnel*, *itératif*, ou *récurif*. Le plan peut être aussi plus ou moins précis quant à la spécification de l'ordre des actions. On parle alors de *plan partiellement ordonné*.

◇ **Définition** : *Le problème du contrôle par planification*

Le problème du contrôle par un système de planification consiste en la construction et la modification d'un *plan*, séquence d'*actions* appliquées sur l'environnement, en vue de satisfaire un *but*.

Pendant longtemps, la planification a été réalisée hors ligne, complètement dé耦lée de l'exécution du plan produit. Celui-ci était construit à partir d'une modélisation précise de l'évolution du monde. Il ne pouvait être exécuté dans l'environnement que sous de très fortes contraintes répondant à celle-ci. On rejoint ainsi le cas de l'automatique où la loi de commande ne peut pas être modifiée en cours de contrôle du processus.

Certains systèmes de planification ont introduit le niveau supplémentaire de *suivi d'exécution* en leur sein, en vue de réaliser d'une part une construction du *plan* en connexion avec l'environnement, et, d'autre part, un amendement dynamique du *plan* tenant compte de l'évolution de ce dernier. Nous voyons apparaître ainsi la possibilité d'une *adaptation* dynamique de la loi de commande du système.

Dans ce qui suit, nous nous intéressons plus particulièrement au contrôleur c.à.d. au mécanisme qui construit le plan d'action. Dans la plupart des cas, l'activité de construction d'un plan, est, elle-même, fortement combinatoire. Elle est souvent envisagée comme une activité de résolution de problèmes. Dans ce cas, la problématique

---

<sup>4</sup>Plusieurs formalismes sont utilisables, par exemple par une liste de conditions, d'ajouts et de retraites comme en *STRIPS*[Fik 71]

du contrôle est relative à la réduction de l'espace de recherche des différents plans possibles, étant donné une description de l'environnement.

## b. Contrôle en résolution de problèmes

En Intelligence Artificielle, une résolution de problèmes peut être abordée selon trois points de vue [Sim 83]: (i) *recherche* dans un espace d'état <sup>5</sup> par des opérateurs, (ii) raisonnement logique par *déduction logique* sur des propositions et selon des modes d'inférences, (iii) *satisfaction de contraintes* consistant en la réduction progressive de l'ensemble des solutions possibles. Dans chacune de ces approches des choix apparaissent à tout moment de la résolution. Ce sont respectivement: (i) l'état à explorer et l'opérateur à utiliser, (ii) la proposition à considérer et le mode d'inférence à employer et enfin (iii) les contraintes à satisfaire et les éléments de solution sur lesquels les appliquer. Ces choix rendent les problèmes souvent NP-complets. L'*objectif*, que le contrôle du système cherche à satisfaire, consiste à trouver une solution au problème en évitant une *explosion combinatoire*.

Amarel a montré l'importance de la représentation d'un problème pour sa résolution [Ama 68]. Etant donnée celle-ci, il est nécessaire de pouvoir contraindre la résolution elle-même [Smi 86a].

Selon le premier point de vue sur la résolution de problèmes, le problème du contrôle consiste à fournir une méthode efficace d'exploration de l'espace d'états qui conduise à une solution. Cette méthode détermine les opérateurs nécessaires, en appliquant des opérations de sélection, voire de planification des séquences d'opérateurs [Hr 85], pour trouver une solution. Très souvent dans la littérature, la présentation de la problématique du contrôle est ramenée à un ensemble de prises de décisions en vue de déterminer l'*ordonnancement* des actions de résolution: le moment de la résolution et l'élément sur lequel appliquer une action, en vue de satisfaire et optimiser un ou plusieurs critères donnés.

◇ **Définition :** *Le problème du contrôle en résolution de problème.*

En résolution de problèmes, le problème du contrôle consiste à déterminer la prochaine opération à appliquer afin d'atteindre une solution en évitant une explosion combinatoire.

Deux aspects se conjuguent ainsi au sein du même système: construction de la solution au problème et contrôle de cette construction. L'évolution des langages de programmation mis en œuvre pour la construction de systèmes de résolution de problème met en évidence une telle dichotomie.

---

<sup>5</sup>Ensemble de symboles relatifs à la solution en construction.

### c. Structures de contrôle

D'un point de vue réalisation, parallèlement à l'accroissement de la dépendance par rapport au domaine d'application des connaissances utilisées<sup>6</sup>, des niveaux de réalisation du contrôle sont apparus dans l'implantation des systèmes de résolution de problèmes :

- *Langages déclaratifs* (ex: *PROLOG* ou *OPS5* par exemple).  
Avec l'introduction de la distinction entre connaissances et mécanisme d'exploitation, le contrôle a pour objet de décider à tout moment de la prochaine connaissance à appliquer étant donné un état de résolution. Par opposition, dans les langages procéduraux, le contrôle est résolu à la conception de l'algorithme en appliquant les instructions de contrôle appropriées pour la définition du séquençement des traitements.
- *Niveau Meta et Niveau Objet*.  
Poursuivant cette séparation, la résolution du problème en tant que telle (*niveau objet*) est distinguée de la détermination des critères de sélection qui la guident (*niveau meta*). On débouche ainsi sur l'expression déclarative des critères de sélection des actions alors que ceux-ci étaient inscrits auparavant dans le mécanisme d'exploitation [Wey 80] [Gen 83] [Dav 80b] [Dav 80a]. En termes réducteurs, c'est l'expression déclarative de la loi de commande.
- *Raisonnement dédié au contrôle*.  
D'autres systèmes sont allés plus loin dans la généralisation du mécanisme d'exploitation permettant ainsi d'étoffer le raisonnement dédié au contrôle. On peut citer dans cette démarche le *Meta Reasoning System* (MRS) [Gen 83] [Rus 85], *Procedural Reasoning System* (PRS) [Geo 87] [Geo 89]. *BB1* [Hr 85] élargit cette démarche en gérant explicitement un autre espace de recherche dédié à la résolution du problème du contrôle, de manière permanente alors qu'il n'apparaît qu'en cas d'impasse dans *SOAR* [Lai 87].

Dans les mécanismes d'exploitation de la plupart de ces systèmes, des fonctions de base sont isolées : *détection* des conflits, *sélection* et *exécution*. La détection des conflits détermine les connaissances applicables sur un état de la résolution (construction de l'ensemble de conflits). La sélection choisit parmi celles-ci une connaissance à appliquer (résolution de conflit). L'exécution engendre un nouvel état en interprétant cette connaissance.

### d. Conclusion

- En résolution de problèmes, le contrôle concerne l'activité du système lui-même.  
Le problème du contrôle consiste à déterminer l'action de résolution à appliquer

---

<sup>6</sup>Lors de l'apparition des premiers systèmes, les connaissances utilisées étaient surtout indépendantes du domaine (weak methods)[Lai 83]. Ce sont la stratégie de l'ensemble support, l'analyse des fins et des moyens, la recherche en profondeur ou en largeur [Nil 82].

étant donné un état d'avancement de celle-ci et un objectif à satisfaire. La sélection de l'action est résolue par l'application de connaissances dépendantes du domaine d'application. L'objectif peut s'exprimer en termes intrinsèques, c.-à-d. limitation de l'utilisation des ressources et en termes propres au problème, correspondant à l'expression de l'état de résolution à atteindre.

- Selon l'implantation des systèmes, deux couches de contrôle sont définies : une couche détermine les critères de sélection qu'utilisent la couche inférieure pour déterminer l'action à appliquer. Des mécanismes de base sont mis en évidence : la détection de conflits et la résolution de conflits.

### 2.1.4 Intelligence Artificielle Distribuée

La problématique du contrôle a été poussée plus loin dans le domaine de l'informatique avec les problèmes rencontrés au sein des systèmes répartis et des systèmes d'*Intelligence Artificielle Distribuée* (IAD) [Gas 88] [Dem 90b] [Dem 91b] [Erc 91] [Boi 90].

L'approche IAD que nous développerons plus en détail dans le chapitre 4, étudie et résout les problèmes rencontrés lorsque plusieurs systèmes ou *agents* interagissent pour la résolution d'un même problème<sup>7</sup> (*Résolution Distribuée de Problèmes*) ou parce qu'ils sont dans le même environnement<sup>8</sup> (*Multi-Agents*).

#### a. Contrôle individuel

Un agent est, comme nous l'avons défini dans l'introduction, un système concret ou abstrait capable de percevoir son environnement sur lequel il peut entreprendre des actions. Pour chaque agent, le problème du contrôle relatif à sa propre activité est le même que celui évoqué précédemment pour un système de résolution de problèmes excepté que l'évolution de l'environnement doit être prise en compte. L'agent est inséré dans un environnement qu'il peut percevoir mais dont il ne maîtrise pas a priori l'évolution.

◇ **Définition :** *Le problème du contrôle individuel.*

Le problème du contrôle au sein d'un agent ou contrôle individuel consiste à déterminer quelle action exécuter à partir de son état de résolution et de l'état de l'environnement.

La résolution de ce problème de contrôle a conduit à l'apparition de deux courants au sein de la conception des agents : les agents *délibératifs* et les agents *réactifs*.

### Délibération

L'approche *délibérative* utilise une représentation interne des opérations, des buts et de l'état de l'environnement. Pour tout nouvel état, elle met en œuvre une planification,

<sup>7</sup>Par exemple un ensemble de systèmes d'interprétation de données répartis à un carrefour routier.

<sup>8</sup>Un ensemble de véhicules routiers par exemple.

afin de déterminer les séquences d'opérations à lui appliquer. Pour tout nouvel état de l'environnement, un agent *délibératif* résout le problème du contrôle en prenant en compte une prédiction sur l'évolution future de l'environnement (plan) et un examen des actions passées pour sélectionner une action à exécuter.

## Réaction

Une autre approche liée aux systèmes *réactifs*, renforce la liaison entre la perception de l'environnement et les actions entreprises sur celui-ci. Ainsi la résolution du problème du contrôle dans un agent réactif ne cherche pas à prédire les situations futures possibles pour déterminer la réponse à une évolution de l'environnement. Celle-ci est immédiate. La sélection des actions est résolue par la suppression, à la conception, des conflits entre actions applicables à une situation donnée (*REX* [Kae 86]), ou par un précodage des mécanismes de sélection au sein de l'agent (*Subsumption Architecture* [Bro 85], *PENGI* [Agr 87], *SONJA* [Cha 90] [Mae 90]).

### b. Contrôle social

Un agent inséré dans une société doit inscrire son activité dans la résolution de celle-ci. Les agents étant *répartis*, une vision globale de l'ensemble de l'activité leur est impossible. Des systèmes de surveillance du trafic routier, par exemple, répartis sur un carrefour ne peuvent pas, sans diminution de leur performance, avoir une vision globale de la circulation : en effet, ne pouvant capter qu'une région du carrefour, ils ne peuvent se construire une vision globale que par communication avec les autres agents. Ceci se déroule au dépend de leur activité de surveillance. D'autre part, les informations qu'ils peuvent avoir ainsi reçues sont rapidement invalidées impliquant une continuelle remise à jour.

Lors de la surveillance du trafic par ces agents, chacun d'eux a reçu une région limitée à surveiller. Le problème de contrôle au niveau de l'agent peut donc être plus facilement résolu. Cependant de nouvelles contraintes apparaissent. En effet, quelle doit être la distribution des différents problèmes à résoudre (sélection des agents et des régions)? Dans notre exemple, ce problème est résolu lors de la conception, dans d'autres cas cette distribution peut être réalisée dynamiquement. Un autre problème est celui des échanges entre les agents : la solution est-elle construite indépendamment par chacun des agents et intégrée à la fin? Ou des échanges réguliers ont-ils lieu entre les agents afin de réduire les incohérences possibles?

Dans l'exemple précédent, les agents sont conçus pour résoudre ce problème de surveillance du trafic routier. Plaçons-nous maintenant dans un cadre général où des agents possédant des capacités plus importantes sont dans un même environnement : des robots se déplaçant dans un couloir reliant différentes pièces. Ce couloir ne permet pas le passage de deux robots de front. Dans cet exemple aussi, une vision globale de l'état de l'environnement est exclue. Nous voyons apparaître un problème de partage de ressources relatif au passage dans le couloir. Les agents voulant emprunter ce couloir

vont donc devoir se *coordonner* pour que la circulation puisse se réaliser sans blocage.

Nous mettons ainsi clairement en évidence une dimension qui était absente dans notre exemple précédent : les agents prennent dynamiquement en compte le but commun qui consiste à résoudre le problème de l'accès à la ressource, et mettent en place un mode de coordination définissant les règles de résolution de ce problème. Les agents doivent établir un équilibre permanent entre leurs intérêts propres et ceux des autres agents. En effet une règle de conduite doit s'établir pour la circulation dans le couloir qui fait prévaloir l'évitement du blocage par rapport, par exemple, à la volonté d'un agent de traverser immédiatement le couloir.

◊ **Définition :** *Le problème du contrôle social.*

Le problème du contrôle social ou contrôle de la société dans un système d'IAD [Gas 89] consiste à déterminer pour tout agent les contraintes à imposer sur ses actions et ses interactions. Ces contraintes établissent un équilibre entre des critères locaux (objectifs de l'agent) et des critères sociaux (objectifs partagés par tous les agents).

Les stratégies proposées pour le résoudre définissent des *protocoles d'interaction* et l'*organisation* (centralisation, décentralisation) qui établit les relations de pouvoir entre les agents.

### c. Conclusion

- En IAD, le contrôle est décomposé selon deux dimensions : le contrôle individuel (au sein d'un agent) et le contrôle social (au sein de la société).
- Au sein d'un agent, la problématique du contrôle se rapproche de celle de la résolution de problème à laquelle s'ajoute l'ouverture de l'agent sur l'environnement pour lequel les contraintes sur son évolution sont diminuées.
- Au sein d'une société, le contrôle concerne la gestion des interactions : on se rapproche des notions utilisées dans la théorie des systèmes hiérarchiques avec les notions d'intervention et de modes de coordination que l'on retrouve exprimées par les protocoles d'interaction et les organisations.

#### 2.1.5 Psychologie cognitive

La *psychologie cognitive* a pour objet d'étude la structure de la connaissance, sa genèse et ses lois de fonctionnement. Son objectif est d'expliquer l'architecture de l'esprit humain et les lois de représentation et de fonctionnement des connaissances [Tib 86].

Certains auteurs [Ric 90] se sont intéressés plus particulièrement aux *activités cognitives mentales finalisées* : activités de compréhension, raisonnement, résolution de problèmes par exemple. Ces activités sont dirigées par des *objectifs* et s'appuient sur une *représentation* de la situation. Elles ne concernent que celles se déroulant entre le

traitement sensoriel et la programmation motrice. Elles sont organisées au sein d'une *architecture cognitive* constituée de différentes fonctionnalités <sup>9</sup>.

### a. Couches de contrôle

L'exécution d'une activité est une combinaison d'appels aux fonctionnalités concernées de l'architecture cognitive. Cette combinaison est le résultat des efforts conjoints des fonctions de *régulation* et de *contrôle*.

#### Régulation

La régulation sélectionne et ordonne les opérations primitives constituant l'activité. Lors de la *sélection*, les objectifs sous-tendant chacune des opérations primitives sont activés ou abandonnés. Selon les objectifs activés, l'*ordonnancement* affecte des priorités aux opérations qui les réalisent, et leur alloue des ressources de traitement : le temps à passer, l'effort à fournir. Les lois gouvernant les différents choix sont une fonction multiplicative de l'importance de l'opération du point de vue des motivations et de son espérance de succès.

#### Contrôle

La fonction de contrôle proprement dite, se décompose en planification et suivi d'exécution. La planification ou programmation des actions met à la disposition de chaque opération les moyens de sa réalisation. Le suivi d'exécution évalue les résultats de l'opération en cours d'exécution et établit un diagnostic en cas d'incidents.

Nous voyons apparaître deux couches de réalisation du contrôle : (i) la sélection des objectifs et l'ordonnancement des opérations, (ii) la sélection des actions dans laquelle se déroule un suivi d'exécution.

### b. Réalisation

L'existence de ces différentes couches de contrôle ne préjuge en rien de la manière dont le contrôle est réalisé. Ceci n'implique pas notamment l'existence d'une instance supérieure, mécanisme unique, qui régit l'ensemble des activités de l'architecture cognitive. L'hypothèse qui semble plutôt prévaloir est une *distribution* des fonctions de contrôle et de régulation au sein de chaque fonctionnalité, fonctions qui interviennent aux différents stades de réalisation de l'activité. La conséquence de cette distribution est que la fonction de suivi d'exécution notamment est en *interaction*, d'une part avec les mécanismes de sélection des objectifs et d'évaluation des résultats et, d'autre part, avec les autres activités se déroulant dans le système. Cette interaction est nécessaire pour construire une image du déroulement de l'activité et l'influencer.

---

<sup>9</sup>Conservation des structures cognitives permanentes, élaboration de décisions d'action, construction de représentations, production d'inférences afin d'augmenter les représentations ou de générer des décisions d'actions, construction de connaissances, régulation et contrôle de l'activité.

### c. Conclusion

- D'un point de vue modélisation, deux couches sont distinguées. Elles concernent (i) la sélection des objectifs et l'ordonnancement des opérations, (ii) la sélection des actions. Le suivi d'exécution réactive la première couche en cas de modification nécessaire du fonctionnement.
- Des interactions se produisent entre ces couches. Une couche réalise les choix utilisés ensuite sur la couche inférieure : la couche *contrôle* met en œuvre la planification des opérations sélectionnées par la couche de *régulation*. Le suivi d'exécution intervient au niveau de la sélection ou de l'abandon des objectifs définis par cette couche.
- La régulation et le contrôle interviennent au sein de toutes les activités. Il n'existe pas d'instance unique régissant le fonctionnement du système.

## 2.2 Nouvelle formulation du contrôle

En Automatique, le problème du contrôle est considéré résolu par la définition d'une loi de commande. Sa résolution se situe donc au niveau de la conception du système. En IA, comme dans la suite de ce rapport, nous nous intéressons à l'aspect dynamique du problème du contrôle. Pour nous, *le problème du contrôle doit être résolu à tout instant de l'activité du système*.

Par ailleurs, nous envisageons *le problème du contrôle au sein d'un seul système*. Comment le système peut-il agir sur son propre comportement? Nous n'excluons pas cependant le cas habituel de l'Automatique où le processus et le contrôleur sont deux entités séparées.

De manière générale, le contrôle d'un système concerne le problème de la *définition* et de la *modification* de son fonctionnement à partir d'une *évaluation* de son état en vue de poursuivre des *objectifs* donnés.

Nous avons vu que, selon les domaines de recherche, cette notion s'exprime par les termes de régulation, de commande, de décision, de coordination, de planification. A partir de la section précédente, nous donnons une formulation générale de la problématique du contrôle sans faire d'hypothèses sur la manière dont le système est réalisé. Nous utiliserons ensuite cette formulation pour définir plus précisément notre cadre d'étude.

### 2.2.1 Préliminaires

#### a. Processus et contrôleur

Nous inspirant de l'Automatique, de la théorie des systèmes hiérarchiques, de la structuration selon les niveaux meta et objet en IA, nous abordons l'étude du *problème du contrôle d'un système* en définissant deux entités conceptuelles : le *processus* et le *contrôleur*.

Le *processus* est constitué des éléments qui le font changer d'état le système. Le *contrôleur*, quant à lui, regroupe l'ensemble des traitements modifiant le comportement du processus.

Ainsi par exemple, dans un système de production, l'interpréteur de règles est le *processus*, alors que le *contrôleur* est le mécanisme qui détermine la règle à appliquer à partir de l'état de la mémoire de travail (cf. figure 2.3).

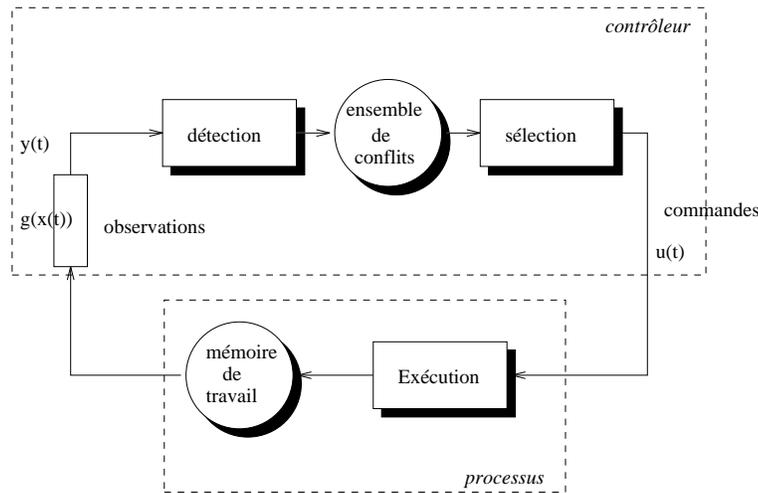


FIG. 2.3 - **Formulation d'un système de production en terme de contrôleur et de processus.** Le contrôleur est constitué des fonctions de construction de l'ensemble de conflits (*détection*) et de résolution de conflits (*sélection*). La commande qu'il émet en direction du processus est la règle instanciée à exécuter.

## Formulation

Nous approfondissons cette conceptualisation en nous inspirant de l'Automatique. Le processus reçoit en entrée les actions ou *commandes* du contrôleur,  $u(t)$ , ainsi que différents signaux de son environnement,  $\omega(t)$ , et change son *état*,  $x(t)$ . Une image de ce dernier est fournie au contrôleur au moyen d'*observations*,  $y(t)$ . Le contrôleur les utilise en appliquant une *loi de commande* ou *politique*,  $\pi$ , pour déterminer l'ensemble des commandes à appliquer au processus afin de satisfaire les objectifs  $o(t)$ .

Soient les ensembles  $X$ ,  $Y$  et  $U$  regroupant respectivement les variables  $x(t)$ ,  $y(t)$  et  $u(t)$ . A chaque instant  $t$ , ces variables définissent respectivement l'état du processus, les observations de l'état et les commandes appliquées au processus. Les évolutions temporelles possibles de ces variables définissent des *trajectoires* ou histoires notées respectivement  $h_x$ ,  $h_y$ , et  $h_u$ . Soient  $H_x$ ,  $H_y$ ,  $H_u$  les ensembles correspondants.

– *équations d'état du processus.*

Soient  $p$  les équations d'état du processus produisant un nouvel état. Nous les

exprimons dans un espace discret par :

$$x(t+1) = p(x(t), u(t), \omega(t))$$

– *fonction d'observation.*

Soit  $g$  la *fonction d'observation* de l'état du processus fournissant  $y(t)$ :

$$y(t) = g(x(t))$$

– *loi de commande.*

Le contrôleur est défini par la *loi de commande*  $\pi$  :

$$u(t) = \pi(y(t))$$

Une loi de commande peut être spécifiée directement par la suite des commandes à appliquer à tout instant. Ceci consiste à fournir un sous ensemble de  $X \times U$ . Cependant, il est souvent plus facile de la définir à partir d'un *objectif*. Celui-ci permet de sélectionner une loi de commande dans  $P$ , ensemble des lois de commande possibles :

$$P = \{\pi : H_y \rightarrow U\}$$

– *objectif.*

L'objectif  $o(t)$  peut exprimer les états du processus que l'on cherche à atteindre sans préciser a priori les moyens pour les atteindre. La loi de commande,  $\pi$ , influe sur le fonctionnement du processus en produisant un ensemble de commandes afin que *son état satisfasse l'objectif*.

L'objectif peut aussi être exprimé par l'intermédiaire d'une fonction d'évaluation de l'état du processus,  $V$ ,  $o(t) = V(y(t))$  où  $V : H_y \rightarrow \mathfrak{R}$ . La loi de commande crée des commandes afin que *l'état du processus maximise cette valeur*.

Le temps  $t$  intervient dans la définition de  $o$  et de  $\pi$  dans la mesure où l'adaptation du fonctionnement du système aux évolutions de l'environnement est nécessaire. Elle implique la mise en œuvre de lois de commande différentes en cours d'activité. Le *choix d'une loi de commande* dans l'ensemble  $P$  est partie intégrante de la tâche du contrôleur alors qu'auparavant elle était assurée par le concepteur.

◇ **Définition :** *Problème du contrôle.*

Le problème du contrôle concerne la définition de la *commande*  $u(t)$  que le contrôleur doit envoyer au processus étant donnés une *loi de commande*  $\pi(t)$  et un ou plusieurs *objectifs*  $o(t)$ . Il recouvre donc à la fois la sélection de la commande, la définition de la loi de commande permettant de la réaliser, et la sélection de l'objectif relatif à sa définition. Nous pouvons définir le contrôleur  $c$  appliquant la loi de commande aux observations comme :

$$u(t+1) = c(\pi(t), o(t), u(t), y(t))$$

Dans le cas où le contrôleur est un système de planification, il est équipé de capacités de prédiction et de raisonnement sur l'évolution de l'environnement. La loi de commande  $\pi(t)$  est le plan engendré en réponse à un but  $o(t)$ . Les commandes  $u(t)$ , envoyées au processus, sont les actions proposées par le plan. En exprimant la loi de commande à l'aide de modèles mathématiques nous nous orientons par contre vers des fonctions plus classiques de régulation de l'Automatique.

Dans notre exemple des systèmes de production, le contrôleur  $c$  est exprimé par les mécanismes de construction de l'ensemble de conflits et de résolution de conflits exprimés au sein de la structure de contrôle (cf. figure 2.3). La loi de commande  $\pi(t)$  du contrôleur est constituée d'une part de la méthode de construction de l'ensemble de conflits et d'autre part des critères de sélection utilisés pour résoudre les conflits. L'objectif  $o(t)$  peut être, par exemple, celui que l'on trouve dans les méthodes *satisficing*: avoir à tout moment une solution satisfaisante. La commande  $u(t)$  est une règle dont les variables sont toutes liées. Les observations  $y(t)$  sont les variables liées aux éléments de la mémoire de travail.

## 2.2.2 Problèmes de contrôle

Nous avons vu précédemment, que le problème du contrôle était dépendant de la définition de la commande  $u(t)$ , de la loi de commande  $\pi(t)$  et de l'objectif  $o(t)$ .

Dans notre étude du contrôle, nous nous fixons comme objectif d'analyse de n'envisager qu'un seul type de choix par contrôleur. Nous définissons ainsi les trois sous problèmes de contrôle suivants : l'un relatif au choix d'*objectif*, un autre au choix de *loi de commande* et le dernier relatif au choix de *commande*. Nous les appelons ainsi respectivement *problème de décision*, *problème d'adaptation* et *problème de commande*. La résolution de chacun d'eux peut faire l'objet de la définition d'un contrôleur.

◇ **Définition :** *Problème de commande.*

Le problème de commande vise à déterminer la commande  $u(t)$  à appliquer au processus par application de la loi de commande  $\pi(t)$  à un ensemble d'observations  $y(t)$  en vue de satisfaire l'objectif  $o(t)$ .

$$u(t+1) = c(y(t), u(t), \pi(t), o(t))$$

◇ **Définition :** *Problème d'adaptation.*

Le problème d'adaptation vise à déterminer la loi de commande  $\pi(t)$ , à appliquer en fonction des modifications intervenant dans l'environnement, et en fonction d'un objectif  $o(t)$ .

$$\pi(t+1) = a(y(t), \pi(t), o(t))$$

◇ **Définition :** *Problème de décision.*

Le problème de décision vise à déterminer l'objectif  $o(t)$  devant contraindre le choix de loi de commande  $\pi(t)$ .

$$o(t+1) = d(y(t), o(t))$$

Dans beaucoup de systèmes existants, il est confondu avec celui de l'adaptation.

◇ **Définition :** *Problème d'observation.*

Lié à la définition de ces différents problèmes, nous définissons le *problème d'observation* comme étant celui de la définition des observations à fournir pour résoudre chacun des problèmes de contrôle. Sa résolution passe par la définition de la fonction  $g$ .

Le problème d'observation est souvent implicite dans les différents domaines et notamment en Intelligence Artificielle. Par exemple, dans les systèmes de production, la fonction d'observation de l'environnement est implantée par la constitution d'un contexte d'activation des règles de production. Ce contexte est ensuite évalué pour déterminer le seuil d'activation de la règle. Les observations sont utilisées pour déterminer la possibilité d'exécution d'une règle.

Dans le cadre du contrôle du raisonnement, l'observation traite aussi de l'accès d'un "résolveur" à ses propres représentations (réflexivité) pour modifier son raisonnement [Man 84] [Mae 87]. Comme nous le verrons par la suite, un prolongement de ce problème est celui de la communication entre agents autonomes interagissant dans le même environnement : quelles informations échanger pour fournir une vision de l'état global de l'environnement au sein de chacun des agents et quelles actions exécuter sur les autres agents ?

◇ **Définition :** *Problème du contrôle.*

Le problème du contrôle est constitué des quatre sous problèmes : commande, adaptation, décision et observation dont la résolution conduit à définir les fonctions suivantes :

$$\begin{array}{ll} \text{commande :} & u(t+1) = c(y(t), u(t), \pi(t), o(t)) \\ \text{adaptation :} & \pi(t+1) = a(y(t), \pi(t), o(t)) \\ \text{décision :} & o(t+1) = d(y(t), o(t)) \\ \text{observation :} & y(t) = g(x(t)) \end{array}$$

### 2.2.3 Couches de contrôle

Selon la résolution du problème du contrôle, nous faisons apparaître des *couches de contrôle* dans l'expression du contrôleur. Chacune de celle-ci résout un sous-problème de contrôle (cf. figure 2.4). Nous les appelons par référence au problème de contrôle qu'elles résolvent : *couche décision*, *couche d'adaptation* et *couche de commande*. Ces couches sont distinguées seulement dans le cas où, respectivement, un choix d'objectif, un choix de loi de commande ou un choix de commande est réalisé dynamiquement en cours de résolution.

Chacune des couches de contrôle manipule des problèmes d'échelle temporelle différente [Mes 70] : le choix de la commande se pose après toute modification de l'état du processus. En revanche, les choix de loi de commande et d'objectif peuvent se poser

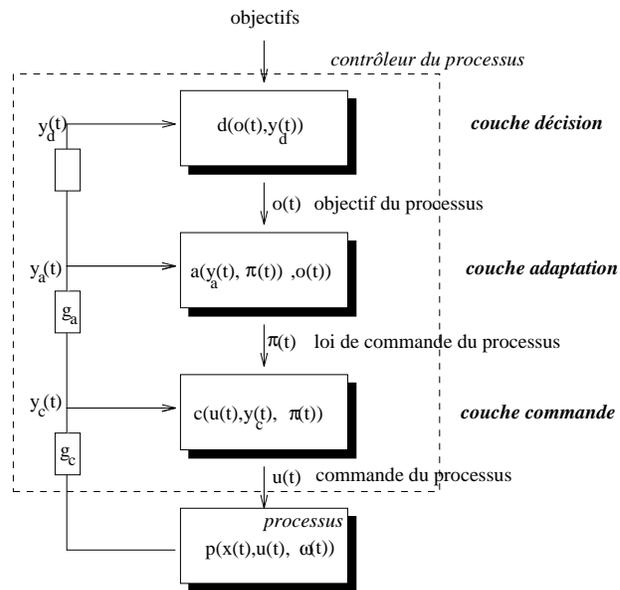


FIG. 2.4 - **Hierarchie de contrôle.** Les couches commande, adaptation et décision sont utilisées dans la construction du contrôleur. Chacune d'elle répond à un sous problème de contrôle spécifique : définition de la commande pour la couche commande, définition de la loi de commande pour la couche adaptation, définition de l'objectif pour la couche décision. Les fonctions d'observation  $g_x$  fournissent les informations sur l'état du processus à chacun des couches.

sur un intervalle de temps plus important, c.-à-d. à chaque fois que la loi de commande courante est invalidée ou que le but est satisfait ou invalidé.

◇ **Définition :** *Couche de commande.*

La couche de commande regroupe l'ensemble des mécanismes résolvant le problème de la commande : définition de la commande.

◇ **Définition :** *Couche d'adaptation.*

La couche d'adaptation apparaît dans le contrôleur si celui-ci comporte des mécanismes définissant la loi de commande en fonction de l'évolution de l'environnement. Elle résout le problème de l'adaptation.

◇ **Définition :** *Couche de décision.*

La couche de décision apparaît dans le contrôleur si celui-ci comporte des mécanismes définissant l'objectif en fonction de l'évolution de l'environnement. Cette couche résout le problème de la décision.

Ainsi par exemple, dans le cadre d'un système de planification, nous assimilons le planificateur à la couche d'adaptation dans le cas où le plan est construit dynamiquement en fonction de l'évolution de l'environnement. La couche commande réalise l'exécution de chacune des actions du plan. Après chaque exécution, l'action suivante indiquée par le plan constitue la nouvelle commande appliquée au processus tant que le suivi d'exécution du plan ne le remet pas en cause. Dès que l'évolution de l'environnement invalide le plan la couche adaptation est réactivée. La couche décision existe si le but à satisfaire peut être modifié dynamiquement par le contrôleur en fonction de l'environnement. Nous ne définissons par contre qu'une couche commande dans le cas où le suivi d'exécution est absent du contrôleur. L'évolution de l'environnement n'influe que sur la construction du plan et non pas sur l'exécution de chacune des actions qui le constituent. Le planificateur est donc défini au sein de la couche commande.

Les mécanismes de contrôle apparaissant sur des couches de contrôle précédentes définissent un contrôleur par rapport à la couche inférieure. Ainsi, dans la suite de notre exposé, nous désignerons les mécanismes d'une couche de contrôle par le terme de *contrôleur* suivi du nom de la couche concernée. De même, nous utiliserons le terme *commande* suivi du nom de la couche concernée pour désigner la sortie du contrôleur intervenant sur cette couche de contrôle. Nous utiliserons *loi de commande* et *objectif* suivis du nom de la couche pour désigner les critères de sélection et l'objectif utilisés par le contrôleur de cette couche. Lorsque ces termes sont utilisés sans complément du nom, ils désignent les éléments contrôlant le processus (cf. figure 2.4). Ainsi, la commande de la couche décision est l'*objectif du processus*. La commande de la couche adaptation est la *loi de commande du processus*. La commande de la couche commande est la *commande du processus*.

## 2.2.4 Définitions supplémentaires

### Structure de contrôle

Sur chacune des couches de contrôle qui existent dans le système, nous trouvons un ensemble de fonctions qu'utilise le contrôleur de cette couche pour déterminer la commande qu'il va émettre. Ces fonctions agissent sur différentes représentations de l'état de l'environnement et du raisonnement. Nous appelons cet ensemble de fonctions la *structure de contrôle*.

◇ **Définition :** *Structure de contrôle.*

La *structure de contrôle* d'un système est l'ensemble des fonctions regroupant les mécanismes de représentation <sup>10</sup> ainsi que ceux implantant les contrôleurs du système existants sur chacune des couches de contrôle.

### Cycle de contrôle

Une notion afférente à la précédente est celle de cycle de contrôle. Celui-ci donne une description du déroulement de l'activité du système par la présentation de l'enchaînement des fonctions inscrites dans la structure de contrôle. Selon le degré d'abstraction des fonctions présentées, nous obtenons plusieurs types de description de cette activité. Les cycles de contrôle de différents systèmes de vision présentés dans [Tso 87] [Lux 85] en sont une illustration. Ces "meta-cycles" de contrôle correspondent à l'enchaînement d'actions prédéfinies pouvant être elles-mêmes construites par un cycle de contrôle mettant en œuvre des mécanismes plus primitifs.

◇ **Définition :** *cycle de contrôle.*

Le *cycle de contrôle* définit l'enchaînement des fonctions constituant la structure de contrôle. Il décrit le déroulement de l'activité du système.

Le cycle de contrôle et la structure de contrôle sont liés à la manière dont le problème de contrôle est résolu. Cette dépendance a été illustrée par exemple dans le système *SONIA* [pap 88] [Col 88], système d'ordonnancement journalier pour un atelier de production, s'appuyant sur le modèle de blackboard de type *BB1* <sup>11</sup>. Dans ce système le cycle de contrôle de *BB1* a du être modifié pour permettre de fixer des paramètres d'une opération après que celle-ci ait été choisie, réalisation impossible dans la mesure où dans *BB1* une opération choisie est immédiatement exécutée. Ce type de problème est encore plus évident dans les applications temps réel dans lequel le cycle de contrôle doit pouvoir être interrompu pour prendre en compte les événements extérieurs [Hr 87] [Col 91].

---

<sup>10</sup>Suppression, modification, création d'un nouvel état par exemple.

<sup>11</sup>Les opérations de résolution sont paramétrées. Ainsi la détermination de ces paramètres s'ajoute au choix de l'opération à exécuter.

## Connaissances de contrôle

Les deux notions présentées ci-dessus sont relatives à l'aspect algorithmique de la résolution du contrôle. Un autre aspect concerne la formulation du contenu des fonctions réalisant les sélections et les modifications de fonctionnement du système. Nous définissons ainsi les *connaissances de contrôle*.

◇ **Définition :** *connaissances de contrôle.*

Nous appelons *connaissances de contrôle*, l'ensemble des connaissances relatives à l'expression des lois de commande et des fonctions qu'utilisent les contrôleurs des différentes couches de contrôle.

Ainsi le contrôleur du processus est défini par une structure de contrôle recevant des fonctions de contrôle appliquant les connaissances de contrôle. L'enchaînement des fonctions de contrôle est régi par le cycle de contrôle. Les fonctions et les connaissances de contrôle se répartissent selon les couches de contrôle existant dans le contrôleur.

## 2.3 Conclusion

Dans ce chapitre, nous avons défini le *problème du contrôle* tel que nous allons l'aborder dans la suite de ce rapport. Nous mettons en évidence conceptuellement au sein du système un *processus* et un *contrôleur*. Le processus est l'ensemble des mécanismes du système le faisant changer d'état. Le contrôleur est l'ensemble des mécanisme définissant la commande du processus. Nous avons mis en évidence quatre sous-problèmes dont la résolution conduit à la définition du contrôle d'un système. Ces quatre sous-problèmes de contrôle sont :

- le *problème de décision* qui définit l'objectif du processus à satisfaire.
- le *problème d'adaptation*, qui étant donné l'objectif du processus, définit la loi de commande du processus utilisée pour résoudre le problème de commande.
- le *problème de commande* définissant la commande du processus à partir de la loi de commande et de l'objectif.
- le *problème d'observation* est transversal aux trois problèmes précédents. Sa résolution définit les observations du processus utilisées pour résoudre chacun des problèmes.

Au sein de la structure de contrôle du système, la prise en compte de ces problèmes conduit à la définition d'une hiérarchie de contrôle constituée de trois *couches de contrôle* : *couche de commande*, *couche d'adaptation*, *couche de décision*. Chacune de ces couches apparaît, dès que la résolution du problème de contrôle qu'elle résoud dépend

de l'évolution de l'environnement, c.à.d. lorsque nous pouvons modifier la commande, la loi de commande ou l'objectif du processus en cours d'activité du système.

Les degrés de liberté du fonctionnement du système dépendent du nombre de couches de contrôle structurant les fonctions de la *structure de contrôle* du système. Les *connaissances de contrôle* sont utilisées par ces fonctions pour déterminer le déroulement du processus.

## Chapitre 3

# Contrôle dans les Systèmes Intégrés de Vision

Nous débutons ce chapitre par diverses définitions relatives au domaine de validation de notre étude du contrôle. Nous exposons ensuite les problèmes liés à la conception d'un *système intégré de vision*. Ayant ainsi défini le contexte de notre étude, nous analysons la manière dont le problème du contrôle est résolu dans quelques systèmes existants en utilisant les définitions et les critères d'analyse mis en évidence dans le chapitre précédent. Ces systèmes ont été choisis en fonction de la problématique définie sur les systèmes intégrés de vision par ordinateur. Ainsi, nous n'avons pas pris en compte les systèmes de vision utilisant une approche fondée sur les réseaux connexionistes ou sur les réseaux de neurones. A partir de cette analyse, nous présentons les aspects du contrôle d'un système intégré de vision que nous abordons dans la suite de ce rapport.

### 3.1 Vision par Ordinateur

#### 3.1.1 Définitions

Un système de perception a pour fonction de *construire* et de *maintenir* une description du monde extérieur à partir d'informations fournies par ses capteurs. Dans notre travail, nous nous intéressons plus particulièrement à la perception *visuelle* et donc à des capteurs de type caméra. Ces derniers fournissent une représentation bidimensionnelle,

instantanée d'une *scène* : l'*image*.

Une *scène* est la portion de l'environnement observée par les capteurs. Elle est le plus souvent tridimensionnelle et est le lieu de manifestation de différents phénomènes : changements d'intensité, occlusions, mouvements du capteur ou des objets, etc.

Une *image* est une matrice bidimensionnelle de pixels consignnant le signal numérisé fourni par la caméra. Chaque *pixel* code par une valeur discrète, l'intensité lumineuse observée<sup>1</sup>.

### a. Informations manipulées

Les informations manipulées au sein d'un système de vision par ordinateur sont de deux types : les *données*, c.-à-d. informations issues des traitements et des capteurs et les *modèles*, ensemble de connaissance a priori exprimant les particularités de l'environnement capté par le système.

## Données

Nous appelons *données* l'ensemble des informations manipulées par les différents traitements afin de construire une description de la scène. Le processus de formation d'images conduit à une importante perte d'information au sein de ces données. Cette perte est essentiellement d'ordre qualitatif dans la mesure où le signal numérisé réunit dans la valeur de chacun des pixels plusieurs facteurs tels l'illumination, la couleur, les reflets, les paramètres de la caméra, la texture. Codant des phénomènes dynamiques se déroulant dans la scène, la validité de ces données est restreinte. En revanche, leur nombre est important<sup>2</sup>, conduisant à des traitements coûteux.

## Modèles

Face à ces informations que fournissent la caméra et les traitements, existe un ensemble d'informations a priori : les modèles. Ils sont utilisés pour construire la description de la scène. Un *modèle* est un ensemble d'*indices visuels* et de *contraintes* exprimant la manière dont un objet ou un phénomène visuel de la scène peut apparaître dans l'image. La définition des indices visuels s'exprime dans les mêmes termes que les données que fournissent les traitements appliqués sur l'image. Ce sont les points d'articulation entre les données et les modèles.

Plusieurs types de modèles sont disponibles. Certains expriment des contraintes physiques ou géométriques, des lois élaborées à partir d'expériences de psychologie expérimentale telles celles à la base de groupements perceptuels [Low 84] [Bie 81] ou des objets génériques *geons* [Bie 85]. D'autres concernent la représentation d'objets en terme de formes, de volumes, de cylindres généralisés [Mar 82] [Bro 81]. La formulation de modèles génériques pour couvrir l'ensemble des possibilités d'apparition des objets

---

<sup>1</sup>L'intensité est codée le plus généralement en 256 niveaux de gris pour des images noir et blanc.

<sup>2</sup>Une image est généralement une matrice 512×512 pixels.

de la scène dans l'image est un problème difficile. Il faut chercher à relier les modèles aux informations issues des images en se détachant de toutes les conditions de prises de vue, de luminosité, etc. Il faut de plus prendre en compte toutes les incertitudes dues à l'imprécision des données. C'est l'objet des recherches menées sur les *invariants* et sur la vision qualitative [Kah 90a].

D'autres modèles enfin, font intervenir des connaissances relatives au type de scène (scène d'intérieur ou d'extérieur par exemple), aux types d'objets en présence, à des hypothèses restrictives sur la scène, absence d'occlusion par exemple, etc.

## b. Traitements

Comme nous l'avons vu, le problème de la vision par ordinateur est lié à l'absence de bijection entre la scène et l'image. Une des tâches les plus importantes dans ce domaine consiste donc à développer des méthodes pour retrouver les informations pertinentes de la scène à représenter à partir des images. Trois groupes sont distingués : les méthodes de *segmentation*, les méthodes d'*inférences de formes* et les méthodes d'*interprétation*.

### Segmentation

Nous trouvons dans ce groupe des traitements dédiés à l'extraction des informations "pertinentes" de l'image. Les données qu'ils fournissent s'expriment en terme d'*indices d'image*, expressions des variations des valeurs d'intensités qui correspondent dans la scène à des informations caractéristiques telles les arêtes, les coins, etc. Barrow et Tenebaum ont défini de tels traitements pour l'obtention de *propriétés intrinsèques* [Bar 78] liées à la texture, à la réflectance, au contraste, etc. Ces propriétés sont appelées intrinsèques du fait de leur indépendance de toute connaissance spécifique sur la scène. Les méthodes d'obtention n'utilisent que des connaissances générales relatives aux contraintes physiques.

### Inférences de formes

Les traitements appliqués sur ces propriétés utilisent des connaissances plus spécifiques sur la scène telles des *modèles* génériques d'objets [Bro 81] ou des méthodes d'*inférence de formes* ("Shape from X"). Celles-ci construisent une description géométrique de la scène en exploitant une propriété intrinsèque et en n'utilisant que des connaissances très générales. Nous n'en donnons pas une description dans le cadre de ce rapport. Le lecteur intéressé peut se référer à [Mar 82], [Dem 86] ou [Alo 90] où elles sont reformulées dans le cadre de la vision active. Chaque méthode aboutit à des aspects précis de la description géométrique. Une représentation complète de la scène ne peut être que l'aboutissement de la combinaison de leurs résultats, voire de leurs interactions.

## Interprétation

Alors que les précédentes méthodes ont un rôle essentiellement de *description*, celles de ce groupe permettent la *désignation*, c.-à-d. l'affectation d'étiquettes à la description obtenue à partir des autres méthodes, ainsi que l'expression de tout un contexte relationnel, fonctionnel. Ces étiquettes font référence aux modèles symboliques liés au type de scène.

Dans les deux derniers groupes de traitements, les modèles utilisés peuvent être importants. Leur association ou *mise en correspondance* avec des indices visuels est extrêmement combinatoire. Différentes propositions d'organisation de cette connaissance sont faites pour faciliter cette tâche (ex : *Alven* [Tso 87]). Cependant la combinatoire de ce mécanisme de recherche dans le graphe des associations possibles entre modèles et indices visuels est NP-Complet dans le cas où aucune information sur les modèles susceptibles d'être utilisés n'est disponible [Lux 85] [Tso 89] [Tso 90]. Le contrôle de ces différents traitements est donc extrêmement important.

### c. Objectifs

Cette importance est majorée selon les objectifs assignés au système. Aloimonos [Alo 86] définit le problème central de la vision par ordinateur comme celui de “*comprendre* en terme de ses propriétés tridimensionnelles, la scène constituée d'un ou plusieurs objets mobiles ou stationnaires à partir d'une ou de plusieurs images prises grâce à une ou plusieurs caméras mobiles ou fixes”. Le terme *compréhension* donne lieu à différents objectifs pour la conception de systèmes de vision.

La description de la scène peut être construite par rapport à des exigences précises émises par un utilisateur : système de navigation autonome, système de surveillance, etc. Un robot se déplaçant dans une salle n'a pas besoin d'avoir une description précise des objets posés sur une table. Une description de la scène en termes d'un réseau d'obstacles lui est suffisante. L'*objectif* du système de vision est alors lié à la satisfaction de buts explicites. Il en découle un mode de fonctionnement appelé *reconnaissance* [Lux 85] [Dem 86] ou *teleonomique* (“purposive”)[Alo 90].

Une autre école de pensée argumente en faveur d'un fonctionnement de *reconstruction* [Mar 82] [Bar 78]. L'objectif du système aboutit alors à la construction d'une description générale de la scène ne répondant à aucun but prédéfini : la description de l'environnement est primordiale par rapport à la recherche d'un objet précis.

### 3.1.2 Système de Vision

Ayant rapidement décrit le contexte de la vision par ordinateur, nous laissons de côté, pour le moment, le débat relatif aux objectifs du système. Nous avons vu que la

construction d'un système de vision conduit à développer :

- un ou plusieurs traitements extrayant des propriétés de l'image ou indices,
- des modèles explicites des phénomènes visuels, du domaine d'application, c.-à-d. forme, couleur, vitesse des objets, etc,
- une ou plusieurs méthodes de mise en correspondance entre ces modèles et les indices.

Les données manipulées sont nombreuses. Des modèles ou des traitements généraux n'existent pas. Une description de la scène ne peut être ainsi que le résultat de la combinaison de plusieurs modèles et traitements. Pour qu'un système de vision puisse aborder des types de scènes différents, l'*intégration* de ces éléments en son sein est donc nécessaire. Celle-ci passe par la définition d'une *structure de contrôle* appropriée qui permette de les combiner.

Les nombreux systèmes fondés sur le paradigme *segmentation-interprétation*, tout comme la diversité des structures de contrôle dont nous pouvons trouver une description dans [Lux 85], [Tso 87], montrent la difficulté de ce problème d'intégration et de définition de la structure de contrôle d'un système de vision.

Diminuant la complexité de la conception d'un système de vision, beaucoup ont limité les représentations et les capacités de traitements utilisées. Certains se sont ainsi cantonnés à un type de scène, à des objets particuliers. Ainsi Roberts [Rob 65], Guzman [Guz 68] se sont intéressés à l'analyse de scènes polyédriques, Barrow [Bar 78], Garvey [Gar 76] à l'analyse de scènes d'intérieur. L'analyse de scènes naturelles est abordée dans [Han 78b], celles de scènes aériennes par [Bro 81], [Nag 79], [Gar 89], [Mat 85], etc.

Une direction de recherche s'est cependant attachée à la résolution de ce problème d'intégration. Une longue période s'est écoulée entre les premières propositions de *systèmes généraux de vision* recueillies dans [Han 78b] et la réapparition du terme de vision *intégrée* [Cro 89]. La difficulté de la tâche l'explique en partie. Sa réapparition s'est faite concurremment avec l'utilisation de la *vision active* qui a introduit les capteurs dans le groupe des traitements à prendre en considération dans le contrôle du système.

## 3.2 Systèmes Intégrés de Vision

Avant de poursuivre plus avant dans notre étude, nous allons préciser et fixer le type de système de vision par ordinateur auquel nous nous intéressons.

### 3.2.1 Définition

Selon Horn [Hor 86], un *Système Général de Vision*<sup>3</sup> doit être capable de traiter l'ensemble des aspects de la vision. Il doit être applicable à tous les problèmes pouvant être résolus à partir d'informations visuelles. Précisant cette opinion, au travers d'un

---

<sup>3</sup>General Purpose Vision System.

bilan des réalisations en vision, Binford [Bin 82] énonce les critères nécessaires pour qualifier un système de général.

- Critère d'*indépendance*: le système peut être confronté à plusieurs domaines d'application, comprenant un nombre important d'objets.
- Critère de *robustesse*: les traitements et les modèles doivent être applicables dans différentes conditions.
- Critère de *représentation*: les traitements doivent s'appuyer sur des primitives fiables pouvant être combinées.

Ces deux définitions ne s'appliquent actuellement à aucun système. Elles se traduisent surtout en terme d'objectif à long terme de la construction de systèmes généraux de vision.

Dans cette étude, nous nous plaçons d'un point de vue plus pragmatique. C'est pourquoi, nous préférons abandonner le qualificatif *général* pour celui d'*intégré*. Notre but est de construire un système ayant les *capacités d'évolution* qui lui permettent d'atteindre les exigences précédentes, à plus ou moins long terme, *sans remise en cause complète de son architecture*. Ceci implique pour celle-ci, d'une part son *indépendance* vis-à-vis du domaine, des classes d'objets, des conditions de prises de vue et, d'autre part, son *ouverture*, c.-à-d. la possession de capacités d'évolution.

De manière plus précise, dans le cadre de cette étude nous définissons un Système Intégré de Vision comme :

◊ **Définition** : *Système Intégré de Vision*

Un Système Intégré de Vision est un système *ouvert* et *indépendant* du type de scène. La notion d'*ouverture* implique, à tout moment, la possibilité d'ajouter ou de retirer du système différentes fonctionnalités (traitements et représentations) sans que son fonctionnement et ses performances ne soient remis en cause. L'*indépendance* du domaine implique la possibilité d'être utilisé dans n'importe quel contexte à tout type de scène.

### 3.2.2 Problématique d'un Système Intégré de Vision

Un Système Intégré de Vision doit être capable de recueillir les connaissances et traitements relatifs à plusieurs types de scènes, de modèles et de fonctionnement. Nous recherchons dans cette section la structure pour organiser les traitements et représentations d'un tel système. Sa base est, encore actuellement, constituée des *niveaux de représentation*. Nous montrons que leur présence dans le système influe sur la décomposition des traitements et sur la définition du cycle de contrôle.

#### a. Représentation

La quantité importante de données et de traitements possibles a suggéré la mise en place d'une hiérarchie de descriptions successives ou *niveaux de représentation* [Mar 82].

C'est Marr, qui, dans les années 75, contestant la structuration restreinte du paradigme segmentation-interprétation, introduit une décomposition plus riche des informations et du processus visuel selon trois types de structures de données ("Primal sketch", "2,5-D sketch", "3-D Model") :

- le *croquis élémentaire* explicite les changements d'intensité lumineuse dans l'image, leur distribution et leurs organisations géométriques,
- le *croquis 2D 1/2* décrit la géométrie des surfaces visibles en terme d'orientations et d'informations de profondeur,
- la *description 3D* décrit les formes en terme de volume d'occupation spatiale ainsi que leur organisation spatiale.

Marr [Mar 82]	Demazeau [Dem 86]	Crowley [Cro 90]	Neumann [Neu 90]
Non défini	Scène	Interprétation	Interprétation de haut niveau
Description 3D	$\simeq$ Objet	$\simeq$ Groupement des caractéristiques	$\simeq$ Objet reconnu
Croquis 2,5 D	$\simeq$ Indices de scène	$\simeq$ Description géométrique de la scène	$\simeq$ Eléments de la scène
Croquis élémentaire	$\simeq$ Indices d'image	$\simeq$ Description géométrique de l'image	$\simeq$ Eléments d'image
Images	$\simeq$ Images	$\simeq$ Images	$\simeq$ Images

TAB. 3.1 - **Niveaux de représentation en Vision par Ordinateur** [Dem 90a]. Au travers de cette table, nous voyons apparaître une définition à peu près similaire du nombre de niveaux de représentation dont dispose un système de vision. Nous avons comparé les différents indices caractérisant chaque niveau, par rapport au modèle proposé par Marr. Alors que la définition des indices sur les niveaux de représentation les plus bas (bas du tableau), semblent converger, ceux du niveau le plus haut sont différents. Cette remarque va dans le même sens que celle que l'on peut faire au sujet de la représentation des connaissances utilisée. Aucune uniformité n'existe à son sujet.

Depuis, de nombreuses propositions prolongeant cette structuration ont été faites. Une présentation de certaines d'entre elles peut être trouvée dans [Dem 86]. Toutes ces propositions conduisent, plus ou moins, aux mêmes ensembles de niveaux de représentation (cf. figure 3.1). D'une manière générale, on distingue trois ensembles [Wee 87] [Smi 86b] :

- *Représentation de bas niveau.*

La scène est décrite en termes d'indices d'image ou de propriétés intrinsèques :

points, lignes de contraste et de reflectance, etc. Eventuellement des regroupements apparaissent : segments, régions, groupements perceptuels (jonctions, parallèles, etc). Ces données expriment une information *bidimensionnelle* dans un repère lié à l'image. Elles ont un degré de changement temporel important. Elles sont utilisées par la plupart des traitements de niveaux supérieurs.

– *Représentations de niveaux intermédiaires.*

Ils sont à l'origine de la plupart des différences de représentation entre les systèmes, notamment en ce qui concerne leur nombre. Les représentations utilisées visent à expliciter la structure *tridimensionnelle* géométrique de la scène. Les niveaux intermédiaires constituent le lien entre les informations en changement continu du niveau inférieur et les représentations stables et progressivement plus dépendantes du domaine qui sont exprimées au niveau supérieur.

– *Représentation de haut niveau.*

Sur ce niveau, la scène est décrite en termes propres au domaine d'application sous une forme le plus souvent symbolique : noms d'objets, relations spatiales, etc.

Un consensus semble vouloir se dégager actuellement sur la nécessité des niveaux de représentation. Il n'existe cependant pas encore de formalisation les concernant. Dans [Dem 86] sont proposés deux critères pouvant caractériser un niveau de représentation : *abstraction* et *décentration*. Ce dernier fait référence au changement de repères spatial et temporel que nous trouvons d'un niveau de représentation à l'autre, tel par exemple le passage d'un repère lié à la scène à celui lié à un objet.

Un des problèmes non résolus encore, est celui du type de représentation à utiliser sur chacun des niveaux. Actuellement, aucun consensus ne se dégage et les représentations utilisées restent très ad hoc. Ce problème est d'autant plus difficile qu'une représentation est fortement dépendante de l'utilisation que l'on veut en faire.

## b. Traitements

Poursuivant la structuration de la représentation, les traitements regroupés dans les ensembles de segmentation, d'inférence de formes et d'interprétation s'organisent selon les niveaux de représentation. Les méthodes de segmentation travaillant à partir de l'image se retrouvent sur les bas niveaux. Les méthodes d'inférence de formes agissent sur les niveaux intermédiaires alors que les méthodes d'interprétation opèrent sur les hauts niveaux. Les niveaux de représentation apparaissent comme le lieu d'échange d'informations entre les traitements qui agissent comme des opérateurs d'abstraction entre les niveaux.

La structure de contrôle du système se superpose aux différents niveaux de représentation et contraint le fonctionnement et l'enchaînement de chacun des traitements. La plupart des cycles de contrôle qui en résultent sont des variations autour du *cycle de perception* [Mac 78]. Chacun développe plus particulièrement l'une ou l'autre des

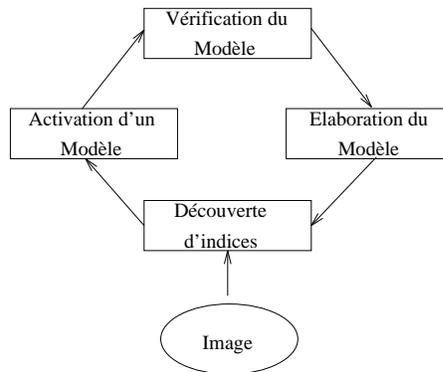


FIG. 3.1 - Cycle de perception. [Mac 78]

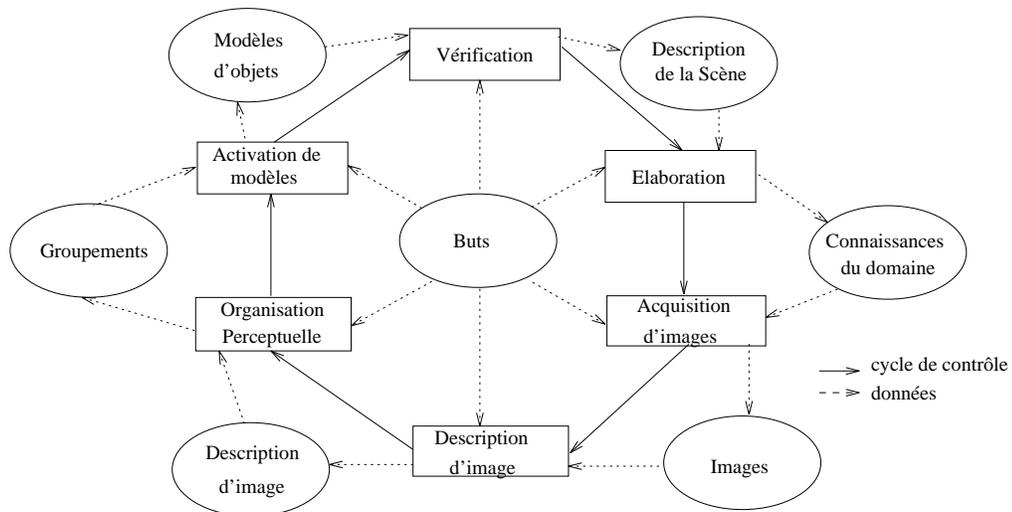


FIG. 3.2 - Cycle de contrôle du système *Vision As Process*. [Cro 90] Ce cycle de contrôle met en place un parcours de tous les niveaux de représentation du système. A la différence du cycle de perception l'acquisition de l'image est incluse dans ce cycle. Les caméras sont ainsi une part intégrante du système. Comme l'ensemble des autres traitements elles sont contrôlées par un ensemble de buts.

phases de ce cycle (cf. figure 3.1 et 3.2) et institue un parcours particulier des niveaux de représentation.

### c. Vision passive/Vision active

Dans la plupart des systèmes généraux de vision proposés dans [Han 78a], les descriptions élaborées s'appuient sur des scènes statiques dans lesquelles les capteurs sont immobiles, sans aucune interaction avec l'environnement. La vision abordée est qualifiée de *passive*, par opposition au terme de vision *active* correspondant à la tendance actuelle que l'on voit apparaître dans certains systèmes [Baj 85][Alo 87][Cro 89]. Dans ce type d'approche, différentes contraintes ajoutées sur la définition des problèmes en vision passive sont compensées par le mouvement imposé au capteur et peuvent donc disparaître. Aloimonos [Alo 87] a montré que le mouvement du capteur permet de mieux poser les problèmes rencontrés dans bon nombre de méthodes d'inférences de forme et d'accroître ainsi leur fiabilité. D'autres avantages sont aussi introduits. C'est ainsi que Ballard [Bal 91] a proposé l'abandon du repère centré caméra pour l'utilisation d'un repère de référence centré sur le point de fixation du capteur. Cette technique facilite l'implantation de certains algorithmes, en particulier celui du calcul du déplacement de la caméra qui peut s'exprimer ainsi en termes qualitatifs.

Cette composante active vient se superposer sur le fonctionnement et les problèmes présents dans le cadre de la vision passive. Bien que contribuant à l'augmentation de la fiabilité, elle introduit les tâches supplémentaires consistant, par exemple, à déterminer à tout moment les valeurs des paramètres du capteur et à faire exécuter ensuite les réglages nécessaires [Tso 90]. Un autre problème important consiste aussi en la gestion de l'accès des différents traitements du système à la ressource limitée que constitue le capteur.

La vision active a cependant des prolongements autres que simplement l'introduction des capteurs dans le cycle de contrôle du système. Les caméras peuvent en effet être considérées comme un nouveau module de traitement fournissant une description de la scène aux traitements du niveau de représentation supérieur. Ceux-ci sont eux-mêmes les capteurs du niveau supérieur. Les conséquences apparaissent plutôt sur la manière d'aborder le fonctionnement du système lui-même. Celui-ci pourra être *continu* (ex : VAP[Cro 89]) ou *ponctuel* (ex : TEA-1[Rim 92b]), *réactif* ou *délibératif* (ex : [Gar 76]), etc. L'élargissement du spectre de fonctionnements envisageables accentue l'importance du contrôle dans un tel type de systèmes.

## 3.3 Exemples de Systèmes Intégrés de Vision

Les systèmes que nous présentons dans cette section ont été choisis à partir de notre définition de système intégré de vision : ils s'affichent comme des systèmes intégrés de vision ou ont, à notre avis, des éléments intéressants quant aux notions d'ouverture ou d'indépendance.

Après une description générale de chaque système, nous présentons plus particulièrement sa résolution du problème du contrôle. Nous nous focalisons sur la description de sa structure de contrôle en utilisant les résultats du chapitre précédent. Nous mettons ainsi en évidence les couches de contrôle abordées et les mécanismes utilisés. Nous présentons rapidement différentes connaissances de contrôle manipulées.

Comme nous l'avons vu, les systèmes de vision actuels, se distinguent par leur prise en compte du capteur : vision passive/active. Nous présentons ainsi, tout d'abord des systèmes de vision passive. Cette distinction est utilisée pour bien mettre en évidence les exigences de contrôle qu'introduit la mobilité du capteur.

### 3.3.1 Propriétés intrinsèques

La description de ce système est issue d'une proposition formulée par Barrow et Tenenbaum [Bar 78] comme prolongements de travaux initiés dans l'extraction des propriétés intrinsèques d'une image. Son fonctionnement est décomposé en deux : *analyse d'image* et *analyse de scène*. L'objectif est de montrer que l'interprétation, se fondant sur des propriétés intrinsèques telles que la couleur, l'orientation, la distance, la texture, la forme ou l'illumination, est moins combinatoire que celle se fondant uniquement sur l'intensité lumineuse. La description faite dans [Bar 78] ne présente en détail que la phase d'extraction des propriétés intrinsèques (analyse d'image). Quelques indications sont fournies concernant l'analyse de scène.

### Représentation et traitement

Les entrées sont constituées d'une ou plusieurs images d'intensité, pour plusieurs points de vues et bandes spectrales. Un ensemble "d'images" de propriétés intrinsèques pour chacun des points de vue est fourni en sortie. Ces images contiennent aussi les informations de discontinuités des valeurs ou du gradient.

La construction de chaque ensemble de propriété intrinsèque pour tout point de l'image d'intensité se déroule par consultation d'un catalogue de correspondances entre valeur d'intensité et propriété. Le résultat est inséré dans l'image de propriétés correspondante. Les traitements construisant chacun de ces ensembles travaillent en parallèle. Une vérification d'incohérence inter-images et intra-images de propriétés est ensuite réalisée afin d'ajuster les valeurs des propriétés intrinsèques. Ce traitement est itéré jusqu'à ce qu'une convergence soit obtenue.

### Contrôle

Chaque traitement d'extraction des propriétés est *dirigé par les données* : les images sont les déclencheurs de son fonctionnement. Ces traitements ne subissent aucun contrôle. Ce dernier intervient lors de la vérification de la cohérence des propriétés intrinsèques créées. Nous distinguons deux aspects :

1. *Vérification de la cohérence Intra-propriétés intrinsèques.*

Lors de cette vérification des valeurs intrinsèques et des contours pour différents

points d'une même image de propriété intrinsèque, le couple contrôleur-processus est le suivant : le contrôleur est le mécanisme de satisfaction de contraintes, le processus est le traitement modifiant la valeur de la propriété intrinsèque qui satisfait la contrainte. Le contrôleur ne met en œuvre que la couche *commande*. Son *objectif* est l'obtention d'une valeur cohérente pour chaque point de l'image. La *loi de commande* est l'ensemble des contraintes relatives à la continuité et aux intervalles de valeurs permises pour une propriété. Les *observations* dont dispose le contrôleur sont les valeurs de propriété.

## 2. Vérification de la cohérence Inter-propriétés intrinsèques.

La vérification se déroule pour un même point pris sur plusieurs images de propriété. Nous retrouvons le même type de couple contrôleur-processus. Le contrôleur utilise aussi la seule couche de *commande*. Son *objectif* est le même que précédemment. La *loi de commande* est exprimée par les contraintes physiques. A la différence du contrôleur précédent, les *observations* proviennent de l'ensemble des images de propriété pour un même point.

## Conclusion

- L'analyse d'image met en œuvre deux processus de satisfaction de contraintes qui ont tous deux un contrôle de couche *commande*. Les informations de retour sont fournies par les valeurs des propriétés. Le critère d'arrêt est la convergence. L'initialisation du traitement se produit par l'arrivée d'une image d'intensité à traiter (*direction par les données*).
- Ces processus n'influent que sur les valeurs des propriétés. Aucun retour n'a lieu sur le mécanisme d'extraction lui-même. Celui-ci en fait consulte le catalogue construit hors ligne, par les tests de tangence et d'invariance.

A la différence de Marr qui s'est intéressé au développement de méthodes d'inférences par elles-mêmes, Barrow et Tenenbaum ont ainsi étudié leurs interactions. Dans le même ordre de démarche, nous pouvons citer la *MIT Vision Machine* [Pog 88] qui vise à construire une architecture pour le fonctionnement parallèle de différents modules de vision primitive<sup>4</sup> et l'intégration de leurs résultats au sein d'une structure commune mettant en évidence les discontinuités visibles de la scène.

### 3.3.2 Procédures visuelles

En 1984, Ullman [Ull 84] propose un modèle pour la vision que l'on peut situer dans les niveaux de représentation intermédiaires vues précédemment. L'objectif de ce modèle est d'étudier le mécanisme d'analyse d'une scène à partir d'une image, en termes de propriétés et de relations spatiales. La notion à la base de ce modèle est celle de *Procédure visuelle*.

---

<sup>4</sup>Early vision.

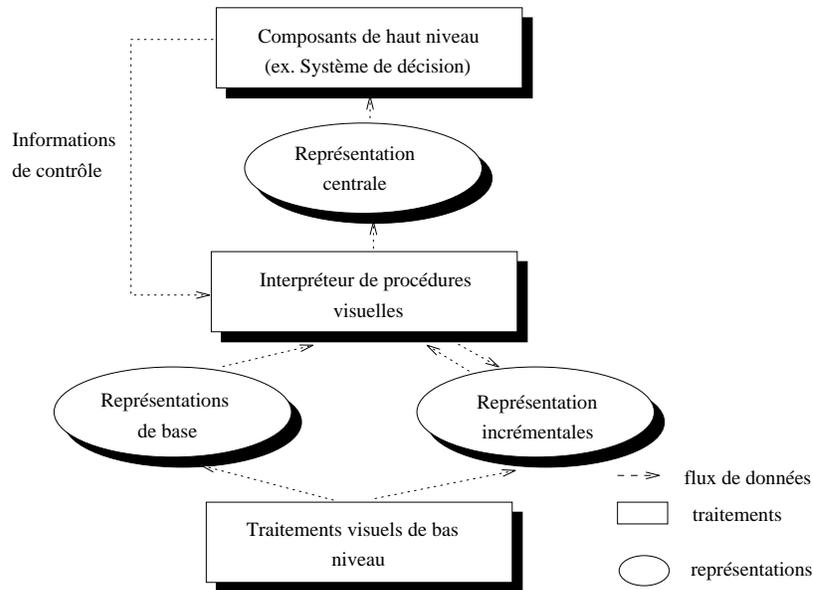


FIG. 3.3 - **Modèle des Procédures Visuelles** d'après [Ull 84]. Les traitements sont encadrés alors que les représentations manipulées sont encadrées. Les flèches en pointillé représentent le flux de données.

Le traitement est décomposé en deux étapes principales (cf. figure 3.3) : une *ascendante* consistant en la création de la *représentation de base* de bas niveau, la seconde, *descendante*, appliquant les procédures visuelles sur cette représentation pour extraire les propriétés spatiales nécessaires pour satisfaire les buts.

## Représentation et traitement

Trois représentations sont définies : (i) la *représentation de base* équivalente au “croquis élémentaire” de Marr, en entrée des procédures visuelles, (ii) la *représentation centrale* recevant les résultats de leur exécution, (iii) la *représentation incrémentale* stockant les résultats intermédiaires pour des utilisations ultérieures.

Les *procédures visuelles* sont des programmes séquentiels, des schémas d'activation d'un ensemble fixé d'opérations visuelles de base. Le canal de communication entre les représentations de base et les composants de haut niveau est réalisé par l'interpréteur qui planifie et exécute les différentes procédures.

## Contrôle

L'application des procédures est fortement dépendante de la tâche et des informations nécessaires au niveau de représentation supérieur où à l'utilisateur du système. Leur activation est donc *dirigée par les buts*. Nous distinguons deux aspects du contrôle

relatifs à deux types de choix : choix d'opération qui se manifeste par la *construction des procédures* et choix de région réalisé par l'*indexation*.

#### 1. *Construction des procédures.*

Elle sélectionne des opérateurs de base et construit leur schéma d'activation en fonction du but exprimé. Le contrôleur est constitué de l'ensemble des traitements construisant le schéma d'activation. Le processus est l'interpréteur de procédures. Le contrôleur aborde seulement la couche *commande*. En effet, l'objectif à résoudre est imposé par le niveau supérieur. La méthode de construction du schéma d'activation, c.-à-d. la loi de commande, n'est pas modifiée en cours d'exécution de la procédure. Une fois un schéma d'activation construit (c.-à-d. la commande), la procédure est exécutée entièrement. Les observations sont les éléments de la représentation centrale.

#### 2. *Indexation.*

L'indexation ("Indexing") choisit la région de l'image sur laquelle la procédure est appliquée. Cette opération se déroule en trois étapes :

- calcul de propriétés en parallèle sur toute la représentation de base. Les propriétés considérées pour l'indexation sont relatives au contraste, à l'orientation, à la couleur, au mouvement, etc.
- examen et comparaison de ces propriétés afin d'inscrire dans la représentation centrale celles qui se différencient des autres : les *exceptions*.
- changement du centre d'intérêt (focus), c.-à-d. la décision d'appliquer la procédure dans telle région en choisissant l'exception qui correspond à la particularité de la procédure.

Le contrôleur relatif au choix de région aborde donc la couche *commande* : le mécanisme de sélection est fixé dans la mesure où il est spécifié dans la procédure. Les informations de retour qui permettent le choix sont les exceptions mises en évidence dans la représentation.

## Conclusion

- Deux dimensions distinctes du contrôle sont mises en évidence : le choix de la *région spatiale* (indexation) dans laquelle travailler, la *sélection des opérations*. Les deux contrôleurs les réalisant abordent la couche *commande*.
- L'indexation est réalisée par la consultation de la représentation de base à la recherche d'exceptions au sein des indices. La détection de tels éléments permet de fixer la région spatiale.
- L'organisation de l'activité est orientée par la satisfaction de *demandes*. A partir de la représentation centrale et d'opérateurs visuels, elle met en œuvre un mécanisme de *planification*, non décrit dans le modèle.

Un tel modèle de traitement de l'information visuelle est souvent évoqué comme la suite de nombreux systèmes présentant l'extraction de propriétés intrinsèques comme l'étape préliminaire. Peu de réalisations complètes existent. Parmi celles-ci, nous pouvons citer les systèmes *Sonja* [Cha 90] et *Pengi* [Agr 87] qui utilisent ce modèle pour coupler l'action et la perception. Leurs systèmes ne travaillent cependant pas sur des données réelles et ne construisent qu'une description spatiale de la scène. Ils utilisent une représentation *deictic* [Agr 87] correspondant seulement aux informations pertinentes pour la tâche en cours. Dans ces systèmes, la sélection des questions à poser au système est implantée par un réseau d'activation et d'inhibition. D'autres personnes utilisent ce modèle pour faire de l'apprentissage de schémas d'activation dans ce niveau intermédiaire tel le système présenté dans [Whi 92].

### 3.3.3 Stratégies perceptuelles

Garvey [Gar 76] propose une approche pour la production de stratégies perceptuelles fondées sur des connaissances a priori, relatives à la scène. Il a ainsi développé un système pour étudier l'apport de l'utilisation d'informations *contextuelles* sur la recherche d'objets dans une scène.

#### Représentation et traitement

Le système est constitué de deux parties : (i) un ensemble d'*opérateurs de description* appliqués sur les images issues de capteur mettant en évidence plusieurs propriétés de l'image telles la couleur, la profondeur, etc., (ii) un *planificateur* qui décide de l'endroit où regarder, et du ou des opérateurs à utiliser en fonction d'un but émis. Les modèles d'objets de la scène sont décrits directement en terme de propriétés de l'image.

Un *opérateur de description* de l'image est constitué de trois composants : l'acquisition, la validation, l'encadrement qui mettent en œuvre une recherche incrémentale de l'objet dans l'image par une succession de tests des indices d'images.

#### Contrôle

Le fonctionnement du système est dirigé par les buts : la création d'un but active le traitement. La satisfaction d'un de ceux-ci se déroule en trois étapes : la création du plan à partir du but et de la description courante de la scène, l'évaluation statique et l'exécution. Le contrôleur du processus de description de la scène est constitué de deux couches de contrôle : l'adaptation et la commande.

##### 1. Couche *adaptation*.

L'élaboration du plan décompose hiérarchiquement le but jusqu'à ce qu'il puisse se traduire en appels directs d'opérateurs de description. Le plan produit est un graphe ET/OU. Sa linéarisation est réalisée dans la couche commande. La commande issue de ce contrôleur est donc le plan produit. Celui-ci n'est pas remis en cause. Ainsi, aucun retour n'est opéré sur cette couche si ce n'est lors de la

satisfaction du but. Le planificateur utilise les connaissances de contrôle liées aux propriétés perceptuelles des objets de la scène, à leurs relations contextuelles, aux caractéristiques de traitement des opérateurs de description. Les résultats partiels produits par l'analyse en cours, sont les observations utilisées pour la construction du plan.

## 2. Couche *commande*.

Après chaque exécution, le plan est réévalué pour déterminer la prochaine action à exécuter. Ce choix dépend du coût local de l'exécution de l'opérateur associé, du coût plus global relatif à la généralité plus importante d'un but, et de la probabilité des résultats corrects à l'issue de l'exécution de l'opérateur. Selon ces critères un processus de marquage par relaxation est utilisé<sup>5</sup>. Le meilleur sous-but exécutable, consistant en un opérateur de description, est choisi. Le résultat de l'exécution est ensuite évalué. Les observations de ce contrôleur sont donc le résultat de l'exécution. Le critère d'arrêt est soit l'échec, soit l'exécution complète du plan. L'initialisation est fournie par la production d'un plan à exécuter provenant de la couche adaptation.

## Conclusion

- Le fonctionnement de ce système est dirigé exclusivement par les *buts*. Il est un des seuls à avoir utilisé une méthode de *planification* explicite pour l'interprétation symbolique. Il est vrai que son système est très limité dans les descriptions utilisées : les scènes d'intérieur avec très peu d'objets. De plus il n'aborde que la vision passive.
- Le contrôle est assuré par la construction d'un plan partiel linéarisé en fonction de l'exécution. Deux couches de contrôle sont utilisées : une concernant la construction du plan (adaptation) et l'autre consistant en sa linéarisation conduisant directement à un choix d'opérateur (commande).

### 3.3.4 Schéma

Le système *SCHEMA* [Dra 87] se situe dans la suite naturelle des recherches effectuées avec le système *VISIONS* <sup>6</sup> [Han 78b]. Il étudie plus spécialement le type de résolution qui peut se mettre en place sur le plus haut niveau de représentation de ce système. Son objectif est l'interprétation de scènes statiques monoculaires d'extérieur.

---

<sup>5</sup>Les informations prenant part au coût d'un nœud sont seulement locales et expriment uniquement des notions de coût d'exécution. On n'exprime pas l'importance que peut avoir un nœud dans la suite du traitement.

<sup>6</sup>Visual Integration by Semantic Interpretation Of Natural Scenes.

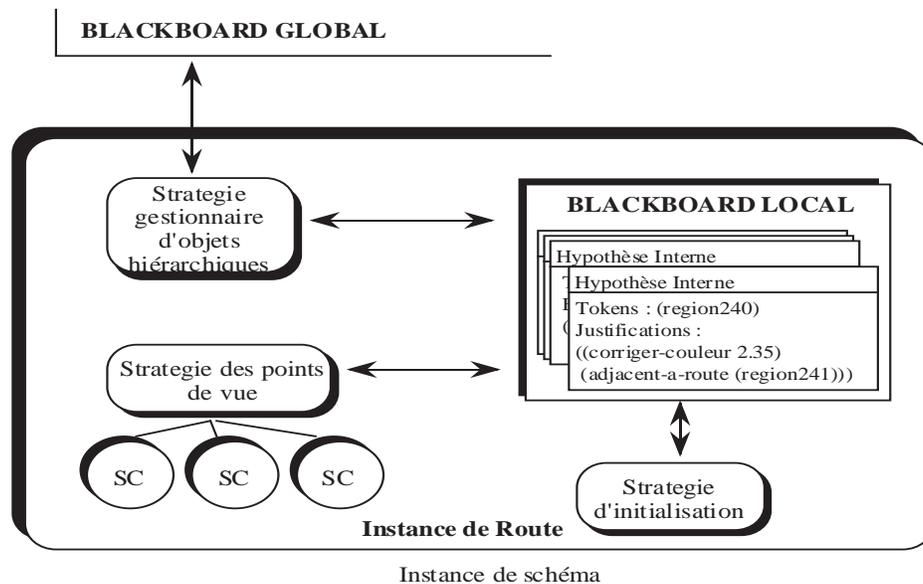
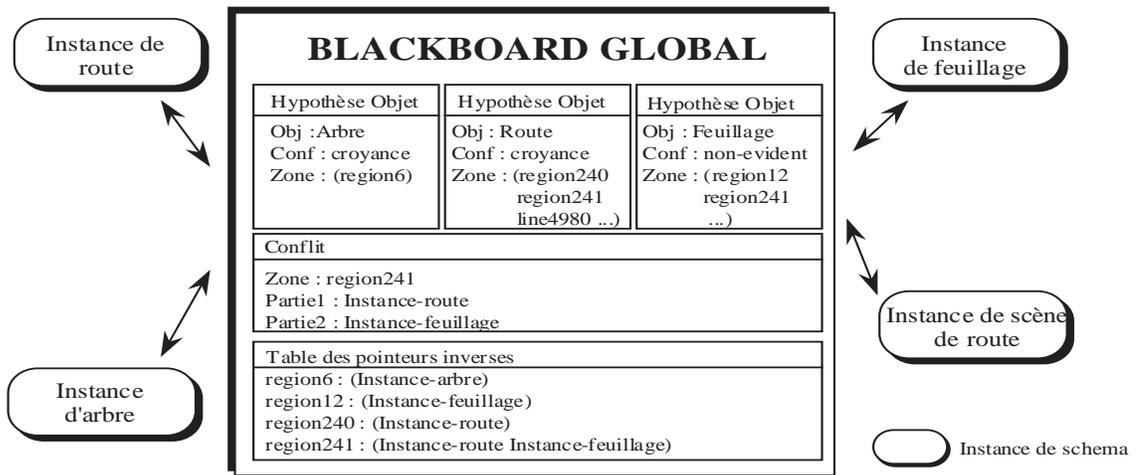


FIG. 3.4 - **SCHEMA** : architecture du système d'après [Dra 87]. La figure du haut représente le système global organisé en différents schémas interagissants au travers du blackboard global. La figure du bas détaille la structure d'un schéma. Celui-ci est constitué d'un blackboard local sur lequel interagissent différentes sources de connaissances.

## Représentation et traitement

La connaissance et les traitements sont répartis en différentes classes relatives aux objets constituant la scène : route, ciel, maison par exemple. Des relations spatiales les relient ainsi qu'une hiérarchie de décomposition *part-of*. Toute classe d'objets est reliée à un *schéma* qui contient les connaissances déductives et les stratégies propres à l'interprétation de l'objet correspondant. La description de la scène est décomposée en deux ensembles : un, global, accessible à tous les schémas, le *blackboard global* et l'autre, local, pour chaque schéma, le *blackboard local* (cf. figure 3.4).

Tout comme la représentation, le traitement est décomposé selon les deux types de blackboard. Le mécanisme général consiste en une création d'*hypothèses et de tests*. Un ensemble de sources de connaissance, de segmentation et d'interprétation génériques sont présents dans le système. Ils opèrent sur les niveaux de représentation inférieurs. Dans les deux types de blackboard, nous retrouvons le mécanisme classique d'activation des schémas ou des sources de connaissances par modification de l'état du tableau noir.

## Contrôle

Nous distinguons deux aspects du contrôle : le contrôle du système lui-même, le contrôle à l'intérieur d'un schéma.

### 1. Contrôle du système.

Le contrôle du système régit l'exécution des différents schémas. Le contrôleur le réalisant aborde la couche commande : le critère de décision consiste à maximiser les coefficients de vraisemblance des hypothèses par propagation des valeurs au sein de la hiérarchie de décomposition. Ce critère est mixé à la résolution des conflits dus à l'occupation d'une même région de l'image par deux hypothèses différentes. La commande émise est l'activation d'un schéma.

Aucun contrôle centralisé n'est exercé dans ce système. Le contrôleur est distribué au sein de chacun des schémas.

### 2. Contrôle à l'intérieur d'un schéma.

Le contrôle à l'intérieur d'un schéma aborde deux couches de contrôle : *adaptation* et *commande*.

- La couche *adaptation* met en œuvre un choix de stratégie d'interprétation qui guide la production des hypothèses sur la couche commande. Ce choix de stratégie cherche à maximiser la valeur de vraisemblance de l'hypothèse d'objet à inscrire sur le blackboard global tout en maintenant la cohérence de la résolution. Chaque stratégie définit : l'ordre d'appel des opérateurs, l'ajout de justifications pour soutenir la création d'hypothèses, la régulation des créations et des tests d'hypothèses au sein du schéma (*prédiction vérification*). Le choix de la région sur laquelle travailler n'est pas fait explicitement. Il est une conséquence des relations spatiales qui sont exprimées entre les hypothèses.

- Le contrôleur de la couche *commande* active les différents opérateurs de traitement de l'image selon le mode spécifié par la stratégie d'interprétation que lui a communiquée la couche adaptation. La description en cours de construction inscrite dans le blackboard local, lui fournit des observateurs.

## Conclusion

- Au sein d'un schéma, un contrôleur aborde les deux couches de contrôle *adaptation* et *commande*. L'approche prise ici rejoint celle de Garvey dans la mesure où tous deux cherchent la mise en œuvre de stratégies spécifiques d'interprétation. La différence réside dans l'élaboration : ici, les stratégies sont prédéfinies<sup>7</sup>, alors que dans Garvey, nous avons un planificateur qui les construit dynamiquement. En plus de la définition d'un centre d'intérêt, nous voyons apparaître la gestion des interactions inter-niveaux définie explicitement par les stratégies.
- En ce qui concerne le système, seule la couche *commande* est utilisée. Du fait de la hiérarchie d'activation, le système est initialisé par l'activation du schéma correspondant à la racine.

### 3.3.5 Medusa

*MEDUSA* [Alo 90] est un système de vision active construit avec des capteurs inertiels, une caméra mobile, un outil de préhension visible de la caméra, le tout monté sur une plate-forme mobile. Le système traite les images tout en se déplaçant et en observant l'environnement.

Deux structures de données principales sont utilisées par tous les composants du système : les images et les champs de flux normal. Un module est dédié à leur extraction à partir des images.

## Contrôle

Un contrôleur central a une vision globale de la résolution et décide de l'activation et de l'exécution des modules constituant le système. Ceux-ci sont des méthodes d'inférence de formes, spécifiques à une tâche donnée. Ils travaillent à partir de l'image et des flux normaux. Les niveaux de représentations ne sont pas explicités. Chaque module met en œuvre ses propres représentations selon ses besoins. Ses traitements sont spécifiés par une méthode particulière de contrôle exprimée au travers d'un algorithme. Donc, seule la couche *commande* est abordée au sein d'un module. Actuellement, les descriptions données de ce système laissent l'aspect contrôle dans le flou le plus total. Le peu d'informations disponibles sur MEDUSA nous conduit ainsi à accompagner notre description de différentes interrogations.

---

<sup>7</sup>Des recherches sont actuellement menées sur l'apprentissage de telles stratégies [Dra 90]

## Conclusion

- Ce système met en œuvre un fonctionnement dans lequel à la fois le contrôle et l'échange de données sont centralisés. Le problème de l'accès aux ressources est ainsi facilement résolu avec toutefois la mise en place d'un important goulot d'étranglement que constitue le contrôleur.
- L'organisation prédominante sur laquelle raisonne le contrôleur est la tâche à réaliser.
- Chaque module de traitement est une méthode spécifique à une tâche.

### 3.3.6 Tea-1

Les activités menées actuellement à Rochester sont en prolongement direct des premières idées proposées dans [Bal 78] et reformulées dans [Bal 91] au travers du paradigme d'*animate vision*: le processus de vision est un ensemble de comportements qu'active un système plus général. Ces comportements mettent en œuvre des algorithmes très spécialisés concernant la tâche et les conditions d'application. Ils représentent uniquement les éléments qui leur sont nécessaires. Dans ce contexte, a été développé le système *TEA-1* [Rim 92b] [Rim 92a]. Il a pour objectif de répondre à des questions par une exploration contrôlée d'une scène (déjeuner) en modifiant la position de la caméra.

## Représentation et traitement

La représentation de la connaissance est organisée selon plusieurs *réseaux de Bayes* exprimant plusieurs hiérarchies : la décomposition ("part-of"), la spécialisation ("is-a"), le réseau de positions des objets dans la scène auquel est ajoutée l'expression de leurs relations spatiales. Un réseau de tâches, enfin, organise celles que peut satisfaire le système.

Les actions sont constituées de conditions d'activations et d'appels à des opérateurs visuels qui soit déplacent la caméra ou la région de plus grande résolution dans l'image, soit réalisent divers traitements d'image. Les actions sont directement reliées aux types d'objets qu'elles permettent de retrouver.

Le fonctionnement du système est orienté par les *demandes*. Les informations sont recherchées dans la scène jusqu'à ce que le degré de confiance dans la réponse obtenue soit suffisant. Le traitement itératif suivant est mis en œuvre : lister toutes les actions exécutables, sélectionner l'action avec la plus haute utilité, l'exécuter, affecter une certitude au résultat et la propager au travers des réseaux de *Bayes*. Seul le réseau de tâches réalise une combinaison des valeurs obtenues sur les autres réseaux.

## Contrôle

Le processus correspond à l'interpréteur des opérateurs visuels. Le contrôleur aborde la seule couche de commande. Il sélectionne les actions selon une fonction d'utilité. Le

critère d'arrêt du traitement s'exprime en terme de valeur de confiance suffisante pour la tâche correspondant à la demande formulée par l'utilisateur. La fonction d'utilité calcule le ratio de la valeur de l'action par son coût. Dans le calcul de cette valeur, sont pris en compte, à partir des réseaux de *Bayes*, l'information produite, l'impact sur la suite du traitement et la réduction possible de la région de l'image dans laquelle travailler. Deux choix sont ainsi combinés dans cette fonction : un choix de région et un choix d'action. La sélection de la région est réalisée à partir des valeurs du réseau de positions. Du fait de la formulation des actions, le contrôle de la caméra est inclus dans le contrôle de la résolution elle-même. Ces deux types de contrôle ne sont donc pas distingués dans le système.

## Conclusion

- Le fonctionnement du système *TEA-1* est dirigé par la demande. Il est initialisé par la génération de celle-ci. Le fonctionnement est *descendant*.
- Le contrôle de la caméra est directement en relation avec le but du système et avec la construction de la représentation.
- Bien que la fonction d'utilité incorpore un mécanisme permettant de prendre en compte une évolution possible du futur, elle reste statique. La seule couche de contrôle abordée est donc celle de la commande. Une des extensions que les auteurs de ce système veulent incorporer consiste en un système de planification qui permettrait de prévoir l'évolution de la résolution sur un plus long terme.

### 3.3.7 Vision As Process

L'objectif du projet "Vision As Process" (*VAP*) [Cro 89] (cf. figure 3.5) est d'étudier le contrôle de la perception dans un système de vision intégré. La vision est abordée comme un *processus* fonctionnant en continu. Lié à ce problème, ce projet traite aussi celui de l'*intégration* de différentes méthodes de vision au sein d'un même système. La *vision active* est abordée par l'utilisation d'un système stéréo de caméras mobiles opérant dans un environnement dynamique. Des contraintes de *temps réel* sont ajoutées au système. Dans cette perspective la limitation des ressources utilisées est primordiale. Dans ce rapport, nous présentons principalement la première version du système sur laquelle certaines de nos études sur le contrôle ont été réalisées (cf. chapitre suivant).

## Représentation et traitement

Le système (cf. figure 3.5) est constitué de quatre niveaux de représentation : l'image, la description 2D, la description 3D, l'interprétation symbolique. Chaque niveau possède un ou plusieurs modules dédiés à la construction de la description de la scène dans sa représentation en utilisant la description du niveau inférieur. Six modules sont utilisés : l'unité de contrôle de la caméra, deux modules construisant une description de l'image en deux dimensions pour respectivement l'image gauche et l'image droite, la

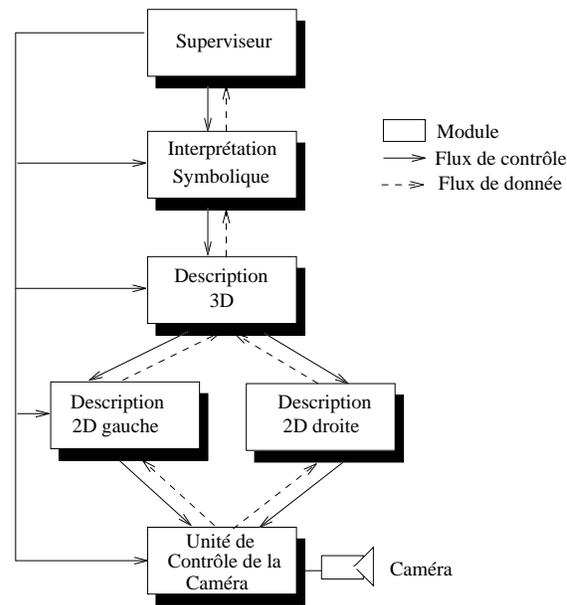
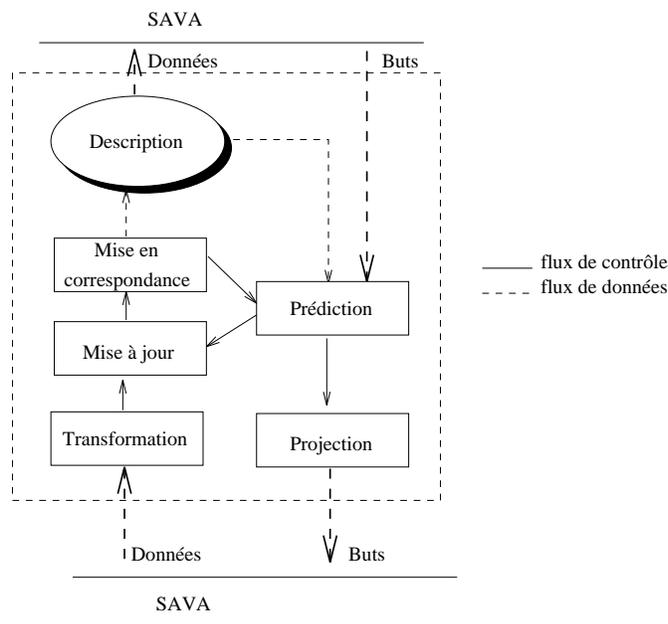
FIG. 3.5 - **Vision As Process : architecture du système** d'après [Cro 89].

FIG. 3.6 - **Vision As Process : module standard** d'après [Cro 89]. Le cycle *mise en correspondance*, *mise à jour* et *prédiction* réalise la cohérence temporelle de la description construite. La fonction de *transformation* traduit les informations en provenance d'un niveau de représentation inférieur en celle du niveau de représentation de module. La fonction de *projection* réalise l'opération inverse.

construction tridimensionnelle de la scène, un module d'interprétation symbolique et un superviseur. Chacun de ces modules a été développé indépendamment sur un même modèle : la transformation des informations reçues dans la description interne (fonction de transformation sur la figure 3.6), la gestion de la cohérence temporelle (cycle mise en correspondance, mise à jour et prédiction sur la figure 3.6), la prise en compte des buts d'un module supérieur et la communication de la description en retour (flèches épaisses sur la figure 3.6), l'émission des buts pour le module inférieur (fonction de projection qui fait la traduction des buts dans la description du module inférieur, figure 3.6). Un squelette d'intégration *SAVA* [Ben 90], [Ber 91] offre une structure de communication et permet une exécution répartie de chacun de ces modules sur différentes machines connectées sur un réseau ethernet.

## Contrôle

Tout comme nous l'avons fait dans le système *SCHEMA*, nous décomposons le contrôle de ce système en deux aspects : contrôle au sein d'un module et contrôle du système.

### 1. Contrôle au sein d'un module.

Le contrôle d'un module concerne la définition des buts, des actions et de la région de la description dans laquelle travailler. Généralement, la couche commande est la seule abordée par les contrôleurs de chacun des modules.

### 2. Contrôle du système.

Le contrôle inter-modules est exprimé par une organisation hiérarchique exprimant l'organisation naturelle que fournissent les niveaux de représentation : plus le niveau de représentation est élevé, plus le module qui l'occupe définit les commandes pour les modules situés sur les niveaux de représentation inférieurs. Surplombant les niveaux de représentation, nous trouvons le superviseur qui définit les buts des autres modules.

Les échanges inter-modules sont constitués de messages fixés mettant en place une stratégie de *prédiction vérification* entre chacun d'eux : un module d'un niveau donné communique des informations en réponse à un but formulé par celui de niveau supérieur. Celui-ci utilise ces informations pour construire sa propre description de scène. En fonction de celle-ci, il redéfinit un nouveau but pour le module de niveau inférieur.

L'introduction du temps et de la dimension de fonctionnement continu ont conduit à l'installation de *modes de commande* entre les modules propres à certaines activités du système. Ces modes de commande installent un module dans un fonctionnement répétitif de production d'informations pour le module de niveau supérieur.

L'accès à la commande du capteur, dans la première version du système, n'est pas complètement déterminé. Celui-ci est couplé avec les différents modules qui permettent

de lui donner des commandes formulées à partir de l'analyse de l'image. Le superviseur s'occupe de l'accès au capteur pour certaines commandes du type "définition de la région à regarder". Les commandes relatives à la valeur de certains paramètres de mise au point peuvent être fixées directement par un module donné. Le module de contrôle de la caméra [Cro 92] est organisé en une hiérarchie de contrôle. La couche supérieure consiste en une sélection d'actions en fonction d'un but.

## Conclusion

- Deux dimensions du contrôle existent : le contrôle au sein d'un module et le contrôle inter-modules.

Le contrôle au sein d'un module est réalisé avec la couche commande. Celle-ci détermine l'action que le module doit exécuter à partir de la région de l'image ou de la scène dans laquelle travailler et des buts fixés.

- Un superviseur central assure le contrôle du système en affectant à chacun la région dans laquelle travailler et la tâche à accomplir. Ce superviseur peut être aussi relayé par chacun des modules en suivant la hiérarchie de contrôle.

## 3.4 Contrôle dans un Système Intégré de Vision

Dans la section précédente, nous avons présenté des systèmes pour lesquels nous avons explicité l'aspect contrôle. Nous avons ainsi mis en évidence les mécanismes influençant leur comportement. Pour chacun de ces systèmes, leurs traitements sont structurés selon une hiérarchie de contrôle se superposant aux niveaux de représentation. Cette approche, apparue avec le paradigme *segmentation-interprétation* s'est généralisée avec l'apparition des niveaux de représentation. La vision active l'a confirmée en introduisant le capteur lui-même dans le cycle de contrôle du système. Ceci est illustré, par exemple, dans [Baj 88], où un système de vision active est présenté en terme d'une hiérarchie contrôleur-processus sur les différents niveaux de représentation.

Dans la table 3.2, nous proposons une classification des systèmes décrits précédemment selon quatre dimensions mettant en lumière leur approche du contrôle.

- La première colonne consigne les niveaux de représentation sur lesquels les couples contrôleur et processus sont définis au sein du système. Cette information n'est pas sans importance dans la mesure où les traitements mis en œuvre sur chacun des niveaux sont de nature différentes. Ainsi, selon le niveau sur lequel le contrôleur et le processus sont définis (au sein du même niveau de représentation ou non), les observations qu'ils échangent, font l'objet d'un changement de représentation ou non. La caméra apparaît dans ce tableau lorsqu'elle est un des processus contrôlés.
- La deuxième colonne présente les couches de contrôle abordées par les contrôleurs définis dans le système. Nous pouvons remarquer tout de suite que la couche

Système	Contrôleur Processus	Couche de Contrôle	Type des observations	Mécanismes de décision
SYSTÈMES DE VISION PASSIVE				
Barrow & Tenenbaum	au sein du même niveau	commande	valeur des propriétés	satisfaction de contraintes
Procédures Visuelles	<i>Activité liée à la sélection des procédures</i>			
	sur des niveaux $\neq$	commande	représentation centrale	planification
	<i>Activité liée à la sélection des Régions</i>			
Garvey	sur des niveaux $\neq$	adaptation	description de la scène	planification
		commande	description de la scène en cours	linéarisation du plan
SCHEMA	<i>Contrôle du système</i>			
	au sein du même niveau	commande	coef. de vraisemblance, conflits	maximisation de vraisemblance, résolution des conflits
	<i>Contrôle au sein d'un Schéma</i>			
	niveaux $\neq$	adaptation	hypothèses créées	maximisation de vraisemblance, cohérence
commande		hypothèses créées	stratégie d'interprétation, cohérence	
SYSTÈMES DE VISION ACTIVE				
<i>MEDUSA</i>	le processus contrôlé est la caméra	inconnu	résultats des méthodes d'inférence de formes spécialisées	inconnu
<i>TEA-1</i>	sur des niveaux $\neq$ et caméra	commande	coef. de vraisemblance,	maximisation de l'utilité
<i>VAP</i>	sur des niveaux $\neq$ et caméra	commande	informations entre niveaux	dépend du contrôleur

TAB. 3.2 - **Contrôle dans des Systèmes Intégrés de Vision.** Les systèmes Vision Passive sont en style normal, les systèmes de Vision Active sont en italique. La première colonne consigne les niveaux de représentation sur lequel le contrôleur et le processus sont définis. La deuxième colonne exprime les couches de contrôle utilisées au sein des contrôleurs ainsi définis. Nous trouvons dans la troisième colonne le type des observations à partir desquelles une commande est créée par les contrôleurs. Les mécanismes de décision présents au sein d'un contrôleur font l'objet de la dernière colonne. Le symbole  $\neq$  signifie différent. Le terme *inconnu* est utilisé lorsque les informations disponibles sur le système ne nous permettent pas de répondre.

décision n'apparaît jamais. Les contrôleurs définis au sein des systèmes de vision que nous avons considérés ne choisissent pas le but à résoudre. Celui-ci leur est donné par un autre système ou par l'utilisateur.

- Le type des observations à la base du contrôle est exprimé dans la troisième colonne.
- Les mécanismes de décision utilisés au sein du contrôleur sont inscrits dans la dernière colonne.

Dans cette section, nous identifions, dans un premier temps, les couples contrôleur-processus existant dans ces systèmes par rapport aux niveaux de représentation. Nous présentons ensuite les couches de contrôle abordés au sein de ces différents contrôleurs. Dans la dernière section nous nous intéressons à l'expression des modes de fonctionnement du système en terme de connaissances de contrôle, des commandes émises et notamment au *centre d'intérêt* ou focus sur lequel les traitements sont focalisés.

### 3.4.1 Contrôleur et processus

Lorsque nous parlons de contrôle en vision, les études réalisées en psychologie cognitive sur l'*attention visuelle* sont souvent évoquées. Cette dernière désigne la capacité d'appliquer préférentiellement les traitements visuels à un sous-ensemble de la scène. S'inspirant de son étude, nous distinguons le *contrôle externe* ("Overt control") et le *contrôle interne* ("Covert control"). Le premier désigne le contrôle de la caméra, le second celui des traitements mis en œuvre pour la construction de la description de la scène.

Pour ces deux types de contrôle, nous mettons en évidence les niveaux de représentation du système. Les traitements opérant sur ceux-ci sont utilisés pour contrôler les traitements du niveau inférieur. L'identification des couples contrôleur-processus, permet de mettre en évidence des modes de fonctionnement privilégiés : prédiction-vérification, génération d'hypothèses et de tests, etc.

#### a. Contrôle externe

Le contrôle externe<sup>8</sup> désigne dans le système visuel les éléments relatifs à la définition de la direction du regard, c.-à-d. les contraintes imposées à l'acquisition du signal visuel. Dans un système de vision, ce type de contrôle se matérialise par le contrôle de la caméra. Celui-ci, dans sa forme la plus générale, fixe les paramètres de prises d'images : les valeurs pour les six degrés de liberté de la *position* de la caméra, pour l'*ouverture*, pour le *focus* et pour le *zoom*. Dans le cas de deux caméras, ces paramètres sont doublés et nous tenons compte par ailleurs des relations entre les caméras, telle la *vergence*.

Le contrôle de la caméra prend différentes formes selon le mode de fonctionnement que l'on veut installer dans le système. Ce sont :

---

<sup>8</sup>Appelé aussi *Gaze Control*.

### Contrôle intra-niveaux de représentation

Le problème du contrôle de la caméra peut être abordé par un ensemble de *réflexes oculaires* réalisant un asservissement des paramètres de prise de vue à partir de mesures réalisées sur les images produites. Contrôleur et Processus sont ainsi définis au sein du même niveau de représentation. Les paramètres concernés sont principalement l'ouverture, le focus, et la vergence. Les fonctions d'estimation de l'état du processus d'acquisition emploient des mesures très rapides sur l'image : maximisation de la variance des pixels, somme des amplitudes des gradients sur différents niveaux d'une pyramide de résolution par exemple [Cro 92]. Le contrôleur est un ensemble de réflexes qui, à partir de mesures opérées sur l'image, engendre des commandes sur les moteurs de réglage des paramètres. Nous approfondirons cette remarque lorsque nous aborderons la réalisation des couches de contrôle présentes dans un contrôleur.

Ce type de contrôle installe un très fort *couplage* entre l'acquisition et l'analyse de l'image. Il se réalise indépendamment de tout but ou tâche assignés au système. Le contrôleur et le processus sont ainsi sur des niveaux de représentation successifs voire même confondus.

### Contrôle inter-niveaux de représentation

Cependant, la caméra pouvant être aussi directement contrôlée par la tâche à réaliser, ceci en fixant sa position et sa direction, le contrôleur peut aussi appartenir à d'autres niveaux de représentation.

Cette séparation du contrôle de la caméra et des modules de vision opérant sur les autres niveaux de représentation a été introduite, par exemple, dans [Baj 91] où un système de vision contrôle l'observation d'une opération de manipulation se déroulant dans la scène. L'opération est modélisée en terme d'un système dynamique à événements discrets. Des traitements de "bas niveau" sont utilisés pour extraire ces événements de l'image : calcul de flux optique, extraction d'informations tridimensionnelles par exemple. Leur détection déclenche un changement d'état du modèle de l'opération et entraîne la prévision de l'évolution probable de l'opération. Celle-ci se traduit alors en terme de mouvements de caméras permettant au système de vision d'observer les prochains événements.

Cette séparation entre le processus d'acquisition et le contrôleur est encore élargie dans d'autres systèmes avec, par exemple, l'utilisation d'informations provenant de la reconstruction stéréo. Le système *TEA-1* place ainsi sur le plus haut niveau de représentation du système le contrôleur déterminant la direction ou le fonctionnement de la caméra (cf. table 3.2).

#### b. Contrôle interne

Le contrôle interne désigne le contrôle de la résolution interne constituée de l'exécution des traitements agissant sur le signal acquis. Différentes études psychologiques

et physiologiques, notamment l'étude des lésions cérébrales, montrent que celui-ci peut très bien être découplé du contrôle externe [Tre 86].

L'interprétation d'images peut s'exprimer comme une suite successive de "mises en correspondance" entre les modèles et les données sur chacun des niveaux de représentation définis dans le système. Différentes méthodes la réalisent. On trouve notamment la propagation de contraintes, les isomorphismes de graphe, le raisonnement probabiliste (ex : *TEA-1*), la génération et le test d'hypothèses (ex : *SCHEMA*). Comme précédemment, les degrés de couplage entre le processus et le contrôleur sont liés aux niveaux de représentation sur lesquels ils sont définis. Cette étude du contrôle interne est commune à la fois aux systèmes de vision passive et à ceux de vision active.

### Contrôle intra-niveaux de représentation

Nous désignons ainsi le cas où *le contrôleur et le processus sont au sein d'un même niveau de représentation*. Les observations et les commandes échangées n'impliquent pas de changement de niveau de représentation. Nous trouvons un tel type de contrôle dans le système *SCHEMA*, dans le cadre de l'activation des différents schémas, au sein du plus haut niveau de représentation du système dans le blackboard global : les observations sont les hypothèses créées, les commandes sont les schémas manipulant ces hypothèses. Le système de *Barrow et Tenenbaum* emploie aussi un tel contrôle au travers de la satisfaction des contraintes de cohérence entre les valeurs de propriétés intrinsèques. Dans *VAP*, ce type de contrôle interne est illustré par le mécanisme de maintien de cohérence temporelle réalisé par un filtre de *Kalman* dans le module de description 2D [Dis 90] : les informations prédites par le filtre de Kalman et celles avec qui elles sont mises en correspondance sont des segments. Un tel contrôle est aussi présent aussi dans l'affinement de la qualité d'indices visuels, comme par exemple dans [Kah 90b] : la production de groupements perceptuels est affinée par leur évaluation, selon une fonction de qualité, entraînant une modification des paramètres de regroupement.

### Contrôle inter-niveaux de représentation

Le contrôleur et le processus sont sur des niveaux de représentation différents. Nous appelons *prédiction-vérification* le mécanisme consistant à contrôler les traitements, à un niveau de représentation donné, par l'utilisation de la description d'un niveau supérieur. La communication d'observations du niveau inférieur est suivie de leur transformation dans la description du contrôleur. Cette transformation peut être l'une des méthodes de traitement que nous avons présentées en début de chapitre : la segmentation, l'inférence de forme ou l'interprétation. La commande produite est fonction de l'adéquation du résultat de la transformation avec la description de la scène manipulée par le contrôleur. La commande contraint les traitements agissant sur le niveau de représentation inférieur à rechercher des informations dans leur description. Celles-ci doivent vérifier ou compléter l'hypothèse issue des observations communiquées au

contrôleur. Ce mode de fonctionnement définit ainsi un cycle de contrôle entre deux niveaux de représentation. Ces modes d'échanges de base se trouvent par exemple entre chacun des modules de *VAP*. Le mode de résolution au sein d'un schéma du système *SCHEMA* avec l'activation des sources de connaissances en est un autre exemple.

### c. Modes de fonctionnement

Une stratégie plus globale régit le cycle de *prédiction-vérification* entre deux niveaux de représentation. Elle est exprimée par les modes de fonctionnement du système qui sont *ascendant* ou *descendant*. Ils définissent la manière dont est initialisée le cycle *prédiction- vérification* dans le système, et la fréquence des retours d'un niveau à l'autre.

– *Le mode ascendant.*

Ce mode privilégie l'aspect *prédiction*, c.-à-d. le flux d'observations d'un niveau de représentation à l'autre. Ainsi un système fonctionnant selon un tel mode oriente ses traitements à partir des informations provenant des niveaux de représentation inférieurs : les images, par exemple, dans le système de Barrow et Tenenbaum. Notre définition de ce mode de fonctionnement ne préjuge en rien de l'utilisation ou non, des connaissances propres au système dans les traitements construisant la description. Elle diffère ainsi de [Baj 88] où un mode ascendant est associé à la non-utilisation dans les traitements de tout modèle dépendant du domaine.

– *Le mode descendant.*

Ce mode privilégie la *vérification*, c.-à-d. le flux de commande d'un niveau de représentation à l'autre. Un système fonctionnant dans un tel mode est dirigé par les commandes issues des niveaux supérieurs. Le mode descendant correspond à un mécanisme d'uniformisation des commandes dans le système, perpendiculairement aux niveaux de représentation : toute commande émise est la conséquence directe de la commande du niveau supérieur.

Nous nous plaçons, ici, du point de vue des traitements s'enchaînant d'un niveau de représentation à un autre. Si nous voulons faire une analogie avec le mode de fonctionnement d'un traitement au sein d'un seul niveau de représentation, nous assimilons le mode ascendant à un fonctionnement *dirigé par les données* (“Data-driven”) et le mode descendant à un fonctionnement *dirigé par les buts* (“Goal-driven”).

Une séparation nette entre ces deux modes de fonctionnement existe dans la plupart des systèmes. Le mode ascendant s'arrête à un niveau de représentation donné pour laisser la place au mode descendant (cf. table 3.3) : la construction d'une abstraction de la scène jusqu'à un certain niveau de représentation, l'interrogation selon les besoins des traitements des niveaux supérieurs. Un mixage des deux modes a été envisagé dans le système *SCHEMA* avec l'intégration de *Goldie* [Koh 87] qui permet d'exécuter des opérateurs de segmentation dans un mode dirigé par les buts. L'un des deux modes peut aussi être totalement absent et fournir un système dans lequel, par exemple, le mode descendant est privilégié : *TEA-1*, *VAP* ou le système de Garvey. Dans de tels

systèmes, en l'absence d'une commande, les images fournies par la caméra ne seront pas prises en compte.

Modes de Fonctionnement	Systèmes
Ascendant	Système de Marr [Mar 82]
Ascendant, Descendant	Barrow & Tenenbaum, Routines Visuelles SCHEMA,
Descendant	Garvey, <i>TEA-1</i> , <i>VAP</i>

TAB. 3.3 - **Modes de fonctionnement d'un système intégré de vision.** Le mode ascendant décrit l'enchaînement des échanges des niveaux les plus bas vers les niveaux supérieurs. Le mode descendant décrit l'enchaînement des échanges des niveaux le plus haut vers les niveaux inférieurs. Ils expriment ainsi quel est le niveau de représentation qui initialise les traitements.

#### d. Conclusion

- L'identification des couples processus-contrôleur conduit à la mise en évidence de deux options possibles. Ces options sont liées aux niveaux de représentations du système : contrôleur et processus appartiennent au même niveau de représentation ou à des niveaux de représentation différents. Nous remarquons que ces deux options sont communes au contrôle externe ou au contrôle interne. Nous pouvons donc, par la suite, supprimer cette distinction.
- La définition du contrôleur et du processus sur des niveaux de représentation différents fait apparaître un mécanisme de base d'échange d'informations : *prédiction-vérification*. Ce cycle lie les traitements de deux niveaux de représentation par une boucle de rétroaction des traitements du niveau supérieur vers ceux du niveau inférieur. Un mode de fonctionnement plus général vient contraindre ce cycle de base en orientant les échanges d'observation ou de commande sur plusieurs niveaux de représentation : modes ascendant et descendant.

### 3.4.2 Couches de contrôle

Nous laissons maintenant de côté le lien existant entre définition des processus-contrôleurs et les niveaux de représentations. Nous nous intéressons à la réalisation d'un contrôleur en terme de couches de contrôle.

Nous pouvons remarquer, à partir du tour d'horizon que nous avons fait, qu'aucun système parmi ceux décrits n'aborde la couche décision (cf. table 3.2). En effet, dans la mesure où tous sont dirigés par le but (cf. table 3.3), la commande issue de cette couche est fixée par l'extérieur. Ces contrôleurs n'ont donc pas à définir leurs buts.

### a. Couche de contrôle unique

Le contrôleur est le plus souvent constitué de la seule couche *commande*. Nous remarquons différents degrés de réalisation des mécanismes conduisant à la définition d'une nouvelle commande en fonction de l'évolution de la description.

### Comportement

La *vision comportementale*<sup>9</sup> restreint la description de la scène obtenue. Cette restriction est définie par rapport au phénomène à contrôler, tels la manipulation, le mouvement d'un objet. Les critères de sélection des traitements de construction de la représentation sont fixés et prédéfinis dans le code du comportement. C'est à ce titre que nous les considérons sur la couche *commande*. Ce type de réalisation est utilisé pour obtenir rapidement une information sur l'évolution de l'environnement perçu. On le retrouve, par exemple, dans les traitements et les représentations de *SONJA* [Cha 90] ou le système de [Whi 92], [Nel 91]. Il est utilisé pour la structurations des traitements tant d'un point de vue transversal aux niveaux de représentation qu'au sein d'un même niveau.

### Sélection

Une autre réalisation de la couche *commande* correspond à une sélection, en fonction de l'évolution de l'environnement, des traitements selon un critère fixé. Elle est utilisée dans la plupart des systèmes présentés (cf. table 3.2). Les réflexes oculaires intervenant dans le contrôle de la caméra, sont classés aussi dans cette catégorie. La couche *Device* définie dans *VAP* [Cro 92] ainsi que l'ensemble des processus coopérants, dédiés à chacun des paramètres de la caméra [Pah 92] en sont des exemples parmi d'autres.

### Planification

Une extension de cette approche sélective est illustrée par le système des *Procédures Visuelles* dans lequel un schéma d'activation est construit à partir d'une description de la scène et d'un but. Cependant, une fois construit, ce schéma est exécuté entièrement sans que l'environnement ne soit repris en compte en cours d'exécution.

Ces trois approches sont utilisées indifféremment en vision active, bien que l'approche comportementale semble être actuellement privilégiée. L'exigence de réaction rapide aux évolutions de l'environnement qui est imposée sur les traitements justifie cette tendance. La vision passive, n'étant pas confrontée à cet aspect, privilégie plutôt la deuxième approche.

---

<sup>9</sup>Le terme de comportement est utilisé ici dans un sens élargi par rapport à celui défini par Brooks [Bro 85] dans la mesure où l'élément déclencheur peut être aussi un but.

## b. Deux couches de contrôle

Des systèmes ont aussi mis en œuvre un contrôle abordant la couche *adaptation* : le système adapte sa loi de commande en fonction de l'évolution de sa résolution ou de l'environnement (cas plus particulier d'un système de vision active). Les mécanismes utilisés dans chacune de ces couches peuvent être du même type que ceux présentés ci-dessus. Nous nous intéressons ici à la présentation des interactions entre les deux couches de contrôle. Parmi celles-ci, deux types sont distingués :

### Planification ou stratégies d'interprétation

Dans le cadre de la vision passive, de tels fonctionnements sont illustrés par le système de Garvey et par la résolution prenant place au sein d'un schéma de *SCHEMA*. Dans le système de Garvey, le plan produit par la couche adaptation est linéarisé au fur et à mesure de l'évolution de la résolution dans la couche commande. Dans *SCHEMA*, la stratégie d'interprétation qu'active un schéma en fonction des buts et de la description courante (couche adaptation), module l'activation des opérateurs de description que déclenche la production d'une nouvelle information sur le blackboard local (couche commande).

### Modes de commande

En vision active, l'aspect continu du fonctionnement est accentué. La dimension temporelle joue un rôle important. Les traitements s'inscrivent dans un fonctionnement cyclique jusqu'à la production d'un événement dans l'environnement : la loi de commande est la même tant qu'elle n'est pas invalidée par l'environnement. Son intérêt, corroboré par différentes expériences psychologiques [Tre 86], apparaît lors de l'observation de scènes dynamiques : fonctionnement de poursuites, ou observations centrées sur un focus jusqu'à ce qu'un événement nouveau se produise.

Ce fonctionnement répétitif se retrouve dans la réalisation du contrôle externe avec les *modes de commande* utilisés dans [Cla 88] [Cla 89]. Dans ce système, plusieurs lois de commandes possibles de la caméra sont installées dynamiquement en fonction de la tâche à satisfaire. Les commandes, tout comme les observations, sont modulées selon la loi de commande en vigueur. A un niveau de représentation plus élevé, nous trouvons, dans le système de Califano [Cal 90], des *mécanismes de preattention* où le changement de fonctionnement de la couche commande est provoqué par une modification de l'environnement. Celle-ci invalide la loi de commande en vigueur. Une nouvelle loi de commande, plus adaptée à l'environnement, est alors redéfinie par la couche adaptation.

### c. Conclusion

La réalisation d'un contrôleur peut aborder une ou deux couches de contrôle :

- Lorsqu'une seule couche de contrôle est utilisée, sa réalisation se situe entre deux extrêmes : l'approche comportementale où l'enchaînement des traitements est fixé à la conception, et la planification où un traitement est sélectionné ou construit à chaque cycle en prenant en compte l'évolution de l'environnement.
- Dans le cas où deux couches de contrôle existent, deux expressions des conditions de passage de la couche commande à la couche adaptation sont définies : (i) l'échec ou la fin d'exécution de la loi de commande dans le cas où elle consiste en une stratégie d'interprétation ou un plan d'interprétation, (ii) l'apparition d'événements dans l'environnement dans le cadre des modes de commande. Dans ce deuxième cas, nous voyons apparaître l'expression explicite de conditions de validités d'une loi de commande.

#### 3.4.3 Connaissances de contrôle

Les modes de fonctionnement mis en place par un système de vision peuvent être fortement contraints ou non<sup>10</sup>. Cet aspect est surtout relatif à la définition et la prise en compte de critères de sélection conduisant le système à utiliser seulement une partie de ses traitements dans son fonctionnement.

- *preattentif*

Nous utilisons le terme de *preattentif* pour désigner un fonctionnement non contraint du système ou d'une partie de celui-ci. Un tel fonctionnement se traduit par une recherche ou une construction exhaustive de la description à un niveau de représentation. Dans la plupart des systèmes, ce niveau est celui de la construction d'indices d'images (ex : Le système de Barrow & Tenenbaum, les *procédures visuelles*).

- *attentif*

Le mode *attentif* caractérise un fonctionnement du système dans lequel les traitements sont sélectionnés. Le système oriente son fonctionnement dans l'attente d'un élément. On retrouve cette tendance par exemple dans les systèmes précédents pour lesquels le fonctionnement est essentiellement descendant.

Tout comme les remarques énoncées dans les sections précédentes, ces deux notions permettent de caractériser la structure interne du système. Les fonctionnements exhibés par ces systèmes offrent des structurations possibles des types de connaissance utilisant cette structure.

---

<sup>10</sup>Nelson [Nel 91] les appelle respectivement stratégie du plus fort engagement et stratégie du moindre engagement.

### a. Reconnaissance/Reconstruction

D'un point de vue fonctionnement on retrouve aussi deux caractéristiques :

– *Reconnaissance.*

La construction de la description de la scène répond à un but précis dont la satisfaction contraint à la fois les traitements et les données. L'interrogation de la description oriente les traitements. Ce fonctionnement est aussi appelé *teleonomique* ("purposif").

– *Reconstruction.*

La construction de la description de la scène est réalisée sans avoir de but défini à priori. L'interrogation de la description n'a aucune influence sur les traitements.

Ces fonctionnements se retrouvent dans les systèmes précédents (cf. table 3.4). C'est ainsi que dans *TEA-1* toute représentation ne sera construite que si elle est utilisable pour la tâche à satisfaire. Le système dans son entier est organisé pour la reconnaissance. En revanche, dans le système de Tenenbaum et Barrow, l'ensemble des propriétés intrinsèques de l'image sera produit indépendamment de toute utilisation ultérieure. Le fonctionnement de reconstruction apparaît ainsi dans la première partie du système.

Types de Fonctionnement	Systèmes
<b>Reconstruction</b>	Barrow & Tenenbaum
<b>Reconnaissance</b>	Routines Visuelles, Garvey, SCHEMA <i>TEA-1 VAP MEDUSA</i>

TAB. 3.4 - Types de fonctionnement : Reconnaissance/Reconstruction.

### b. Localisation/Identification

L'expression des lois de commande des systèmes précédents fait apparaître l'utilisation de deux notions qui selon la réalisation sont explicités ou non : sélection de l'opération à exécuter et sélection de la région dans laquelle l'exécuter. La mise en évidence et l'étude de cette dichotomie sont réalisées, par exemple, dans [Swa 91] au travers de l'indexation par la couleur.

Le système des *Procédures Visuelles* pratique un choix de région par l'indexation des procédures visuelles selon des caractéristiques de l'image. Un calcul complet de ces caractéristiques sur toute l'image permet ainsi la sélection de la région en fonction de la procédure choisie. Celle-ci est déterminée à partir d'un objet à retrouver. Garvey réalise lui aussi un choix de la région après avoir sélectionné les opérateurs de description. Ceux-ci sont déterminés aussi à partir de l'objet à chercher. Dans ces deux systèmes, la sélection de la région est réalisée à partir d'un calcul d'un ensemble de propriétés

sur toute l'image. Le choix de la région est ensuite une conséquence du modèle d'objet attendu dans le système.

Le système *TEA-1* exprime dans la représentation de ses connaissances un réseau de position mis à jour à l'exécution de chaque action. Les valeurs de ce réseau sont prises en compte dans le calcul de la fonction d'utilité permettant de réaliser le choix de l'action. A la différence des systèmes précédents cette notion rentre en ligne de compte pour la sélection du traitement. Dans le système *VAP*, une autre approche est encore utilisée. La région d'intérêt peut être fixée indépendamment des actions à réaliser. A la différence des autres systèmes elle peut apparaître explicitement dans l'expression de ses buts.

Nous distinguons ainsi :

– *Identification.*

L'identification est la recherche d'un objet parmi un ensemble de modèles disponibles dans une région donnée.

– *Localisation.*

La localisation est l'opération complémentaire de l'identification : rechercher la région dans laquelle un objet connu est situé. Par cette définition, la localisation ne désigne pas uniquement l'opération de détermination de tous les paramètres spatiaux de l'objet consistant en leur transformations géométriques [Lux 85].

### c. Centres d'intérêt

Dans ces modes de fonctionnement des éléments privilégiés apparaissent : les centres d'intérêt. Un *centre d'intérêt* ou *focus* est un élément sur lequel l'ensemble des traitements du système est concentré. A partir de celui-ci différents critères de sélection sont établis.

Tenant compte des caractéristiques précédentes relatives au fonctionnement du système, le centre d'intérêt possède des dimensions *spatiales*, *sémantiques* ou *temporelles* :

– *Centre d'intérêt spatial.*

La spécification de la région sur laquelle restreindre les traitements prend différentes formes selon le niveau de représentation (portion de la scène, portion de l'image, etc). Selon la représentation des connaissances elle peut être exprimée quantitativement ou qualitativement.

– *Centre d'intérêt sémantique.*

La focalisation sur un objet s'exprime en terme de propriétés intrinsèques, de contextes, de types d'objets, etc. Cette expression est fortement dépendante des mécanismes de représentation à la disposition du système.

– *Centre d'intérêt temporel.*

Ce dernier type de centre d'intérêt impose une restriction sur le déroulement du fonctionnement alors que les deux précédents dictent des restrictions sur les éléments sur lesquels appliquer les traitements.

Selon l'expression du centre d'intérêt prépondérant dans la prise de décision, on dira que le système est dirigé par la tâche ("Task driven"), par les propriétés ("Feature driven"), par le contexte ("Context driven"), etc. La décision est ensuite combinée avec des critères de coût (évaluation en temps), d'exactitude et de robustesse des traitements.

Cette notion de centre d'intérêt se traduit au sein des traitements de chacun des niveaux de représentation, mais aussi dans la combinaison des traitements entre les niveaux de représentation. Les systèmes présentés peuvent exprimer cette combinaison lors de la conception (ex : MEDUSA où des méthodes sont exprimées sur les niveaux de représentation allant de l'image au niveau le plus élevé). Le système *VAP* en revanche met en place cette combinaison en la traduisant sous forme d'interaction entre chacun des modules : à un instant donné, tous les modules travaillent sur la même région d'intérêt. Seule son expression change d'un niveau à l'autre. Nous trouvons aussi cette dimension dans les autres systèmes (cf. table 3.5).

Centre d'intérêt	Systèmes
Région d'intérêt	VAP
Modèles d'Objets	SCHEMA, Garvey
Propriétés Intrinsèques	Barrow et Tenenbaum, MEDUSA
Tâche	TEA-1, MEDUSA

TAB. 3.5- Expression d'un centre d'intérêt privilégié dans l'organisation du système.

### 3.5 Conclusion

Dans ce chapitre, nous avons défini le domaine de validation du problème du contrôle. Nous nous situons dans le cadre d'un Système Intégré de Vision, c'est-à-dire un système dans lequel une description de la scène est construite à différents niveaux de représentation. Bien que ces niveaux soient une structure d'organisation des traitements et des représentations, aucune structure de contrôle commune ne se dégage réellement dans ces systèmes. L'étude précédente des différents systèmes nous permet, cependant, de dégager les remarques suivantes :

- L'identification des processus et des contrôleurs s'exprime par rapport aux *niveaux de représentation* du système. A partir de cette structuration, des modes de contrôle entre ces différentes entités apparaissent : cycle de *prédiction-vérification* entre les niveaux de représentation, *génération et test* d'hypothèses au sein des niveaux de représentation. Ces modes de contrôle se traduisent au sein du système

en modes de fonctionnement définissant le flux de contrôle entre les traitements d'un niveau à l'autre : ascendant, descendant.

- Au sein de chacune de ces entités constituant les processus ou les contrôleurs, plusieurs couches de contrôle sont visibles. Leur réalisation met en évidence plusieurs réalisations possibles. Nous ne pouvons donc pas dégager de principe général pour leur définition.
- Du point de vue de l'expression des connaissances de contrôle, deux remarques se dégagent : (i) les critères de sélection font apparaître des dimensions spatiales, sémantiques, temporelles, (ii) plusieurs modes se combinent pour reproduire les fonctionnements de *localisation* et d'*identification* puis de *reconnaissance* ou de *reconstruction*. Cette remarque s'applique à l'expression du contrôle tant du point de vue du système que de celui de chacune des entités.

Bien que présentées séparément dans les sections précédentes, ces différentes remarques ne sont pas indépendantes. Le fonctionnement d'un système de vision résulte de leur combinaison. Les systèmes de vision actuels limitent les flux de contrôle possibles en particularisant leur structure de contrôle à une combinaison particulière (cf. table 3.4). Ainsi, l'architecture d'un système de vision ayant pour objectif l'étude du contrôle de la perception doit prendre en compte les problèmes suivants :

□ **Problème 1** *Identification des couples contrôleur-processus.*

Ce problème concerne l'organisation des traitements du système en terme de processus et de contrôleur. Les niveaux de représentation sont un facteur important d'organisation. Il est donc naturel de les utiliser comme élément de référence pour la définition des contrôleurs et des processus.

- Définition d'un processus : quels niveaux de représentation servent à sa définition ? Combien de niveaux recouvre-t-il ? Combien de processus existent dans le système ?
- Définition des contrôleurs : Etant donné un processus à contrôler, sur quels niveaux de représentation définir le contrôleur ? Combien de niveaux de représentation recouvre-t-il ? Y-a-t'il un ou plusieurs contrôleurs dans le système ?

□ **Problème 2** *Architecture d'un contrôleur.*

Étant donné un contrôleur quelles couches de contrôle utiliser pour le réaliser : couche commande, couche adaptation ou couche décision ? Transversalement à cette question, quelles sont les observations à prendre en compte pour le fonctionnement de chacune des couches de contrôle ?

□ **Problème 3** *Connaissances de contrôle.*

À partir des types de fonctionnements reconnaissance/reconstruction et leurs traductions en localisation/identification, quelles connaissances de contrôle utiliser ?

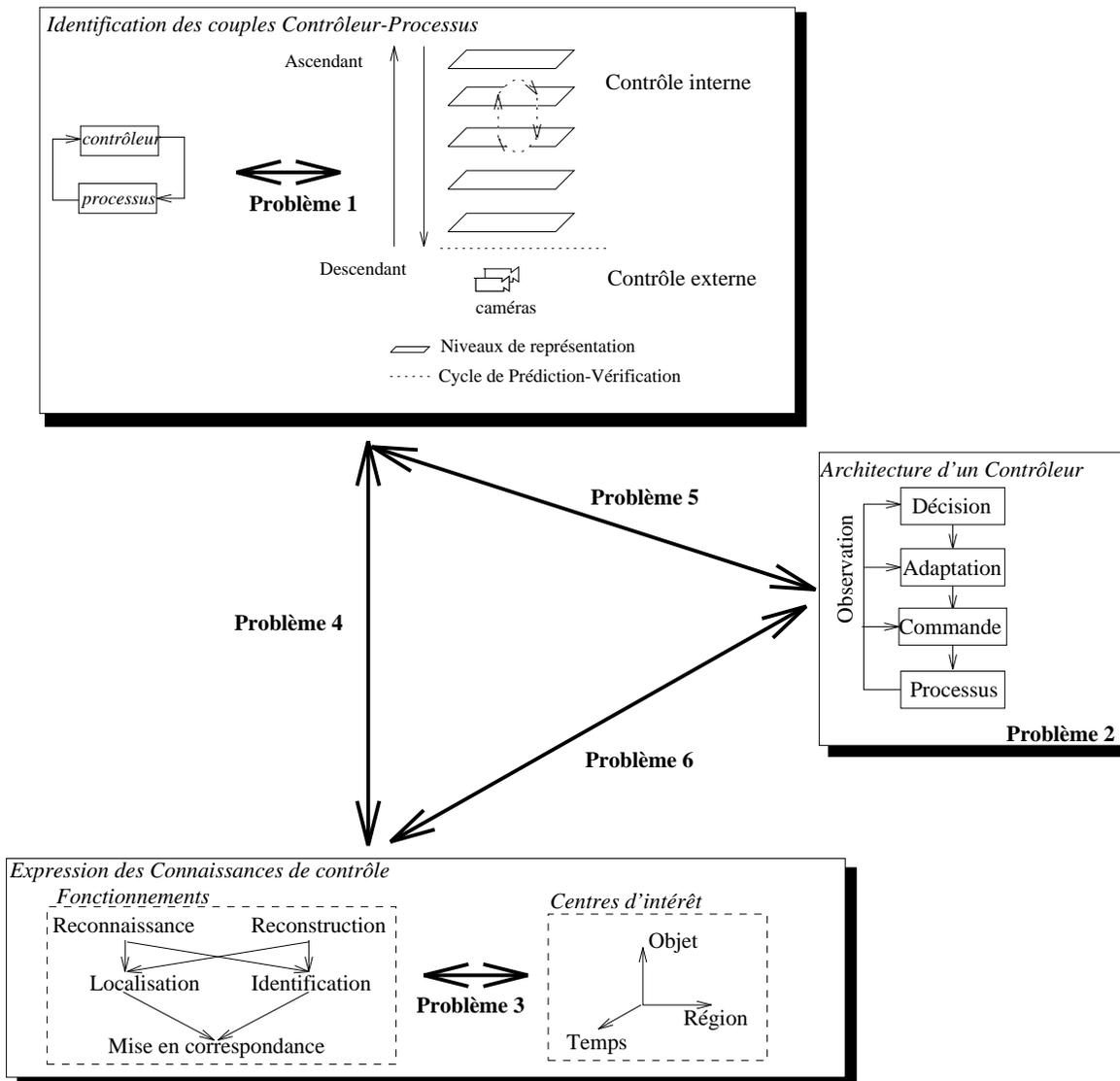


FIG. 3.7 - **Problèmes du contrôle dans un Système Intégré de Vision** Le contrôle d'un système intégré de vision s'exprime: (i) par une structuration du système en terme de couples contrôleur-processus (problème 1) avec des modes d'interaction particuliers, (ii) le choix d'une architecture de contrôle pour chacun des modules de traitement (problème 2), (iii) l'expression des fonctionnements en connaissances de contrôle (problème 3), (iv) l'expression des connaissances de contrôle par rapport au système dans sa globalité (problème 4), (v) la structuration de chacune des entités selon leur activité dans le système (problème 5), (vi) l'expression des connaissances de contrôle au sein de chacune des entités (problème 6).

Une autre question plus critique pour le développement de la structure de contrôle est relative à la définition des critères de sélection à utiliser.

Nous étudions maintenant les interactions entre chacun des problèmes présentés ci-dessus.

□ **Problème 4** *Relations entre connaissances de contrôle et la définition des contrôleur-processus au sein du système.*

Ce problème possède deux aspects dûs à la prise en compte des niveaux de représentation :

- Inter-niveaux de représentation. Existe-t-il une uniformité de définition des centres d'intérêt entre les niveaux de représentation ?
- Intra-niveaux de représentation. Quelles sont les relations entre les traitements fournissant une description au sein d'un même niveau de représentation ? Y-a-t'il un ou plusieurs centre d'intérêt définis ? Plusieurs centres d'intérêt au sein d'un niveau lié à l'existence, par exemple, de plusieurs méthodes d'inférences de forme.

Cette problématique a plusieurs répercussions au niveau des contrôleurs : comment permettre une activité dirigée par les données, par les buts ? Au niveau du système comment mixer dynamiquement les modes ascendant et descendant ? Comment permettre à un processus de fonctionner avec ou sans centre d'intérêt ? (preattentif, attentif)

□ **Problème 5** *Architecture d'un module par rapport à sa fonction de contrôleur ou de processus dans le système.*

En fonction de la définition du processus et du contrôleur, quelle est la formulation du contrôleur en terme de couches de contrôle ? La définition de l'architecture des contrôleurs en terme de couches de contrôle est-elle uniforme dans le système. Selon les niveaux de représentation recevant les contrôleurs et les processus, le problème d'observation est en parti résolu : une commande est élaborée à partir de la description issue du niveau sur lequel le processus est défini.

□ **Problème 6** *Relations entre connaissances de contrôle et l'architecture définie pour chacun des modules.*

Quelles sont les couches de contrôle utilisées pour définir les dimensions spatiales, sémantiques et temporelles d'un centre d'intérêt ? Les décisions par rapport à ces deux aspects doivent-elles être abordées selon les mêmes couches de contrôle ?



## Chapitre 4

# Propositions pour le contrôle d'un Système Intégré de Vision

Dans ce chapitre, nous exposons, dans un premier temps, le résultat d'expériences de contrôle réalisées lors du développement du système *VAP*. Ces expériences nous ont permis de formuler les exigences que doit satisfaire la structure de contrôle du système que nous proposons dans le chapitre suivant.

Nous présentons ensuite le projet *SATURNE* [Dem 86]. Son objectif est de définir une *architecture* stable d'*intégration* de plusieurs méthodes d'inférence de formes. Notre travail contribue à la définition du contrôle dans cette architecture tout en étendant *SATURNE* vers un système dynamique.

Ces deux systèmes proposent des réponses aux problèmes évoqués en conclusion du chapitre 3. Leurs apports complémentaires (intégration pour *SATURNE* et contrôle pour *VAP*), nous ont conduit à nous tourner vers la formulation de *modèles généraux* s'appuyant sur l'*Intelligence Artificielle Distribuée*. Nous présentons rapidement ce domaine de recherches afin, d'une part de définir le vocabulaire utilisé et, d'autre part, de présenter les motivations pour ce type d'approche.

Nous concluons en présentant les propositions à la base de la structure de contrôle de notre système. Celles-ci seront reprises et développées dans le chapitre suivant lors de la présentation des modèles.

## 4.1 Expériences de contrôle dans le système VAP

Le projet *VAP* est un projet Européen (BRA3038) dans lequel plusieurs partenaires<sup>1</sup> se sont associés pour construire un *système intégré de vision*. Ses objectifs sont la construction d'un système de vision *active* fonctionnant en continu et en temps réel.

Le système est décomposé en six modules s'inscrivant chacun sur un niveau de représentation : l'unité de contrôle de la caméra, la description 2D (un module pour l'image gauche et un module pour l'image droite), la description tridimensionnelle, l'interprétation symbolique et le superviseur (cf. figure 3.5). Ce dernier module est dédié à la définition du contrôle du système. L'abstraction liée aux niveaux de représentation induit une hiérarchie de contrôle dans le système.

En plus du développement du contrôleur de la caméra et d'une partie du module de construction de la description 2D, le LIFIA a le rôle d'*intégreur* du système. C'est ainsi qu'un *Squelette d'Application pour la Vision Active* (SAVA) a été développé. Celui-ci reçoit chacun des modules développés séparément et permet de tester le fonctionnement du système ainsi obtenu. SAVA permet une exécution répartie des modules sur plusieurs stations de travail (Sun 3 et Sparc) reliées par un réseau ethernet. Ecrit en C, il fonctionne sous environnement UNIX Berkeley BSD 4.2. Il offre un langage de communication et un module standard (cf. figure 3.6, décrit dans le chapitre précédent) dans lequel chacun des modules peut être exprimé. Des cartes de traitement spécialisées (extraction de contours, calcul de pyramide de résolution et suivi d'indices) ont été développées au LTIRF et intégrées dans le système. Une tête binoculaire développée au LIFIA, montée sur le bras articulé d'un robot mobile est ajoutée. Elle sert de banc d'études de la vision active.

Dans le cadre de ce projet, nous avons eu deux axes de travail : *intégration* des modules et *étude du contrôle*.

Nous avons ainsi participé au développement du *squelette* d'intégration pour les différents modules développés par chacun des partenaires. L'utilisation de deux versions de celui-ci, SAVA1.0 [Ben 90] et SAVA2.0 [Ber 91], a conduit à la formulation de principes pour l'intégration des modules au sein du système. Nous les retrouvons dans la conception de la structure de contrôle de l'architecture de notre système.

L'autre aspect de notre travail est directement lié à l'étude du contrôle dans ce système. Nous avons élaboré un *cadre d'expérimentation* sur la base de SAVA et d'un sous ensemble du système VAP. Nous nous sommes plus particulièrement intéressés aux interactions entre les modules d'*interprétation* et de *description 2D*, ainsi qu'aux répercussions induites sur le fonctionnement de chacun d'eux.

Nous avons ainsi testé une des solutions possibles aux problèmes présentés dans la conclusion du chapitre précédent.

---

<sup>1</sup>University of Surrey (Grande Bretagne), Aalborg University (Danemark), Royal Institute of Technology (Suède), Linköping University (Suède) et enfin le LTIRF et le LIFIA deux laboratoires de l'Institut National Polytechnique de Grenoble.

### 4.1.1 Contrôle au sein de la segmentation et de l'interprétation

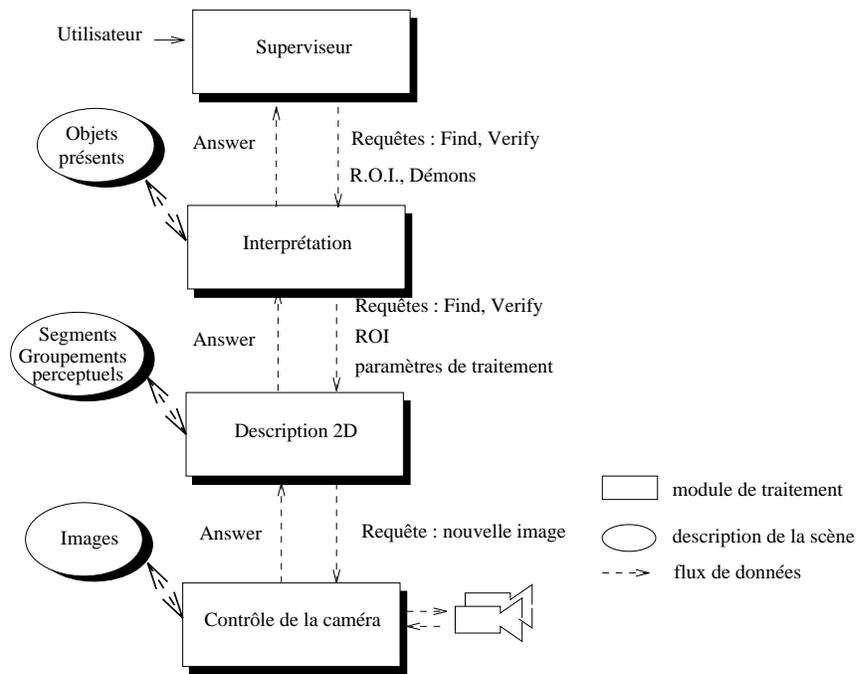


FIG. 4.1 - **Cadre d'expérimentation pour l'étude du contrôle.** Le cadre d'expérimentation pour l'étude du contrôle est constitué d'un sous-ensemble des modules de VAP : caméra, description-2D, interprétation, superviseur. Ces quatre modules communiquent entre eux au travers de SAVA.

### Cadre d'expérimentation

Le cadre d'expérimentation utilisé est constitué de quatre modules communiquant au travers de SAVA (cf. figure 4.1) :

- Un module de *commande de la caméra* acquiert dynamiquement les images. Dans nos expériences, le problème du contrôle de la caméra n'est pas abordé. Ce module ne fait qu'une acquisition d'images dans une scène dynamique sans que nous ne définissions ou ne modifions les paramètres d'acquisition tels la vergence, le focus ou l'ouverture.
- Le module de *description 2D* exprime la description de la scène en termes de segments à différents niveaux d'une pyramide Multi-résolutions à partir de l'image fournie par le module précédent.
- Le module d'*interprétation* interroge et contrôle le module de description 2D pour construire sa description. Cette dernière s'exprime en terme d'une description géo-

métrique bidimensionnelle (formes géométriques simples). Le module d'interprétation n'est pas plus développé dans la mesure où nous nous intéressons seulement aux interactions entre les modules.

- Un module *superviseur* constitue l'interface pour l'utilisateur qui définit les buts et le centre d'intérêt spatial (Région d'intérêt R.O.I.) dans lesquels travaille le système. Ceux-ci sont transmis au module d'interprétation ou au module de description 2D.

Nous présentons les fonctionnalités de contrôle des modules de description 2D et d'interprétation. Etant réduites au minimum, les fonctions de contrôle des modules *superviseur* et *commande de la caméra* ne sont pas présentées.

### a. Module de description 2D

Une première version de ce module a été développée par M. Sjölin [Sj 90]. Nous avons ensuite assuré la maintenance et l'intégration de modifications en fonction des exigences induites par nos expériences.

Ce module, écrit en C, est exprimé dans le module standard de SAVA (cf. figure 3.6). Il construit une description en terme de segments de contrastes sur laquelle une maintenance temporelle est réalisée : suivi de ces segments d'une image à l'autre [Dis 90]. La description est faite à plusieurs niveaux d'une pyramide multi-résolutions [Cro 84] (cf. figure 4.3). Le système de communication peut accéder à cette description par accès associatif au travers de groupements perceptuels [Cal 91].

## Contrôle

L'ensemble des traitements s'insère dans le cycle de contrôle représenté par la figure 4.5. Les centres d'intérêt suivants modifient le fonctionnement : les paramètres de traitement appliqués sur chacune des procédures, la région d'intérêt.

- *Paramètres de traitement.*

Chacun des traitements (segmentation, suivi, groupement perceptuel) a un fonctionnement paramétré (valeurs de seuils, de qualité par exemple). La valeur de ces paramètres est fixée exclusivement par le module d'interprétation à partir des résultats que lui fournit ce module.

- *Région d'intérêt.*

La *région d'intérêt* (R.O.I)<sup>2</sup> (cf. figure 4.4) est l'autre centre d'intérêt mis en place. Dans ce module, six paramètres la définissent : son centre  $(x, y)$ , sa largeur en  $x$  et en  $y$  ( $dx, dy$ ), le premier niveau  $k$  de la pyramide Multi-résolutions qu'elle concerne et le nombre de niveaux pris en compte à partir de celui-ci  $dk$ . A l'exclusion des traitements effectuant la maintenance temporelle, tous les traitements

---

<sup>2</sup>Définition spatiale du sous-ensemble de la description sur lequel concentrer les traitements.

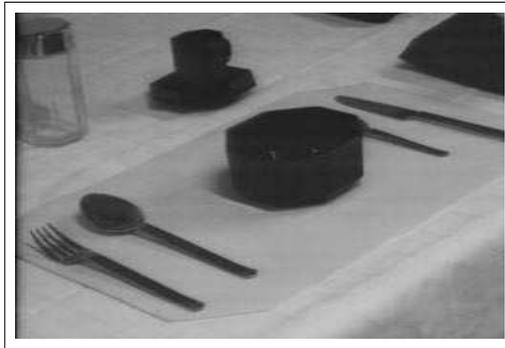


FIG. 4.2 - Image d'une scène

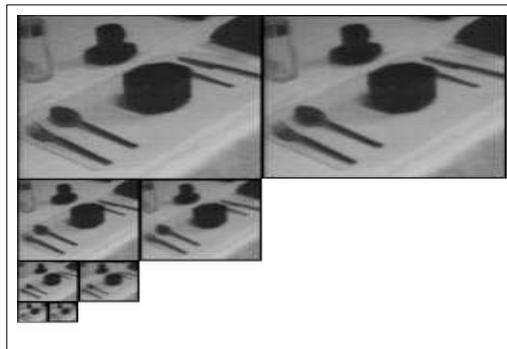


FIG. 4.3 - **Pyramide multi-résolutions.** Cette pyramide est obtenue par une cascade de convolutions de l'image avec des filtres binomiaux suivies de rééchantillonnage de l'image obtenue.

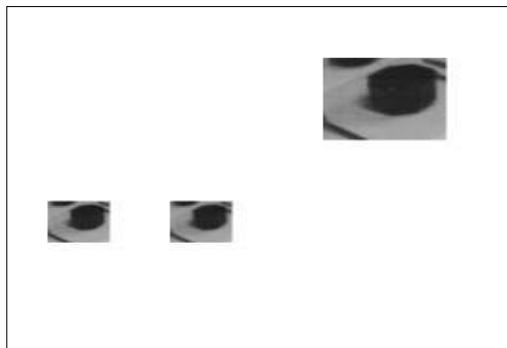


FIG. 4.4 - **Région d'intérêt dans la pyramide multi-résolutions.** La région d'intérêt sur les niveaux 4, 5 et 6 de la pyramide.

L'ensemble des traitements du module description 2D s'effectueront dans ces régions de l'image.

---

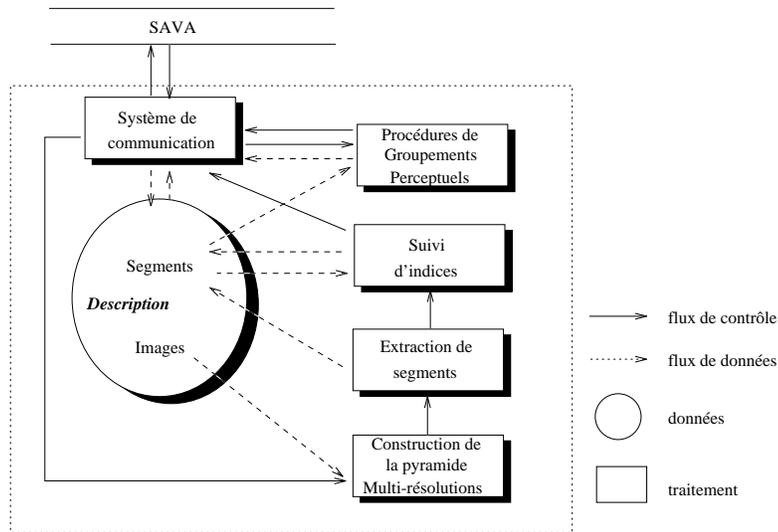


FIG. 4.5 - **Architecture du module de description 2D.** Pour toute nouvelle image reçue, une pyramide multi-résolutions est construite. Les segments sont ensuite extraits dans cette pyramide, sur les niveaux et dans les régions spécifiés par la région d'intérêt. Ils sont ensuite mis en correspondance avec la description issue de l'image précédente afin de réaliser un suivi de ces segments d'une image à l'autre. La description ainsi obtenue est interrogée par le système de communication en utilisant un mode d'accès associatif selon différents groupements perceptuels. Cette interrogation de la description a un mode de fonctionnement répétitif possible dans lequel régulièrement des messages sont émis. Ce module est écrit dans le Module standard : la construction de la pyramide ainsi que l'extraction de segments correspondent à la fonction de transformation, le suivi d'indice est constitué des trois fonctions mise à jour, prédiction et mise en correspondance.

du module sont paramétrés par la R.O.I. : la pyramide et la description ne sont construites que dans cette région.

Ces centres d'intérêt étant fixés par le module d'interprétation, le contrôle de ce module sur lui-même est inexistant. Les expériences ont cependant montré que la disparition d'un indice de la description peut être due à une réelle disparition de l'objet dans la scène tout comme à une mauvaise valeur de paramètre de segmentation ou de suivi. Il serait donc intéressant d'installer une rétroaction des traitements de cohérence temporelle sur les traitements du module lui-même. Dans ce cadre d'expérimentation, le fait que seul le module d'interprétation puisse réaliser cette opération, implique, d'une part des échanges de données entre ces deux modules, et, d'autre part la présence au sein du module d'interprétation de connaissances relatives à des traitements du module de description 2D. Le développement des diverses stratégies de contrôle (utilisation de la pyramide Multi-résolutions pour la recherche d'indices par exemple) est freiné au sein de ce module. Il ne peut se faire que par l'intermédiaire du module d'interprétation.

## b. Module d'interprétation

Le module d'interprétation (cf. figure 4.6) est bâti sur une architecture de blackboard *GBB* [Cor 88] écrite en *Common Lisp*.

### Représentation

Les différentes représentations manipulées au sein de ce module sont organisées en trois tableaux. Les éléments s'y inscrivant s'expriment sous la forme classique de *frames* [Min 75] avec différents attributs et sont liés entre eux. Un coefficient de vraisemblance est ajouté à chacun. Tous les éléments sont organisés dans les tableaux selon leurs coordonnées et leur date de création :

– *tableau du domaine.*

Il est décomposé en un panneau de *données* et en un panneau d'*hypothèses*. Ce dernier est décomposé en niveaux d'abstraction : segments, groupements, plans, objet. Il regroupe les informations créées localement, à partir d'autres hypothèses ou des données reçues. Le panneau de *données* contient les informations en provenance des autres modules : superviseur et module de description 2D. Une représentation explicite d'une partie de la description des autres modules est ainsi disponible. Tout but à satisfaire est inscrit comme une hypothèse sur le blackboard du domaine avec un coefficient de vraisemblance négatif.

– *tableau de contrôle.*

Il est organisé selon les niveaux classiques que l'on trouve dans *BB1* [Hr 85]. Les informations de contrôle sont distinguées en *région d'intérêt* et en critères de sélection. Seuls les premiers niveaux de ce tableau de contrôle ont été utilisés dans le cadre de ces expériences : une représentation explicite des problèmes résoudre est ainsi absente dans ce module.

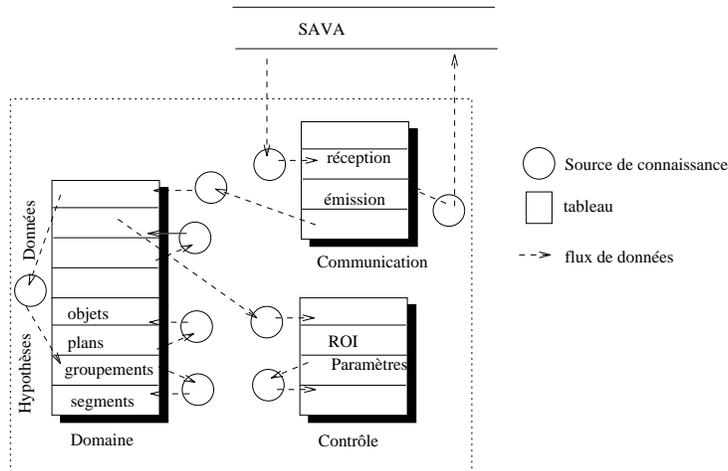


FIG. 4.6 - **Architecture du module d'interprétation.** Cette architecture est implantée à l'aide d'une architecture de blackboard. Les informations manipulées sont représentées par trois tableaux : tableau du domaine constitué des panneaux de *donnée* et d'*hypothèses*, tableau de communication, tableau de contrôle. Différentes sources de connaissances regroupent les connaissances permettant de manipuler ces informations.

– *tableau de communication.*

Il regroupe les requêtes relatives aux centres d'intérêt (requêtes de contrôle) ou aux informations constituant la description (requêtes de donnée) ainsi que les réponses reçues ou à émettre. Il gère la communication de ce module avec les autres modules du système.

## Traitements

La résolution est organisée selon le paradigme de *génération et de vérification* d'hypothèses. Les sources de connaissances du domaine sont classées en deux groupes selon qu'elles réalisent une génération ou une vérification d'hypothèse. Elles expriment les connaissances relatives aux modèles d'objet (formes géométriques simples).

- sources de connaissance de génération : leur exécution crée de nouvelles hypothèses sur le tableau de domaine ou augment le coefficient de vraisemblance d'hypothèses préexistantes. Des liens issus de cette hypothèse sont créés en direction des données ou des hypothèses qui la supportent.
- sources de connaissance de vérification. Deux types de vérifications sont possibles selon le déclencheur de la source de connaissance : but à satisfaire ou hypothèse dont le coefficient de vraisemblance n'est pas assez élevé. Dans ce dernier cas, l'hypothèse doit être vérifiée soit en recherchant des indices visuels particuliers dans la description locale en parcourant les liens entre les hypothèses, soit en

demandant des informations supplémentaires au module de description 2D.

Deux sources de connaissance interviennent sur le blackboard de communication : source de connaissance d'émission et source de connaissance de réception. Elles réalisent la traduction, dans le langage de représentation de ce module, du contenu des messages exprimé dans le langage de communication par l'utilisation d'un répertoire pour chacune des interactions.

## Contrôle

Le cycle de contrôle est celui des architectures de blackboard : (i) déclenchement des sources de connaissance à partir des événements créés par l'exécution de la source de connaissance du cycle précédent, (ii) activation des sources de connaissance déclenchées par exécution de leur partie condition, (iii) sélection des sources de connaissance activées, (iv) exécution de la partie action de la source de connaissance choisie.

Dans le cadre de notre expérience, le critère de sélection des sources de connaissance activées est fixe. Il consiste à privilégier la source de connaissance portant sur l'élément d'un tableau ayant la valeur absolue de coefficient de vraisemblance la plus élevée. La représentation d'un but avec un coefficient de vraisemblance négatif permet de réaliser un fonctionnement dirigé soit par les données, soit par les buts.

Parallèlement à l'utilisation de ce critère de sélection, une R.O.I. définit l'ensemble des éléments des tableaux qui peuvent être pris en compte pour la résolution. Cette opération est facilitée du fait de leur indexation selon leurs coordonnées spatiales. Des expériences ont été réalisées pour que le module puisse modifier lui-même sa R.O.I et puisse aussi définir la R.O.I du module de description 2D. Les modules d'interprétation et de description 2D ne fonctionnant, à un instant donné, que dans une seule région de l'image, la création d'une R.O.I entraîne la désactivation de la précédente.

Cette phase coûteuse de sélection des sources de connaissance est nécessaire lors de la génération et de la vérification d'une hypothèse. En revanche, lorsque l'hypothèse est établie, son suivi doit être réalisé automatiquement selon un plan déterminé. L'architecture de blackboard que nous avons utilisée ne permet pas d'installer dynamiquement la répétition d'un ensemble de fonctionnements correspondant à l'exécution d'une séquences de sources de connaissance. Cette lacune est détournée en partie par le mécanisme de démon mis en place dans le module de description 2D. Celui-ci permet la mise en place par le module d'interprétation d'un mode d'émission automatique d'éléments de la description 2D : par exemple, à chaque cycle, envoie au module d'interprétation des segments ayant une qualité de 1 situés dans la R.O.I. Bien que le contrôle de ce module soit coûteux, il est surchargé, en plus, par la définition des paramètres et de la R.O.I du module de description 2D.

## 4.1.2 Contrôle dans le système

### a. Protocole d'interprétation

Le système SAVA offre un mécanisme de communication par messages entre les modules. Les actes de communication mis en œuvre sont très restreints. Ils sont décomposés en deux ensembles qui, selon la terminologie de *VAP*, sont :

– *Requêtes de données.*

Dans *VAP*, ces requêtes expriment des échanges de but entre les modules. Dans toute requête, la R.O.I. peut être spécifiée. En cas d'absence de celle-ci dans la requête, la R.O.I. existant dans le module récepteur est utilisée. Les primitives du langage de communication sont :

- *Find*: une interaction de ce type déclenche la recherche d'un élément de la description satisfaisant la spécification précisée après le mot clef *find*. Par exemple, (*find jonction 1.57 1.32 10*) déclenche la recherche de jonctions faisant un angle droit, l'incertitude permise sur cette valeur est de 1.32, les deux extrémités de la jonction sont distantes au maximum de 10. La réponse à une telle requête est la liste des identificateurs des données trouvées.
- *Verify*: vérification de la présence d'un indice satisfaisant les caractéristiques précisées dans l'interaction. La syntaxe utilisée est similaire à la précédente.
- *Get*: recherche des paramètres d'un indice de la description dont l'identificateur est communiqué dans le message. Ce message permet par exemple la mise à jour des informations d'une donnée préalablement communiquée.

– *Requêtes de contrôle.*

Elles sont définies à partir des deux primitives *Get* et *Set* conduisant à obtenir ou à fixer la valeur des centres d'intérêt du module récepteur : R.O.I. et paramètres de fonctionnement. Par exemple, (*set seuil-haut 10*) fixe 10 comme valeur du seuil haut de la segmentation par seuillage par hysteresis utilisé dans le module de description 2D.

Une requête de contrôle supplémentaire installe, à la demande du module d'interprétation, des fonctionnements répétitifs (démons) dans le module de description 2D. Sa nécessité est apparue dans des tâches de suivi d'objet : une fois que le module d'interprétation a établi le lien entre des segments et un modèle d'objet, son suivi dans la scène est réalisé dans le module 2D par le suivi des segments correspondant à l'objet. Ainsi par ce mécanisme de démon, l'interprétation installe des traitements particuliers dans le module description 2D pendant un nombre variable de cycles de contrôle.

### b. Organisation

Un autre aspect du contrôle s'exprime dans la *hiérarchie de contrôle* régissant l'activité du système. Celle-ci se trouve en partie exprimée dans les interactions précédentes

mais aussi dans la constitution des modules eux-mêmes.

– *Expression de la hiérarchie dans les interactions.*

Le fonctionnement d'un module est fixé exclusivement par les interactions en provenance du module situé sur le niveau de représentation supérieur. Ces interactions précisent le but, les paramètres de fonctionnement ou la R.O.I. Les *échanges* de donnée ne sont possibles que par réponse à une requête. Cette particularité est due au contexte de *VAP* qui ne prend en compte que des interactions inter-niveaux selon un fonctionnement dirigé par les buts.

– *Expression de la hiérarchie au sein des modules.*

L'autre élément exprimant la hiérarchie de contrôle se manifeste dans chacun des modules par le principe guidant leur résolution. La détection de conflits entre des hypothèses dans une description d'un niveau de représentation donné engendre des requêtes de vérification vers le module situé sur le niveau de représentation inférieur. Ainsi la description de celui-ci est construite à partir des exigences de cohérence du module supérieur. Il en est de même lors d'informations manquantes. Dans le cadre de notre expérimentation, la cohérence de la résolution est assurée par le module d'interprétation qui se charge de spécifier les buts et les paramètres de fonctionnement au module de description 2D (cf. figure 4.1).

### 4.1.3 Evaluation des expériences de contrôle dans VAP

Les capacités de contrôle dans cette première expérimentation sont le résultat de choix de solution par rapport aux problèmes 1, 2, et 3 du chapitre précédent et de limitations imposées par SAVA. Elles ont conduit à la solution suivante :

- Une hiérarchie de contrôle définit les couples *contrôleur/processus* dans le système (*Problème 1*) : le module d'interprétation définit le but, la région d'intérêt, R.O.I. et les paramètres de contrôle du module de description 2D. Les interactions possibles entre les modules reproduisent le cycle de prédiction-vérification. Un mode descendant est inscrit dans le système.

Un fonctionnement séquentiel limite le découplage des niveaux de représentation. Chaque module réalise ses traitements à la fréquence de ceux du module supérieur. Or les informations manipulées à un niveau désignent des phénomènes dans la scène dont la vitesse de modification est différente : un objet (niveau scène) peut être présent alors que certains de ses contours peuvent disparaître (niveau description 2D).

- Le module d'interprétation n'utilise que la couche commande (*Problème 2*). Aucune couche de contrôle n'apparaît dans le module de description 2D. Cet aspect est dû à la définition de la hiérarchie de contrôle : le module description 2D est entièrement contrôlé par le module d'interprétation. De ce fait, de nombreuses interactions sont mises en place entre ces deux modules conduisant à un très fort couplage entre eux.

Pour réaliser des tâches de suivi, la mise en place d'un fonctionnement automatique et répétitif est nécessaire dans un module. Ainsi le mécanisme de "démons" a été introduit dans le module de description 2D. Ce mécanisme permet la répétition de l'interrogation de la description selon les mêmes critères d'un cycle à l'autre. D'une part ce mécanisme est réservé à l'interrogation de la description et d'autre part il accentue le couplage entre les deux modules en permettant au module d'interprétation d'installer des fonctionnements sur plusieurs cycles dans le module de description 2D.

Du point de vue de l'intégration, cette solution est difficilement viable dans la mesure où des connaissances importantes relatives au fonctionnement d'un module doivent être présentes dans le module situé sur le niveau de représentation supérieur.

- La reconnaissance est exprimée principalement dans notre cas par un fonctionnement d'identification (find) et de mise en correspondance (verify). Les centres d'intérêt utilisés sont exprimés par la R.O.I. et les paramètres des traitements (*Problème 3*). La R.O.I. se retrouve sur tous les niveaux de représentation.
- L'émission d'un but et d'une R.O.I. par un module se répercute dans le module inférieur en une succession de spécification de R.O.I. et de buts déduits de ceux reçus (*Problème 4*). Ainsi, à un instant donné, le système peut être décrit par une *barre de traitement* ou *focalisation* correspondant à la R.O.I. et au but en cours d'examen. Les centres d'intérêt sont uniformisés dans le système à tout instant.
- Le module d'interprétation n'a qu'une couche de contrôle pour la définition de ses critères de sélection des actions. Une couche adaptation a été définie pour la définition de la R.O.I. Le module de description 2D n'a qu'un rôle de processus : il est complètement sous la dépendance du module d'interprétation (*Problème 5*).
- Les deux types de centre d'intérêt ont été utilisés dans le module d'interprétation. La couche adaptation a été utilisée par rapport à la définition de la R.O.I. Celle-ci a été modifiée dynamiquement en fonction de la résolution. Cet aspect est réalisé alors que seule la couche commande est abordée dans le choix des actions de ce module, le critère de sélection est fixé (*Problème 6*).

## 4.2 Le projet de recherches SATURNE

Le projet de recherches *SATURNE* [Dem 86] a pour objectif clairement avoué la construction d'un système général de vision. Son principal axe de recherche est l'étude et le développement de la notion de niveaux de représentations fondée sur les deux critères d'abstraction et de décentration.

Les traitements de ce système sont des méthodes d'inférences de forme (reflet, texture, stereo, couleur par exemple). Chaque méthode est élaborée en explorant au

maximum les résultats qu'elle produit indépendamment de toute connaissance sur la scène et de tout échange avec d'autres méthodes.

Le but est d'intégrer ces méthodes au sein d'une structure commune dont les *niveaux de représentations* sont le moyen de communication. Leur inscription sur ces niveaux conduit à leur décomposition en un ensemble d'entités, intersection des niveaux et de la méthode d'inférence (cf. figure 4.7). Une entité construit donc la description de la scène à un niveau de représentation donné selon une méthode d'inférence de formes. Les niveaux et les méthodes d'inférence auxquelles elles appartiennent constituent des liens privilégiés de communication entre elles. Cette communication par les niveaux s'élargit aussi à d'autres systèmes intelligents tels, par exemple, un bras manipulateur dont l'architecture est organisée de manière similaire selon des niveaux de spécification de la tâche à accomplir [PT 86].

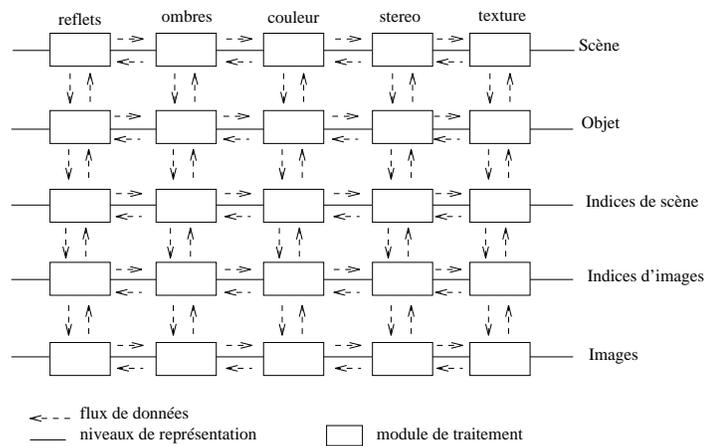


FIG. 4.7 - **SATURNE: architecture du système.** Les méthodes d'inférence de formes s'inscrivent sur les niveaux de représentation du système. Elles définissent ainsi les traitements de base qui sont l'intersection d'une méthode d'inférence avec les niveaux de représentation. [Dem 86]

Le mode de fonctionnement de ce système vise à reproduire *reconnaissance* et *reconstruction*. L'étude du contrôle n'est pas du tout abordée dans ce système qui ne traite actuellement que la vision passive.

### 4.3 Conclusions par rapport à VAP et SATURNE

A partir de ces deux systèmes, nous pouvons dégager des principes guidant la spécification de la structure de contrôle de notre système.

### 4.3.1 Architecture du Système

#### a. Eléments d'organisation

##### Niveaux de représentation

*SATURNE* propose les niveaux de représentation comme structure d'organisation de la connaissance et des traitements. Nous avons vu que d'un point de vue contrôle dans *VAP*, cette structure est aussi utilisée pour la définition des couples contrôleur-processus.

Les niveaux de représentation différencient les traitements en ceux *transformant* une description d'un niveau en celle d'un autre niveau, et en ceux augmentant ou *enrichissant* la description au sein d'un même niveau. Tous ne sont pas des lieux d'échanges entre les traitements. Ainsi, les traitements peuvent ne pas utiliser dans leur exécution le même nombre de niveaux de représentation. C'est le cas, par exemple, de *SCHEMA* ou des *Procédures Visuelles*.

##### Focalisation

Dans le cadre de nos expériences présentées en début de ce chapitre, nous avons mis en évidence la répercussion de la R.O.I. et du but sur chacun des niveaux de représentation du système. Ces échanges de commandes et d'observations relatifs à un centre d'intérêt, entre contrôleurs et processus situés sur des niveaux de représentation différents, constituent un second élément de structuration des traitements du système. Nous retrouvons aussi dans *SATURNE* de telles "barres" de traitement qui correspondent aux méthodes d'inférence de formes dans lesquelles des interactions du même type que *VAP* sont envisageables : commandes spécifiques à un type d'inférence. Cette structuration est visible aussi dans d'autres systèmes. Elle correspond aux propriétés intrinsèques [Bar 78], à des objets de la scène [Dra 87] ou à des tâches [Alo 90] (cf. figure 4.8).

Selon les systèmes (cf. figure 4.8), le regroupement modulaire des traitements conduit à l'existence de modules soit sur un niveau de représentation (*VAP*), soit sur l'ensemble des niveaux (*SCHEMA*, *MEDUSA*), soit sur l'intersection d'un niveau de représentation et d'une focalisation (*SATURNE*).

#### b. Interactions

Les interactions entre chacun des modules du système *VAP* sont spécifiques à ce système. Elles ne sont pas utilisables par exemple pour la réalisation du mode de fonctionnement que veut mettre en place le système *SATURNE*. En effet, la présence de plusieurs modules sur un même niveau de représentation dans ce système introduit la nécessité d'envisager d'autres interactions que celles réalisant la hiérarchie de contrôle de *VAP*. Ce sont par exemple des échanges de données entre deux méthodes d'inférence de forme ou entre différents traitements au sein d'un même niveau. Ainsi, la définition

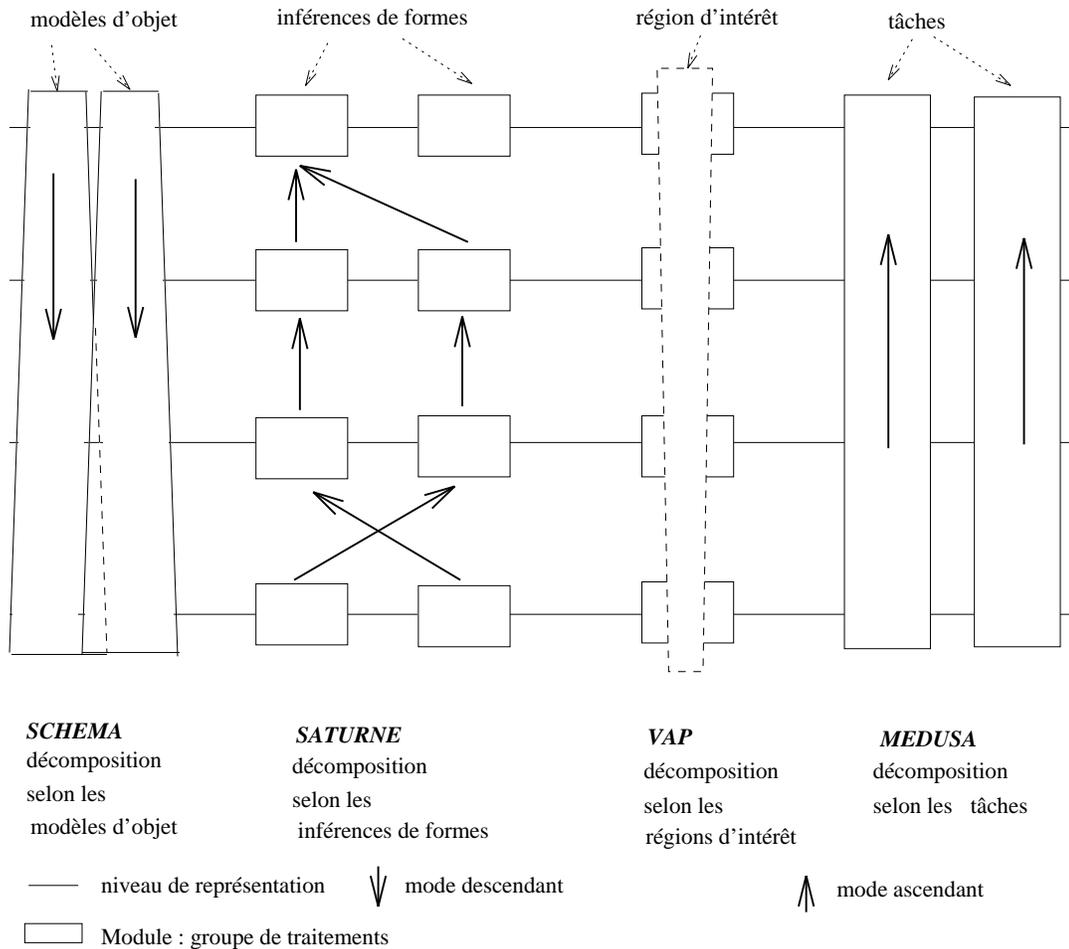


FIG. 4.8 - **Comparaisons.** [Boi 91] Expression des dimensions de structuration de systèmes intégrés de vision. Nous voyons ainsi apparaître les niveaux de représentation et les focalisations s'installant transversalement. Nous avons fait apparaître en flèches pleines les modes de fonctionnement : ascendant ou descendant. Les modules représentent l'ensemble des traitements regroupés autour d'une fonctionnalité propre.

d'un langage d'interaction plus élaboré du point de vue de l'expression du contrôle que dans le cas de *VAP* est nécessaire.

### 4.3.2 Module général

Une des tâches les plus importantes pour une intégration est la construction d'un module standard dans lequel les différents traitements puissent être exprimés.

Après divers tests et expériences dans l'intégration des modules, un module standard a été conçu et implanté [Ber 91] au sein de SAVA2.0 afin d'aider le travail d'intégration. Le module formulé est fortement inspiré de l'architecture du module de description 2D (cf. figure 3.6). Ses capacités de contrôle sont ainsi, comme nous l'avons vu, très restreintes. Ce module standard limite ainsi les améliorations locales possibles du contrôle d'un module. Le développement d'un module standard facilitant l'intégration doit permettre l'utilisation des trois couches de contrôle en fonction de la complexité du contrôle requis.

Une structure supplémentaire doit être alors nécessaire pour restreindre l'utilisation des différentes couches de contrôle au sein de chacun de ces modules. Elle permet ainsi l'installation, par exemple, de la hiérarchie de contrôle utilisée dans nos expériences. Alors que dans *VAP*, cette hiérarchie est installée définitivement dans le système, l'architecture de *SATURNE* exige de changer dynamiquement en cours de résolution les relations entre chacune des modules.

Ces deux systèmes font apparaître un ensemble de sous-systèmes interagissant entre eux sur les niveaux de représentation. Nos expérimentations ont mis en évidence deux expressions du contrôle : au sein de chacun de ces sous-systèmes et au sein de leurs interactions dans le système. La définition des interactions et des relations entre chacun des sous-système doit pouvoir être réalisée explicitement en cours de fonctionnement du système. Afin de formuler le modèle à la base de la structure de contrôle d'un tel système, nous nous tournons vers l'Intelligence Artificielle Distribuée dans laquelle nous trouvons des modèles et des concepts permettant de réaliser de telles exigences.

## 4.4 Intelligence Artificielle Distribuée

L'Intelligence Artificielle Distribuée (IAD) est au centre de plusieurs courants issus de la psychologie, la sociologie, la biologie et l'informatique. Ces études sont utilisées pour modéliser et construire des systèmes informatiques dans lesquels les capacités de traitement et de représentation sont distribuées dans un ensemble de sous-systèmes appelés *agents*. L'ensemble de ces agents constitue une *société*. Selon les ressources en possession de chacun, des résolutions entreprises, les agents interagissent entre eux. Le plus souvent ces interactions sont des échanges d'information.

Dans la suite de ce rapport, nous nous attachons à donner une définition claire et précise des termes que nous utilisons en IAD <sup>3</sup>. En effet, la plupart du vocabulaire employé est issu des domaines de recherche cités ci-dessus. En psychologie cognitive ou en philosophie de la pensée, nous trouvons par exemple les notions de *croyance*, d'*intentions*. Ces notions permettent d'apporter des abstractions intéressantes et utiles pour la description, l'analyse et la conception de comportements de systèmes complexes telles une société humaine, la vie biologique, etc. Leur utilisation en IAD, permet de décrire le fonctionnement et l'organisation de systèmes, indépendamment de leur implantation. Il faut cependant leur donner une base objective solide. L'IAD est encore loin de réaliser dans le concret toute la puissance sémantique qui peut être attribuée à ces notions. De ce fait, ils reçoivent une sémantique limitée qui varie selon les auteurs. Avant de poursuivre, nous définissons ainsi plus précisément les notions : *agent* et *société*.

#### 4.4.1 Définition Agent/Société

Un système peut être décrit et modélisé selon deux points de vue [Boi 90], [Dem 90c] : *macroscopique* et *microscopique*. Le premier observe l'activité du système à partir des échanges qu'il a avec l'extérieur. Il caractérise le système comme une entité agissant dans l'environnement et ayant un problème à résoudre avec des connaissances et des capacités de traitement limitées. Nous décrivons le système en terme d'*agent*. Le point de vue microscopique, en revanche, se rapproche du système pour distinguer, en son sein, les sous-systèmes et les échanges qui ont lieu entre eux. Le système est ainsi décrit par l'ensemble de ses agents (les sous-systèmes) et de leurs échanges. Nous nous tournons alors vers une description en terme de *société*.

En IAD, nous trouvons souvent des termes différents pour désigner les entités agissant dans le système. Les termes utilisés sont issus d'une des disciplines scientifiques, source d'inspiration de l'IAD : spécialistes ou experts, sources de connaissances ou modules ou règles, ant, acteur. Nous utilisons le terme d'*agent* de préférence à ces termes, dans la mesure où ceux-ci font référence à une implantation particulière. Plusieurs définitions d'agent existent actuellement dans la littérature. Cependant, à notre avis, elles sont trop restrictives sur la structure du système [Nil 87], [Sho 90] ou sur son fonctionnement [Nil 87], [Col 91]. Notre définition fait abstraction de toute implantation. Elle se rapproche de celle trouvée dans [Fer 91a] ou [Dem 90c].

##### ◇ Définition : *Agent*.

Un agent est un système informatique <sup>4</sup> placé dans un environnement qu'il est capable de *percevoir*, de représenter, même partiellement, et sur lequel il est capable d'entreprendre des *actions*. Ces dernières constituent l'unique moyen pour l'agent de signifier son existence dans l'environnement.

<sup>3</sup>Une part de ce travail de définition a été menée en collaboration avec le groupe de travail *PLEIAD* regroupant plusieurs chercheurs en IAD de laboratoires Grenoblois [PLE 92].

<sup>4</sup>Matériel ou logiciel.

Dans cette définition, nous incluons la notion d'*autonomie* : capacité d'adaptation de l'agent aux évolutions de l'environnement tout en agissant en accord avec ses propres objectifs. Les objectifs ou buts peuvent être explicites et faire l'objet d'un raisonnement (cas des agents délibératifs) ou implicites (cas des agents réactifs) [Rus 89].

◇ **Définition** : *Environnement d'un agent*.

L'environnement d'un agent désigne les autres agents et le monde extérieur.

◇ **Définition** : *Société*.

Une société est un ensemble d'agents, auxquels est offerte la possibilité d'interagir, et leurs interactions.

Cette définition se distingue des plus courantes dans la mesure où beaucoup d'auteurs notent dans leur définition, d'une part l'obligation pour les agents d'interagir et, d'autre part, l'inclusion implicite d'une dimension d'*organisation* de la société.

#### 4.4.2 Description Agent/Société

Tout agent (toute société) possède deux composantes : une *structure* et des mécanismes de traitement qui, par utilisation de la structure, permettent à l'agent (à la société) d'exhiber un comportement. Nous donnons une brève description de ceux-ci pour fixer le cadre de cette discipline. Dans le chapitre suivant qui présente les différents modèles à la base de la structure de contrôle de notre système, nous préciserons la définition de ces termes.

### Agent

#### Structure

La structure d'un agent regroupe l'ensemble des *connaissances*, des *données*, des représentations (*hypothèses*, *buts*) et des *actions* [Dem 90c] qu'un agent manipule. Les *données* sont l'ensemble des informations perçues sur l'environnement. Les *buts* sont les moteurs de l'action de l'agent pour lesquels diverses *actions* sont exécutées. Les *connaissances* font référence à l'ensemble des informations que l'agent possède ou acquiert. Il les utilise pour déterminer les actions à entreprendre dans l'environnement à partir d'une représentation de celui-ci (*hypothèses*) obtenue à partir d'une interprétation des *données*.

#### Mécanisme de traitement

Les mécanismes de traitement sont décomposés en deux sous ensembles : *raisonnement* et *décision*. Les mécanismes de raisonnement élaborent, à partir des connaissances, des buts et des hypothèses, les possibilités d'actions de l'agent. Les mécanismes de décision, en fonction des buts et des hypothèses, sélectionnent les actions à réaliser.

### **Contrôle individuel**

Le problème du contrôle au sein de cet agent ou *contrôle individuel* consiste à déterminer quelle action exécuter à partir de ses représentations de la résolution et de l'environnement.

### **Société**

La société est décrite selon le même canevas de description que celui utilisé pour l'agent :

### **Structure**

La société possède une structure constituée du *réseau d'interactions* entre les agents, d'un *langage d'interaction*, de *protocoles d'interaction* définissant les enchaînements d'interaction et d'une *organisation* contraignant les relations entre les différents agents. Le réseau d'interactions est le moyen par lequel les agents interagissent. Il définit toutes les possibilités d'échanges <sup>5</sup> que les agents peuvent utiliser. L'organisation définit les types d'interactions et de résolution que les agents peuvent entreprendre. Les protocoles d'interaction sont décomposés en protocoles de résolution et en protocoles d'organisation. Les *protocoles de résolution* sont les protocoles utilisés pour les échanges ayant trait au partage de données, de buts, etc. Ces protocoles contribuent à la progression de la résolution en définissant l'enchaînement de telles interactions. Les *protocoles d'organisation* concernent les protocoles utilisés pour définir une nouvelle organisation dans la société, pour résoudre des conflits.

### **Mécanismes de traitement**

Les mécanismes de traitement définissent la manière et le moment d'utilisation de la structure définie. Nous distinguons aussi un mécanisme de raisonnement qui correspond aux *échanges de base* (hypothèses, buts, actions par exemple) régis par les protocoles de résolution. Le mécanisme de décision utilise l'ensemble des protocoles d'organisation du type résolution de conflits, réseau de contrats [Smi 81], [Cha 91] [Kre 91] [Ber 92] qui permettent de réorganiser la société.

### **Contrôle social**

Le *contrôle social* ou contrôle de la société consiste à déterminer pour tout agent les contraintes à imposer sur ses actions et ses interactions. Ces contraintes établissent un équilibre entre des critères locaux (objectifs de l'agent) et des critères sociaux (objectifs partagés par tous les agents).

---

<sup>5</sup>Ici le terme d'échange est pris au sens large : communication, perception, conflits de ressources, etc.

### 4.4.3 Problèmes abordés par l'Intelligence Artificielle Distribuée

L'IAD s'est avérée bien adaptée pour différentes applications qui présentent les exigences suivantes :

- *Utilisation de connaissances et de mécanismes de traitements différents pour la résolution d'un problème.*

Cette exigence fait référence à la possible *hétérogénéité* des agents : ceux-ci peuvent avoir des structures et des mécanismes de traitement différents.

- *Distribution des mécanismes de traitement et des représentations.*

Ce sont, par exemple, la répartition de capteurs sur un territoire [Cor 83b] [Cam 83]. Des contraintes *temporelles* peuvent aussi obliger à répartir les traitements : effectuer, par exemple, au même moment deux tâches différentes pour utiliser ensuite les résultats produits.

- *Augmentation des capacités du système en terme de fiabilité, d'ouverture, de flexibilité.*

Les agents et le système sont conçus de telle sorte qu'à tout moment, un nouvel agent puisse être ajouté ou retiré du système sans que les performances globales en soient gravement affectées. Cette dimension d'ouverture, recouvre les capacités d'adaptation, de dynamisme, de non-dégradation des performances en cas de défaillance d'un des éléments. La distribution des fonctionnalités, des représentations, du contrôle améliorent la flexibilité et les capacités d'adaptation du système en fonction des contextes d'exécution. Plusieurs décisions complémentaires ou antagonistes peuvent être explorées, plusieurs représentations, cohérentes ou non, peuvent exister.

### 4.4.4 Recherches actuelles

Dans le domaine de l'IAD, différentes directions de recherches complémentaires sont actuellement visibles.

#### Recherches au niveau des agents

Au niveau des agents, les recherches s'intéressent à la définition de structures et de mécanismes de traitements. Deux tendances très fortes se sont dégagées ces dernières années avec les recherches sur les agents *réactifs* et celles sur les agents *délibératifs* ou *cognitifs*. Nous en avons déjà donné un aperçu dans le cadre du chapitre sur le contrôle.

#### Agents réactifs

Les bases [Ste 90] du développement d'agents réactifs résident dans les études physiologiques, éthologiques [Lor 90], de systèmes physiques, mettant en avant la réalisation de comportements complexes par interaction d'un nombre important de systèmes

primaires (ex: fourmi). Cette approche a d'importants liens avec de récents développements des réseaux de neurones ou des systèmes connexionistes. Ces derniers s'appuient sur la théorie des automates, et de la conception de circuits. Les systèmes réactifs s'appuient, par contre, sur les notions issues d'intelligence artificielle, [Bro 85], [Ste 90], [Mae 89], [Ros 86], [Kae 86], [Agr 87], [Cha 90], [Fer 91b], [Dro 91], [Dem 91a]. Ainsi les agents réactifs abordent des problèmes dans lesquels la relation avec l'état de l'environnement est primordiale. Cette relation est exprimée par la mise en place d'un comportement prédéfini<sup>6</sup>. Les interactions de l'agent avec l'environnement et avec les autres agents sont privilégiées dans la mesure où elles sont à l'origine de la progression de la résolution. Les sociétés de tels agents comportent un nombre important d'agents ayant une structure le plus souvent *homogène*<sup>7</sup>.

### Agents délibératifs

Ces agents sont aussi qualifiés de cognitifs ou intentionnels. Ils s'inscrivent dans la poursuite des recherches menées en IA avec la prise en compte des interactions avec l'environnement. Ces agents mettent en œuvre un raisonnement guidé par un ensemble d'intentions, de buts afin de déterminer un ensemble d'actions possibles à prendre dans leur environnement à partir d'une représentation explicite de celui-ci<sup>8</sup>. Dans une telle approche, l'accent est mis sur la présence d'une représentation interne. Celle-ci est à la base du fonctionnement de l'agent. A la différence des agents réactifs, ils peuvent prendre des initiatives sans que l'apparition d'événement dans leur environnement ne soit nécessaire. Ils sont exprimés dans différents formalismes logiques [Gen 87] [Sho 90] [Nil 87] [Wer 89] ou autres [Geo 87] [Bur 91]. Les sociétés de tels agents comprennent un petit nombre d'agents ayant chacun d'importantes capacités, des structures *homogènes* ou *hétérogènes*. L'accent est mis essentiellement sur la structure de ces agents, dans la mesure où la résolution de la société en dépend.

### Agents Mixtes

Selon Parunak [Par 90], la distinction entre agents réactifs et délibératifs doit être dépassée. En effet le monde qui nous entoure nous donne des exemples de l'intégration de ces approches, au sein des mêmes systèmes. C'est ainsi que, depuis quelques années, des recherches sont menées sur l'intégration réactif et délibératif [Geo 89], [Fer 91c], [Kis 91], [Lev 90], que nous appellerons par la suite *mixtes*.

## Recherches au niveau de la Société

Les recherches menées au niveau *agent* ont une répercussion sur la société et sur la complexité des outils mis en œuvre. Au niveau de la société, les recherches les plus

<sup>6</sup>"Having a propositional attitude is being in a state that an external observer interprets as you having this propositional attitude" [Bro 85].

<sup>7</sup>Tous les agents ont la même structure et le même mécanisme de traitement.

<sup>8</sup>"Having propositional attitude is having a data structure in the corresponding module" [Bro 85].

importantes concernent l'élaboration de protocoles d'interaction exprimant différentes tâches [Kre 91] [Con 86] et sur l'expression des organisations [Pat 87] [Bon 90] [Bou 92]. Nous trouvons deux tendances importantes actuellement dans ce domaine : *Résolution Distribuée de Problèmes* et *Multi-Agents*.

### **Résolution Distribuée de Problèmes**

La *Résolution Distribuée de Problèmes* qualifie des systèmes dans lesquels plusieurs agents coopèrent pour atteindre un but, identique à tous. Ce but est la raison d'être d'un tel système. Toute la conception du système, c.-à-d. la définition des agents, de la société, est orientée dans l'optique de la résolution de ce problème <sup>9</sup>.

### **Multi-Agents**

L'approche *Multi-Agents*, comme la précédente, met en jeu des agents interagissant dans une société, sans avoir a priori de problème à résoudre, sinon celui de la gestion de leurs interactions de ressource. La notion de multi-agents est utilisée pour la première fois au travers de l'étude de Konolige et Nilsson : "Multiple Agent Planning Systems", [Kon 80]. Les premiers problèmes abordés font ainsi référence à la gestion des interactions entre les différents agents, et plus particulièrement lors de problèmes de planification entre plusieurs agents.

Un système multi-agents est donc motivé par la recherche de méthodes pour gérer les interactions entre différents agents, agissant dans une même société, mais ayant chacun des buts différents et n'étant pas, a priori, définis pour résoudre un but commun <sup>10</sup>.

## **Recherches au niveau implantation**

Divers outils sont offerts pour la mise en œuvre des modèles proposés par ces différents courants de recherche. Les premiers à être utilisés et conçus pour de telles études sont les langages *acteurs* [Hew 73], les architectures de *blackboard* [Nii 86a] [Nii 86b] [Eng 87]. Depuis, plusieurs plate-formes *multi-agent* ont été développées à partir de différentes applications : *DVMT* [Cor 83a], *MICE* [Mon 92], *MAGES* [Bou 91], [Bou 92], *MACE* [Gas 87], *MAPS* [Bau 92] entre autres.

## **4.5 Principes généraux à la base de la structure de contrôle**

Plusieurs systèmes généraux de vision se sont appuyés sur une architecture de blackboard [Eng 87]. Ce sont par exemple les systèmes de [Nag 79], [Mer 90], *MESSIE* [Gar 89], *SCHEMA* [Dra 89], *SKIDS* [Dek 89][Lac 92]. Cette architecture a l'avantage d'offrir un cadre général dans lequel viennent interagir les sources de connaissance. Un

<sup>9</sup>Rosenschein qualifie la démarche à la base de tels systèmes comme "using communities of agents to find solutions"[Ros 84].

<sup>10</sup>Rosenschein qualifie l'approche à la base de ces systèmes comme "using artificial intelligence to find solutions for communities of agents".

développement incrémental est possible. Cependant la plupart du temps, les modes d'interaction sont peu développés et figés. Les sources de connaissance ne peuvent influencer l'une sur l'autre que de manière indirecte.

L'approche que nous utilisons nécessite une modélisation d'agents et d'interactions beaucoup plus conséquentes. Nous retrouvons une telle démarche dans les systèmes de [Lan 89] qui est dédié à une application ou de [Bau 92] utilisé pour la construction d'un sous-ensemble restreint d'un système de vision.

#### 4.5.1 Organisations possibles du système

Différentes structurations d'un système intégré de vision sont possibles. Utilisant la terminologie définie précédemment, nous définissons les agents possibles et les éléments de la société qui en découlent. Selon les options choisies, le contrôle du système prend des expressions différentes.

##### Agents

Les niveaux de représentation sont une importante source d'organisation du système, c'est pourquoi, nous définissons :

◇ **Définition :** *Agent Niveau.*

Nous définissons un *agent niveau* comme l'ensemble des traitements se déroulant sur un niveau de représentation du système.

Au cours de notre étude, un second élément d'organisation est apparu. Il est relatif à l'uniformité des centres d'intérêt qui existent sur l'ensemble des niveaux de représentation.

◇ **Définition :** *Agent Focus.*

Nous définissons un *agent focus* comme l'ensemble des traitements agissant sur l'ensemble des niveaux de représentation, et pour lesquels une uniformité de focus est visible dans leur fonctionnement ou dans leur conception.

Une unité de traitement et de représentation élémentaire peut être ainsi définie en prenant l'intersection des deux agents précédents.

◇ **Définition :** *Agent de Base*

Les intersections des agents focus et agents niveaux définissent les *agents de base*. L'activité d'un tel agent consiste à interpréter la scène à un niveau de représentation donné, selon un centre d'intérêt fixé.

##### Interactions au sein de la société

Les décompositions selon les niveaux et selon le centre d'intérêt définissent les flux d'informations au sein du système :

◇ **Définition :** *Transformation.*

Nous appelons *transformation* les échanges d'informations d'un niveau à l'autre,

c'est-à-dire les traductions d'une représentation à une autre.

Nous préférons ce terme à celui d'inférence en référence aux méthodes d'inférence de formes [Dem 86]. Son sens est trop marqué d'un point de vue IA. Ceci étant, nous retrouvons le sens que lui assignait Demazeau bien que nous n'imposions pas de direction à l'échange : la transformation peut se dérouler d'un niveau inférieur à un niveau supérieur et inversement.

◇ **Définition** : *Enrichissement*.

Nous appelons *enrichissement* [Dem 86] la production d'informations au sein d'un niveau de représentation, à partir d'informations du même niveau.

Selon la définition des agents du système, ces types d'échanges apparaissent liés à l'activité de communication entre agents ou liés à l'activité de résolution de l'agent.

### Conséquences pour le contrôle du système

Ces structurations possibles des traitements et représentations en agent conduisent à différentes formulations du problème de contrôle.

- *agent-niveau* : le contrôle rencontré au sein d'un niveau de représentation est géré par l'agent. Le contrôle des interactions inter-niveaux est du ressort du contrôle de la société.
- *agent-focus* : le contrôle pour un centre d'intérêt donné est du ressort du contrôle de l'agent. Les interactions intra-niveaux sont du ressort du contrôle de la société.
- *agent-base* : le contrôle de la société recouvre à la fois le contrôle intra et inter-niveaux. Le contrôle de l'agent est diminué dans la mesure où son activité concerne un niveau de représentation selon un centre d'intérêt donné.

Dans ces trois cas, nous voyons apparaître cette dichotomie entre contrôle social et contrôle individuel. Selon les cas, l'un des types de contrôle est prépondérant.

#### 4.5.2 Synthèse

A partir des définitions d'agent et de société, nous reformulons les problèmes présentés dans le chapitre précédent en décomposant les aspects propres à l'agent et ceux propres à la société. Nous les rappelons ici brièvement.

□ **Problème 1** *Identification des couples contrôleur-processus.*

Ce problème concerne l'organisation des traitements du système en terme de processus et de contrôleur. Les niveaux de représentation sont un facteur important d'organisation. Il est donc naturel de les utiliser comme élément de référence pour la définition des contrôleurs et des processus. Sur quels niveaux de représentation définir les processus et les contrôleurs? Combien de niveaux recouvrent-ils? Quel est le nombre de processus et de contrôleurs existants dans le système?

□ **Problème 2** *Architecture d'un contrôleur.*

Étant donné un contrôleur quelles sont les couches de contrôle à utiliser pour le réaliser : couche commande, couche adaptation ou couche décision? Transversalement à cette question, quelles sont les observations à prendre en compte pour le fonctionnement de chacune des couches de contrôle?

□ **Problème 3** *Connaissances de contrôle.*

A partir des types de fonctionnements reconnaissance/reconstruction et leurs traductions en localisation/identification, quelles sont les connaissances de contrôle à utiliser?

□ **Problème 4** *Relations entre connaissances de contrôle et la définition des contrôleur-processus au sein du système.*

Quelle est la définition des centres d'intérêt inter-niveaux et intra-niveaux de représentation? Quelles sont les relations entre les traitements fournissant une description au sein d'un même niveau de représentation?

□ **Problème 5** *Architecture d'un module par rapport à sa fonction de contrôleur ou de processus dans le système.*

Quelle est la définition de l'architecture d'un module en terme de couches de contrôle dans le système, en fonction de son rôle de contrôleur ou de processus

□ **Problème 6** *Relations entre connaissances de contrôle et l'architecture définie pour chacun des modules.*

Définition des dimensions spatiales, sémantiques et temporelles d'un centre d'intérêt par les couches de contrôle?

### a. Propositions pour l'architecture d'un agent

Dans le cadre de cette thèse, nous considérons les agents comme étant des *agents de base*. Cette décision est motivée par la volonté d'avoir la structure la plus flexible possible. Comme nous l'avons vu les deux autres types d'agent (niveau et focus) intègrent une partie de la gestion des interactions au sein de l'agent.

Les agents de base seront ainsi le module de description 2D ou le module d'interprétation spécialisés dans la construction de la description pour une région donnée ou pour une caractéristique donnée. Nous considérons le module de contrôle de la caméra aussi comme un de ces agents.

Afin de résoudre le *Problème 2* en permettant d'aborder différents types de contrôle dans chacun des agents, nous faisons la proposition suivante relative à la définition de l'architecture de contrôle de chacun des agents :

□ **Proposition 1** : *Architecture de contrôle.*

Les mécanismes de traitement et la structure de l'agent possèdent les couches de commande, d'adaptation et de décision pour la résolution du problème du contrôle. L'agent définit ainsi ses objectifs, sa loi de commande et sa commande en fonction de l'évolution de l'environnement.

Nos premières expérimentations ont mis en évidence la nécessité d'installer dynamiquement dans un agent des comportements répétitifs tout comme des comportements de recherche. La répétition se traduit, par exemple, par l'application d'une loi de commande sur plusieurs cycles. La recherche se traduit par une modification dynamique de celle-ci. Actuellement peu d'éléments existent sur les conditions régissant le changement de la loi de commande dans le cadre d'un fonctionnement répétitif, si ce n'est la détection d'un échec de la commande par exemple. Un mécanisme gérant le passage d'une couche de contrôle à une autre en fonction de critères précis doit être inscrit dans l'agent.

□ **Proposition 2 :** *Gestion explicite du passage d'une couche de contrôle à l'autre.*

Des mécanismes de traitement particulier seront mis en place afin de pouvoir étudier et gérer le passage d'une couche de contrôle à une autre, dans le sens ascendant : commande vers adaptation, adaptation vers décision. Dans le sens descendant, le passage d'une couche à l'autre est déclenché par génération de la commande pour la couche inférieure.

Nous venons de définir la nécessité des couches de contrôle, de mécanismes de changement dynamique d'une couche à l'autre au sein de l'architecture de contrôle d'un agent. Il nous reste à préciser nos exigences par rapport à l'initialisation des traitements au sein d'un agent.

□ **Proposition 3 :** *Fonctionnement dirigé par les données et par les buts.*

Un agent pouvant être contrôleur ou processus au sein du système, il doit pouvoir être dirigé soit par les données (observations), soit par les buts (commandes). Un agent utilisant les trois couches de contrôle devra utiliser le même mécanisme de traitement pour un fonctionnement dirigé par les données ou par les buts.

La résolution des *Problèmes 3 et 6* entraîne la modification du centre d'intérêt selon l'état de la résolution. Par ailleurs, plusieurs centres d'intérêt sont présents et ont des gestions différentes.

□ **Proposition 4 :** *Existence de plusieurs centres d'intérêt.*

Au sein de l'architecture de contrôle d'un agent, les mécanismes gérant les centres d'intérêt permettent un découplage de ceux-ci tant du point de vue de leur prise en compte que des couches de contrôle abordées pour leur définition.

## b. Propositions pour l'architecture de la société

La résolution du *problème 1* se reporte sur la définition des organisations et des protocoles d'interaction.

□ **Proposition 5 :** *Définition des organisations en terme de processus-contrôleur.*

L'organisation exprimée dans le modèle de société définit les relations entre les agents en terme de contrôleur-processus. Selon l'organisation définie, un agent aura ainsi un rôle de processus ou de contrôleur dans le système par rapport aux

autres agents. L'organisation permet d'aborder, par exemple, un contrôle centralisé, décentralisé, hiérarchique. L'organisation peut être modifiée dynamiquement selon les échanges et les modifications dans l'environnement.

□ **Proposition 6 :** *Interactions réalisent la prédiction/vérification.*

Les mécanismes de *prédiction-vérification* sont exprimés par les types d'interaction primitifs possibles entre les agents.

□ **Proposition 7 :** *Protocoles d'interaction ascendant et descendant.*

Les protocoles d'interaction définissent la structure d'interaction entre les agents et expriment les modes de fonctionnement ascendants et descendants.

La combinaison des deux modes est réalisée par la mise en place de plusieurs organisations relatives à des résolutions particulières du système : organisation hiérarchique selon un centre d'intérêt donné pour un fonctionnement de reconnaissance, une organisation hétérarchique pour un fonctionnement de reconstruction. L'expression de ces deux modes de fonctionnement par les organisations et les protocoles permet la reconfiguration du flux de contrôle entre les agents du système.

La solution aux *problèmes 3 et 4* conduit à la proposition suivante :

□ **Proposition 8 :** *Expression des connaissances de contrôle dans les organisations et protocoles.*

Exprimer les éléments nécessaires au niveau des protocoles, des interactions et l'organisation pour que le système puisse fonctionner selon différents centres d'intérêt.

Le *problème 5* conduit à la proposition suivante :

□ **Proposition 9 :** *Modification de l'architecture d'un agent en fonction de l'organisation définie.*

L'expression des organisations définissant la structure de contrôle de la société spécifie les couches de contrôle que les agents peuvent mettre en œuvre dans leur fonctionnement local.



## Chapitre 5

# Modèles pour l'étude du contrôle dans un Système Intégré de Vision

Nous présentons dans ce chapitre les outils utilisés pour notre étude du contrôle dans un Système Intégré de Vision à partir des propositions faites dans le chapitre précédent.

Dans la première section, nous présentons le *modèle d'agent social* pour la définition et la structuration des traitements au sein de chacun des agents. Au sein de ce modèle, nous isolons un *modèle d'agent individuel* et les *capacités sociales*. Celles-ci sont explicitées au travers d'un *modèle de société* exprimant la gestion et la structuration des interactions entre les agents. Ces modèles définissent le cadre de notre étude du contrôle.

A partir de l'étude faite au chapitre 2, nous explicitons les facteurs liés à la modification du comportement d'un *agent* et de la *société* dans chacun des modèles. Nous débouchons ainsi sur un *modèle de contrôle individuel* (contrôle au sein du modèle d'agent individuel) et sur un *modèle de contrôle social* (contrôle entre les agents au sein de la société). Chacun de ces modèles est organisé autour des couches de *commande*, d'*adaptation* et de *décision* mises en évidence dans le chapitre 2.

Dans la dernière section, nous particularisons ces deux modèles à partir de l'étude

relative à la vision effectuée dans les chapitres précédents. Nous montrons comment ces modèles répondent aux propositions du chapitre précédent. A l'issue de ce chapitre, nous avons ainsi défini le *modèle de contrôle individuel* et *modèle de contrôle social* pour un *Système Intégré de Vision*.

## 5.1 Modèles

Comme nous l'avons noté dans le chapitre précédent, trois axes de recherche existent en IAD au niveau des modèles d'agent : agents *réactifs*, agents *délibératifs* et agents *mixtes*.

Nous avons choisi un modèle d'agent *délibératif*, proche des modèles que l'on trouve dans la littérature [Dem 90c] [Bur 91]. Ce choix est motivé par la possibilité de mettre en évidence plus facilement qu'au sein d'un modèle d'agent réactif, les trois couches de contrôle présentées ci-dessus. Un agent délibératif possède, en effet, un ensemble de traitements et représentations pour l'élaboration de plans et de buts.

La conception d'un agent *réactif* conduit en revanche à la définition d'un ensemble d'actions en relation directe avec les entrées issues de l'environnement. De tels agents n'abordent ainsi généralement que la couche *commande*. Dans celle-ci, les critères de sélection utilisés ne peuvent pas être modifiés dynamiquement en cours d'activité. Cependant, ces agents ont l'avantage d'être ouverts sur l'environnement : toute modification de celui-ci provoque l'exécution immédiate d'une action prédéfinie. Une action par défaut recouvre les cas imprévus. L'intérêt d'obtenir une réponse immédiate a été mis en évidence dans nos premières expériences sur le contrôle dans *VAP* au travers du mécanisme d'interrogation automatique de la description. (cf. chapitre 4)

Nous aurions ainsi été plus intéressés par un modèle d'agent *mixte*, c.-à-d. intégrant les aspects réactifs et délibératifs. L'absence de résultats, actuellement, sur ce type de modèle a empêché ce choix. Toutefois, cette problématique nous intéresse et nous présenterons quelques aspects de notre approche pouvant conduire au développement de tels modèles.

Le *modèle d'agent social* (cf. figure 5.1) que nous considérons possède ainsi des connaissances, des représentations et des mécanismes de traitement qui lui permettent de raisonner sur ses actions dans l'environnement, de décomposer un problème en différents sous-problèmes, de construire par exemple des plans, de tenir compte de ses buts et de l'évolution de l'environnement. Agissant au sein d'une société, il possède aussi des éléments relatifs à la mise à jour des connaissances et au raisonnement sur les autres agents. Il possède aussi des capacités pour interagir avec les autres agents.

Au sein de ce modèle, nous distinguons un *modèle d'agent individuel* (composants sur fond noir dans la figure 5.1) constitué des traitements et des représentations impliquées dans l'activité propre de l'agent indépendamment de toute interaction avec les autres agents. Les *capacités sociales* (composants sur fond gris dans la figure 5.1)

de l'agent, découlent de l'adjonction des mécanismes de traitement exprimés dans le *modèle de société*. Ceux-ci sont liés à la gestion des interactions (*protocoles*) et de la coordination (*organisation*).

### 5.1.1 Modèle d'Agent Social

Nous définissons un agent social par le 10-uplet suivant (cf figure 5.1):

$$\begin{aligned}
 \textit{Agent} &= \langle KR, L, Pr, Org, PS, RS, DS, CS, ES, PM \rangle \\
 \textit{PM} &= \langle \textit{capteur}, \textit{récepteur}, \textit{interprétation}, \textit{interprétation-dialogue}, \textit{évaluation}, \\
 &\quad \textit{raisonnement}, \textit{décision}, \textit{engagement}, \textit{exécution}, \textit{exécution-dialogue}, \\
 &\quad \textit{effecteur}, \textit{émetteur}, \textit{organisation} \rangle
 \end{aligned}$$

Les *connaissances* sont symbolisées par  $KR$ . Chaque *représentation* de l'environnement ou de l'activité de l'agent est symbolisée par une majuscule faisant référence à la sémantique de son contenu, suivie de la lettre  $S$ . Lorsqu'elle est particularisée par rapport à un descripteur, *descripteur1*, nous la représentons par son symbole suivi du descripteur séparé par un point ( $XS.descripteur1$ ).  $XS.descripteur1[descripteur2]$  est une particularisation par rapport au descripteur supplémentaire, *descripteur2*. Les mécanismes de traitement sont écrits en minuscules. Par abus de langage, nous appelons *fonction* les mécanismes de traitement et *état* une représentation de l'environnement ou de l'activité de l'agent.

Dans ce modèle, du fait de leur utilisation privilégiée dans les systèmes multi-agents, nous faisons apparaître un capteur particulier correspondant à la fonction *récepteur* et un effecteur particulier qui est la fonction *émetteur*. Notre modèle n'impose pas cependant a priori de limitations sur le nombre de capteurs et d'effecteurs.

Nous notons  $\Omega$  l'*environnement* de l'agent. Celui-ci est constitué des autres agents et du monde extérieur.

$KR$  est la représentation des *connaissances*. Cet ensemble est décomposé en connaissances individuelles  $KR.ind$  et en connaissances sociales  $KR.soc$ . Les connaissances individuelles sont utilisées par la fonction de *raisonnement*. Les connaissances sociales sont relatives aux connaissances de l'agent pour l'utilisation de ses capacités sociales.

$L$  est le langage d'interaction utilisé par les agents au sein de la société;  $Pr$  est l'ensemble de protocoles qu'ils peuvent utiliser et  $Org$  est l'ensemble des rôles qui définissent les liens d'interaction existant entre les agents. Un rôle est une spécialisation imposée sur le fonctionnement des agents. Nous présenterons plus en détail ces éléments dans le cadre du modèle de société.

$PS$  est l'*état de perception* recueillant les données fournies par la fonction *capteur* sur  $\Omega$  et les messages reçus par la fonction *récepteur*;  $RS$  est l'*état de raisonnement*, représentation de l'environnement dans le langage de représentation de l'agent en termes

d'hypothèses, de plans et de buts; *DS* est l'état de décision dans lequel sont représentées les intentions possibles *DS.possibilités* et les intentions choisies *DS.choix*. Une intention exprime une relation existant entre les buts, les plans, les actions ou les opérations que l'agent peut entreprendre et son état de raisonnement; l'état d'engagement *CS* regroupe les éléments contrôlant le fonctionnement de l'agent tant d'un point de vu individuel *CS.ind* que social *CS.soc*. *ES* est l'état d'exécution constitué des messages correspondant aux actions de communication et des données, traductions des autres actions dans les termes propres à l'effecteur.

Les fonctions *capteur* et *récepteur* mettent à jour *PS* à partir, respectivement, de la perception de  $\Omega$  et des messages envoyés par les autres agents. Les données fournies par le *capteur* sont traitées par la fonction d'interprétation qui les traduit dans le langage de représentation de l'agent et les inscrit dans *RS.non-évalué*. La fonction d'interprétation-dialogue traduit à l'aide de *L* et de *KR.soc*, les messages reçus et les inscrit dans *RS* et *DS* selon la nature de l'interaction exprimée dans le message. Son fonctionnement est contraint par les engagements exprimés dans *CS.soc*.

L'état de raisonnement ayant été ainsi mis à jour, la fonction d'évaluation détecte les incohérences en son sein à partir des critères inscrits dans *CS.ind* et génère *RS.évalué*. La fonction de raisonnement définit les intentions possibles en utilisant les connaissances *KR.ind* et *RS.évalué* et les critères inscrits dans *CS.ind*. La fonction de décision choisit les intentions de *DS.possibilités* et les inscrit dans *DS.choix* à partir des critères inscrits dans *CS.ind*. La fonction d'engagement met à jour l'état de raisonnement *RS.non-évalué* et *CS.ind*.

Dans le cas où l'intention choisie est relative à une action, la fonction d'exécution inscrit les données, traduction de l'action, dans *ES*. Celles-ci sont exécutées sur l'environnement  $\Omega$  par l'effecteur. Si l'intention choisie est relative à une action de communication, la fonction *exécution-dialogue* traduit cette intention en message à partir de *L*, *Pr* et de *KR.soc* et l'inscrit dans *ES*. L'émetteur communique ce message aux autres agents.

La fonction d'organisation inscrit les restrictions sur l'activité de résolution et de dialogue dans *CS.soc* à partir de l'organisation *Org*, *KR.soc* et des éléments inscrits dans *CS.soc*.

Dans le cycle de contrôle de l'agent, nous isolons le cycle de résolution, consistant en l'enchaînement des fonctions d'évaluation, de raisonnement, décision, d'engagement. Ce cycle est interrompu lorsque la décision concerne une action sur l'extérieur. Le cycle de contrôle (cf. figure 5.2) est défini par l'enchaînement des fonctions: *capteur*, *récepteur* sur l'environnement, *interprétation* des éléments perçus, *interprétation-dialogue* des messages, *résolution* ou *organisation*, puis *exécution-dialogue* et *exécution* à partir des résultats de la *résolution* ou de l'*organisation*, action sur l'environnement par l'intermédiaire des fonctions *effecteur* ou *émetteur*.

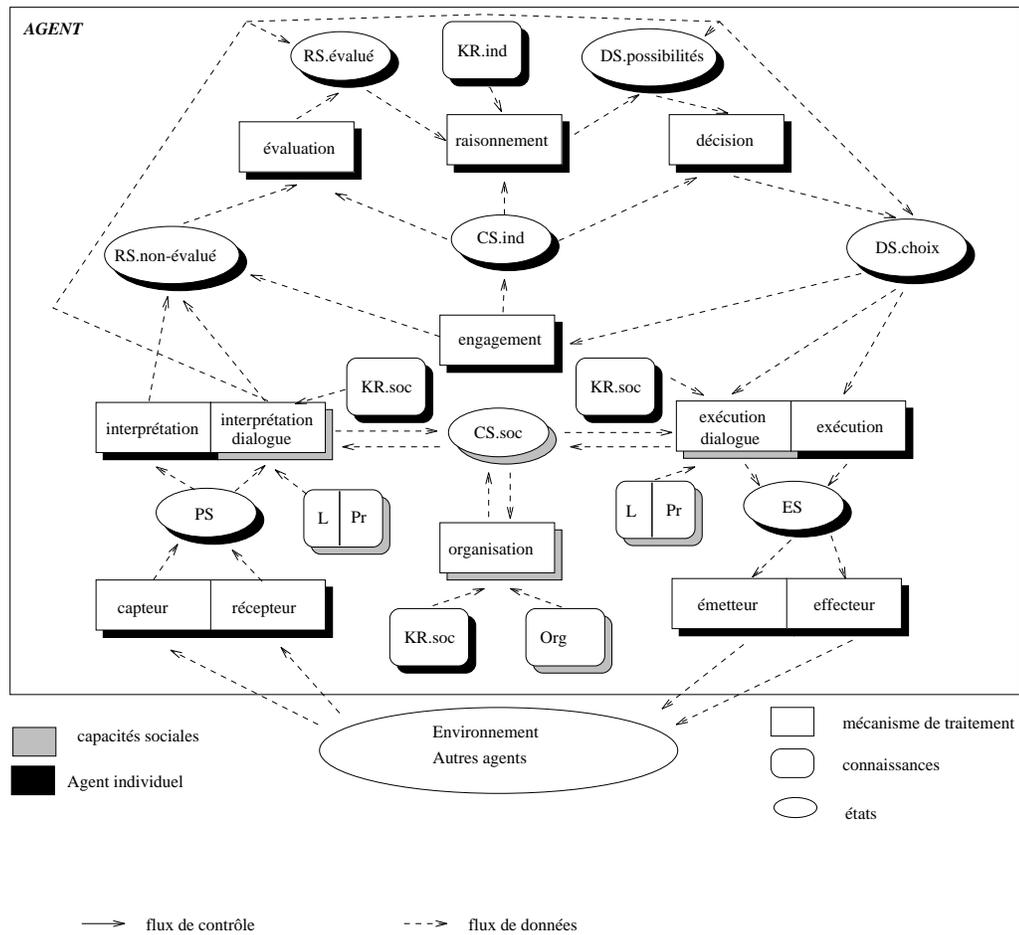


FIG. 5.1 - **Architecture de l'agent social.** L'architecture est constituée des états, représentations des perceptions *PS*, du raisonnement *RS*, des décisions *DS*, des engagements *CS* et des exécutions *ES*. Les connaissances utilisées pour les modifier sont représentées dans *KR*, dans le langage d'interaction *L*, dans les protocoles d'interaction *Pr* et dans les organisations *Org*.

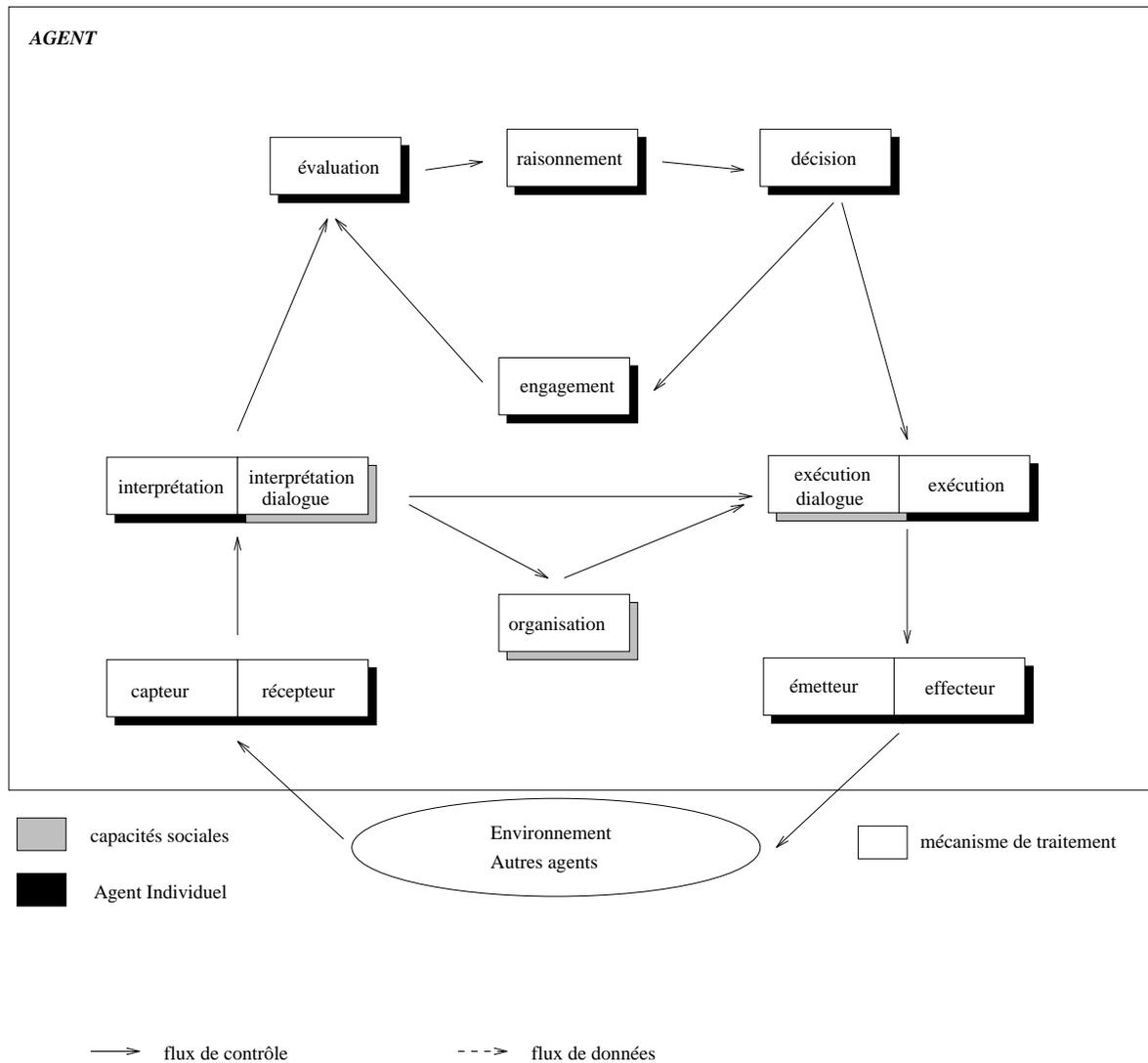


FIG. 5.2 - Cycle de contrôle du modèle d'agent social.

Nous débutons par la *représentation des connaissances*  $KR$  qui définit les *capacités de raisonnement et d'actions de l'agent*. Nous présentons ensuite les états de *perception*  $PS$ , d'*exécution*  $ES$ , de *raisonnement*  $RS$ , de *décision*  $DS$  et d'*engagement*  $CS$ . Nous terminons par les *mécanismes de traitement*  $PM$  utilisés pour construire ces états à partir des connaissances en possession de l'agent.

Le temps n'apparaît pas explicitement dans la présentation de ce modèle. Chaque état est mis à jour à chaque cycle selon différents modes de gestion que nous détaillons dans la section suivante sur le contrôle.

*Pour la présentation de ce modèle, nous illustrons les notions introduites par un exemple volontairement simple : trois agents R2D2, C2D2 et T2D2, équipés chacun d'un bras manipulateur, de capacités de perception et de communication, sont dans un environnement constitué d'une table sur laquelle reposent différents cubes, portant soit la lettre A, soit la lettre B, soit la lettre C, et des triangles. R2D2 et C2D2 ne savent que manipuler des cubes. T2D2 ne sait que manipuler les triangles. Nous exprimons notre modèle par rapport à R2D2.*

### a. Représentation des connaissances ( $KR$ )

L'ensemble des connaissances  $KR$  est décomposé en deux sous ensembles : connaissances individuelles  $KR.ind$  et connaissances sociales  $KR.soc$ . Ils sont constitués des ensembles de *faits* (**fait**), de *compétences* (**comp**), de *tactiques* (**tact**) et de *stratégies* (**strat**):

$$\begin{aligned} KR &= KR.ind \cup KR.soc \\ KR.ind &= KR.ind[\mathbf{fait}] \cup KR.ind[\mathbf{comp}] \cup KR.ind[\mathbf{tact}] \cup KR.ind[\mathbf{strat}] \\ KR.soc &= KR.soc[\mathbf{fait}] \cup KR.soc[\mathbf{comp}] \cup KR.soc[\mathbf{tact}] \cup KR.soc[\mathbf{strat}] \end{aligned}$$

### Connaissances individuelles ( $KR.ind$ )

#### Faits ( $KR.ind[\mathbf{fait}]$ )

L'ensemble de faits ou connaissances factuelles,  $KR.ind[\mathbf{fait}]$  désigne l'ensemble des informations sur le domaine et sur lui-même en possession, a priori, de l'agent.

*Nous inscrivons par exemple dans cet ensemble les ressources en la possession de l'agent [Bur 91]. Une ressource désigne les éléments utilisables et nécessaires pour la production de nouvelles informations : éléments propres à l'agent R2D2, tel son bras manipulateur, ses capacités de communication.*

#### Compétences ( $KR.ind[\mathbf{comp}]$ )

L'ensemble de compétences,  $KR.ind[\mathbf{comp}]$  est décomposé en connaissances sur les actions, **actions** et en connaissances sur la résolution, **opérations**. L'exécution d'une

**action** conduit à l'utilisation des effecteurs de l'agent alors que l'exécution d'une **opération** modifie l'état de raisonnement. Leurs sous-ensembles respectifs sont notés :  $KR.ind[comp][action]$  et  $KR.ind[comp][opération]$ .

$$KR.ind[comp] = KR.ind[comp][action] \cup KR.ind[comp][opération]$$

- $KR.ind[comp][action]$  désigne l'ensemble des compétences en relation avec l'**action** conduisant à une exécution par la fonction *effecteur* de l'agent. Il définit les *capacités d'action* de l'agent. Parmi celles-ci, les compétences de **communication** se traduisent en appels à l'émetteur et au récepteur permettant à l'agent de communiquer avec les autres agents. Leur représentation par des compétences les fait apparaître explicitement dans les décisions d'action de l'agent : celui-ci peut ou non, recevoir ou émettre des messages [Mar 90].

*Des connaissances d'actions de R2D2 expriment les conditions permettant de prendre des actions de déplacement des cubes dans l'environnement. Elles sont représentées par exemple par :*

*(si (libre?X) et (libre bras) alors (prendre ?X))*

*(si (libre?Y) et (dans bras ?X) alors (poser ?X ?Y))*

*La partie condition teste la possibilité d'utilisation de cette action dans l'environnement. Par exemple, (si (libre ?X) et (libre bras)) signifie que l'objet ?X ne doit rien avoir de posé dessus et le bras manipulateur doit être vide.*

*Une compétence de communication de R2D2 utilisée lorsqu'il doit déplacer un triangle est la suivante :*

*(si (triangle?X) alors (communiquer déplacer?X))*

*La fonction d'exécution traduira cette action dans le langage d'interaction et enverra ce message à l'agent T2D2 pour que celui-ci déplace le triangle ?X.*

- $KR.ind[comp][opération]$  désigne l'ensemble des connaissances déductives permettant de mettre à jour l'état de raisonnement de l'agent. Il définit les *capacités de raisonnement* de l'agent. Parmi celles-ci, nous trouvons les compétences lui permettant de raisonner sur les *actions* des autres agents.

*L'agent R2D2 pourra ainsi avoir des opérations consistant à simuler l'exécution d'une action réelle pour envisager les conséquences possibles. Une telle opération peut être représentée par la reformulation des actions précédentes avec une liste d'ajouts, de retraites et de conditions. Son exécution consistera à modifier l'état de raisonnement conformément aux spécifications des listes d'ajout et de retrait.*

### **Tactiques** ( $KR.ind[tact]$ )

Les *tactiques* appartenant à  $KR.ind[tact]$  sont les connaissances utilisées par la fonction de raisonnement pour contraindre le fonctionnement de l'agent. Elles expriment, par exemple, des critères de sélection des opérations ou des actions à exécuter.

*De tels critères peuvent par exemple vouloir privilégier les actions de communications par rapport à celles conduisant à l'utilisation du bras manipulateur.*

### Stratégies ( $KR.ind[**strat**]$ )

Les *stratégies* appartenant à  $KR.ind[**strat**]$  sont les connaissances déductives utilisées par la fonction de *raisonnement* pour contraindre le fonctionnement de l'agent à un plus long terme que les tactiques. Elles expriment, par exemple, des critères de sélection de tactiques.

$KR.ind[**strat**]$  et  $KR.ind[**tact**]$  définissent les *capacités de décision* de l'agent. Elles sont utilisées par la fonction de *raisonnement* pour déterminer les buts, ou plans possibles. Elles peuvent aussi être des plans ou des séquences de critères de contrôle prédéfinis. Dans ce cas, elles comportent des conditions d'activation évaluées dans la fonction de raisonnement sur l'état de raisonnement.

### Connaissances sociales ( $KR.soc$ )

Les connaissances sociales de l'agent sont relatives aux connaissances pour l'utilisation des capacités sociales : *exécution-dialogue*, *interprétation-dialogue* et d'*organisation*. On retrouve la même structuration en ensemble de faits, compétences, tactiques et stratégies.

#### b. Etat de perception ( $PS$ )

L'*état de perception*,  $PS$ , représente les informations issues de la *perception* de l'environnement par l'agent, appelées **données**, et celles envoyés par les autres agents, appelées **messages**.

$$PS = PS[**donnée**] \cup PS[**message**]$$

Un **message** est un ensemble d'informations exprimées dans le langage d'interaction  $L$ , commun à l'ensemble des agents. Son expression est explicitée dans le modèle de contrôle social.

La traduction des **données** en éléments utilisables par l'agent est réalisée par la fonction d'*interprétation*, alors que celle des **messages** est réalisée par la fonction d'*interprétation-dialogue* en utilisant le langage d'interaction  $L$  et les protocoles d'interaction  $Pr$ .

*Les données qu'inscrit R2D2 dans son état de perception sont des images fournies par sa caméra. R2D2 ne manipulant que des informations symboliques du type (sur  $A B$ ), il devra analyser ces images pour en déduire une représentation de l'environnement dans ces termes.*

**c. Etat d'exécution ( $ES$ )**

L'état d'exécution,  $ES$ , est la représentation de l'ensemble des traductions des **actions** à exécuter dans l'environnement en **données**, commandes pour l'**effecteur** concerné par l'**action**. Dans le cadre de la traduction d'**actions** de type **communication** dans le langage d'interaction  $L$ , il comporte les **messages**.

$$ES = ES[donnée] \cup ES[message]$$

La traduction des **actions** de type **communication** est assurée par la fonction d'*exécution-dialogue* alors que la traduction des autres **actions** est assurée par la fonction d'*exécution*.

**d. Etat de raisonnement ( $RS$ )**

L'état de *raisonnement*,  $RS$ , représente les informations manipulées par l'agent pour représenter : l'environnement perçu (fournis par l'*interprétation* de  $PS[donnée]$  et par l'*interprétation-dialogue* de  $PS[message]$ ) et les traitements entrepris localement (issus de la fonction d'*engagement*). Nous décomposons  $RS$  en informations *non-évaluées* et en informations *évaluées* selon qu'elles ont été soumises ou non à la fonction *évaluation*. Nous les distinguons en **hypothèses**, en **conflits**, en **buts** ou en **plans**.

$$RS = RS.non-évalué \cup RS.évalué$$

$$RS = RS[hypothèse] \cup RS[conflit] \cup RS[but] \cup RS[plan]$$

**Hypothèses ( $RS[hypothèse]$ )**

Les **hypothèses** appartenant à  $RS[hypothèse]$  sont la modélisation de l'environnement et de l'activité de l'agent.

*L'agent R2D2 a pour tâche de déplacer les cubes. Il représente une partie de son environnement par une liste de relations entre chacun de ces cubes. Une hypothèse exprimant que le cube A est sur le cube B sera représentée par (sur A B).*

Les **accointances** sont les hypothèses relatives à un autre agent de la société. L'expression d'une accointance est orientée par les intérêts propres de l'agent. Uniformisant cette notion, nous incluons aussi une description de l'agent lui même.

*L'agent R2D2 représente par exemple [Ber 92], l'accointance agent C2D2 à la suite de l'introduction de celui-ci dans la société par :*

(accointance (fonction C2D2 déplacer cube))

(accointance (action C2D2 poser prendre))

(accointance (structure C2D2 bras))

*exprimant que la fonctionnalité de cet agent est de déplacer des cubes, en utilisant les actions poser et prendre pour la réaliser, qui utilisent un bras manipulateur comme effecteur.*

*Il peut aussi inclure dans cette accointance le fait que l'agent T2R2 est en train de déplacer un triangle, c.-à-d. risque de collision possible au cas où R2D2 veut aussi déplacer un objet sur la table. Cette information est utilisée, par exemple, pour planifier les actions relatives au mouvement de son bras manipulateur.*

### Conflits ( $RS[\mathbf{conflict}]$ )

Les **conflits** appartenant à  $RS[\mathbf{conflict}]$  expriment les incohérences détectées par la fonction d'évaluation entre les informations au sein de l'état de raisonnement. Suivant la proposition faite par Galliers [Gal 90], nous inscrivons explicitement dans l'état de raisonnement de l'agent, toute incohérence détectée. Nous l'utilisons comme indicateur pour la remise en cause d'une décision. Nous y reviendrons dans le cadre du modèle de contrôle individuel.

### Buts ( $RS[\mathbf{but}]$ )

Les **but**s appartenant à  $RS[\mathbf{but}]$  expriment les objectifs que doit satisfaire l'agent. Leur satisfaction peut consister à atteindre une configuration particulière de l'environnement.

*Le but de l'agent R2D2 est d'obtenir une représentation de l'environnement telle que le cube A soit sur B lui-même sur C : (but (sur A B) et (sur B C))*

*Des objectifs plus généraux s'expriment en terme de critères à minimiser : minimiser le nombre de déplacements de cubes par exemple.*

### Plans ( $RS[\mathbf{plan}]$ )

Les **plans** appartenant à  $RS[\mathbf{plan}]$  expriment une suite d'**actions** ou d'**opérations** à utiliser pour satisfaire les **but**s de l'agent.

*Pour le but précédent, la fonction de raisonnement en utilisant les tactiques représentées dans  $KR.ind[\mathbf{tact}]$  construit le plan constitué de la séquence d'actions suivante :*

*(prendre B);(poser B C);(prendre A);(poser A B)*

*où prendre et poser sont l'expression des actions que peut effectuer l'agent. Ce plan a été élaboré ou activé à la suite d'une configuration particulière de l'environnement.*

Dans notre cas, un **plan** pourra aussi représenter une suite de critères sélectionnant les actions qu'entreprend l'agent : privilégier les actions de type *prendre*; puis privilégier les actions de type *poser* par exemple.

Nous adoptons la notion élargie que l'on trouve habituellement en IA en permettant à un plan d'être exécuté soit sur l'environnement, acception habituelle, soit sur l'état de raisonnement de l'agent. Dans le cas où l'agent est équipé de capacités d'apprentissage,

les **plans** exécutés avec succès, par exemple, peuvent être inscrits dans l'ensemble  $KR.ind[\mathbf{tact}]$  ou  $KR.ind[\mathbf{strat}]$ . Nous n'envisageons pas ce cas dans notre modèle. Nous citons cette éventualité pour montrer la relation entre les **plans** et les **tactiques** et **stratégies**.

#### e. Etat de décision ( $DS$ )

L'état de décision ( $DS$ ) représente les *intentions*. Les *intentions* expriment la relation existant entre les **buts**, les **plans**, les **actions** ou les **opérations** que l'agent peut entreprendre, avec son état de raisonnement  $RS$ . Elles sont le résultat de mise en rapport des éléments de  $KR.ind[\mathbf{strat}]$ , de  $KR.ind[\mathbf{tact}]$  ou de  $KR.ind[\mathbf{comp}]$  avec l'état de *raisonnement* par les fonctions de *raisonnement* et de *décision*. Nous les répartissons dans deux ensembles : *ensemble de possibilités* noté  $DS.possibilités$  et *ensemble de choix* noté  $DS.choix$ . A l'issue de la fonction de *raisonnement* les intentions sont inscrites dans  $DS.possibilités$ . Elles sont inscrites dans  $DS.choix$  à l'issue de la fonction de *décision*.

$$\begin{aligned} DS &= DS.possibilités \cup DS.choix \\ DS &= DS[\mathbf{action}] \cup DS[\mathbf{opération}] \cup DS[\mathbf{plan}] \cup DS[\mathbf{but}] \end{aligned}$$

##### – $DS.possibilités$

Il regroupe les *intentions* relatives aux **opérations**, aux **actions**, aux **plans**, et aux **buts** possibles étant donné la configuration de l'état de raisonnement  $RS$ .

$$\begin{aligned} DS.possibilités &= DS.possibilités[\mathbf{opération}] \cup DS.possibilités[\mathbf{action}] \cup \\ &DS.possibilités[\mathbf{plan}] \cup DS.possibilités[\mathbf{but}] \end{aligned}$$

Ainsi dans l'exemple précédent, l'action donnée ci-dessus (paragraphe sur les connaissances) pour l'agent R2D2 est inscrite dans  $DS.possibilités[\mathbf{action}]$ , ensemble des possibilités, si les conditions exprimées dans l'action dont elle est issue sont satisfaites dans l'environnement  $RS$  : le bras manipulateur est libre et le cube peut être déplacé. Cette intention exprime que cette action est possible étant donné l'état de raisonnement. Une fois choisie, elle exprime que cette action sera exécutée par le bras manipulateur.

##### – $DS.choix$

Il regroupe les *intentions* que le mécanisme de *décision* a choisi pour contraindre le fonctionnement de l'agent (**but**, **plan**) ou pour modifier son état de raisonnement (**opération**) ou pour agir sur l'environnement (**action**).

$$\begin{aligned} DS.choix &= DS.choix[\mathbf{opération}] \cup DS.choix[\mathbf{action}] \cup DS.choix[\mathbf{plan}] \cup \\ &DS.choix[\mathbf{but}] \end{aligned}$$

Une intention décrivant à plus long terme le comportement de l'agent R2D2 désigne le but (but (sur  $A B$ ) et (sur  $B C$ )). Son choix implique sa satisfaction en produisant un plan et l'exécution des actions correspondantes par l'agent. Pour chaque nouvel état de

*RS, R2D2 regarde les actions qui sont possibles et crée des intentions correspondantes. Il les sélectionne en utilisant le plan construit.*

### f. Etat d'engagement (*CS*)

L'état d'engagement (*CS*) représente les *engagements* contrôlant le fonctionnement de l'agent tant d'un point de vu individuel *CS.ind* que social *CS.soc*.

- Un *engagement* individuel est la traduction d'une intention relative à un **but** ou un **plan** en critères de contrôle conditionnant le fonctionnement des fonctions d'évaluation, de raisonnement, de décision.
- Un *engagement* social conditionne l'activité sociale de l'agent en agissant sur les fonctions d'interprétation-dialogue, d'exécution-dialogue, et d'organisation.

Nous explicitons *CS.ind* dans le cadre du modèle de contrôle individuel et *CS.soc* dans le cadre du modèle de contrôle social.

$$\begin{aligned} CS &= CS.ind \cup CS.soc \\ CS.ind &= CS.ind[\mathbf{plan}] \cup CS.ind[\mathbf{but}] \end{aligned}$$

*L'agent R2D2 engage le but qu'il a choisi en le traduisant en différents critères de contrôle qui conditionnent son action de construction et de sélection de plans afin de satisfaire ce but. Lors de l'engagement du plan, un ensemble de critères sera exprimé. Ils contraindront la fonction de décision lors de la sélection des compétences à exécuter.*

### g. Mécanismes de traitement (*PM*)

Les mécanismes de traitement (notés *PM*) utilisent la représentation des connaissances *KR* pour mettre à jour les états de *perception PS*, de *raisonnement RS*, de *décision DS*, d'*engagement CS* et d'*exécution ES* de l'agent. Nous présentons ces fonctions dans leur ordre d'utilisation dans le cycle de contrôle : *capteur* et *récepteur*, *interprétation* et *interprétation-dialogue*, *évaluation*, *raisonnement*, *décision*, *engagement*, *exécution* et *exécution-dialogue*, *effecteur* et *émetteur*, *organisation*.

#### Capteur

La fonction *capteur* applique les capteurs de l'agent sur son environnement  $\Omega$ . Les *données* perçues sont inscrites dans l'état de *perception PS* de l'agent.

$$\text{capteur} : \Omega \rightarrow PS[\mathbf{donnée}]$$

#### Récepteur

La fonction *récepteur* reçoit les *messages* envoyés par les autres agents. Elle les inscrit dans *PS*.

$$\text{récepteur} : \Omega \rightarrow PS[\text{message}]$$

## Interprétation

La fonction d'*interprétation* traduit les **données** perçues inscrits dans *PS* dans les termes de la représentation interne de l'agent.

$$\text{interprétation} : PS[\text{donnée}] \rightarrow RS.\text{non-évalué}$$

## Interprétation-dialogue

La fonction d'*interprétation-dialogue* traduit les **messages** reçus inscrits dans *PS* à l'aide du langage d'interaction *L*, dans les termes de la représentation interne de l'agent, c.à.d. en terme d'**hypothèses**, de **buts** ou de **plans**. Son fonctionnement est contraint par *CS.soc*, *Pr* et par les connaissances sociales de l'agent *KR.soc*. Les échanges prenant place entre les agents peuvent conduire à la modification du comportement de l'agent récepteur. Ainsi, celui-ci peut par exemple recevoir un **plan** à exécuter. Ce **plan** est alors inscrit dans l'*ensemble de choix*.

$$\text{interprétation-dialogue} : PS[\text{message}] \times L \times CS.\text{soc} \times \\ KR.\text{soc} \times Pr \rightarrow RS \cup DS$$

## Evaluation

La fonction *évaluation* détecte les incohérences entre les éléments de l'état de raisonnement mis à jour par la fonction d'*engagement* ou par les fonctions d'*interprétation*, d'*interprétation-dialogue*. Comme nous le verrons dans le modèle de contrôle individuel, son fonctionnement est contraint par le contenu de *CS.ind*. A l'issue de l'évaluation de l'état de raisonnement, les **conflits** détectés sont ajoutés à l'état de raisonnement.

$$\text{évaluation} : RS.\text{non-évalué} \times RS.\text{évalué} \times CS.\text{ind} \rightarrow RS.\text{évalué}$$

## Raisonnement

La fonction *raisonnement* construit l'ensemble des *intentions* possibles (*DS.possibilités*) à partir de l'état de raisonnement (*RS.évalué*) et des *faits* (*KR.ind[fait]*), des *compétences* (*KR.ind[comp]*), des *tactiques* (*KR.ind[tact]*) et des *stratégies* (*KR.ind[strat]*) de l'agent. Son fonctionnement est contraint par les engagements *CS.ind* (cf. section suivante sur le contrôle). Trois formulations de cette fonction *raisonnement* peuvent être envisagées : examen des **buts**, des **plans** (planification), des **compétences** qui sont possibles étant donné l'état de *raisonnement*.

$$\text{raisonnement} : RS.\text{évalué} \times KR.\text{ind} \times CS.\text{ind} \rightarrow DS.\text{possibilités}$$

## Décision

La fonction de *décision* choisit les *intentions* de l'*ensemble de possibilités* ( $DS.possibilités$ ) à partir des *engagements* définis dans  $CS.ind$ . Les intentions choisies sont inscrites dans l'*ensemble de choix* ( $DS.choix$ ). Comme pour la fonction précédente, son fonctionnement est contraint par les engagements  $CS.ind$ . Elle peut être aussi décomposée en trois aspects : décision sur les **but**s, décision sur les **plan**s et décision sur les **action**s ou **opération**s. Nous la décrivons en détail dans la section sur le contrôle.

$$d\acute{e}cision : DS.possibilités \times CS.ind \rightarrow DS.choix$$

## Engagement

Si *intentions* choisies inscrites dans l'*ensemble de choix* ( $DS.choix$ ) sont des **opération**s, la fonction d'*engagement* les exécute et met à jour l'*état de raisonnement* ( $RS$ ) de l'agent. Dans le cas où elles sont relatives à des **but**s ou des **plan**s, elle les traduit en *engagements* et met à jour l'ensemble d'*engagement*  $CS.ind$ . Nous explicitons et justifions la formulation de cette fonction dans la section sur le contrôle.

$$\begin{aligned} engagement & : DS.choix[opération] \rightarrow RS.non-évalué \\ engagement & : DS.choix[plan] \rightarrow CS.ind[plan] \\ engagement & : DS.choix[but] \rightarrow CS.ind[but] \end{aligned}$$

## Exécution

La fonction d'*exécution* traduit une *intention* relative à une **action** de l'*ensemble de choix*  $DS.choix$  de l'agent en une **donnée** pour l'*effecteur*.

$$exécution : DS.choix[action] \rightarrow ES[donnée]$$

## Exécution-dialogue

La fonction d'*exécution-dialogue* traduit une *intention* relative à une **action** de type **communication** de l'*ensemble de choix* ( $DS.choix$ ) de l'agent en un **message**. Son fonctionnement est contraint par les protocoles d'interaction  $Pr$ , le langage d'interaction  $L$ , les engagements sociaux de  $CS.soc$  et par les connaissances sociales  $KR.soc$

$$\begin{aligned} exécution-dialogue & : DS.choix[communication] \times L \times Pr \times \\ & CS.soc \times KR.soc \rightarrow ES[message] \end{aligned}$$

## Effecteur

La fonction *effecteur* applique sur l'environnement l'*effecteur* correspondant à la ou les **action**s traduites en **données**.

$$\text{effecteur} : ES[\text{donnée}] \rightarrow \Omega$$

### Emetteur

La fonction *émetteur* émet les **messages** inscrits dans *ES* en direction des autres agents.

$$\text{émetteur} : ES[\text{message}] \rightarrow \Omega$$

### Organisation

La fonction d'*organisation* définit l'organisation *CS.soc* de l'agent à partir de l'organisation *Org* préexistante dans la société, des engagement sociaux inscrits dans *CS.soc* et des connaissances sociales *KR.soc*. Nous l'explicitons dans le cadre du modèle de contrôle social.

$$\text{organisation} : CS.soc \times Org \times KR.soc \rightarrow CS.soc$$

### h. Conclusion

En conclusion, le modèle d'agent social que nous utilisons par la suite est (cf. figure 5.2).

$$\begin{aligned} \text{Agent} &= \langle KR, L, Pr, Org, PS, RS, DS, CS, ES, PM \rangle \\ PM &= \langle \text{capteur}, \text{récepteur}, \text{interprétation}, \text{interprétation-dialogue}, \text{évaluation}, \\ &\quad \text{raisonnement}, \text{décision}, \text{engagement}, \text{exécution}, \text{exécution-dialogue}, \\ &\quad \text{effecteur}, \text{émetteur}, \text{organisation} \rangle \end{aligned}$$

– Connaissances

$$\begin{aligned} KR &= KR.ind \cup KR.soc \\ KR.ind &= KR.ind[\text{fait}] \cup KR.ind[\text{comp}] \cup KR.ind[\text{tact}] \cup KR.ind[\text{strat}] \\ KR.soc &= KR.soc[\text{fait}] \cup KR.soc[\text{comp}] \cup KR.soc[\text{tact}] \cup KR.soc[\text{strat}] \end{aligned}$$

– Etats

$$PS = PS[\text{donnée}] \cup PS[\text{message}]$$

$$ES = ES[\text{donnée}] \cup ES[\text{message}]$$

$$RS = RS.non-évalué \cup RS.évalué$$

$$RS = RS[hypothèse] \cup RS[conflit] \cup RS[but] \cup RS[plan]$$

$$DS = DS.possibilités \cup DS.choix$$

$$DS = DS[action] \cup DS[opération] \cup DS[plan] \cup DS[but]$$

$$DS.possibilités = DS.possibilités[opération] \cup DS.possibilités[action] \cup$$

$$DS.possibilités[plan] \cup DS.possibilités[but]$$

$$DS.choix = DS.choix[opération] \cup DS.choix[action] \cup DS.choix[plan] \cup$$

$$DS.choix[but]$$

$$CS = CS.ind \cup CS.soc$$

$$CS.ind = CS.ind[plan] \cup CS.ind[but]$$

– Mécanismes de traitement

$$capteur : \Omega \rightarrow PS[donnée]$$

$$récepteur : \Omega \rightarrow PS[message]$$

$$interprétation : PS[donnée] \rightarrow RS.non-évalué$$

$$interprétation-dialogue : PS[message] \times L \times CS.soc \times$$

$$Pr \times KR.soc \rightarrow RS \cup DS$$

$$évaluation : RS.non-évalué \times RS.évalué \times CS.ind \rightarrow RS.évalué$$

$$raisonnement : RS.évalué \times KR.ind \times CS.ind \rightarrow DS.possibilités$$

$$décision : DS.possibilités \times CS.ind \rightarrow DS.choix$$

$$engagement : DS.choix[opération] \rightarrow RS.non-évalué$$

$$engagement : DS.choix[but] \rightarrow CS.ind[but]$$

$$engagement : DS.choix[plan] \rightarrow CS.ind[plan]$$

$$exécution : DS.choix[action] \rightarrow ES[donnée]$$

$$exécution-dialogue : DS.choix[communication] \times L \times Pr \times$$

$$CS.soc \times KR.soc \rightarrow ES[message]$$

$$effecteur : ES[donnée] \rightarrow \Omega$$

$$émetteur : ES[message] \rightarrow \Omega$$

$$organisation : CS.soc \times Org \times KR.soc \rightarrow CS.soc$$

### 5.1.2 Modèle d'Agent Individuel

Au sein de ce modèle d'agent social, nous isolons le modèle d'agent individuel constitué des mécanismes de traitement, des états et des connaissances réalisant l'activité de *résolution*, fonctionnement *individuel* de l'agent (cf. figure 5.1 et 5.2 composants sur fond noir):

$$Agent_{ind} = \langle KR.ind, PS, RS, DS, CS.ind, ES, PM \rangle$$

$$PM = \langle \text{capteur, récepteur, interprétation, évaluation, raisonnement,} \\ \text{décision, engagement, exécution, effecteur, émetteur} \rangle$$

Les fonctions d'*exécution*, d'*interprétation*, d'*effecteur*, d'*émetteur*, de *capteur* et de *récepteur* sont présentes dans ce modèle dans la mesure où celles-ci ne concernent que l'action dans l'environnement de l'agent sans aucune interaction avec les autres agents.

### 5.1.3 Modèle de société

Dans la littérature informatique, nous trouvons peu ou pas de modèles explicites de société. En effet, du fait de la distribution des capacités propres à la société au sein de chacun des agents, la modélisation de la société se trouve incluse dans la modélisation de ceux-ci. C'est par exemple le cas de [Lev 90] [Bur 91][Bou 92] ou *MACE* [Gas 87]. D'autres auteurs ont pris en compte la notion de société dans leur formalisation en introduisant les notions de connaissances, de buts, de plans mutuels et étudient ainsi l'effet des communications sur les états mentaux des agents [Coh 90] [Rao 91b] [Wer 89].

Dans la description qui suit, suivant la démarche entreprise dans [PLE 92], nous suivons le même canevas de description que celui pris pour le modèle d'agent. Nous décrivons ainsi la *structure* de la société et les *mécanismes de traitements*. Cette présentation explicite du modèle de société a pour objectif de séparer le *contrôle individuel* du *contrôle social* que nous utilisons pour notre étude dans les sections suivantes.

Nous définissons le *modèle de société* par le 6-uplet :

$$société = \langle \mathcal{A}, L, Pr, Org, CS.soc, PM \rangle$$

$$PM = \langle \text{dialogue, organisation} \rangle$$

$$\text{dialogue} = \langle \text{interprétation-dialogue, exécution-dialogue} \rangle$$

$$Pr = Pr[\mathbf{résolution}] \cup Pr[\mathbf{organisation}]$$

$$Org = Org[\mathbf{résolution}] \cup Org[\mathbf{dialogue}] \cup Org[\mathbf{organisation}]$$

$\mathcal{A}$  est l'ensemble des agents de la société exprimés dans le modèle d'agent individuel  $Agent_{ind}$ ,  $L$  est le langage d'interaction définissant les *interactions*,  $Pr$  est l'ensemble des protocoles qu'utilisent les agents pour interagir,  $Org$  est l'ensemble des organisations possibles de la société,  $CS.soc$  est l'état d'engagement social,  $PM$  est l'ensemble des mécanismes utilisant les éléments précédents. Ayant présenté dans le cadre du modèle d'agent social, les mécanismes de traitement et l'état d'engagement social, nous détaillons ici, le langage d'interaction  $L$ ,  $Pr$  et  $Org$ .

#### a. Langage d'interaction ( $L$ )

Le langage d'*interaction*,  $L$ , intervient dans l'*interprétation* des éléments de l'état de *perception*  $PS$  en éléments de l'état de *raisonnement* et dans la traduction des éléments

de l'état de *décision* en éléments de l'état d'*exécution*.

Nous définissons une *interaction* comme toute action effectuée par un agent qui provoque en retour une action ou un choix d'un autre agent [Sea 69]. Dans le cadre de notre modèle, une *interaction* conduit à l'échange d'un des éléments des états de *raisonnement* ou de *décision* de l'agent traduit dans le langage de interaction  $L$ . Une interaction concerne donc les échanges d'*hypothèses*, *plans*, *buts* et *actions* formulés dans les termes du langage  $L$ . Le contenu d'un message peut s'inscrire dans l'état de *raisonnement* de l'agent ou dans son état de *décision*. Dans le cadre de la section suivante, nous nuancerons légèrement cette présentation. Actuellement celle-ci nous suffit.

## b. Protocoles d'interactions ( $Pr$ )

Un *protocole d'interaction* définit l'enchaînement des *interactions* entre les agents de la société. Nous en distinguons deux groupes : *protocoles de résolution*,  $Pr[\mathbf{résolution}]$  et *protocoles d'organisation*,  $Pr[\mathbf{organisation}]$ .

$$Pr = Pr[\mathbf{résolution}] \cup Pr[\mathbf{organisation}]$$

### Protocoles de résolution ( $Pr[\mathbf{résolution}]$ )

Les *protocoles de résolution* définissent l'enchaînement des interactions entre les agents conduisant à l'apport de nouvelles *hypothèses*, de nouveaux *plans*, de nouveaux *buts* ou de nouvelles *actions* au sein d'un agent.

### Protocoles d'organisation ( $Pr[\mathbf{organisation}]$ )

Les *protocoles d'organisation* définissent les enchaînement d'interaction entre les agents qui participent à la mise en place d'une *organisation* dans la société. Un exemple d'un tel protocole est le *Contract Net Protocol*[Smi 81] qui au travers d'une négociation entre les agents permet de mettre en place des relations entre eux : un agent est contractant d'un autre, etc.

## c. Organisation ( $Org$ )

L'*organisation* ( $Org$ ) définit la structure de gestion des interactions entre les agents. Nous la définissons comme un ensemble de *rôles*. Un *rôle* est une spécialisation imposée sur le fonctionnement des agents de la société. Nous distinguons trois types de rôle relatifs aux trois activités que peut entreprendre un agent inclus dans une société : *résolution*, *dialogue* et *organisation*. Nous présenterons de manière plus détaillée ces éléments dans la section relative au modèle de contrôle social.

$$Org = Org[\mathbf{résolution}] \cup Org[\mathbf{dialogue}] \cup Org[\mathbf{organisation}]$$

### Rôle de résolution ( $Org[\mathbf{résolution}]$ )

Les rôles appartenant à cet ensemble contraignent le fonctionnement des fonctions de *raisonnement* et de *décision* de l'agent en limitant les éléments des états de *raisonnement*  $RS$  et de *décision*  $DS$  que l'agent peut utiliser. Par exemple, dans le cadre d'une organisation hiérarchique, un rôle de résolution s'applique sur les agents d'un niveau au travers de l'interdiction de prendre certaines décisions et en les obligeant à les transmettre au niveau supérieur. Une autre expression de ce rôle oblige ces agents à exécuter l'ensemble des ordres provenant du niveau supérieur.

*Dans la société constituée des trois agents, les rôles de résolution désignent, par exemple, les liens d'autorité qui régissent leurs interactions : R2D2 ne peut pas demander à l'agent C2D2 de déplacer des cubes, par contre il doit exécuter toutes les demandes de la part de T2D2.*

### Rôle de dialogue ( $Org[\mathbf{dialogue}]$ )

Les rôles de dialogue limitent les protocoles d'interaction que peut utiliser l'agent dans la société.

*Ainsi toujours dans le cadre de notre exemple, l'interdiction de communiquer avec un agent situé au même niveau sans passer par le niveau supérieur est un rôle de dialogue.*

### Rôle d'organisation ( $Org[\mathbf{organisation}]$ )

Les rôles d'organisation définissent quelle est l'action possible d'un agent dans la construction d'une nouvelle organisation, c'est-à-dire dans la définition et la distribution des deux rôles précédents.

## 5.2 Modèles de contrôle

Ayant présenté le modèle d'agent social à partir duquel, nous avons isolé le modèle d'agent individuel et ses capacités sociales entrant dans le cadre du modèle de société, nous détaillons les aspects du contrôle relatifs, respectivement, au modèle d'agent individuel et au modèle de société. La réunion de ces deux éléments constituent le modèle de contrôle social. Dans cette optique, nous rappelons ici rapidement les notions introduites dans le chapitre 2.

Deux entités sont distinguées au sein d'un système : le *contrôleur* et le *processus*. Le contrôleur définit la *commande* qui contraint le fonctionnement du processus. Le fonctionnement du contrôleur est contraint lui-même par un *objectif* et une *loi de commande*. La *loi de commande* est un ensemble de critères de contrôle utilisés pour sélectionner

la commande parmi celles qui sont possibles étant donnée une *observation* de l'état du processus. Nous avons défini trois *couches de contrôle* au sein du contrôleur, résolvant chacune un choix spécifique :

- La couche *décision* définit l'*objectif* du processus.
- La couche *adaptation* définit sa *loi de commande*.
- La couche *commande* définit sa *commande*.

Nous considérons dans un premier temps le contrôleur et le processus comme étant au sein du même agent en faisant toute abstraction des interactions qui peuvent se produire dans son environnement. Nous définissons ainsi le *modèle de contrôle individuel* à partir du modèle d'agent individuel de la section précédente.

Dans un second temps, nous définissons le *modèle de contrôle social* à partir du modèle de société précédent en faisant abstraction de l'activité de résolution de l'agent et en considérant les couples (contrôleur, processus) comme étant attachés à des agents différents. Nous explicitons ainsi dans le modèle de société, les modules de *dialogue* et d'*organisation* et l'état d'engagement social *CS.soc*.

### 5.2.1 Modèle de contrôle individuel

Ce modèle est basé sur les propositions, énoncées en conclusion du chapitre 4 :

- **Proposition 1 :** *L'architecture de contrôle d'un agent est organisée selon la hiérarchie des trois couches de contrôle : commande, adaptation et décision.*

La structure et les mécanismes de traitement de l'agent sont explicités selon cette hiérarchie afin de permettre la définition des intentions relatives aux *buts*, aux *plans* et aux *actions* et *opérations* en tenant compte de l'évolution de l'environnement.

- **Proposition 2 :** *La gestion explicite du passage d'une couche de contrôle à l'autre est réalisée par la fonction de bascule.*

Le passage d'une couche de contrôle à l'autre est réalisé, d'une part par une représentation explicite des *engagements* régissant le fonctionnement de la couche inférieure, et d'autre part par un mécanisme de *bascule* gérant la validité des engagements et déclenchant le changement de couche.

#### a. Contrôleur et Processus

Au sein du système constitué par le modèle d'agent individuel  $Agent_{ind}$ , nous distinguons les mécanismes liés au processus et ceux liés au contrôleur :

$$Agent_{ind} = \langle Processus, Contrôleur \rangle$$

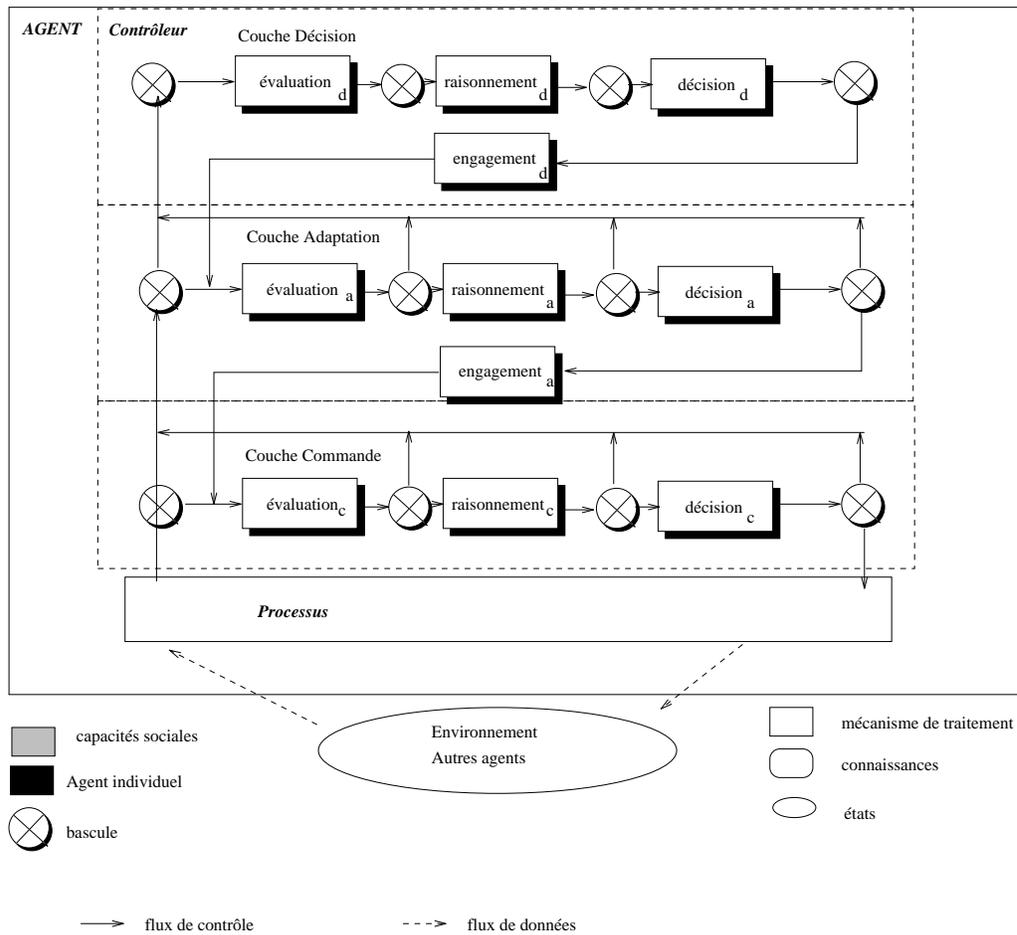


FIG. 5.3 - **Modèle de contrôle individuel.** Le modèle de contrôle individuel organise les représentations, connaissances et mécanismes de traitement du modèle d'agent individuel selon la hiérarchie de trois couches de contrôle : décision dédiée à la définition des engagements relatifs aux *buts*, adaptation dédiée à la définition des engagements relatifs aux *plans*, commande dédiée à la définition des engagements relatifs aux *opérations* ou aux *actions*.

## Processus

Le processus est constitué par :

$$\text{Processus} = \langle PS, ES, RS.\text{non-évalué}, DS.\text{choix}[\text{opération}], DS.\text{choix}[\text{action}] \\ \text{capteur}, \text{interprétation}, \text{exécution}, \text{effecteur}, \text{engagement} \rangle$$

Il correspond soit :

- à la modification de l'état de *raisonnement* par la fonction d'*engagement* appliquée à l'*intention* choisie relative à une **opération** (équivalente à la commande),
- à la modification de l'environnement par l'exécution d'une *intention* relative à une **action** par les fonctions d'*exécution* et d'*effecteur*. La perception de l'environnement est obtenue par les fonctions de *perception* et d'*interprétation*.

Dans les deux cas, *RS.non-évalué* est la perception de l'état du processus. La commande appliquée au processus est l'*intention* relative à une **opération** ou à une **action** que l'agent a choisie d'exécuter.

## Contrôleur

Le contrôleur est constitué par :

$$\text{Contrôleur} = \langle KR.\text{ind}, RS.\text{non-évalué}, RS.\text{évalué}, DS.\text{possibilités}, DS.\text{choix}, \\ CS.\text{ind}, \text{évaluation}, \text{raisonnement}, \text{décision}, \text{engagement}, \text{bascule} \rangle$$

La fonction d'*engagement* est incluse au sein du contrôleur lorsque celle-ci concerne des *intentions* relatives aux **buts** ou aux **plans**. La fonction de *bascule* permet le passage d'une couche à l'autre. Le contrôleur définit l'*intention* relative à une **opération** ou à une **action** à appliquer au processus étant donné l'état de *raisonnement* *RS.non-évalué*.

## Cycle de contrôle

Le *cycle de contrôle* de l'agent définit la séquence itérative des fonctions constituant les couches de contrôle permettant au processus de changer l'état de *raisonnement* *RS*. Dans la présentation qui suit, nous indexons les états de l'agent par le numéro de cycle de contrôle exécuté. Nous considérons qu'un nouveau cycle de contrôle débute à tout nouvel état de *raisonnement*, c'est-à-dire après toute exécution de l'*intention* relative à une **opération** ou à une **action** par le processus. Ainsi, l'état de *raisonnement* *RS.non-évalué* à un cycle  $t$  contient les éléments créés au cycle  $t - 1$  par la fonction d'*engagement* relative à une **opération** et ceux provenant de l'environnement (fonction d'*interprétation*)

## b. Structure et traitements d'une couche de contrôle

Nous décomposons selon les trois couches de contrôle la structure et les traitements du contrôleur. Nous les indexons avec la première lettre de la couche de contrôle sur laquelle ils sont considérés<sup>1</sup> :  $c$  pour commande,  $a$  pour adaptation et  $d$  pour décision. La lettre  $e$  désigne les éléments manipulés par le processus. La notation  $x - 1$  (resp.  $x + 1$ ) est utilisée pour désigner la couche inférieure (resp. supérieure) à la couche  $x$ .

Un contrôleur agissant sur une couche de contrôle est défini par le 10-uplet suivant (cf. figure 5.4) :

$$\text{Contrôleur}_x = \langle RS.\text{non-évalué}, RS_x.\text{évalué}, DS_x, CS_x.\text{ind}, KR_x.\text{ind}, \\ \text{bascule}_x, \text{évaluation}_x, \text{raisonnement}_x, \text{décision}_x, \text{engagement}_x \rangle$$

Le fonctionnement des fonctions d'évaluation, de raisonnement et de décision est similaire à celui que nous avons défini dans le modèle dans la section précédente. Il est régi par les engagements définis par la couche de contrôle supérieure. La fonction de bascule s'intercale entre chacune d'elles et évalue la validité de ces engagements. Elle réactive la couche supérieure en cas d'invalidation d'un engagement.

## Connaissances

Nous répartissons les connaissances  $KR.\text{ind}$  de l'agent selon les trois couches de contrôle de la manière suivante :

- couche décision : Les stratégies interviennent sur les couches de décision.

$$KR_d.\text{ind} = KR.\text{ind}[\text{strat}] \cup KR.\text{ind}[\text{fait}]$$

- couche adaptation : Les tactiques interviennent sur la couche d'adaptation.

$$KR_a.\text{ind} = KR.\text{ind}[\text{tact}] \cup KR.\text{ind}[\text{fait}]$$

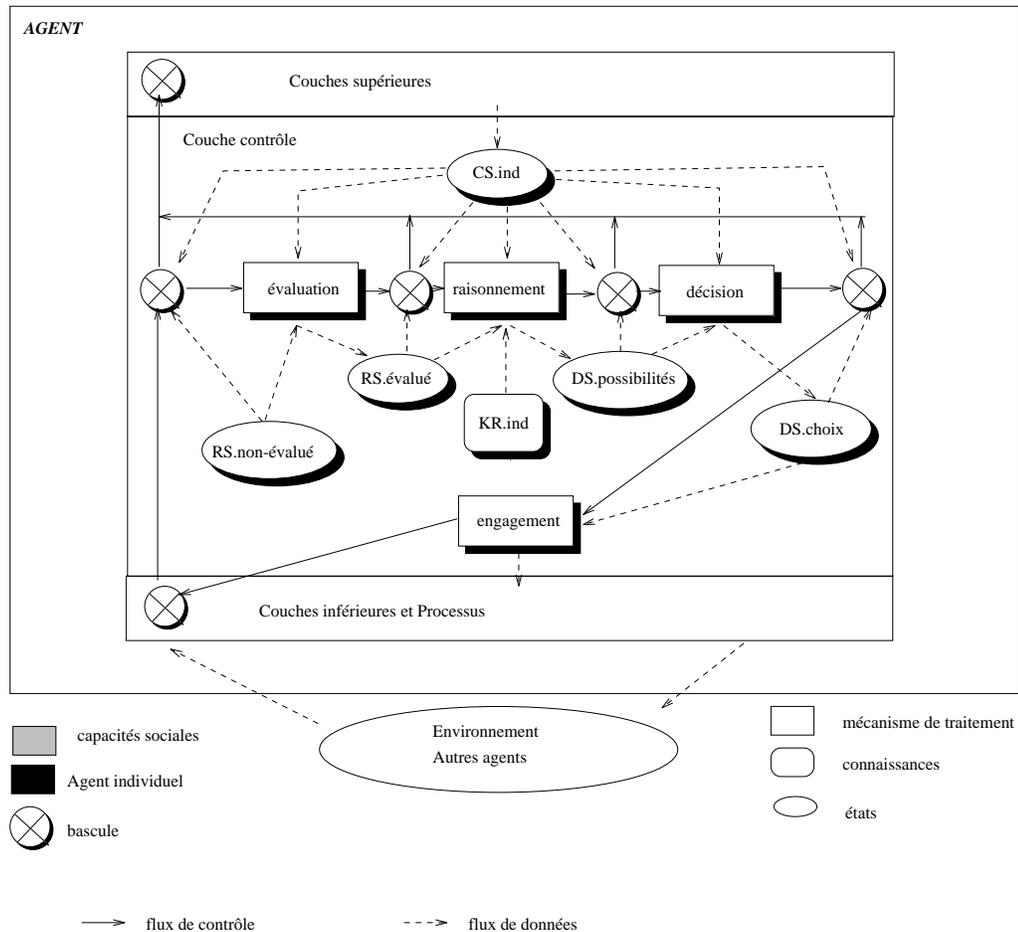
- couche commande : Les compétences interviennent sur la couche de commande.

$$KR_c.\text{ind} = KR.\text{ind}[\text{comp}] \cup KR.\text{ind}[\text{fait}]$$

## Etat de raisonnement

L'état de raisonnement n'est pas réparti à priori sur les différentes couches de contrôle. L'ensemble des éléments qu'il contient est accessible par les fonctions de toutes les couches de contrôle.

<sup>1</sup>Nous préférons utiliser ce type de notation plutôt que de leur affecter des noms particulier du type *behave*, *react*, *plan* comme ceci est fait par exemple dans [Lev 90]. Ces termes font trop référence à des techniques d'implantation particulières.



**FIG. 5.4 - Architecture d'une couche de contrôle du modèle de contrôle individuel.** Le contrôleur d'une couche de contrôle est constitué des fonctions d'évaluation, de raisonnement, de décision d'engagement et de bascule. La fonction d'évaluation détecte les incohérences de l'état de raisonnement. La fonction de raisonnement construit l'ensemble de possibilités. La fonction de décision choisit l'intention. La fonction d'engagement traduit cette intention en critères d'incohérence, d'activation, de sélection et de validité qui définissent le contrôle de la couche inférieure. La fonction de bascule effectue le changement de couche de contrôle en évaluant la validité des engagements définis par la couche supérieure. Ceux-ci régissent le fonctionnement de l'ensemble des fonctions de la couche.

## Etat de décision

L'état de décision est distingué selon les couches de contrôle considérées (cf. figure 5.5).

- couche décision: Les éléments de l'état de *décision* manipulés sur cette couche sont les *intentions* relatives aux **buts**.

$$\begin{aligned} DS_d.possibilités &= DS.possibilités[\mathbf{but}] \\ DS_d.choix &= DS.choix[\mathbf{but}] \end{aligned}$$

- couche adaptation: Les éléments de l'état de décision manipulés sur cette couche sont les *intentions* relatives à un **plan**.

$$\begin{aligned} DS_a.possibilités &= DS.possibilités[\mathbf{plan}] \\ DS_a.choix &= DS.choix[\mathbf{plan}] \end{aligned}$$

- couche commande: Les *intentions* relatives à une **opération** ou à une **action** sont manipulées sur la couche *commande* et sont exécutées par les fonctions d'*engagement* ou d'*exécution*.

$$\begin{aligned} DS_c.possibilités &= DS.possibilités[\mathbf{opération}] \cup DS.possibilités[\mathbf{action}] \\ DS_c.choix &= DS.choix[\mathbf{opération}] \cup DS.choix[\mathbf{action}] \end{aligned}$$

## Etat d'engagement

Un *engagement* intervenant sur une couche de contrôle  $x$  est décomposé en quatre critères : **incohérence**, **activation**, **sélection** et **validité**. Ceux-ci interviennent respectivement sur les fonctions  $évaluation_x$ ,  $raisonnement_x$ ,  $décision_x$  et  $bascule_x$ . Cette décomposition a pour objectif d'exprimer les éléments de contrôle propres à chaque fonction.

$$\begin{aligned} CS_x.ind &= CS_x.ind[\mathbf{sélection}] \cup CS_x.ind[\mathbf{validité}] \cup \\ &\quad CS_x.ind[\mathbf{activation}] \cup CS_x.ind[\mathbf{incohérence}] \\ \text{où } x &\in \{d, a, c\} \end{aligned}$$

- $CS_x.ind[\mathbf{incohérence}]$ : conditions de création d'un **conflit** dans l'état de *raisonnement* par la fonction d' $évaluation_x$ .

Ces critères expriment les conditions de prise en compte d'une incohérence dans l'état de *raisonnement* entre les éléments entrant  $RS.non-évalué$  et ceux présents  $RS.évalué$ . Leur expression permet ainsi de faire varier dynamiquement la notion d'incohérence au sein d'un agent.

- $CS_x.ind[\mathbf{activation}]$ : conditions régissant l'inscription d'une *intention* dans l'ensemble  $DS_x.possibilités$  par la fonction de  $raisonnement_x$ .

Ces critères permettent la modification du cardinal de l'*ensemble des possibilités*. Ils correspondent, par exemple, à un seuil appliqué aux coefficients de vraisemblance des hypothèses satisfaisant les conditions d'activation d'opérations exprimées sous forme de règle de production.

- $CS_x.ind[**sélection**]$ : conditions d'inscription d'une *intention* dans  $DS_x.choix$  par la fonction de *décision*.
- $CS_x.ind[**validité**]$ : conditions de remise en cause des *engagements* conduisant à l'activation de la couche supérieure  $x + 1$  par la fonction de *bascule* $_x$ .

Ces critères expriment les conditions de validité des critères précédents : **activation**, **incohérence**, **sélection** par rapport aux états de *raisonnement* et de *décision* de l'agent. Cette notion est à rapprocher des *engagements*<sup>2</sup> qui apparaissent dans les formalisations d'architecture *BDI* [Rao 91a] [Sin 91a] [Coh 90]. Dans celles-ci, l'engagement exprime les ressources temporelles et de calcul que l'on est prêt à dépenser pour l'exécution de l'intention.

Ces critères sont *extrinsèques* dans la mesure où ils sont définis par la fonction d'*engagement* $_{x+1}$  de la couche supérieure  $x + 1$ . Ils ont leur équivalent, d'un point de vue *intrinsèque*, inscrits dans les fonctions correspondantes de la couche  $x$ . Ceux-ci ne peuvent être changés. Par exemple, les critères agissant sur la couche *décision* ne pouvant être définis par une couche supérieure sont *intrinsèques*.

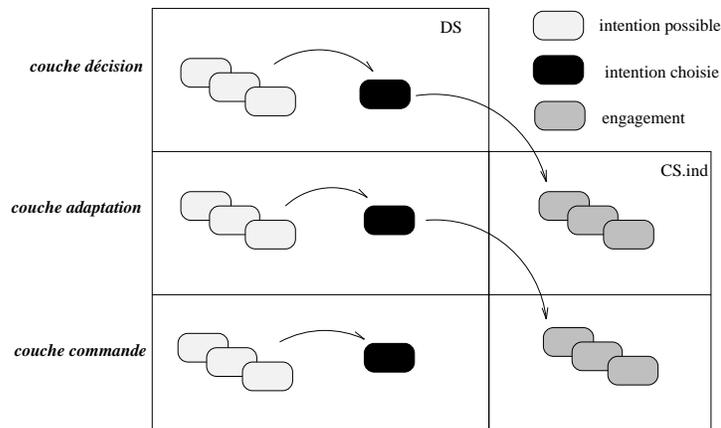


FIG. 5.5 - Répartition des états de décision et d'engagement sur les couches de contrôle. Les états de *décision*, d'*engagement* sont structurés selon les couches de contrôle. Les intentions relatives aux stratégies et aux tactiques sont traduites en engagements exprimant des critères de sélection, de validité, d'activation et d'incohérence. Les intentions relatives à une opération ou à une action sont exécutées par le processus

<sup>2</sup>commitment

## Mécanismes de traitements

### – *évaluation<sub>x</sub>*

Elle définit les **conflits** de  $RS[\mathbf{conflit}](t)$  entre les éléments de l'état *raisonnement non-évalué*  $RS.non-évalué(t)$  de l'agent à l'issue du cycle ( $t$ ) avec les éléments de l'état de *raisonnement évalué* du cycle précédent ( $t-1$ )  $RS.non-évalué(t)$  par application des critères appartenant à  $CS_x.ind[\mathbf{incohérence}](t)$

$$\begin{aligned} \text{évaluation}_x : RS.non-évalué(t) \times RS.évalué(t-1) \\ \times CS_x.ind[\mathbf{incohérence}](t) \rightarrow RS.évalué(t) \end{aligned}$$

Les **conflits** détectés peuvent ou non être traités par les fonctions de *raisonnement<sub>x</sub>* ou de *décision<sub>x</sub>* ou par celles d'une couche supérieure. Ce traitement dépend de l'application et des critères de  $CS_x.ind[\mathbf{validité}]$  correspondants.

La fonction d'évaluation résout une partie du problème d'observation inclus dans notre problématique du contrôle : ses sorties (conflits) sont utilisées pour la construction de la nouvelle commande ou pour sa remise en cause.

### – *raisonnement<sub>x</sub>*

Elle construit l'ensemble des *possibilités*  $DS_x.possibilités(t)$  en utilisant les critères appartenant à  $CS_x.ind[\mathbf{activation}](t)$  étant donné l'état de *raisonnement évalué* ( $RS.évalué(t)$ ) et les connaissances  $KR_x.ind$ .

$$\begin{aligned} \text{raisonnement}_x : RS.évalué(t) \times \\ KR_x.ind \times CS_x.ind[\mathbf{activation}](t) \rightarrow DS_x.possibilités(t) \end{aligned}$$

### – *décision<sub>x</sub>*

Elle applique les critères appartenant à  $CS_x.ind[\mathbf{sélection}](t)$  sur l'ensemble des *possibilités* ( $DS_x.possibilités(t)$ ) et choisit l'*intention* qui est inscrite dans l'ensemble de *choix* ( $DS_x.choix(t)$ ).

$$\text{décision}_x : DS_x.possibilités(t) \times CS_x.ind[\mathbf{sélection}](t) \rightarrow DS_x.choix(t)$$

### – *engagement<sub>x</sub>*

Elle traduit l'*intention* inscrite dans l'ensemble de *choix* ( $DS_x.choix(t)$ ) en critères d'**incohérence**, d'**activation**, de **sélection** et de **validité** dans l'ensemble d'engagement de la couche inférieure ( $CS_{x-1}.ind(t)$ ).

$$\text{engagement}_x : DS_x.choix(t) \rightarrow CS_{x-1}.ind(t) \quad \text{où } x \in \{d, a\}$$

Dans le cas où l'on est sur la couche commande, la fonction d'engagement ne fait pas de traduction en termes de critères. Elle exécute l'*intention* relative à l'**opération**. Cette fonction appartient au processus.

– *bascule<sub>x</sub>*

Elle applique les critères de  $CS_x.ind[validité]$  sur les différents états de la couche de contrôle  $x$  en s'intercalant entre chacune des fonctions. Si le résultat de la bascule est à 1, le cycle de contrôle est modifié, la couche de contrôle supérieure est activée et le critère est invalidé.

$$bascule_x : (RS(t) \cup DS_x(t)) \times CS_x.ind[validité](t) \rightarrow \{0, 1\}$$

**remarque :**

Au sein du contrôleur d'une couche de contrôle, les choix réalisés peuvent prendre différentes formes. Les fonctions de *décision* et de *raisonnement* sont décomposées en une succession de fonctions appliquant un seul critère de **sélection** et d'**activation** sur l'état de *raisonnement* et sur l'ensemble de *possibilités* (proposition 4). Nous exprimons ainsi ce que l'on trouve dans les systèmes de production avec les choix d'état (critère de **sélection** relatif à la version de l'état de raisonnement considéré), choix d'objet (critère de **sélection** portant sur les **hypothèses**, les **buts**, les **plans**), choix d'action (critère de **sélection** portant sur les *intentions*).

**c. Conclusion**

Nous pouvons ainsi formuler le modèle de contrôle individuel comme suit :

$$Agent_{ind} = \langle Processus, Contrôleur \rangle$$

## – Le processus

$$Processus = \langle PS, RS.non-évalué, DS.choix[opération], DS.choix[action], ES\ captteur, interprétation, exécution, effecteur, engagement \rangle$$

$$captteur : \Omega \rightarrow PS[donnée](t)$$

$$interprétation : PS[donnée](t) \rightarrow RS.non-évalué(t+1)$$

$$exécution : DS.choix[action](t) \rightarrow ES[donnée](t)$$

$$effecteur : ES[donnée](t) \rightarrow \Omega$$

$$engagement_c : DS.choix[opération](t) \rightarrow RS.non-évalué(t+1)$$

## – Le contrôleur

$$Contrôleur = \langle Contrôleur_d, Contrôleur_a, Contrôleur_c \rangle$$

où  $d$  est la couche de décision,  $a$  la couche d'adaptation,  $c$  la couche de commande, tel que :

$$\forall x \in \{d, a, c\}$$

$$\text{Contrôleur}_x = \langle KR_x.ind, RS.non\text{-évalué}, RS_x.évalué, DS_x, CS_x.ind, \\ \text{bascule}_x, \text{évaluation}_x, \text{raisonnement}_x, \text{décision}_x, \\ \text{engagement}_x \rangle$$

$$KR_d.ind = KR.ind[\mathbf{strat}] \cup KR.ind[\mathbf{fait}]$$

$$KR_a.ind = KR.ind[\mathbf{tact}] \cup KR.ind[\mathbf{fait}]$$

$$KR_c.ind = KR.ind[\mathbf{comp}] \cup KR.ind[\mathbf{fait}]$$

$$RS = RS.évalué \cup RS.non\text{-évalué}$$

$$RS = RS[\mathbf{hypothèse}] \cup RS[\mathbf{plan}] \cup RS[\mathbf{but}] \cup RS[\mathbf{conflit}]$$

$$DS_d.possibilités = DS.possibilités[\mathbf{but}]$$

$$DS_d.choix = DS.choix[\mathbf{but}]$$

$$DS_a.possibilités = DS.possibilités[\mathbf{plan}]$$

$$DS_a.choix = DS.choix[\mathbf{plan}]$$

$$DS_c.possibilités = DS.possibilités[\mathbf{opération}] \cup DS.possibilités[\mathbf{action}]$$

$$DS_c.choix = DS.choix[\mathbf{opération}] \cup DS.choix[\mathbf{action}]$$

$$CS_x.ind = CS_x.ind[\mathbf{sélection}] \cup CS_x.ind[\mathbf{validité}] \cup$$

$$CS_x.ind[\mathbf{activation}] \cup CS_x.ind[\mathbf{incohérence}]$$

$$\text{où } x \in \{d, a, c\}$$

$$\forall x \in \{d, a, c\}$$

$$\text{évaluation}_x : RS.non\text{-évalué}(t) \times RS.évalué(t-1) \\ \times CS_x.ind[\mathbf{incohérence}](t) \rightarrow RS.évalué(t)$$

$$\text{raisonnement}_x : RS.évalué(t) \times \\ KR_x.ind \times CS_x.ind[\mathbf{activation}](t) \rightarrow DS_x.possibilités(t)$$

$$\text{décision}_x : DS_x.possibilités(t) \times CS_x.ind[\mathbf{sélection}](t) \rightarrow DS_x.choix(t)$$

$$\text{bascule}_x : (RS(t) \cup DS_x(t)) \times CS_x.ind[\mathbf{validité}](t) \rightarrow \{0, 1\}$$

$$\text{engagement}_x : DS_x.choix(t) \rightarrow CS_{x-1}.ind(t)$$

$$\text{où } x \in \{d, a\}$$

Supposons que l'agent R2D2 ait construit sur la couche d'adaptation le plan constitué de la séquence d'actions suivante : (prendre B); (poser B C); (prendre A); (poser A B)

où prendre et poser sont l'expression des actions que peut effectuer l'agent. L'intention relative à ce plan est choisie et est traduite par la fonction d'engagement en critères de sélection du type : (choisir les actions de type prendre cube de lettre B), (choisir les actions de type poser de type prendre cube de lettre B) ...

Sur la couche commande, étant donné l'état de raisonnement un ensemble d'actions sont possibles, la fonction de décision les choisit en fonction du critère de sélection. Une fois celui-ci satisfait, l'autre critère est activé, etc. L'agent R2D2 intervenant dans un environnement ouvert, son environnement peut être modifié sans qu'il en soit la cause. Ainsi en définissant ses critères de sélection, il a défini un critère de validité exprimant les conditions sous lesquelles l'exécution de ce plan doit être poursuivie.

Les critères de validité attachés aux critères de sélection précédents peuvent être de la forme : (libre B) (libre C), ... Le passage à la couche adaptation sera ainsi déclenché par la bascule lorsque l'une de ces conditions sera vérifiée dans l'environnement. Un nouveau plan sera ainsi créé.

### 5.2.2 Modèle de contrôle social

Dans cette section, nous considérons une société exprimée dans le modèle précédent. Chaque agent a sa propre activité de traitements pouvant être dans un cadre local (individuel) ou global au système (social) en entrant en interaction avec les autres agents. Les *interactions* sont les influences qu'un agent exerce sur le fonctionnement des autres. La problématique du contrôle social consiste à obtenir un *comportement global cohérent* à partir des résolutions locales entreprises par chacun des agents et de la limitation et de la gestion des interactions qu'ils peuvent mettre en place. Les propositions énoncées en conclusion du chapitre 4 nous conduisent à contraindre la définition du modèle de société.

□ **Proposition 9 :** *Modification de l'architecture d'un agent en fonction de l'organisation définie.*

La définition de la structure de contrôle de la société permet de préciser et de modifier les couches de contrôle possédées par chacun des agents.

□ **Proposition 5 :** *Définition des organisations en terme de processus-contrôleur.*

Pour cela, nous structurons le modèle de *société* selon les trois couches de contrôle sur lesquelles nous répartissons les représentations et les traitements présentés dans le modèle de société (cf. figure 5.6).

- couche décision : Les fonctions d'*organisation* et de *dialogue* (*interprétation-dialogue* et *exécution-dialogue*) définissent, en utilisant *Org* et *Pr*, les rôles d'*organisation* et les rôles de *dialogue* pour l'activité d'organisation, respectivement dans  $CS_a.soc[\mathbf{organisation}]$  et  $CS_a.soc[\mathbf{dialogue}]$  qui déterminent les capacités de l'agent en terme de construction de l'organisation.
- couche adaptation : construction de l'organisation en utilisant les fonctions *organisation* et *dialogue*, *Org* et *Pr* $[\mathbf{organisation}]$  et en étant contraints par  $CS_a.soc$ .

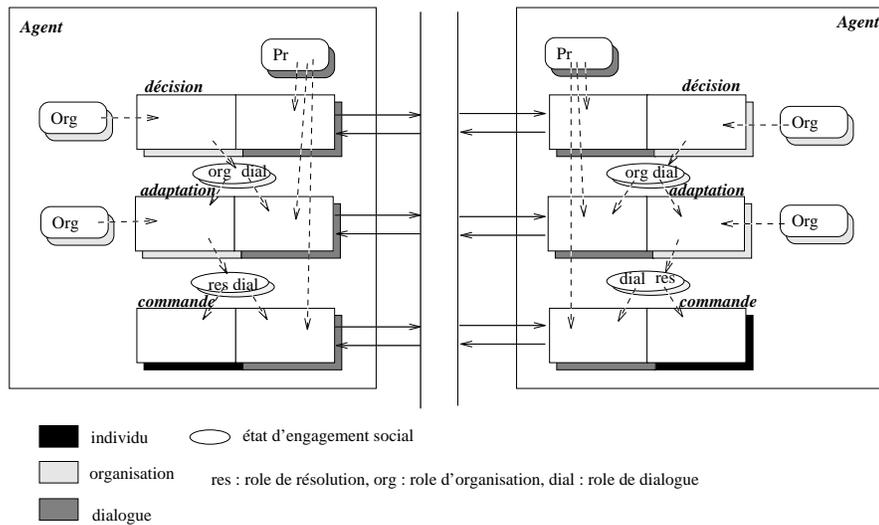


FIG. 5.6 - **Modèle de contrôle social.** Dans le modèle de contrôle social, les capacités sociales des agents sont organisées selon les trois couches de la hiérarchie de contrôle. Le modèle de contrôle individuel s'inscrit sur la couche commande. La fonction d'*organisation* est décomposée en deux fonctions agissant respectivement sur la couche adaptation et sur la couche décision. Les fonctions d'*interprétation-dialogue* et d'*exécution-dialogue* sont décomposées en différentes fonctions agissant sur chacune des couches.

Cette couche inscrit les rôles de résolution et les rôles de dialogue dans les ensembles  $CS_c.soc[\mathbf{résolution}]$  et  $CS_c.soc[\mathbf{dialogue}]$ . (proposition 5)

- couche commande : La couche commande est constituée des  $Agent_{ind}$  et des mécanismes de *dialogue* mettant en œuvre l'activité de résolution. Cette activité est contrainte par les rôles de résolution et de dialogue inscrits respectivement dans  $CS_c.soc[\mathbf{résolution}]$  et dans  $CS_c.soc[\mathbf{dialogue}]$ . Des mécanismes de bascules *social bascule<sub>social</sub>* et *individu bascule<sub>individu</sub>* rendent opératoires les rôles de résolution établis entre les agents.

Les rôles de résolution  $CS_c.soc[\mathbf{résolution}]$  sont définis en terme de liens d'interaction exprimant les relations contrôleur-processus existant entre les agents. Ils font référence au modèle de contrôle individuel précédent. (proposition 9).

Les rôles de dialogue  $CS_c.soc[\mathbf{dialogue}]$  définissent les protocoles représentés dans  $Pr[\mathbf{résolution}]$  utilisables par le module de dialogue étant donné l'organisation existante.

- Le processus contrôlé est le *processus* défini dans le modèle d'agent individuel de chacun des agents.

La fonction de *dialogue* gère les interactions se déroulant dans la société. Les protocoles d'interaction définissent l'enchaînement des interactions entre les agents. Dans le cadre de la définition du modèle de contrôle pour un système intégré de vision, nous montrerons comment ceux-ci peuvent répondre aux propositions 6 et 7. Son contrôle est régi par les rôles de *dialogue*.

Dans ce qui suit, nous présentons la couche commande de ce modèle. Nous définissons ainsi la notion de *lien d'interaction* à partir de laquelle, nous définissons les rôles de résolutions et les bascules *social bascule<sub>social</sub>* et *individu bascule<sub>individu</sub>*. Nous présentons ensuite la fonction de *dialogue*.

### a. Liens d'interaction

Un lien d'interaction définit une relation entre deux agents. A partir du modèle de contrôle individuel précédent, nous caractérisons cette relation par rapport à la couche de contrôle, *nature*, et par rapport à l'état abordé (*RS* ou *DS*), *force*, dans lesquels le lien d'interaction aboutit ou prend naissance. Selon leur valeur, nous définissons deux groupes de liens : *lien de contrôle* et *lien d'observation* (cf. figure 5.7).

La définition d'un lien est relative aussi à un *sujet* d'interaction. Celui-ci est propre à l'application. Sa prise en compte permet de définir des liens d'interaction par sujet d'interaction. On peut par exemple se rapprocher des notions de *groupes d'intérêts* exprimés dans [Num 91].

*Dans le cadre des agents R2D2, C2D2 et T2D2, les sujets d'interaction consistent en demande d'aide, en conflit de ressource lorsque deux des agents veulent utiliser le même cube, etc.*

Pour ne pas surcharger les notations dans ce qui suit, nous ne faisons pas apparaître cette notion de *sujet*, tout en gardant à l'esprit que l'ensemble des liens sont définis par rapport à un *sujet*.

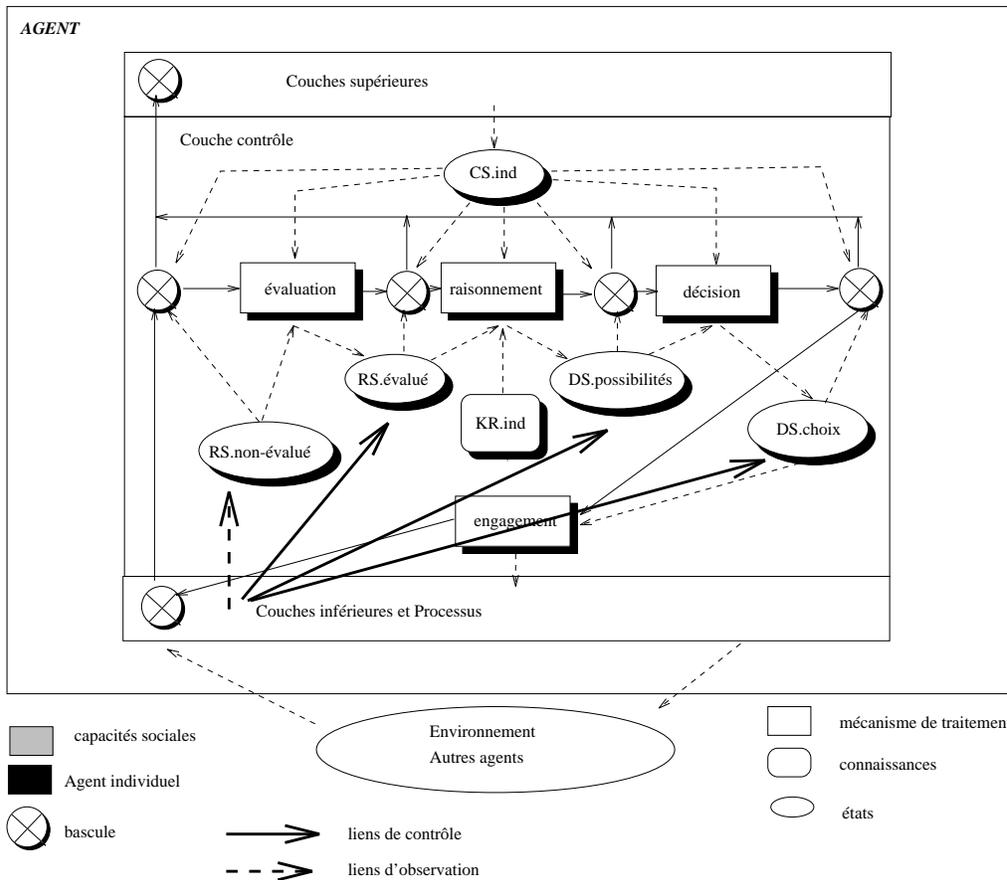


FIG. 5.7 - Effets possibles d'un lien d'interaction. Les liens de contrôle en flèches accentuées de trait plein aboutissent dans l'un des états de la couche de contrôle. Les liens d'observation en flèches hachurées inscrivent leur contenu dans l'état de raisonnement.

## Nature

La *nature* d'un lien d'interaction définit la couche de contrôle sur laquelle le contenu de l'interaction est traité ou représenté: décision, adaptation, commande. Le contenu de l'interaction entre deux agents est soit un **but** (couche décision), un **plan** (couche adaptation), une **action** (couche commande), ou une **hypothèse** (description de l'environnement). Par abus, nous exprimons la *nature* par ces termes.

$$nature \in \{hypothèse, action, plan, but\}$$

## Force

La *force* d'une interaction définit l'état abordé par le lien d'interaction au sein de la couche de contrôle spécifiée par la *nature*. Afin de spécifier différents degrés d'interaction, la force a une valeur comprise dans deux intervalles dont les bornes fixent l'état abordé.

$$force \in [informative..non-directive] \cup [non-directive..directive]$$

- *force* = *directive*.
- *nature*  $\neq$  *hypothèse* :  
Le contenu de l'interaction est inscrit dans l'*ensemble de choix*  $DS_x.chaix$ . La couche de contrôle  $x$  est fixée par la *nature*. Le contenu de l'interaction est donc choisi sans que la fonction de *décision* de l'agent n'ait pu appliquer les critères locaux de *sélection*<sup>3</sup>.
- *nature* = *hypothèse* :  
Le contenu de l'interaction s'inscrit dans l'état de raisonnement *RS.évalué* sans que la fonction d'*évaluation* ne puisse évaluer le contenu.
- *force*  $\in$  [*non-directive..directive*]<sup>4</sup>.
  - *nature*  $\neq$  *hypothèse* :  
le contenu de l'interaction peut être inscrit dans l'*ensemble de possibilités*  $DS_x.possibilités$ . La couche de contrôle  $x$  est fixée par la *nature*. Dans ce cas là il est déclaré possible sans que la fonction de *raisonnement* de cette couche ne l'ait traité. En revanche, il sera traité par la fonction de *décision*. L'agent décide en fonction de ses critères locaux de *sélection* de son choix ou non. Une nuance peut être exprimée par cette force en permettant au contenu d'être traité par la fonction de *raisonnement* qui peut ainsi décider de son inscription ou non dans l'*ensemble de possibilités*.
  - *nature* = *hypothèse* :  
Nous retrouvons la même discussion que précédemment au sujet du court-circuit de la fonction d'*évaluation*.
- *force*  $\in$  [*informative..non-directive*]  
Le contenu de l'interaction va s'inscrire dans l'état de *raisonnement non-évalué*

<sup>3</sup>Les liens d'interactions de ce type sont connues sous le nom de *maître-esclave*.

<sup>4</sup>Nous trouvons dans ce cadre de figure, les systèmes utilisant par exemple des protocoles de  *négociation* développés ces dernières années [Smi 81] [Con 86]: négociation de résultats (nature est une hypothèse) ou négociation de buts, de plans. Entre les bornes *non-directive* et *directive* s'intercalent plusieurs nuances de contrôle telles par exemple la volonté de négocier l'adoption du contenu, voire même de fournir les éléments pour persuader l'agent (cf. le système *PERSUADER* [Syc 89]).

de l'agent *RS.non-évalué*. Il sera ensuite traité par l'ensemble des fonctions des couches de contrôle *évaluation, raisonnement* et *décision*<sup>5</sup>.

D'autres forces du type *informative* peuvent être utilisées. Elles expriment le contexte d'émission de l'interaction : *warning* lors d'une interaction contenant des informations importantes par exemple.

	Force	Lien d'observation	Lien de contrôle Contrôle croissant →		
		Informative	Non Directive	Directive	
<b>Nature</b>					
<b>but</b>		échange but	décision non directive	décision directive	contrôle croissant
<b>plan</b>		échange plan	adaptation non directive	adaptation directive	↓
<b>action</b>		échange action	commande non directive	commande directive	↑ information
<b>hypothèse</b>		échange hypothèse	hypothèse non directive	hypothèse directive	croissante

TAB. 5.1 - **Degrés d'un lien d'interaction.** Liens de contrôle et d'observation. Entre les deux bornes de forces non directive et directives s'intercalent d'autres forces propres au contrôle que l'on veut instituer dans l'application. Entre les deux bornes de forces informative et non-directive s'intercalent d'autres forces relatives au degré d'information.

## Liens d'observation

Un lien d'observation est un lien tel que la *force* est un élément de l'intervalle [*informative..non-directive*].

### Degrés d'information d'un lien d'observation

Selon la *nature* du lien d'observation, nous proposons un classement de ces informations selon la relation "*plus informative que*", notée  $>$  :

***but > plan > action > hypothèse***

## Lien de contrôle

Un lien de contrôle est un lien d'interaction tel que sa *force* est comprise dans l'intervalle [*non-directive..directive*]

<sup>5</sup>Ce type d'échange est présent dans la plupart des systèmes. Il est appelé *partage de résultats* (la nature est une hypothèse). Le système *DVMT* a aussi mis en œuvre des échanges de plans partiels pour faciliter la coordination des agents [Dur 87b] [Dur 87a].

Un agent recevant une interaction empruntant un lien de contrôle insère le contenu de celle-ci dans l'un des états *RS.évalué*, *DS<sub>x</sub>.possibilités* ou *DS<sub>x</sub>.choix* sans que les fonctions correspondant à leur gestion ne soient activées (cf. figure 5.7 où de tels liens apparaissent par une flèche accentuée)

### Degrés de contrôle d'un lien de contrôle

Les couches de contrôle sont organisées selon une hiérarchie qui exprime une dimension d'importance décroissante de la commande qu'elles produisent. Ainsi nous avons, en notant  $\succ$  la relation "est plus critique".

***hypothèse*  $\succ$  *action*  $\succ$  *plan*  $\succ$  *but***

A notre sens, l'insertion d'une hypothèse est plus critique dans la mesure où le mécanisme d'exécution de l'agent est complètement court-circuité : son état de *raisonnement* est changé sans en être la cause.

Dans la table 5.1, nous retrouvons la classification proposée dans [Les 81] avec les notions de *externally directed* correspondant à une force ***directive*** et *autonomous* pour une force ***informative***. Nous avons ajouté le degré de contrôle par ***hypothèse*** que l'on peut inscrire explicitement dans l'état de raisonnement *RS.évalué* sans qu'une détection d'incohérence ne soit effectuée. Cette notion n'est pas explicitée dans la plupart des systèmes. Son introduction accroît ainsi la flexibilité des échanges en permettant, entre autres, de faire la distinction entre certitude d'une hypothèse et volonté de l'agent qui a émis l'hypothèse. Dans la plupart des systèmes, une hypothèse très sûre de l'avis de son émetteur est incorporée par le récepteur de l'interaction. La *force* de l'interaction introduit des nuances de contrôle possibles entre les agents.

### Contrôle centralisé/ Contrôle décentralisé

Dans ce qui précède, nous avons essentiellement considéré deux agents. En élargissant les liens à plusieurs agents, nous pouvons ainsi caractériser l'*étendue de contrôle* qu'un agent peut exercer dans la société, à partir du cardinal de l'ensemble d'agents qu'il contrôle  $\mathcal{R}_{agent}$ . Cette notion se combine avec les degrés de contrôle et donne ainsi plusieurs types de contrôle dans la société qui enrichissent les notions classiques extrêmes de *contrôle centralisé* et de *contrôle décentralisé*.

*Relation de contrôle entre deux agents*

Un agent X *contrôle* un agent Y s'il existe un *lien de contrôle* dont X est la source et Y la cible, pour lequel la *nature* et la *force* sont telles que :

$$\begin{aligned} \textit{nature} &\in \{\mathbf{hypothèse}, \mathbf{action}, \mathbf{plan}, \mathbf{but}\} \\ \textit{et force} &\in [\mathbf{non-directive..directive}] \end{aligned}$$

Nous distinguons ainsi l'ensemble  $\mathcal{R}_X$ , ensemble des agents contrôlés par X.

$$\begin{aligned} \mathcal{R}_X &= \{Y \in \mathcal{A} : X \textit{ contrôle } Y\} \\ &\mathcal{A} \textit{ est l'ensemble des agents.} \end{aligned}$$

NB : Nous dirons qu'il existe un contrôle *local* pour l'agent X si  $X \in \mathcal{R}_X$ .

– *Contrôle centralisé.*

Il s'exprime en instituant des interactions de contrôle directives à partir d'un agent en direction de tous les autres agents de la société.

$$\exists X \in \mathcal{A} : \mathcal{R}_X = \mathcal{A} \text{ et } \forall Y \in \mathcal{A} - \{X\} : \mathcal{R}_Y = \emptyset$$

– *Contrôle décentralisé.*

Un contrôle décentralisé est un contrôle dans lequel il n'existe pas d'agent contrôlant tous les autres.

$$\forall X \in \mathcal{A} : \mathcal{R}_X \neq \mathcal{A}$$

– *Contrôle hiérarchique.*

Un contrôle hiérarchique est un contrôle décentralisé particulier dans lequel la relation de contrôle est une relation d'ordre.

– *Contrôle hétérarchique.*

Aucune relation de contrôle n'existe entre les agents. Ainsi :

$$\forall X \in \mathcal{A} : \mathcal{R}_X = \{X\}$$

## b. Rôles de résolution

Le *rôle de résolution* définit les interactions possibles entre les agents de la société pour mettre en œuvre l'activité de résolution. Il restreint au sein de l'agent l'utilisation des fonctions d'*évaluation<sub>x</sub>*, de *raisonnement<sub>x</sub>*, de *décision<sub>x</sub>* du modèle d'agent individuel et active les fonctions de *dialogue*. Il est défini par un *lien d'interaction* défini par la *nature*, la *force* et exprimé par rapport à un *sujet* d'interaction dont l'expression dépend des échanges se déroulant dans le système. Son expression est indépendante des agents existant dans le système.

$role \in CS_c.soc[\mathbf{résolution}] \quad role ::= \langle \mathbf{nature}, \mathbf{force} \rangle \langle \mathbf{sujet} \rangle$
--

L'état d'engagement social  $CS_c.soc[\mathbf{résolution}]$  représente les rôles de résolution de l'agent. Des critères de *validité* sont aussi représentés dans l'état d'engagement social  $CS_c.soc$  afin de permettre la remise en cause de chacun de ces rôles et de réactiver la couche *adaptation*.

Un agent peut être la source ou la cible d'un lien d'interaction. Nous distinguons les rôles de résolution suivants :

- $CS_c.soc[\mathbf{résolution}][\mathbf{social}]$  : expression des liens d'interaction que l'agent peut recevoir. Le descripteur *social* marque la source du lien : des autres agents (société) vers l'agent (individu)<sup>6</sup>.

---

<sup>6</sup>Ce rôle est exprimé par exemple dans le *DVMT* par les *aires d'intérêt*, expression de la région de

- $CS_c.soc[**résolution**][**individu**]$  : expression des liens d'interaction pour lesquels l'agent doit être la source. Le descripteur **individu** marque l'origine du lien : agent (individu) vers les autres agents (société).

Ces rôles explicitent les traitements que l'agent ne peut pas mener localement. Il est contraint d'interagir avec les autres agents. Nous avons fait ce choix pour restreindre explicitement les capacités de raisonnement et de décision de l'agent en précisant les couches de contrôle qu'il peut utiliser.

Nous n'avons pas exprimé les relations de contrôle que l'agent peut mettre en œuvre avec les autres agents : il peut a priori émettre l'ensemble des interactions qu'il désire. L'interdiction d'interaction de cet agent avec un autre agent est mise en œuvre par ce dernier au travers des rôles représentés dans l'état d'engagement social  $CS_c.soc[**résolution**][**social**]$  : liens d'interaction qu'il peut recevoir.

Si le rôle de résolution individu n'exprime pas de restriction, un agent peut exécuter localement toute fonction de toute couche de contrôle pour toute connaissance. L'expression des rôles permet de déterminer les sujets (connaissances et échanges) pour lesquels l'agent n'est contrôlé par aucun autre agent (utilisation de toute fonction et de toute couche de contrôle possible). Il exerce ainsi un contrôle local pour ce sujet.

A partir de la définition de ces deux rôles, nous pouvons remarquer qu'une interaction entre deux agents  $X$  et  $Y$  ne peut s'établir que si les rôles appartenant à  $CS_c.soc[**résolution**][**social**]$  du récepteur  $Y$  acceptent l'interaction et si les rôles  $CS_c.soc[**résolution**][**individu**]$  de l'émetteur  $X$  provoquent l'émission de l'interaction en direction de  $Y$ .

### c. Bascules sociales

Nous définissons les mécanismes de  $bascule_{social}$  et  $bascule_{individu}$  pour interpréter les rôles de résolution et contraindre le fonctionnement de l'agent. Elles utilisent respectivement les rôles représentés dans les états  $CS_c.soc[**résolution**][**social**]$  et  $CS_c.soc[**résolution**][**individu**]$ .

#### bascule social

La bascule de la société vers l'individu,  $bascule_{social}$ , (cf. figure 5.8) interprète les rôles représentés dans  $CS_c.soc[**résolution**][**social**]$  à la sortie de la fonction d'*interprétation-dialogue*.

- Si son résultat est 1, le modèle de contrôle individuel est activé : représentation du contenu de l'interaction dans l'un des états de l'agent,  $RS.non-évalué$ ,  $RS.évalué$ ,  $DS_x.possibilités$ ,  $DS_x.choix$  selon la *nature* et la *force* exprimée par l'interaction.

---

l'environnement couverte par ses capteurs. L'organisation est peu développée : hétérarchie. La notion de filtres dans *PACO* [Dem 91a] est un exemple exprimant une relation indirecte de l'organisation au travers de l'ouverture de l'agent à l'environnement. Les relations entre les agents sont la conséquence de cette ouverture. C'est ce que l'on appelle phénomène d'*émergence* dans les sociétés d'agents réactifs.

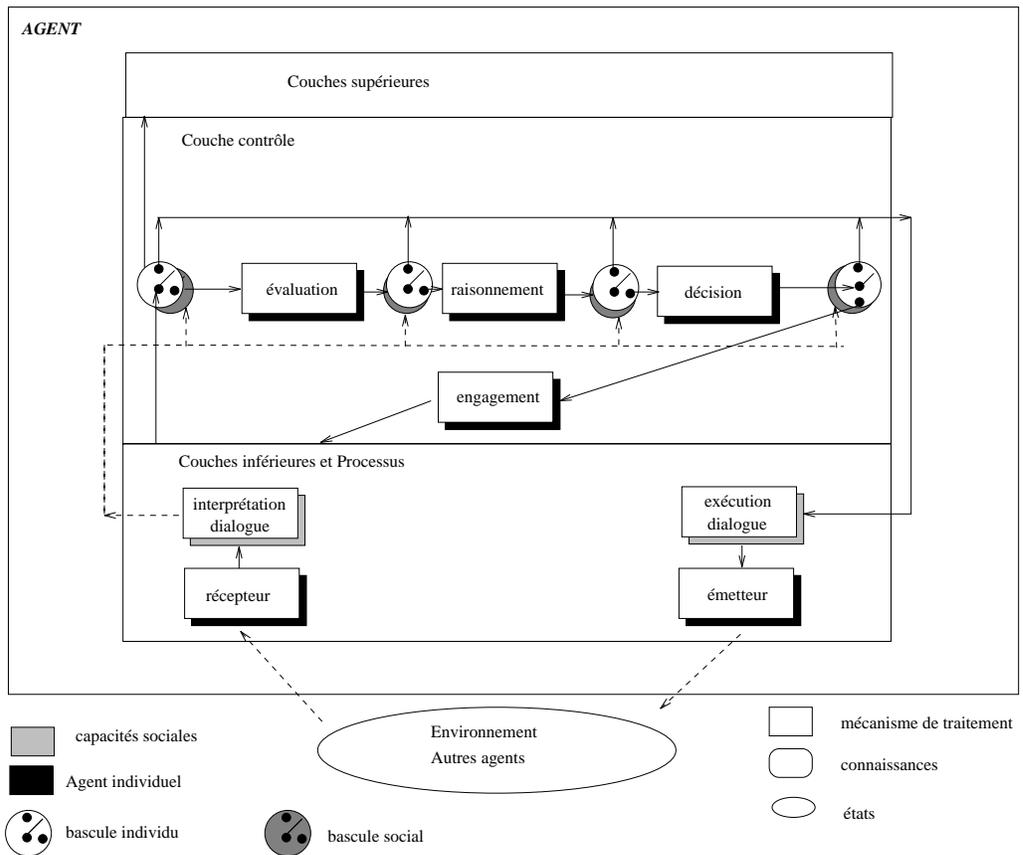


FIG. 5.8 - **Basculés sociaux.** En grisé sont représentées les bascules société (autres agents) vers individu (agent) : activation du module de résolution et représentation des informations en provenance des autres agents dans l'un des états *RS.évalué*, *RS.non-évalué* ou *DS<sub>x</sub>* de l'agent. En blanc, sont représentées les bascules individu (agent) vers société (autres agents) qui activent les fonctions de *dialogue*. Pour raison de clarté, nous n'avons pas fait apparaître les bascules inter-couches de contrôle sur le schéma.

- Si son résultat est nul, il n'est pas activé : l'interaction est rejetée. Le rôle de résolution empêche un tel lien d'interaction pour ce sujet d'interaction.

$$\boxed{\begin{aligned} \text{bascule}_{\text{social}} &: \text{interprétation-dialogue}(PS[\mathbf{message}]) \times \\ &CS_c.\text{soc}[\mathbf{résolution}][\mathbf{social}](t) \rightarrow \{0, 1\} \end{aligned}}$$

### bascule individu

La fonction  $\text{bascule}_{\text{individu}}$ , (cf. figure 5.8) s'intercale entre chacune des fonctions d'une couche de contrôle du modèle de contrôle individuel. Elle interprète les rôles représentés dans  $CS_c.\text{soc}[\mathbf{résolution}][\mathbf{individu}]$ .

- Si la fonction de bascule est vérifiée, la fonction *exécution-dialogue* est activée. Une interaction est créée en direction d'un autre agent. Le contenu de cette interaction est constitué de l'élément ayant satisfait la fonction de bascule.
- Sinon le traitement se poursuit au sein du modèle de contrôle individuel.

$$\boxed{\begin{aligned} \text{bascule}_{\text{individu}} &: (RS.\text{non-évalué}(t) \cup RS.\text{évalué}(t) \cup \\ &DS.\text{possibilités}(t) \cup DS.\text{choix}(t)) \times \\ &CS_c.\text{soc}[\mathbf{résolution}][\mathbf{individu}](t) \rightarrow \{0, 1\} \end{aligned}}$$

La fonction  $\text{bascule}_{\text{individu}}$  contraint l'agent à générer une interaction en direction d'un autre agent en fonction du lien existant avec celui-ci, exprimé dans le rôle. Des stratégies propres à l'agent permettent d'activer sa partie dialogue en créant des *intentions* relatives à des *actions* de type *communication*.

Ces deux fonctions permettent de faire la balance entre les exigences propres de l'agent et celles des autres agents au sujet de la résolution se déroulant dans la société. Elles sont dirigées par deux types de critères : extrinsèques exprimées par le rôle de *résolution*, intrinsèques formulées par une *rationalité* qui lui enjoint soit de refuser tout ce qui vient de l'extérieur, soit d'accepter tout ce qui est demandé par l'extérieur<sup>7</sup>.

### d. Contrôle du dialogue

Les fonctions d'*interprétation-dialogue* et d'*exécution-dialogue* traitent les *messages* échangés entre les agents exprimés dans le langage d'interaction  $L$ . Leur fonctionnement est contrôlé par les *protocoles d'interaction* et les *rôles de dialogue*. Les *protocoles d'interaction* définissent l'enchaînement des interactions. Les rôles de dialogue représentés dans  $CS_a.\text{soc}[\mathbf{dialogue}]$  ou  $CS_c.\text{soc}[\mathbf{dialogue}]$  définissent les protocoles d'interaction que l'agent peut utiliser.

<sup>7</sup>On retrouve par exemple la notion de *volontariat* (benevolent agent) longtemps utilisée en IAD.

## Langage d'interaction

Dans cette étude, nous nous plaçons dans le cadre de l'utilisation d'un répertoire d'*actes de communication*. Selon les auteurs, ces actes de communication sont plus ou moins complexes [Ros 82] [Sia 91] [Kre 91]. Certains utilisent les forces d'*élocutions* mises en évidence dans les *Speech Acts* [Sea 69] [Coh 79]. On les trouve dans [Cam 90] [Cha 91] [Ber 92] [Sin 91b].

Le langage d'interaction  $L$  est constitué d'un langage de *communication*  $LC$  et d'un langage de *représentation des connaissances*  $LKR$ .

$$L = LKR \cup LC$$

### *Langage de communication*

Le *langage de communication*  $LC$  est constitué des termes utilisés pour la définition de la *destination*, du *mode de communication*, du *type*, du *lien* d'une interaction. L'ensemble de ces informations est utilisé : par la fonction *émetteur* pour transmettre le message; par la fonction *bascule<sub>social</sub>* pour accepter ou non l'interaction; par la fonction *interprétation-dialogue* pour traduire le contenu de l'interaction.

#### – *destination*

La destination peut être *ciblée* ou *diffusion*<sup>8</sup>. En mode d'émission ciblé, l'interaction contient explicitement le nom de l'agent. Celui-ci est utilisé par le mécanisme de routage pour communiquer l'interaction. C'est par exemple, le mode de communication utilisé dans les langages acteurs.

#### – *mode de communication*

Le mode de communication peut être *synchrone* ou *asynchrone*.

#### – *type de l'interaction*

Le type d'une interaction exprime trois interactions primitives de base : ***request***, ***answer***, ***inform***.

***inform*** : envoi d'informations produites par l'agent vers un autre agent. Aucune réponse n'est attendue.

***request*** : demande d'informations avec attente de réponse.

***answer*** : envoi d'informations en réponse à une requête

Ces deux dernières primitives correspondent à l'expression d'un protocole de communication restreint : une requête est suivie d'une réponse. Ces primitives sont à la base de la définition des protocoles.

---

<sup>8</sup>Broadcasting

– *lien d'interaction*

Le lien exprime la *nature* et la *force* du lien d'interaction que l'interaction établit. Nous les exprimons au sein de l'interaction pour que l'agent récepteur connaisse explicitement la relation que veut établir l'émetteur avec lui.

– *nature.*

La nature de l'interaction désigne les *hypothèses*, les *buts*, les *plans* ou les *intentions*.

– *force.*

La force est exprimée par une force d'élocution [Sea 69], comprise dans l'intervalle [*informative..directive*].

$$force \in [informative..non-directive] \cup [non-directive..directive]$$

Nous ne définissons pas complètement les forces d'élocution dans la mesure où elles dépendent des besoins de l'application. Elles seront ainsi affinées en fonction des nuances nécessaires dans les interactions entre les agents.

***Langage de représentation des connaissances***

Le *Langage de représentation des connaissances (LKR)* définit le contenu propositionnel de l'interaction. Il est dépendant de l'application. Il intervient dans l'expression du *sujet* d'interaction.

**Protocoles d'interaction**

Les protocoles d'interaction sont distingués en deux sous-ensembles : protocoles de résolution et protocoles d'organisation selon qu'ils définissent les enchaînements d'interaction respectivement pour les échanges de résolution entre les agents, ou pour les échanges conduisant à la définition de l'organisation entre les agents.

$$Pr = Pr[résolution] \cup Pr[organisation]$$

Un protocole d'interaction définit l'enchaînement des interactions entre les agents de la société pour un type de résolution. Il peut être exprimé par un réseau de transitions augmentées (ATN) [Bou 92] dans lequel :

- les états sont les *états de conversation*, configuration de l'échange entre les agents à la réception ou à l'émission d'une interaction.
- les transitions sont la spécification des interactions permises pour changer d'état. Ainsi la spécification de la transition fait référence à l'interaction dont les champs *destination*, *mode de communication*, *type nature*, *force* et *contenu* sont complètement définis ou non.
- les conditions sont des conditions sur les changements d'états portant sur l'état de conversation.

Le choix entre plusieurs transitions fait l'objet d'une sélection de la part de l'agent.

## Rôle de dialogue

Nous distinguons deux types de rôle de dialogue selon les contraintes qu'ils expriment : rôle de dialogue pour l'organisation et rôle de dialogue pour la résolution.

- Les rôles de dialogue représentés dans  $CS_a.soc[\mathbf{dialogue}]$  définissent les interactions et les protocoles de  $Pr[\mathbf{organisation}]$  utilisables par l'agent lors de la définition de l'organisation par interaction avec les autres agents.
- Les rôles de dialogue représentés dans  $CS_c.soc[\mathbf{dialogue}]$  en fonction des sujets d'interaction définissent les protocoles de  $Pr[\mathbf{résolution}]$  utilisables par l'agent lors de la résolution.

$$\begin{aligned} role \in CS_c.soc[\mathbf{dialogue}] \quad role = \text{protocole sujet} \quad \text{où } \text{protocole} \in Pr[\mathbf{résolution}] \\ role \in CS_a.soc[\mathbf{dialogue}] \quad role = \text{protocole sujet} \quad \text{où } \text{protocole} \in Pr[\mathbf{organisation}] \end{aligned}$$

*sujet* est comme dans le cas de la définition du rôle de résolution propre à l'application.

## e. Conclusion

Le modèle de contrôle social, dont nous n'avons explicité que la couche *commande* peut être ainsi résumé par la formulation suivante :

$$\begin{aligned} agent &= \langle Agent_{ind}, L, Pr, Org, CS.soc, PM \rangle \\ PM &= \langle organisation, dialogue, bascule_{social}, bascule_{individu} \rangle \end{aligned}$$

## 5.3 Modèles de Contrôle Individuel et Social d'un Système Intégré de Vision

Pour la suite de notre exposé, nous nous plaçons dans le cadre d'une société constituée d'*agents de base*<sup>9</sup> d'un Système Intégré de Vision bâtis sur les modèles de contrôle individuel et social précédents. Un agent est donc l'ensemble des traitements et des informations relatifs à un *niveau de représentation* et à une *focalisation* donnée.

- *niveaux de représentation*

Un niveau de représentation, noté  $l$ , est défini par des *contraintes d'abstraction* et des *contraintes de décentration* [Dem 86] appliquées aux différentes informations manipulées dans le système.

---

<sup>9</sup>Le passage d'un agent-base à un agent-niveau ou à un agent-focus consiste à reformuler les protocoles et organisations du modèle de société dans les connaissances, représentations et traitements du modèle de contrôle individuel. Par exemple les protocoles d'interactions entre des agents de base ayant une même focalisation se reformulent en terme de stratégies d'interprétation au sein de l'agent focus correspondant.

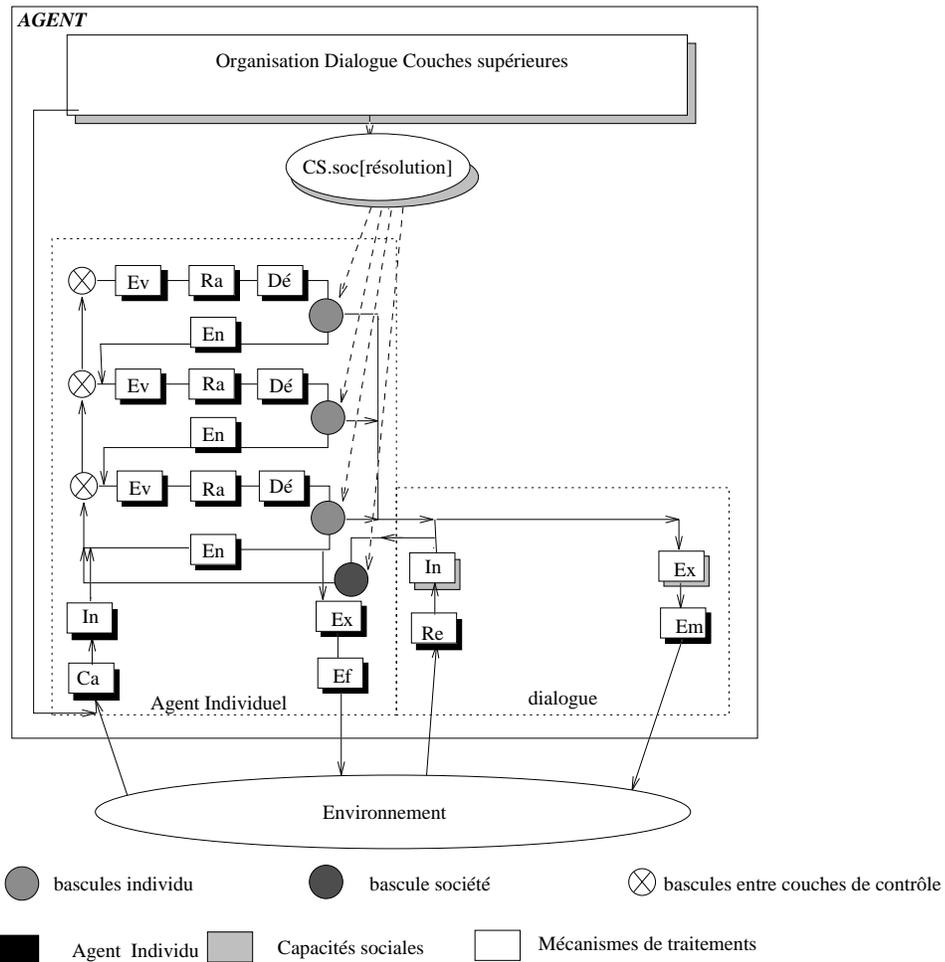


FIG. 5.9 - **Modèle opératoire de contrôle social et individuel.** *Ev*: évaluation, *Ra*: raisonnement, *Dé*: décision, *En*: engagement, *In*: interprétation, *Ca*: capteur, *Em*: émetteur, *Re*: récepteur, *Ef*: effecteur, *Ex*: exécution.

– *focalisations*

Une *focalisation* notée  $f$  est le centre d'intérêt privilégié lors de la conception de l'agent ou en cours d'exécution. Cette focalisation est commune à un ensemble d'agents. Les focalisations possibles font référence aux centres d'intérêt que nous avons mis en évidence dans le chapitre 3 :

- centre d'intérêt sémantique : nous distinguons un centre d'intérêt *objet* exprimé en terme de modèles, en types d'indice (couleur, texture, mouvement, etc), et un centre d'intérêt *tâche* : suivre un objet, etc.
- centre d'intérêt spatial : nous le désignons par *région* exprimant les dimensions spatiales où effectuer les traitements.
- centre d'intérêt temporel : nous le désignons par *durée* expression des dimensions temporelles des traitements.

Ainsi, par exemple, un agent sera constitué des traitements, de représentation et des connaissances relatifs à la texture sur le niveau d'indices d'images. Un autre agent sera, par contre, constitué des traitements, représentations et connaissances relatifs aux reflets sur le niveau indices de scènes.

- Ce cadre d'étude est choisi du fait des exigences induites par le projet de recherches *SATURNE*. Dans celui-ci, en effet, le système de vision est la composition de tels agents : sur un niveau de représentation, chaque agent est relatif à une méthode d'inférence de formes particulière (focalisation).
- Du fait de notre cadre d'expérimentation issu de *VAP*, nous nous intéressons principalement aux échanges inter-niveaux de représentation entre plusieurs agents dans la société, pour une même focalisation. Nous présenterons l'expression de ces échanges dans la section consacrée au *modèle de société*.

Afin d'instancier le modèle d'agent précédent, nous nous plaçons dans le cadre suivant :

– *Modèle*

Soient  $M$  l'ensemble des *modèles* qu'utilise le système. Les termes utilisés pour la définition des modèles participent à la définition des langages de représentation et d'interaction  $L$  entre les agents.

*Modèle, indices visuels*

Un *modèle*  $m_j$ , peut être défini comme un ensemble d'*indices visuels*  $iv_{j,i}$ , de *contraintes absolues*,  $Tabs_{j,k}$  et de *contraintes relationnelles*,  $Trel_{j,n}$ .

$$m_j = \{iv_{j,i}, Tabs_{j,k}, Trel_{j,n}, i, k, n \in N\}$$

Les contraintes *absolues*,  $Tabs_{j,k}$ , expriment des conditions portant sur l'indice visuel lui-même telles par exemple valeur, orientation. Les contraintes *relationnelles*,  $Trel_{j,n}$ , expriment des relations géométriques, physiques ou spatiales entre

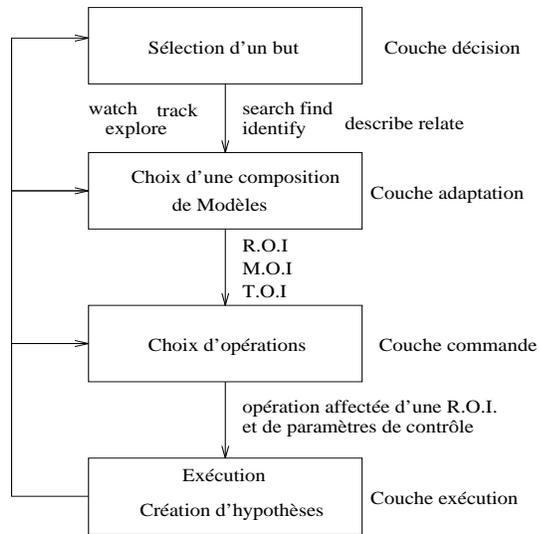


FIG. 5.10 - Modèle de contrôle individuel pour un système intégré de vision.

les indices visuels. Ainsi un modèle de *carré* peut être exprimé en terme d'indices visuels de type *segment* qui vérifient des contraintes relationnelles de parallélisme, de jonction et les contraintes absolues du type couleur rouge, longueur d'un côté est égale à 10 cm.

Les modèles peuvent être organisés selon une hiérarchie de décomposition dont les feuilles font directement référence aux informations fournies par l'agent de niveau inférieur. Ainsi, une hypothèse de présence dans la scène du modèle de carré sera générée dans le cas où les segments (indices visuels) qui le décrivent seront présents dans l'image issue du niveau de représentation inférieur (observations). Nous avons ainsi une mise en correspondance entre les observations de l'image et des indices visuels du modèle de carré.

#### – *description*

Nous appelons  $D$ , descriptions, l'ensemble des informations qui constituent la description de la scène dans le système. Les descriptions expriment une relation entre les modèles et l'environnement perçu.

### 5.3.1 Modèle de contrôle individuel d'un système intégré de Vision

Définissant le modèle d'agent comme étant un agent de base, nous considérons ici descriptions, modèles et traitements relatifs à un niveau de représentation  $l$  et une focalisation  $f$ .

Dans le cadre de ce modèle, nous restreignons le modèle d'agent au niveau des capteurs et effecteurs : seuls les *récepteurs* et les *émetteurs* sont présents. De la même

manière, nous supprimons les fonctions d'*exécution* et d'*interprétation* dans la mesure où dans le cadre ainsi défini, nous n'avons que des communications entre agents.

### a. Représentation des connaissances

#### Connaissances individuelles $KR.ind$

##### Fait ( $KR.ind[**fait**]$ )

Les connaissances factuelles sont relatives au niveau de représentation  $l$  et à la focalisation  $f$  propres à l'agent :

$$KR.ind[**fait**] = M^{l,f}$$

Les modèles expriment les connaissances utilisées a priori par les *opérations* pour produire de nouvelles *hypothèses* à partir des *données* ou d'*hypothèses* existantes.

##### *remarque :*

La distinction selon la focalisation intervient différemment selon la définition de  $f$ . Par exemple, pour  $f$  équivalent à une *région* spatiale, les agents peuvent, a priori, avoir les mêmes modèles dans la mesure où seules les informations de localisation peuvent varier dans les agents. Une focalisation  $f$  exprimée en terme d'*indices visuels* définit les agents comme des parties d'une méthode d'inférence de forme. Chaque agent a ainsi des modèles relatifs à cet indice.

##### Compétences ( $KR.ind[**comp**]$ )

Les compétences correspondent aux *opérations* *intra-niveaux* de représentation pour le niveau  $l$ . Elles sont distinguées en *opérations* de *fusion* et en *opérations* d'*enrichissement* relatives à la focalisation  $f$ .

$$KR.ind[**comp**][**opération**] = KR.ind[**comp**][**fusion**] \cup KR.ind[**comp**][**enrichissement**]$$

– *Opération d'enrichissement* ( $KR.ind[**comp**][**enrichissement**]$ )

Elle construit de nouvelles hypothèses à partir des hypothèses existantes.

*Ces opérations mettent en œuvre par exemple un parcours ascendant ou descendant de la hiérarchie des modèles et des liens exprimés entre les modèles pour affiner la description. Elles utilisent des connaissances sur les indices qui sont intéressants à chercher pour conclure sur des hypothèses.*

– *Opération de fusion* ( $KR.ind[**comp**][**fusion**]$ )

Elle assure la résolution locale des conflits détectés par la fonction d'*évaluation*.

*Ces opérations peuvent par exemple, rechercher les hypothèses relatives à un même modèle et n'ayant pas les mêmes associations (indices visuels, observations). Ces hypothèses sont combinées en une seule en réunissant les indices visuels. La fusion pourra porter par exemple sur deux hypothèses relatives au modèle de carré décrit par des jonctions d'une part, et par des parallèles d'autre part. Ce pourra être aussi des hypothèses relatives à un même modèle avec des indices couleur et texture.*

### Tactiques ( $KR.ind[tact]$ )

Les tactiques sont distinguées en tactiques permettant de réaliser les différentes tâches mises en évidence dans le chapitre 3 auxquelles est ajoutée une dimension temporelle.

$$KR.ind[tact] = KR.ind[tact][identification] \cup KR.ind[tact][localisation] \cup \\ KR.ind[tact][vérification] \cup KR.ind[tact][description] \cup \\ KR.ind[tact][surveillance]$$

Ces dimensions séparent les connaissances prenant part à la génération des critères de contrôle de l'agent. Les tactiques concernent uniquement la définition des critères de contrôle différents de  $f$  et relatifs aux éléments manipulés dans  $l$ .

### Stratégies ( $KR.ind[strat]$ )

Les stratégies sont distinguées à l'aide des différentes tâches mises en évidence dans le chapitre 3 auxquelles est ajoutée une dimension temporelle.

$$KR.ind[strat] = KR.ind[strat][identification] \cup KR.ind[strat][localisation] \cup \\ KR.ind[strat][vérification] \cup KR.ind[strat][description] \cup \\ KR.ind[strat][surveillance]$$

Ces stratégies concernent uniquement la définition des critères de contrôle différents de  $f$  et relatifs aux éléments manipulés dans  $l$ .

### Connaissances sociales ( $KR.soc$ )

Les connaissances sociales sont les connaissances relatives une autre focalisation et à un autre niveau de représentation que ceux concernés par l'agent.

*Ainsi, par exemple, les connaissances permettant d'obtenir la forme à partir de la texture sont des connaissances sociales pour l'agent travaillant sur les reflets.*

Les **faits** ( $KR.soc[fait]$ ), les **compétences** ( $KR.soc[comp]$ ), les **tactiques** et **stratégies** ( $KR.soc[tact]$  et  $KR.soc[strat]$ ) sont du même type que leurs analogues de  $KR.ind$  tout en étant relatives à une autre focalisation et à un autre niveau de représentation.

**b. Etat de perception (PS)**

L'état de perception est constitué des descriptions communiquées entre les agents.

$$PS = D^{l,f}$$

*Dans l'exemple précédent, les segments extraits de l'image (niveau de représentation  $l-1$ ) sont traduits par l'agent construisant la description de la scène en terme de modèles de carrés (niveau de représentation  $l$ ).*

**c. Etat de raisonnement (RS)****Hypothèses (RS[hypothèse])**

Les **hypothèses** constituent la description de la scène construite par l'agent. Elles sont organisées selon les dimensions de *région*, d'*objet*, de *durée* que nous avons définies ci-dessus. Celles-ci deviennent les **données** pour l'agent de niveau de représentation supérieur.

Une **hypothèse** peut être exprimée par une *association*,  $\mathcal{A}^{l,f}$ , d'un modèle  $j$   $m_j^{l,f}$  et de données  $k$   $o_k^{l,f}$ , satisfaisant une ou plusieurs contraintes du modèle  $Trel_{j,k}^{l,f}$  ou  $Tabs_{j,k}^{l,f}$ .

$$\mathbf{hypoth\grave{e}se} = (\mathcal{A}^{l,f}) \quad \mathcal{A}^{l,f} = \{(iv_{j,i}^{l,f}, o_k^{l,f}) : Trel_{j,m}^{l,f} \vee Tabs_{j,m}^{l,f}\}$$

Un *coefficient de vraisemblance* exprimant la confiance ou la plausibilité de l'hypothèse peut être ajouté à cette association afin d'exprimer l'incertitude inhérente à tout système de vision. Cette valeur fournit une mesure pour comparer les hypothèses entre elles qui peut être utilisée par exemple par les critères de **sélection**, d'**activation**, d'**incohérence** ou de **validité** qu'expriment les *engagements* des couches adaptation et décision.

**Buts (RS[but])**

Un **but** est la combinaison de tâches avec, selon celles-ci, la spécification de centres d'intérêt de type *objet*, *région* ou *durée*. A partir de notre étude du chapitre 3, nous distinguons les buts d'*identification*, de *localisation*, de *vérification* (exprimant la combinaison des buts d'identification et de localisation), de *description*, (affinant une description de la scène), de *surveillance* (faisant remplir à un agent la fonction de surveillance).

$$RS[\mathbf{but}] = RS[\mathbf{but}][\mathbf{identification}] \cup RS[\mathbf{but}][\mathbf{localisation}] \cup RS[\mathbf{but}][\mathbf{v\acute{e}rification}] \cup RS[\mathbf{but}][\mathbf{description}] \cup RS[\mathbf{but}][\mathbf{surveillance}]$$

– *buts de localisation* (RS[**but**][**localisation**])

Ces buts s'expriment comme suit : étant donné un *modèle*,  $m_j^{l,f}$ , trouver la *région*

dans laquelle se trouve l'hypothèse, association de  $m_j^{l,f}$  avec des hypothèses. Son expression détermine ainsi une caractéristique permettant de sélectionner directement un modèle.

$$RS[\mathbf{but}][\mathbf{localisation}] = \{find\ objet, track\ objet\}$$

où *objet* fait référence à une spécification de modèle.

**find** : Déterminer la région dans laquelle se trouve l'hypothèse relative à un modèle  $m_j^{l,f}$ .

**track** : Suivre le modèle  $m_i^{l,f}$  dans la scène. Il est la reformulation temporelle du but précédent. Il implique la modification dynamique de la région dans laquelle une hypothèse relative au modèle a été trouvée à un instant donné en se fondant sur une prédiction de la nouvelle position du modèle trouvé à l'instant précédent.

– *buts d'identification* ( $RS[\mathbf{but}][\mathbf{identification}]$ )

Étant donné une région de la scène, trouver le ou les modèles  $m_j^{l,f}$  pouvant être associé à l'ensemble d'observations issues de cette région.

$$RS[\mathbf{but}][\mathbf{identification}] = \{identify\ région, explore\ région\}$$

où *région* désigne la spécification d'une région spatiale de la scène.

**identify** : Rechercher le modèle correspondant aux observations perçues dans la *région*.

**explore** : Décrire l'évolution temporelle de la région. Un tel but implique une succession d'*identification*.

– *but de vérification* ( $RS[\mathbf{but}][\mathbf{vérification}]$ )

Étant donné un modèle,  $m_j^{l,f}$ , et une région, vérifier l'existence de l'hypothèse, association du modèle  $m_i^{l,f}$  avec les données présentes dans la *région*.

$$RS[\mathbf{but}][\mathbf{vérification}] = \{verify\ objet\ région\}$$

où *objet* est la spécification d'un modèle, *région* la spécification d'une région spatiale de la scène.

– *buts de surveillance* ( $RS[\mathbf{but}][\mathbf{surveillance}]$ )

Mise en place d'un fonctionnement répétitif de surveillance d'un modèle ou d'une région. Nous introduisons les buts :

$$RS[\mathbf{but}][\mathbf{surveillance}] = \{watch\ objet\ durée, watch\ région\ durée\}$$

où *objet* est la spécification d'un modèle, *région* la spécification d'une région spatiale de la scène, *durée* est la spécification d'une durée.

**watch** : Fixer les traitements sur les *événements* persistants, ou au contraire éphémères. La notion d'événement est importante. Un événement désigne toute apparition d'hypothèse ou d'ensemble d'hypothèses ayant une configuration particulière. Leur expression intervient dans la définition des bascules au sein de chacun des agents permettant de remettre en cause des critères de contrôle.

– *buts de description* ( $RS[**but**][**description**]$ )

Trouver les valeurs des *attributs* d'un modèle associé à une hypothèse ainsi que les relations de celle-ci avec d'autres hypothèses relatives à la présence d'autres modèles dans la *région*. Nous introduisons les mots clefs :

$$RS[**but**][**description**] = \{describe\ objet\ région, relate\ objet\ région\}$$

**describe** : Décrire l'hypothèse telle qu'elle apparaît dans la *région*.

**relate** : Trouver les relations entre des hypothèses.

### Plans ( $RS[**plan**]$ )

Les plans se décomposent selon les dimensions exprimées au sein des buts et des stratégies. Nous distinguons ainsi les plans relatifs à l'*identification*, la *localisation*, la *vérification* la *description*, la *surveillance*.

$$RS[**plan**] = RS[**plan**][**identification**] \cup RS[**plan**][**localisation**] \cup RS[**plan**][**vérification**] \cup RS[**plan**][**description**] \cup RS[**plan**][**surveillance**]$$

#### d. Etat de décision $DS$

L'état de décision est l'ensemble des *intentions*. On peut les organiser selon les *régions*, les *objets*, les *durées* et les *tâches*.

#### e. Etat d'engagement individuel ( $CS.ind$ )

L'état d'engagement pour les couches décision, adaptation et commande organise les critères d'*incohérence*, d'*activation*, de *sélection* et de *validité*, selon les dimensions spatiales (*région*), temporelles (*durée*) sémantiques (*objet*) et de tâches (*tâche*).

$$\forall x \{a, c\}$$

$$CS_x.ind = CS_x.ind[\mathbf{TOI}] \cup CS_x.ind[\mathbf{MOI}] \cup CS_x.ind[\mathbf{ROI}] \cup CS_x.ind[\mathbf{POI}]$$

- $CS_x.ind[\mathbf{TOI}]$ : ensemble des tâches d'intérêt<sup>10</sup>, exprimant les engagements relatifs à la tâche à réaliser. Cette tâche est soit **identification**, **localisation**, **vérification**, **identification**, **description**, **surveillance**.
- $CS_x.ind[\mathbf{MOI}]$ : ensemble des modèles d'intérêt<sup>11</sup>, exprimant les engagements relatifs au modèle à rechercher.
- $CS_x.ind[\mathbf{ROI}]$ : ensemble des régions d'intérêt<sup>12</sup> exprimant les critères de contrôle relatifs à la région dans laquelle réaliser les traitements.
- $CS_x.ind[\mathbf{POI}]$ : ensemble des périodes temporelles d'intérêt<sup>13</sup> exprimant les critères de contrôle relatifs aux dimensions temporelles des traitements.

Les possibilités d'expression des  $CS_x.ind[\mathbf{ROI}]$  et  $CS_x.ind[\mathbf{POI}]$  se définissent par exemple par des critères numériques. En revanche, les expressions des  $CS_x.ind[\mathbf{MOI}]$  sont nombreuses. Cette richesse d'expression est due à la multiplicité des indices visuels de description des modèles.

Ainsi par exemple une *intention* appartenant à  $DS_d.choix$  pourra se formuler en termes d'engagements s'inscrivant dans  $CS_a.ind[\mathbf{ROI}]$ ,  $CS_a.ind[\mathbf{MOI}]$ ,  $CS_a.ind[\mathbf{POI}]$  ou  $CS_a.ind[\mathbf{TOI}]$  selon les termes dans lesquels sont exprimés le **but** auquel elle est reliée.

## f. Mécanismes de traitement

### Récepteur et émetteur

La fonction *récepteur* est distinguée en : (i) *récepteur-niveau*, réception des messages provenant des agents situés sur le même niveau de représentation, (ii) *récepteur-focus*, réception des messages provenant des agents situés dans la même focalisation. Nous réalisons la même distinction pour la fonction *émetteur*: *émetteur-niveau*, *émetteur-focus*.

$$\begin{array}{l} \text{récepteur-focus} : \Omega \rightarrow PS[\mathbf{message}] \\ \text{récepteur-niveau} : \Omega \rightarrow PS[\mathbf{message}] \\ \text{émetteur-focus} : ES[\mathbf{message}] \rightarrow \Omega \\ \text{émetteur-niveau} : ES[\mathbf{message}] \rightarrow \Omega \end{array}$$

- les échanges au sein du niveau  $l$  sont reçus par la fonction de *récepteur-niveau* et émis par la fonction d'*émetteur-niveau*.

<sup>10</sup>Task Of Interest (TOI)

<sup>11</sup>Model Of Interest (MOI).

<sup>12</sup>Region Of Interest (ROI).

<sup>13</sup>Period Of Interest (POI)

- les échanges avec les agents appartenant à la focalisation  $f$  sont reçus par la fonction de *récepteur-focus* et émis par la fonction d'*émetteur-focus*.

Cette distinction est introduite pour pouvoir privilégier tel ou tel type d'interaction. Les liens d'interaction entre les agents ne sont pas les mêmes dès que l'on communique au sein d'un même niveau de représentation, c'est à dire avec des informations que l'on n'a pas à traduire ou interpréter ou que l'on communique au sein d'un focus. De manière beaucoup plus naturelle, dans un système de vision, les communications au sein d'une focalisation sont relatives à l'échange d'informations de contrôle.

## Interprétation-dialogue

La fonction d'*interprétation-dialogue* utilise les fonctions de traduction du langage d'interaction et des opérations permettant le changement de représentations d'un niveau à l'autre (opérations de transformations).

$$\begin{aligned} \textit{interprétation-dialogue} : PS[\mathbf{message}] \times KR.soc \times L \times \\ CS_c.soc[\mathbf{dialogue}] \times Pr[\mathbf{résolution}] \rightarrow RS \cup DS \end{aligned}$$

La méthode de résolution mise en œuvre consiste souvent en une *transformation* des données (changement de repère par exemple, expression dans un vocabulaire utilisé dans les modèles) et en une *mise en correspondance* avec les modèles prédéfinis.

*Ainsi, par exemple, les opérations de **transformation** génèrent les hypothèses à partir de l'association du contenu des **messages** provenant des agents de niveau inférieur  $l-1$  avec les indices visuels de modèles du niveau  $l$  de l'agent. Les associations produites satisfont des contraintes exprimées dans le modèle.*

## Exécution-dialogue

La fonction d'*exécution-dialogue* utilise les fonctions de traduction du langage d'interaction  $L$  et des opérations de changement de représentation *inter-niveau*(opérations de projection).

$$\begin{aligned} \textit{exécution-dialogue} : DS.choix[\mathbf{action}] \times L \times CS_c.soc[\mathbf{dialogue}] \times \\ KR.soc \times Pr[\mathbf{résolution}] \rightarrow ES[\mathbf{message}] \end{aligned}$$

Elle réalise la traduction des éléments de l'agent dans les termes des modèles,  $M^{l-1,f}$  qu'utilise l'agent de niveau inférieur  $l-1$ .

## Evaluation

La fonction d'*évaluation* est constitué d'un ensemble d'opérations de *cohérence* détectant les conflits entre des informations produites localement et par d'autres agents. Elle met en œuvre la détection des cohérences spatiale, sémantique et temporelle.

Elle peut être exprimée comme la combinaison de fonctions d'évaluations relatives aux engagements de types **ROI**, **MOI**, **TOI** et **POI**.

### Raisonnement

La fonction de *raisonnement* met en œuvre une combinaison de mécanismes dédiés respectivement à chacun des engagements de type **ROI**, **MOI**, **TOI** et **POI**.

### Décision

La fonction de *décision* met en œuvre une combinaison de mécanismes dédiés respectivement à l'utilisation de chacun des engagements relatifs à **ROI**, à **MOI**, à **TOI** et à **POI**.

### Engagement

La fonction d'*engagement* met en œuvre une combinaison de mécanismes dédiés à la gestion, respectivement de chacun des engagements de type **ROI**, **MOI**, **TOI** et **POI**.

$$\text{engagement}_x : DS_x.\text{choix} \rightarrow CS_{x-1}.\text{ind}[\mathbf{ROI}] \cup CS_{x-1}.\text{ind}[\mathbf{POI}] \cup CS_{x-1}.\text{ind}[\mathbf{TOI}] \cup CS_{x-1}.\text{ind}[\mathbf{MOI}]$$

### Bascule

La fonction de *bascule* est décomposée en fonctions testant la validité des engagements représentés dans  $CS_x.\text{ind}[\mathbf{ROI}]$ ,  $CS_x.\text{ind}[\mathbf{MOI}]$ ,  $CS_x.\text{ind}[\mathbf{TOI}]$  et  $CS_x.\text{ind}[\mathbf{POI}]$ . La remise en cause d'un seul de ces types d'engagement est ainsi possible.

Les fonctions de bascule sont un outil pour exprimer les conditions de validité d'un critère de contrôle en fonction, par exemple, des régions de l'image non explorées, des régions en conflits, des hypothèses en conflits.

### g. Conclusion

Le modèle de contrôle individuel d'un système intégré de vision s'exprime ainsi comme :

$$\begin{aligned} \text{Agent} &= \langle KR, L, Pr, Org, PS, RS, DS, CS, ES, PM \rangle \\ \text{PM} &= \langle \text{récepteur, interprétation-dialogue, bascule, évaluation,} \\ &\quad \text{raisonnement, décision, engagement, exécution, exécution-dialogue,} \\ &\quad \text{émetteur} \rangle \end{aligned}$$

Nous rappelons ici, les seuls éléments qui ont été particularisés par le domaine d'application.

– Connaissances

$$\begin{aligned}
KR.ind[\mathbf{fait}] &= M^{l,f} \\
KR.ind[\mathbf{comp}][\mathbf{opération}] &= KR.ind[\mathbf{comp}][\mathbf{fusion}] \cup \\
&\quad KR.ind[\mathbf{comp}][\mathbf{enrichissement}] \\
KR.ind[\mathbf{tact}] &= KR.ind[\mathbf{tact}][\mathbf{identification}] \cup KR.ind[\mathbf{tact}][\mathbf{localisation}] \cup \\
&\quad KR.ind[\mathbf{tact}][\mathbf{vérification}] \cup KR.ind[\mathbf{tact}][\mathbf{description}] \cup \\
&\quad KR.ind[\mathbf{tact}][\mathbf{surveillance}] \\
KR.ind[\mathbf{strat}] &= KR.ind[\mathbf{strat}][\mathbf{identification}] \cup KR.ind[\mathbf{strat}][\mathbf{localisation}] \cup \\
&\quad KR.ind[\mathbf{strat}][\mathbf{vérification}] \cup KR.ind[\mathbf{strat}][\mathbf{description}] \cup \\
&\quad KR.ind[\mathbf{strat}][\mathbf{surveillance}]
\end{aligned}$$

– Etats

$$\begin{aligned}
PS &= D^{l,f} \\
RS &= RS[\mathbf{hypothèse}] \cup RS[\mathbf{conflit}] \cup RS[\mathbf{but}] \cup RS[\mathbf{plan}] \\
RS[\mathbf{but}] &= RS[\mathbf{but}][\mathbf{identification}] \cup RS[\mathbf{but}][\mathbf{localisation}] \cup \\
&\quad RS[\mathbf{but}][\mathbf{vérification}] \cup RS[\mathbf{but}][\mathbf{description}] \cup \\
&\quad RS[\mathbf{but}][\mathbf{surveillance}] \\
RS[\mathbf{plan}] &= RS[\mathbf{plan}][\mathbf{identification}] \cup RS[\mathbf{plan}][\mathbf{localisation}] \cup \\
&\quad RS[\mathbf{plan}][\mathbf{vérification}] \cup RS[\mathbf{plan}][\mathbf{description}] \cup \\
&\quad RS[\mathbf{plan}][\mathbf{surveillance}] \\
DS_c &= DS[\mathbf{action}] \cup DS[\mathbf{opération}] \\
DS_a &= DS[\mathbf{plan}] \\
DS_d &= DS[\mathbf{but}] \\
\forall x \{a, c\} \\
CS_x.ind &= CS_x.ind[\mathbf{TOI}] \cup CS_x.ind[\mathbf{MOI}] \cup CS_x.ind[\mathbf{ROI}] \cup CS_x.ind[\mathbf{POI}]
\end{aligned}$$

– Mécanismes de traitement

$$\begin{aligned}
\text{récepteur-focus} &: \Omega \rightarrow PS[\mathbf{message}] \\
\text{récepteur-niveau} &: \Omega \rightarrow PS[\mathbf{message}] \\
\text{émetteur-focus} &: ES[\mathbf{message}] \rightarrow \Omega \\
\text{émetteur-niveau} &: ES[\mathbf{message}] \rightarrow \Omega \\
\text{interprétation-dialogue} &: PS[\mathbf{message}] \times KR.soc \times L \times \\
&\quad CS_c.soc[\mathbf{dialogue}] \times Pr[\mathbf{résolution}] \rightarrow RS \cup DS \\
\text{exécution-dialogue} &: DS.choix[\mathbf{action}] \times L \times CS_c.soc[\mathbf{dialogue}] \times \\
&\quad KR.soc \times Pr[\mathbf{résolution}] \rightarrow ES[\mathbf{message}]
\end{aligned}$$

$$\begin{aligned}
& \forall x \in \{d, a, c\} \\
& \text{évaluation}_x : RS.\text{non-évalué}(t) \times RS.\text{évalué}(t-1) \\
& \quad \times CS_x.\text{ind}[\text{incohérence}](t) \rightarrow RS.\text{évalué}(t) \\
& \text{raisonnement}_x : RS.\text{évalué}(t) \times \\
& \quad KR_x.\text{ind} \times CS_x.\text{ind}[\text{activation}](t) \rightarrow DS_x.\text{possibilités}(t) \\
& \text{décision}_x : DS_x.\text{possibilités}(t) \times CS_x.\text{ind}[\text{sélection}](t) \rightarrow DS_x.\text{choix}(t) \\
& \text{bascule}_x : (RS(t) \cup DS_x(t)) \times CS_x.\text{ind}[\text{validité}](t) \rightarrow \{0, 1\} \\
& \text{engagement}_x : DS_x.\text{choix} \rightarrow CS_{x-1}.\text{ind}[\mathbf{ROI}] \cup CS_{x-1}.\text{ind}[\mathbf{POI}] \cup \\
& \quad CS_{x-1}.\text{ind}[\mathbf{TOI}] \cup CS_{x-1}.\text{ind}[\mathbf{MOI}]
\end{aligned}$$

Les fonctions d'évaluation, de raisonnement, de décision et de bascule sont distinguées par rapport à *ROI*, *TOI*, *MOI* et *POI*. Sur la couche décision, elles déterminent le **but** à résoudre, et l'expriment en engagements s'inscrivant dans  $CS_a.\text{ind}[\mathbf{ROI}]$ ,  $CS_a.\text{ind}[\mathbf{MOI}]$  et  $CS_a.\text{ind}[\mathbf{TOI}]$  et  $CS_a.\text{ind}[\mathbf{POI}]$  permettant de sélectionner et de construire les plans d'interprétations utilisés sur la couche adaptation. La couche adaptation élabore et sélectionne l'ensemble de **plans** à partir des engagements exprimés dans  $CS_a.\text{ind}[\mathbf{ROI}]$ ,  $CS_a.\text{ind}[\mathbf{MOI}]$  et  $CS_a.\text{ind}[\mathbf{TOI}]$  et  $CS_a.\text{ind}[\mathbf{POI}]$ . Elle construit les engagements pour la couche de commande et les inscrit dans les états d'engagement individuels  $CS_c.\text{ind}[\mathbf{ROI}]$ ,  $CS_c.\text{ind}[\mathbf{MOI}]$ ,  $CS_c.\text{ind}[\mathbf{TOI}]$  et  $CS_c.\text{ind}[\mathbf{POI}]$ . Ceux-ci sont utilisés pour la sélection des **opérations** d'enrichissement ou de fusion à exécuter. Les *basculés* de la couche commande vers la couche adaptation permettent de mettre en place le mécanisme de création et de tests d'hypothèses en testant la validité de la région ou du modèle sélectionné.

### 5.3.2 Modèle de contrôle social d'un Système Intégré de Vision

#### a. Langage d'interaction

La définition du langage d'interaction utilise les termes manipulés par l'ensemble de modèles du système *M* et les termes utilisés pour exprimer les buts, les plans.

#### b. Rôles de résolution

Les rôles de résolution sont relatifs au fonctionnement du système que l'on veut mettre en place. Ils sont propres aux fonctionnements de reconnaissance ou de reconstruction.

$$\begin{aligned}
Org[\mathbf{résolution}] &= Org[\mathbf{résolution}][\mathbf{reconstruction}] \cup \\
& \quad Org[\mathbf{résolution}][\mathbf{reconnaissance}]
\end{aligned}$$

Les rôles de résolution individu interdisent aux agents d'utiliser certaines couches de

leur modèle de contrôle individuel.

*Une organisation pour la reconnaissance est, par exemple, une structure hiérarchique dans laquelle les **buts** à résoudre sont définis successivement par chacun des agents, d'un niveau de représentation à l'autre, pour une même focalisation (le sujet noté  $l + 1, f$ ).*

$role \in CS_c.soc[**résolution**][**reconnaissance**][**social**]$

$role = \langle nature, force \rangle l + 1, f$

$force \in [non-directive, directive]$

$nature \in \{hypothèse, but, plan, action\}$

*Le mode de fonctionnement de reconstruction s'exprime, par exemple, en permettant à un agent de traiter l'ensemble des informations qui lui parviennent en privilégiant ses propres buts fixés par l'extérieur ou par lui même.*

Les liens d'interaction inter-niveaux existant entre les agents d'une même focalisation correspondent plus naturellement à des liens de contrôle. Les liens d'interaction intra-niveaux établissent plus généralement des liens d'observation entre les agents. On distingue ainsi les rôles de résolution selon deux dimensions : niveau de représentation et focalisation.

$$CS_c.soc[**résolution**] = CS_c.soc[**résolution**][**niveau**] \cup CS_c.soc[**résolution**][**focus**]$$

Nous laissons cependant la possibilité d'exprimer au sein de l'organisation du système de vision l'ensemble des relations possibles entre les agents.

Pour chacune de ces dimensions, les rôles de résolution sont distingués selon les dimensions similaires à celles des engagement :

$$CS_c.soc[**résolution**] = CS_c.soc[**résolution**][**TOI**] \cup CS_c.soc[**résolution**][**POI**] \cup CS_c.soc[**résolution**][**MOI**] \cup CS_c.soc[**résolution**][**ROI**]$$

Le **MOI**, la **ROI**, la **TOI** ou la **POI** sont définis en respectant la focalisation  $f$  à laquelle appartient l'agent. Un agent pour lequel, par exemple,  $f$  définit une région de l'image, ne pourra définir une **ROI** qu'au sein de  $f$ . Dès que celle-ci n'est plus incluse dans  $f$ , une interaction sera émise en direction de l'agent ayant une focalisation compatible avec la définition de la **ROI**. De la même manière un **MOI** s'exprimant dans des termes autres que ceux utilisés au sein du niveau de représentation  $l$  de l'agent, fera l'objet d'une interaction.

### c. Protocoles d'interaction

Les protocoles d'interaction mis en place entre les agents sont distingués selon les dimensions *niveau* et *focalisation* introduites au sein de l'agent.

$$Pr = Pr[**focus**] \cup Pr[**niveau**]$$

Cette distinction permet de définir une gestion pour les interactions au sein d'un niveau

de représentation et une gestion pour les interactions entre les niveaux de représentation au sein d'une focalisation.

### Protocole niveau ( $Pr[niveau]$ )

L'expression de ces protocoles reproduit le cycle de *génération d'hypothèse et de test* entre les agents au sein d'un niveau de représentation.

### Protocole focus ( $Pr[focus]$ )

L'expression de ces protocoles exprime le cycle de *prédiction-vérification* que nous avons mis en évidence dans le chapitre 3.

Au sein de cet ensemble, nous distinguons les protocoles en ceux installant un flux de contrôle ascendant et en ceux installant un flux de contrôle descendant entre les modules.

$$Pr[focus] = Pr[focus][ascendant] \cup Pr[focus][descendant]$$

- Protocole focus descendant ( $Pr[focus][descendant]$ )

Un exemple d'un tel protocole est :

$$\begin{aligned} \text{protocole} &\in Pr[focus][descendant] \\ \text{protocole} &= \{state0 \{ \text{request you } \langle \text{force, nature} \rangle \langle \text{TOI ROI}^{l',j} \text{ MOI}^{l',j} \\ &\quad \text{POI}^{l',j} \rangle \rightarrow state1, \\ &\quad \text{OR end} \} \\ &\quad state1 \{ \text{answer you } \langle \text{informative, hypothèse} \rangle \langle \text{resultat} \rangle \\ &\quad \rightarrow state0 \} \end{aligned}$$

avec  $l' < l$  et  $\text{force} \in [\text{non-directive, directive}]$

La valeur *you* exprime que l'interaction à émettre ne peut pas faire l'objet d'une diffusion. Elle doit être ciblée. La valeur *end* signifie la fin du dialogue, la valeur *resultat* demande à l'agent de répondre avec les résultats ayant satisfait le but.

L'agent du niveau  $l$  peut envoyer une requête à un agent de niveau inférieur  $l'$  mais appartenant à la même focalisation. Celui-ci répond à partir des éléments de sa description. (observations permettant de générer une nouvelle commande).

- Protocole focus ascendant ( $Pr[focus][ascendant]$ )

Un exemple d'un tel protocole est :

$$\begin{aligned} \text{protocole} &\in Pr[focus][ascendant] \\ \text{protocole} &= \{state0 \{ \text{inform you } \langle \text{informative, nature} \rangle \langle \text{TOI ROI}^{l',j} \\ &\quad \text{MOI}^{l',j} \text{ POI}^{l',j} \rangle \rightarrow state1 \} \\ &\quad state1 \{ \text{request you } \langle \text{force, nature} \rangle \langle \text{TOI ROI}^{l',j} \\ &\quad \text{MOI}^{l',j} \text{ POI}^{l',j} \rangle \rightarrow state2, \\ &\quad \text{inform you } \langle \text{force, nature} \rangle \langle \text{POI ROI}^{l',j} \text{ MOI}^{l',j} \end{aligned}$$

```

      POIl',j > → state0,
      end}
state2 { answer you < informative, nature > < POI ROIl',j
MOIl',j POIl',j > → state1}}

```

avec  $l' < l$  et  $force \in [\mathbf{non-directive}, \mathbf{directive}]$

*L'agent du niveau  $l'$  peut envoyer une information à un agent de niveau supérieur. Celui-ci a plusieurs possibilités pour lui répondre qui n'impliquent pas forcément la mise en place du cycle de prédiction vérification.*

L'expression de ces protocoles peut bien entendu être augmentée selon les stratégies d'interaction que l'on veut utiliser dans le système. On peut ainsi obtenir des protocoles de la complexité de ceux développés dans [Ber 92] pour l'expression de doute ou de la recherche d'informations.

#### d. Rôle de dialogue

De même manière similaire au rôle de résolution, nous distinguons les rôles de dialogue selon les dimensions relatives aux organisations : reconnaissance, reconstruction et niveau, focus.

$$\begin{aligned}
 CS_c.soc[\mathbf{dialogue}] &= CS_c.soc[\mathbf{dialogue}][\mathbf{reconnaissance}] \cup \\
 &\quad CS_c.soc[\mathbf{dialogue}][\mathbf{reconstruction}] \\
 CS_c.soc[\mathbf{dialogue}] &= CS_c.soc[\mathbf{dialogue}][\mathbf{niveau}] \cup CS_c.soc[\mathbf{dialogue}][\mathbf{focus}]
 \end{aligned}$$

Le rôle de dialogue définit le mode d'interaction à privilégier. Ainsi on pourra, par exemple, interdire à tout agent d'utiliser le protocole de perception ascendant dans une organisation pour la reconnaissance.

## 5.4 Conclusion

Dans ce chapitre, nous avons défini les modèles de contrôle individuel et social pour un Système Intégré de Vision.

A partir du modèle d'agent social défini dans la première section, nous avons isolé : (i) un modèle d'agent individuel regroupant les connaissances, états et mécanismes de traitement propres à l'activité de l'agent sans interaction avec les autres agents, (ii) un modèle de société incluant les capacités sociales exprimées dans le modèle d'agent social.

Les modèles de contrôle ont été ensuite explicités à partir de ces deux modèles. Le modèle de contrôle individuel ainsi formulé distingue le contrôle organisé selon une hiérarchie de contrôle et d'un processus constitué des fonctions d'engagement d'opération, d'exécution, d'effecteur, capteur et interprétation. Au sein du contrôleur, nous avons mis en place sur chacune des couches de contrôle les mécanismes nécessaires (décision, raisonnement) à la résolution du problème de contrôle qu'elles abordent. Les

mécanismes de bascule inter-couches et d'évaluation résolvent le problème d'observation en explicitant un ensemble d'informations utilisables pour redéfinir une nouvelle commande (conditions de validité, conflits). Au travers du mécanisme de bascule inter-couches, existe la possibilité d'installer des fonctionnements répétitifs au sein de l'agent en privilégiant une couche de contrôle. Le modèle de contrôle individuel ainsi exhibé est fondé sur les propositions 1, 2 et 3 du chapitre précédent.

Nous appuyant sur les couches de contrôle mises en évidence au sein de ce modèle, nous avons explicité les liens d'interaction définissant les interactions pouvant exister entre les agents. Nous les avons exprimés à partir de l'hypothèse selon laquelle un agent contrôle un autre agent s'il définit les commandes issues des couches de contrôle de l'autre agent. Nous avons introduit les notions de nature et de force afin d'exprimer les degrés de contrôle pouvant se mettre en place entre les agents. A partir de celles-ci, nous avons exprimé : (i) l'organisation qui restreint les traitements et descriptions utilisées par les agents, (ii) les protocoles qui restreignent les échanges entre les agents. Le modèle de contrôle social est fondé sur les propositions 5, 6 et 9.

Les modèles de contrôle social et individuel que nous obtenons à ce stade de notre étude ne sont pas spécifiques à notre domaine d'application. Nous pouvons donc espérer que ces modèles puissent être utilisés dans le cadre d'autres études utilisant des systèmes Multi-Agents.

L'intervention de notre domaine d'application au sein de ces modèles a consisté à préciser les différentes connaissances et représentations que l'on peut s'attendre à trouver dans un système intégré de vision. Les modèles obtenus sont bâtis sur la proposition 4 du chapitre précédent en ce qui concerne le contrôle individuel, et sur la proposition 7, 8 en ce qui concerne le modèle de contrôle social.



## Chapitre 6

# Réalisation

Dans ce chapitre, nous présentons, dans un premier temps, le prototype du *Module Standard d'Agent* (cf. figure 6.1) construit à partir des modèles de contrôle individuel et social pour un Système Intégré de Vision obtenus en conclusion du chapitre précédent. Son implantation a été réalisée en C et en CLIPS<sup>12</sup> en utilisant le squelette d'application *SAVA* [Ber 91] pour la communication physique et l'exécution répartie des agents sur plusieurs machines *SUN* connectées par un réseau *Ethernet*.

Ce Module Standard d'Agent a été utilisé pour réaliser quelques expériences de contrôle à partir de la reformulation des modules du cadre d'expérimentation du chapitre 4, (cf. figure 4.1): description bidimensionnelle en *AGENT-DESCRIPTION-2D*, interprétation en *AGENT-INTERPRETATION*. Les modules superviseur et caméra ont été simplement encapsulés par les mécanismes sociaux du module standard. Nous avons étudié :

- le *contrôle individuel au sein de l'AGENT-DESCRIPTION-2D et de l'AGENT-INTERPRETATION*. Nous présentons l'utilisation des couches commande, adaptation et décision au sein de ces deux agents.
- le *contrôle social*. La définition de *protocoles d'interaction* et d'*organisations ré-*

---

<sup>1</sup>C Language Interface for Production Systems

<sup>2</sup>CLIPS offre un mécanisme de représentation de faits et de règles de productions. Son choix est motivé par son approche "système de production". De plus, son interfaçage avec le langage C est efficace et facile à mettre en œuvre.

gulant les interactions entre les agents a permis de valider ce modèle.

## 6.1 Module Standard d'Agent

Le prototype du Module Standard d'Agent réalisé est fondé sur les modèles de contrôle social et individuel mis en évidence à la fin du chapitre précédent. L'architecture de ce module standard d'agent est décomposée en :

- un *module-individu* (cf. figure 6.1, composants sur fond noir) qui implante le modèle de contrôle individuel. Il est constitué des fonctions d'évaluation, de raisonnement, de décision, d'engagement, agissant sur les états de raisonnement  $RS$ , de décision  $DS$  et d'engagement  $CS.ind$ . Les fonctions et les états de décision et d'engagement sont distingués selon les trois couches de décision, d'adaptation et de commande.
- un *module-dialogue* (cf. figure 6.1, composants sur fond gris foncé) bâti à partir du modèle du dialogue. Le module-dialogue est constitué des fonctions *récepteur* et *émetteur*, des fonctions d'interprétation-dialogue, d'exécution-dialogue mettant à jour les états de perception  $PS$  et d'exécution  $ES$ . Ces deux fonctions sont dédiées à la gestion des interactions en utilisant les protocoles  $Pr$  et langages  $L$  d'interaction.
- un *module-organisation* (cf. figure 6.1, composants sur fond gris clair) implanté par une fonction de définition des engagements sociaux de *résolution* et de *dialogue* à partir de la consultation de la bibliothèque d'organisation  $Org$ . Ne recherchant à valider que la couche commande du modèle de contrôle social, c.-à-d. à partir d'une organisation définie en terme de rôles de résolution, contrôler le dialogue et la résolution qui se déroulent au sein de l'agent, nous avons simplifié cette fonction en remettant en cause manuellement les organisations définies.
- un *séquenceur* (cf. figure 6.1) définit le séquençement des modules et, pour un module donné, le séquençement de ses fonctions. L'interprétation des engagements sociaux (cf. figure 6.1, engagements sur fond gris clair) par la *bascule-social* et par la *bascule-individu* gère respectivement le passage du module-dialogue au module-individu et du module-individu au module-dialogue. L'évaluation des conditions de validité des engagements individuels (cf. figure 6.1, engagements sur fond noir) par les bascule-contrôles déclenche ou non le passage entre couches de contrôle au sein du module-individu. L'exécution de la *bascule-organisation* permet une remise en cause possible des engagements de résolution, de dialogue et d'organisation (cf. figure 6.1, engagements sur fond gris clair). l'organisation.

Dans ce Module Standard, les états de perception  $PS$ , d'exécution  $ES$ , de raisonnement  $RS$ , de décision  $DS$  et d'engagement  $CS$  sont implantés par des structures de données représentant les *données* (DONNEE), les *hypothèses* (HYPOTHESE), les *buts*





(BUT), les *plans* (PLAN), les *intentions* (INTENTION), les *engagements* (ENGAGEMENT). Les connaissances *KR* sont représentées sous la forme de structures CONNAISSANCE ayant une partie condition testant les valeurs des structures implantant les états, et une partie action créant ou modifiant ces structures.

### 6.1.1 Séquenceur

La fonction *séquenceur* exécute chacune des fonctions du cycle courant. Après l'exécution de chaque fonction, elle exécute la *bascule-individu*, *bascule-social* et la *bascule-contrôle* (cas où module courant est le module-individu). Leur satisfaction conduit à un changement du module courant ou de la couche courante. Selon ces modifications, le séquenceur exécute la première fonction du cycle du module ou de la couche activés. Si aucun changement n'est intervenu, la fonction suivante du cycle est exécutée.

La structure ETAT-CYCLE décrit l'état du cycle de contrôle manipulé par la fonction *séquenceur* (cf. figure 6.2). Elle est constituée des attributs :

- ETAT-CYCLE
  - *fonction*: nom de la fonction courante du cycle,
  - *couche*: *decision*, *adaptation* et *commande* (nom de la couche de contrôle courante du cycle)
  - *module*: *individu* (module-individu), *dialogue* (module-dialogue), *organisation* (module-organisation)

Un changement de couche de contrôle modifie la valeur de l'attribut *couche* (noté ec.couche) et inscrit le nom de la première fonction du cycle de contrôle de la couche dans l'attribut *fonction*. Un changement de module inscrit : le nom du nouveau module dans l'attribut *module* (noté ec.module), la valeur *commande* dans l'attribut *couche* (noté ec.couche), la valeur de la première fonction du cycle de la couche du module dans l'attribut *fonction* (noté ec.fonction).

Le séquenceur du module d'agent standard exécute l'algorithme suivant. Définissons : *cycle-c* la liste des fonctions réalisant un cycle, *exec fonction* appliquée sur ETAT-CYCLE qui exécute la fonction inscrite dans ec.fonction, *first fonction* donnant la première fonction du cycle, *up fonction* donnant la couche supérieure à une couche donnée, *down fonction* donnant la couche inférieure à une couche donnée, *cycle fonction* retournant le cycle de contrôle d'un module pour une couche donnée.

- *cycle*(individu, couche) = (évaluation, raisonnement, décision, engagement), où couche peut avoir les valeurs commande, adaptation ou décision.
- *cycle*(dialogue, commande) = (exécution-dialogue, émetteur, récepteur, interprétation-dialogue).
- *cycle*(organisation, commande) = (organisation).

```

fonction sequenceur
  /** DEFINITION DE L'ORGANISATION */
  ec.module = "organisation"; ec.fonction = "organisation";
  ec.couche = "commande";
  exec(ec);
  /** INITIALISATION: MODULE INDIVIDU */
  ec.module = "individu"; ec.couche = "commande";
  cycle-c = cycle(ec.module,ec.couche);
  tant que vrai faire
    si (bascule-individu() = vrai) alors
      /** PASSAGE DE L'INDIVIDU AU DIALOGUE */
      ec.module = "dialogue"; ec.couche = "commande";
      cycle-c = cycle(ec.module,ec.couche);
    sinon
      si (bascule-controle() = vrai) alors
        /** CHANGEMENT DE COUCHE */
        ec.couche = up(ec.couche);
        cycle-c = cycle(ec.module,ec.couche);
      sinon
        si (bascule-social() = vrai) alors
          /** PASSAGE DIALOGUE A INDIVIDU */
          ec.module = "individu"; ec.couche = "commande";
          cycle-c = cycle(ec.module,ec.couche);
        sinon
          si (cycle-c = null) alors
            /** FIN DU CYCLE */
            si (ec.module = "individu") alors
              si (ec.couche ≠ "commande") alors
                /** DESCENTE D'UNE COUCHE */
                ec.couche = down(ec.couche);
                cycle-c = cycle(ec.module,ec.couche);
              sinon
                /** FIN DU CYCLE INDIVIDU. PASSAGE A DIALOGUE */
                ec.module = "dialogue"; ec.couche = "commande";
                cycle-c = cycle(ec.module,ec.couche);
            fsi
          sinon
            /** DIALOGUE A INDIVIDU */
            ec.module = "individu"; ec.couche = "commande";
            cycle-c = cycle(ec.module, ec.couche);
          fsi
        fsi
      fsi
    fsi
  fin tant que
fin fonction

```

Le prototype étant écrit en CLIPS, toute fonction du cycle, ainsi que celle du *sequenceur*

est implantée par une ou plusieurs règles de production interprétées par les fonctions de filtrage, de sélection et d'interprétation de CLIPS. Le cycle de contrôle de l'agent est ainsi implanté par un enchaînement de contextes d'activation correspondant à chacune des fonctions, modules et couches. L'exécution d'une fonction d'une couche d'un module correspond à l'activation d'un contexte : toute prémisse d'une règle possède une condition testant l'égalité des valeurs des attributs de ETAT-CYCLE avec les noms de la fonction, de la couche et du module qu'elle implante. Chaque fonction est une base de règles mettant en place ses propres stratégies d'examen des règles : saturation, lex, recency par exemple.

Nous détaillons l'architecture du module standard d'agent, en présentant successivement le *module-individu*, l'*état d'engagement social*, le *module-dialogue* et enfin le *module-organisation*.

### 6.1.2 Module Individu

Au sein du module individu, nous présentons :

- les structures de données implantant les états de *raisonnement RS*, de *décision DS* et d'*engagement CS*.
- les connaissances : *compétences*, *tactiques*, *stratégies* exprimées sous forme de règles de production.
- les fonctions du cycle de contrôle.

#### a. Etats

Les structures implantant les constituants de ces états possèdent différents attributs. Ces différentes structures sont implantées par les *templates* de CLIPS, similaires à des structures de frames. La création, la destruction et la modification des contenus des structures sont réalisées par l'utilisation des fonctions de CLIPS pour la gestion de ces templates. La structure HYPOTHESE représente les *hypothèses*, la structure BUT représente les *buts*, la structure PLAN représente les *plans*, la structure INTENTION représente les *intentions* et la structure ENGAGEMENT représente les *engagements*. Par la suite pour faire la distinction entre structure, attribut et valeurs, nous utilisons la convention typographique suivante : STRUCTURE, *attribut*, *valeur*. Chacune de ces structures possède deux attributs utilisés pour la gestion des représentations. Ce sont :

- *cycle* : numéro du cycle de contrôle dans lequel la structure est créée.
- *id* : identificateur de la structure. Affecté lors de sa création, il est unique.

#### Liens

Une structure *lien* est définie afin d'exprimer les relations entre les structures. Ces liens sont utilisés à la fois par les connaissances de l'agent, mais aussi par les fonctions

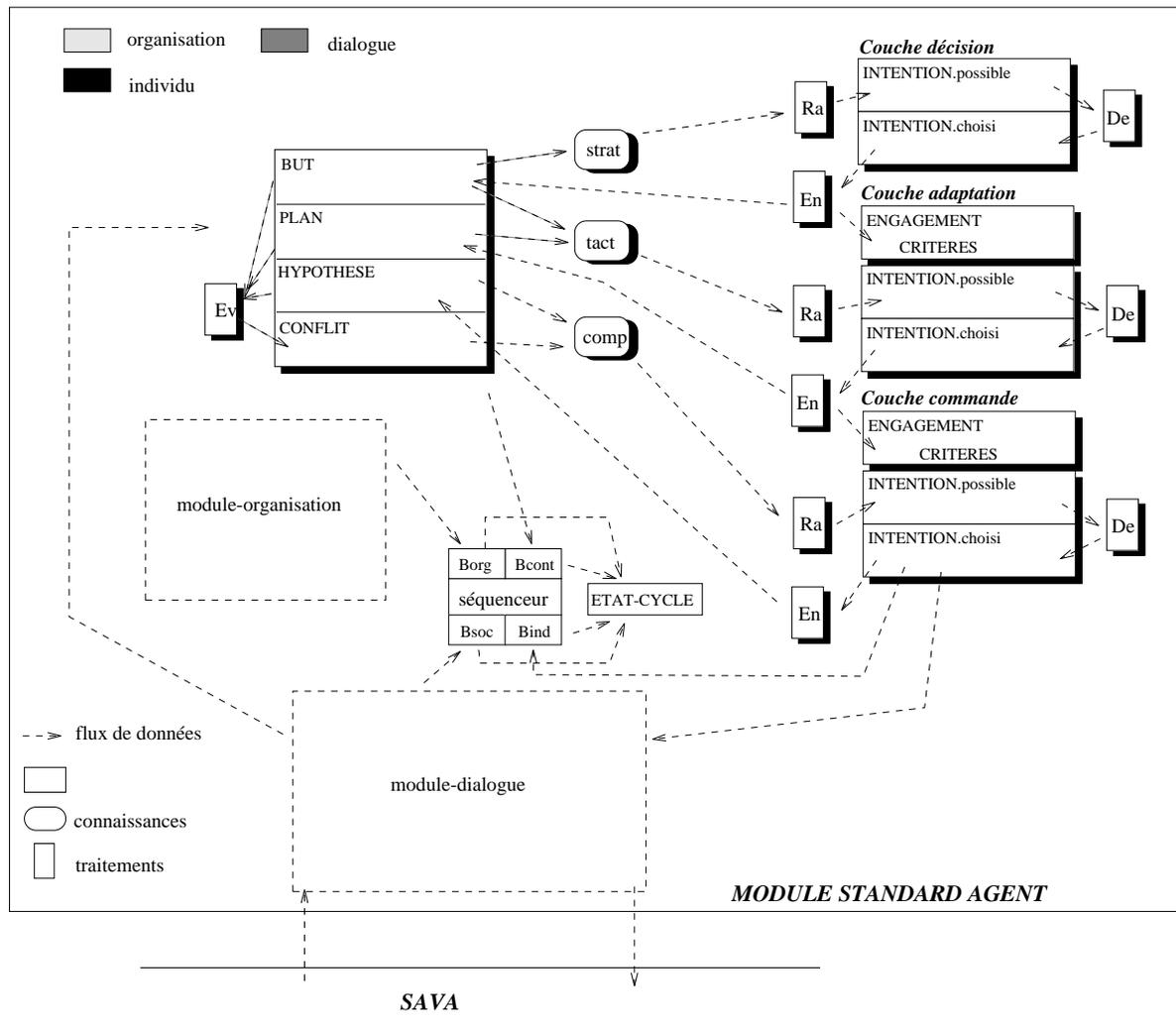


FIG. 6.3 - Architecture du Module Individu pour un Agent d'un Système Intégré de Vision. Les symboles suivants correspondent aux fonctions: *Ev* fonction d'évaluation, *Ra* fonction de raisonnement, *De* fonction de décision, *En* fonction d'engagement, *Borg* bascule d'organisation, *Bcont* bascule inter-couche de contrôle, *Bind* bascule individu, *Bsoc* bascule sociale. Les connaissances de l'agent sont: *tact* tactiques, *comp* compétences, *strat* stratégies. L'état de raisonnement est constitué des représentations des *buts*, *plans*, *hypothèses* et *conflits*. L'état de décision est représenté par les *intentions*. L'état d'engagement est représenté par les *engagements* et *critères*.

de traitement. Pour chaque lien créé, un lien inverse est créé. Les attributs de cette structure sont :

- LIEN
  - *type* : noms des types de lien permettant de distinguer les liens entre eux. Les valeurs permises dans l'implantation actuelle sont : *est-conflit*, *a-conflit*, *supporte*, *supporte-par*, *satisfait*, *satisfait-par*, *realise*, *realise-par*, *invalide*, *invalide-par*. La sémantique de ces différents liens est introduite dans la suite de ce rapport.
  - *source* : *id* de la structure source du lien
  - *dest* : *id* de la structure cible du lien

Les liens suivants sont communs à toutes les structures :

- *invalide* (inverse *invalide-par*) relie les structures de l'état de raisonnement à l'ENGAGEMENT qu'elles invalident. Ce lien est géré par la fonction de *bascule-controlé*.
- *realise* (inverse *realise-par*) relie les INTENTIONS AUX BUTS OU PLANS qu'elles désignent. Il relie aussi les ENGAGEMENTS AUX INTENTIONS qui ont conduit à leur création.

## Etat de raisonnement

L'état de raisonnement (cf. figure 6.3) est constitué d'hypothèses, de buts, de plans, de conflits. Les structures les représentant possèdent un attribut *statut* dont la valeur exprime la distinction entre *RS.évalué* et *RS.non-évalué*. La fonction d'évaluation y inscrit les valeurs suivantes :

- *non-integree* : l'élément est dans *RS.non-évalué*. Il ne peut être pris en compte que par la fonction d'évaluation. Cette valeur est la valeur par défaut de toute structure créée.
- *conflit*, *rejete*, *integre* : l'élément appartient à *RS.évalué*. La valeur est *conflit* si l'incohérence liée à l'élément n'a pas encore été résolue, *rejete* si l'élément n'est pas accepté à l'issue de la résolution de conflit, *integre* si l'élément est intégré dans l'état de raisonnement.

Plusieurs liens sont définis pour marquer les interdépendances entre les structures lors de la résolution (cf. figure 6.4) :

- *a-conflit* (inverse *est-conflit*) relie tout élément en conflit avec la structure CONFLIT représentant le conflit créé.
- *supporte* (inverse *supporte-par*) relie la structure aux structures que les connaissances ont créées à partir d'elle. Les structures inscrites dans la *source* ou la

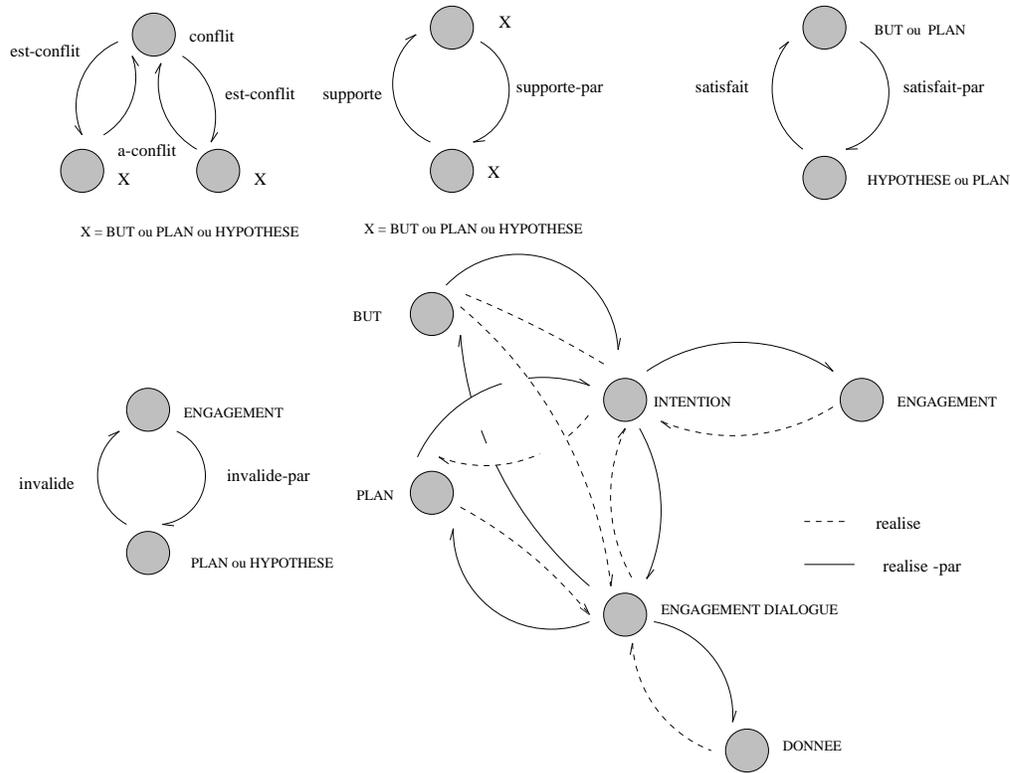


FIG. 6.4 - Liens entre les structures de l'architecture du Module Standard d'Agent.

*cible* de ce lien doivent être de même type (HYPOTHESE avec HYPOTHESE, BUT avec BUT, ...). Ainsi, ce lien relie un but aux sous-buts créés, un plan aux sous-plans, une hypothèse aux hypothèses

- *satisfait* (inverse *satisfait-par*) relie les structures d'HYPOTHESES aux BUTS ou PLANS qu'elles satisfont.

### hypothèse

Les *hypothèses* sont la modélisation de la description de la scène que construit l'agent. Elles sont représentées par la structure HYPOTHESE constituée des attributs :

- HYPOTHESE
  - *type* : symbole ou ensemble de symboles exprimant la sémantique du contenu de la structure,
  - *valeur* : contenu de la structure. Celui-ci peut être constitué de plusieurs champs selon l'application.

Dans l'exemple 6.1, l'HYPOTHESE représente un segment (valeur de *type*) dont la description est obtenue par le contenu de *valeur* : segment dont les extrémités ont pour coordonnées (10, 10) et (30, 40). La valeur 2 est l'identificateur du segment dans l'AGENT-DESCRIPTION-2D.

---

```
(hypothese
  (cycle 12) (id hyp-39) (statut integre)
  (type SEGMENT)
  (valeur 2 10 10 30 40)
  (supporte hyp-43)
)
```

Exemple 6.1: **Exemple d'hypothese.** Exemple d'une HYPOTHESE de segment dans l'AGENT-INTERPRETATION. Le lien *supporte* montre qu'elle a été utilisée pour créer l'hypothèse *hyp-43*.

### but

Les buts expriment les objectifs que doit satisfaire l'agent. Leur satisfaction consiste à atteindre une configuration particulière d'hypothèses. Un but est représenté par la structure BUT possédant les mêmes attributs qu'une HYPOTHESE. Le *type* d'un BUT est construit à partir de l'expression des buts mise en évidence au chapitre précédent. Les valeurs sont donc par exemple FIND CUBE, IDENTIFY ROI, etc. La *valeur*

est la spécification numérique du but. Dans l'exemple 6.2 le BUT de type IDENTIFY ROI, a pour valeur de l'attribut *valeur* la région à identifier (région dont le centre est au centre du repère et dont les demi longueurs en x et en y sont égales à 50).

---

```
(but
  (cycle 19) (id but-9) (statut integre)
  (satisfait-par plan-12)
  (type IDENTIFY ROI)
  (valeur ROI 0 0 50 50)
)
```

Exemple 6.2: **Exemple de but.** Exemple de but d'identification traité par l'AGENT-INTERPRETATION.

---

### plan

Les *plans* expriment les relations entre les compétences de l'agent et les buts qu'elles contribuent à satisfaire. Un plan est exprimé selon une expression similaire à celle du but. Selon l'application et selon sa complexité, d'autres structures peuvent être utilisées pour représenter les différentes étapes du plan. Nous en verrons un exemple dans le cadre de l'AGENT-INTERPRETATION. Un plan est une séquence d'étapes à exécuter. Chacune d'elle représente une action à exécuter (cf. exemple 6.3 étape-10 est la première étape du plan. Dans celle-ci existe un lien vers une autre étape.)

---

```
(plan
  (cycle 3) (id plan-10) (status integre)
  (type IDENTIFY ROI)
  (a-etape etape-10))
```

Exemple 6.3: **Exemple de plan**

---

### conflits

Les conflits sont générés par la fonction d'évaluation. Ils regroupent les informations relatives à un conflit entre des éléments de l'état de *raisonnement*. La structure CONFLIT

(cf. exemple 6.4) comporte les attributs :

– CONFLIT

- *statut*: *traite* ou *non-traite* selon que le conflit a été résolu ou non. La valeur *non-traite* est la valeur par défaut.
- *type*: *hypothese*, *but*, *plan*, *accointance* selon les structures participant au conflit.

Le lien *est-conflit* regroupe les éléments participant au conflit. Dans le cadre de notre exemple (cf. exemple 6.4), les hypothèses d'existence d'un groupement parallèle *hyp-12* et d'un groupement jonction *hyp-34* pour les deux mêmes segments sont en conflit.

---

```
(conflit
  (cycle 15) (id conflit-23) (statut non-traite)
  (type hypothese)
  (est-conflit hyp-12 hyp-34)
)
```

Exemple 6.4: **Exemple de conflit**

---

## Etat de décision

L'état de décision *DS* est constitué des *intentions*. Une intention désigne une relation entre une stratégie, une tactique ou une compétence et l'état de raisonnement. Les couches *commande*, *adaptation*, *décision* structurent les intentions appartenant à cet état. Les intentions qui s'y inscrivent sont : possibles (dans *DS.possibilités*), choisies (dans *DS.choix*). La structure INTENTION représente les éléments de cet état. (cf. figure 6.3). Selon la couche sur laquelle elle apparaît, une INTENTION représente une stratégie, une tactique ou une compétence executables. La structure d'une INTENTION est similaire à celle d'un *KSAR* [Hr 85]. Elle comporte les attributs suivants :

– INTENTION

- *module*: module dans lequel la structure est utilisée *individu*, *dialogue* ou *organisation*.
- *statut*: *possible* si l'INTENTION appartient à l'ensemble de possibilités, *choisie* si elle appartient à l'ensemble de choix, *traite* une fois l'INTENTION exécutée par la fonction d'*engagement*. Cet attribut est modifié dans les fonctions de *raisonnement* et de *décision*.

- *couche*: nom de la couche sur laquelle intervient l'intention (*commande*, *adaptation*, ou *decision*).
- *priorité*: valeur numérique utilisée par la fonction de décision pour sélectionner l'intention.
- *type*: type de l'élément auquel fait référence l'intention : *but*, *plan* ou *operation*; *request*, *inform* ou *answer* lorsque l'intention fait référence à une action de communication.
- *action*: liste d'actions déduites de la stratégie, de la tactique ou de la compétence ayant créé cette intention. Ces actions seront interprétées par la fonction d'engagement dans le contexte d'activation inscrit dans l'attribut *contexte*.
- *ressource*: liste de symboles faisant référence au *type* des éléments produits par l'intention (hypothèses, buts, plans, conflits, actions). Le contenu de cette liste est utilisé pour choisir l'intention. Dans l'exemple 6.5, cette ressource exprime que cette intention va produire des connexions (JUNC).
- *contexte*: contexte relatif à l'état de raisonnement pour l'exécution du contenu de l'attribut *action*, variables liées, etc. Dans l'exemple 6.5, les valeurs du *contexte* sont les identificateurs des deux hypothèses de segment à tester (*hyp-12* et *hyp-14*). Entre ces deux valeurs s'intercalent les valeurs respectives du coefficient de vraisemblance de l'hypothèse. Les trois valeurs terminant la liste définissent l'angle, l'incertitude et la distance maximale entre les deux extrémités de la jonction.

Les liens suivants sont utilisés :

- *realise* relie l'intention inscrite sur la couche décision (resp. couche adaptation) au BUT (resp. au PLAN) à partir desquels elle a été créée.
- *realise-par* relie une intention choisie inscrite sur la couche de décision ou d'adaptation à l'engagement présent sur la couche inférieure créé par la fonction d'engagement.

## Etat d'engagement individuel

L'état d'engagement individuel (cf. figure 6.3) est décomposé selon chacune des couches de contrôle. Sur celles-ci sont représentées les *engagements*, expression de critères de contrôle du comportement individuel *CS.ind*. Ces engagements conditionnent le fonctionnement des fonctions de la couche sur laquelle ils sont inscrits au travers de critères d'*incohérence* (fonction d'évaluation), d'*activation* (fonction de raisonnement), de *sélection* (fonction de décision), de *validité* (fonction de bascule). Le critère de *sélection* a été décomposé en *focus* et *politique*. Le *focus* exprime un critère de sélection, la *politique* exprime une règle de composition des focus. Nous représentons un engagement par la structure ENGAGEMENT, constituée des attributs suivants. Cette structure est reliée à différents CRITÈRES exprimant les critères de contrôle. Nous exprimons dans tout

---

```
(intention
  (module individu) (couche commande)
  (type operation) (cycle 12) (id int-30) (statut choisi)
  (priorite 60) (action calcul-jonction-action)
  (ressource JONCTION)
  (contexte hyp-12 40 hyp-14 50 1.57 1.32 10)
```

Exemple 6.5: **Exemple d'intention** . dans l'*AGENT-INTERPRETATION*. Le contenu de l'attribut *action* est l'identificateur de la partie action constituant la compétence à exécuter.

---

engagement une condition de terminaison et une condition de validité. Cette dernière est testée par la fonction *bascule-contrôle*.

– ENGAGEMENT

- *module*: *organisation*, *dialogue* ou *individu*
- *statut*: *choisi* lors de la création de l'ENGAGEMENT, *invalide* si les conditions de validité n'ont pas été vérifiées, *traite* si sa condition de terminaison exprimée par *condition-terminaison* a été satisfaite.
- *type*: type de l'engagement. La valeur prise fait référence aux types d'engagement: région d'intérêt, (*ROI*), modèle d'intérêt (*MOI*), tâche d'intérêt (*TOI*) ou période temporelle d'intérêt (*POI*), mis en évidence dans le modèle de contrôle pour un système intégré de vision.
- *couche*: couche de contrôle sur laquelle l'engagement contraint le fonctionnement des fonctions: *commande*, *adaptation*, *decision*.
- *condition-terminaison*: liste de conditions de terminaison de l'engagement (conditions normales de suppression de l'engagement).
- *condition-validité*: liste de conditions de validité de l'engagement portant sur la configuration des différents états (conditions anormales de remise en cause de l'engagement).

Les liens suivants sont utilisés :

- *realise* relie cet engagement à l'intention de la couche supérieure à partir de laquelle il a été créé.
- *prochain* (inverse *precedent*) relie les engagements qui doivent être activés les uns après les autres.
- *invalide-par* relie l'engagement aux structures de l'état de raisonnement l'ayant invalidé.

- *a-critere* (inverse *est-critere*) relie l’engagement aux critères qu’il exprime.

Les critères de contrôle sont représentés par la structure CRITERE (cf. exemple 6.6) possédant les attributs suivants :

– CRITERE

- *type*: type de l’engagement, *focus*, *politique*, *activation*, *incoherence*.
- *valeur*: valeur du critère. Nous présentons des exemples de mots clefs s’inscrivant dans cet attribut lors de la description de chacun des agents.

```
(engagement
  (module individu) (couche commande)
  (cycle 12) (id eng-12) (statut choisi)
  (type ROI)
  (realise int-30)
  (a-critere cont-3 cont-4 cont-5)
  (condition-terminaison false)
  (condition-validite SEGMENT-IN-ROIP))
```

```
(critere
  (module individu) (couche commande)
  (cycle 12) (id cont-2)
  (type focus)
  (valeur 0 0 3 30 30 3)
  (est-critere eng-12)
)
```

Exemple 6.6: **Exemple d’engagement.** Cet engagement, une fois mis en œuvre, sera toujours actif (condition terminaison est False) excepté si les segments présents dans la description de l’agent sortent de la région d’intérêt (condition validité SEGMENT-IN-ROIP))

## b. Représentation des connaissances

L’ensemble des connaissances est constitué de *faits*, de *compétences*, de *tactiques* et de *stratégies*. Les faits sont représentés par des HYPOTHESES. Les compétences, les tactiques et les stratégies sont représentées par la structure CONNAISSANCE. Cette dernière est constituée d’une partie *condition*, d’une partie *action* et des attributs suivants :

– CONNAISSANCE

- *couche*: *commande*, *adaptation décision* selon qu’elle représente une compétence, une tactique ou une stratégie.

- *module*: *individu*, *dialogue*, *organisation* selon qu'elle représente une connaissance individuelle, une connaissance de dialogue ou une connaissance d'organisation.
- *condition*: liste de conditions portant sur les structures de l'état de raisonnement. Les structures d'ENGAGEMENT apparaissent dans ces conditions lorsque la connaissance est relative à une tactique ou à une compétence. L'évaluation de la partie condition construit la *ressource* et le *contexte* de l'INTENTION par la liaison des variables exprimées dans cette partie condition.
- *action*: création, modification ou destruction de structures de l'état de raisonnement. Dans le cas où la CONNAISSANCE exprime une tactique ou une stratégie, elle crée des ENGAGEMENTS dans la couche de contrôle inférieure.

Nous implantons la partie condition et la partie action d'une compétence, d'une tactique ou d'une stratégie avec des règles CLIPS distinctes.

- La règle exprimant la partie condition représente les conditions de la compétence, de la stratégie ou de la tactique dans la prémisse de la règle. La partie action représente la construction du contexte d'activation avec inscription d'un identificateur spécifique testé dans la partie condition de la règle implantant la partie action.
- La règle implantant la partie action teste en prémisse l'identificateur correspondant à la partie condition de la connaissance qu'il implante. La partie action implante la partie action de la connaissance.

### c. Mécanismes de traitements

Les mécanismes de traitements agissent sur les structures décrites précédemment. Le séquençage de ces mécanismes est défini par la fonction *séquenceur*. Le module-individu est constitué des fonctions d'évaluation, de raisonnement, de décision, d'engagement et de bascule que l'on retrouve sur chacune des couches. Dans le cadre de ce prototype, leur algorithme est le même d'une couche à l'autre. Seuls les états et connaissances sur lesquels elles travaillent sont différents.

Par la suite nous désignons le contenu de l'attribut d'une structure par : `STRUCTURE.attribut`. Pour simplifier les schémas algorithmiques, par abus de langage tout ce qui est désigné par ENGAGEMENT ne concerne que les engagements individuels, c.-à-d. les engagements tels que : `ENGAGEMENT.module = individu`.

### Evaluation

La fonction d'évaluation détecte les incohérences entre les éléments des états de raisonnement et ceux produits par la fonction d'engagement de la couche de commande ou par la fonction d'interprétation-dialogue. Elle applique la fonction *evalue* dont l'algorithme est donné ci-dessous, sur tout élément de l'état de raisonnement X dont le

*statut* a la valeur **non-integre** avec les éléments Y ayant déjà été évalués (valeur de *statut* différente de **non-integre**) en appliquant les critères d'incohérence (valeur de l'attribut *valeur* des CRITERE tels que : la valeur de *type* est **incoherence**, ils sont reliés à l'ENGAGEMENT de *statut choisi* agissant sur la même *couche* que la fonction). Si aucun conflit n'a été détecté, la valeur **integre** est inscrite dans le *statut* de X. La fonction *detecter* retourne la valeur vrai si une incohérence a été détectée entre la structure X et la structure Y, en fonction des critères. La fonction *creer-conflit* crée une structure CONFLIT et remplit la valeur des attributs en fonction des valeurs des attributs de X et de Y. La fonction *creer-lien* crée une structure LIEN entre les deux structures données en paramètre.

```

fonction evalue(X,Y,critere)
  si (detecter(X,Y,critere) = vrai) alors
    conflit = creer-conflit(X,Y);
    creer-lien("est-conflit", conflit, X);
    creer-lien("est-conflit", conflit, Y);
    X.statut = conflit;
    Y.statut = conflit;
    retourner(vrai);
  fsi
  retourner(faux);
fin fonction

```

Un ensemble de règles CLIPS détectant les conflits entre buts, plans, hypothèses implantent cette fonction d'évaluation.

## Raisonnement

La fonction de raisonnement applique les compétences, les tactiques ou les stratégies sur l'état de raisonnement courant pour créer les intentions possibles en fonction du critère d'activation de l'engagement courant présent sur la couche de la fonction (valeur de l'attribut *valeur* des CRITERE tels que : la valeur de *type* est **activation**, ils sont reliés à l'ENGAGEMENT de *statut choisi* agissant sur la même *couche* que la fonction). La fonction raisonnement applique la fonction *raisonne* sur toute connaissance X appartenant à la couche. La fonction *active* définit si la connaissance est activée étant donné le critère d'activation. Elle inscrit dans *contexte* les éléments de l'état de raisonnement satisfaisant les conditions. La fonction *creer-intention* définit le contenu des attributs *contexte* et *ressource* de l'INTENTION.

```

fonction raisonne(X,critere)
  si (active(X.condition,critere,contexte) = vrai) alors
    intention = creer-intention(X,contexte);
    intention.statut = possible;
    creer-lien("realise",intention,contexte);

```

```

    fsi
fin fonction

```

## Décision

La fonction de décision sélectionne les intentions X parmi celles qui sont possibles en appliquant les critères de sélection de l'engagement courant de la couche considérée (focus et politique). Nous notons *focus* le contenu de l'attribut *valeur* du CRITERE de *type focus*. Nous notons *politique* le contenu de l'attribut *valeur* du CRITERE dont l'attribut *type* a pour valeur *politique*. La fonction *calcul-priorite* calcule la priorité d'une intention en appliquant le critère focus. La fonction *choisi-intention* choisit une intention en appliquant le critère politique. L'algorithme de la fonction décision est le suivant :

```

fonction decision(couche,focus,politique)
    pour tout X : X est INTENTION && X.couche = couche &&
        X.statut = possible faire
        calcul-priorite(X,focus);
    finpour
    intention = choisi-intention(politique);
    intention.statut = choisi;
fin fonction

```

## Engagement

Si la couche considérée est la couche décision ou adaptation, la fonction d'engagement traduit l'intention choisie en critères de contrôle pour les fonctions de la couche inférieure. Si la couche est la couche commande et si l'intention est relative à une opération, la fonction d'engagement crée de nouveaux éléments dans l'état de raisonnement. Si l'intention est relative à une action, elle active le module-dialogue. La fonction engagement fait appel à la fonction *execute* qui évalue la partie action d'une intention dans le contexte d'activation de celle-ci. La fonction *activer-module* modifie les valeurs des attributs de ETAT-CYCLE en fonction du nom de module donné en paramètre. La fonction *test* teste la condition donnée en paramètre et inscrit dans contexte les éléments confrontés à la condition.

```

fonction engagement(couche)
    pour tout X : X est INTENTION && X.statut = choisi &&
        X.couche = couche faire
        si X.type ∈ {request, answer, inform} alors
            activer-module("dialogue");
        sinon
            execute(X.action,X.contexte);
    fsi

```

```

    X.statut = traite;
finpour
/** TEST CONDITION TERMINAISON DES ENGAGEMENTS **/
pour tout Y: Y est ENGAGEMENT && Y.statut = choisi faire
    si (test(X.condition-terminaison,condition) = vrai) alors
        Y.statut = traite;
    fsi
finpour
fin fonction

```

### bascules inter-couches de contrôle

La fonction de bascule-contrôle évalue les conditions de validités de l'engagement courant sur les différents états de la couche de contrôle courante en s'intercalant entre chacune des fonctions de la couche. La fonction *activer-couche* modifie les valeurs des attributs de ETAT-CYCLE en fonction de la valeur de son paramètre couche.

```

fonction bascule-contrôle(couche)
    pour tout X: X est ENGAGEMENT && X.statut = choisi &&
        X.couche = couche faire
        si (test(X.condition-validite,contexte) = faux) alors
            creer-lien("invalide-par",X,contexte);
            X.statut = invalide;
        fsi
        activer-couche(up(couche));
    finpour
fin fonction

```

### 6.1.3 Etat d'engagement social

Ayant décrit le module-individu, nous décrivons l'état d'engagement social. Cet état (cf. figure 6.5) est constitué des engagements inscrits dans *CS.soc*. Ils expriment les rôles de *résolution*, de *dialogue* et d'*organisation*. La structure représentant un engagement social est similaire à celle d'un engagement individuel. Nous retrouvons ainsi :

– ENGAGEMENT

- *module: organisation.*
- *statut: choisi* lors de la création de l'ENGAGEMENT, *invalide* si le critère de validité a été satisfait, *traite* si sa condition de terminaison exprimée par *condition-terminaison* a été satisfaite.
- *couche*: couche sur laquelle s'applique l'engagement. Dans le cas de notre implantation actuelle, seule la valeur *commande* est inscrite dans cet attribut.

- *type: role-resolution, role-dialogue, role-organisation* pour exprimer les différents rôles dans le cadre du contrôle social.
- *condition-termination*: liste de conditions de terminaison de l'engagement (conditions normales).
- *condition-validité*: liste de conditions de validité de l'engagement (conditions de remise en cause). Cet attribut est testé par la fonction *bascule-organisation*.
- *valeur*: expression du rôle. Nous donnerons l'expression de ces rôles dans le cadre de nos expériences de contrôle ci-après.
  - Role de résolution. Les valeurs *social* et *individu* inscrites dans l'attribut *valeur* de la structure d'ENGAGEMENT social font la distinction entre les éléments appartenant à l'ensemble  $CS_c.soc[résolution][social]$  et ceux appartenant à l'ensemble  $CS_c.soc[résolution][individu]$ . Les autres valeurs de cet attribut sont constituées de la *nature*, de la *force* et du *sujet*. (cf. exemple 6.9. 6.10)
  - Role de dialogue. les valeurs sont le nom de protocole (cf. exemple 6.11).
  - Role d'organisation: Dans le cadre de notre réalisation, nous n'avons pas abordé l'expression de ces rôles.

#### 6.1.4 Module-dialogue

Au sein du module de dialogue, nous présentons :

- Le langage d'interaction et les protocoles d'interaction.
- La structure DONNEE utilisée pour la représentation des messages reçus *PS* et à émettre *ES*. La structure ACCOINTANCE, représentation des autres agents, utilisée pour définir les récepteurs des messages.
- les fonctions *récepteur*, d'*interprétation-dialogue*, d'*exécution-dialogue* et *émetteur*.

##### a. Langage et Protocoles d'interaction

###### langage d'interaction

Le langage d'interaction est constitué d'un langage de communication et d'un langage de représentation des connaissances. Nous présentons ici les éléments du langage de communication relatifs à la définition de la *nature*, de la *force*. Les autres constituants du langage de communication seront rapidement rappelés. Nous présentons les éléments du langage de représentation des connaissances utilisées dans la définition du *contenu* d'une interaction dans le cadre de nos expériences de contrôle.

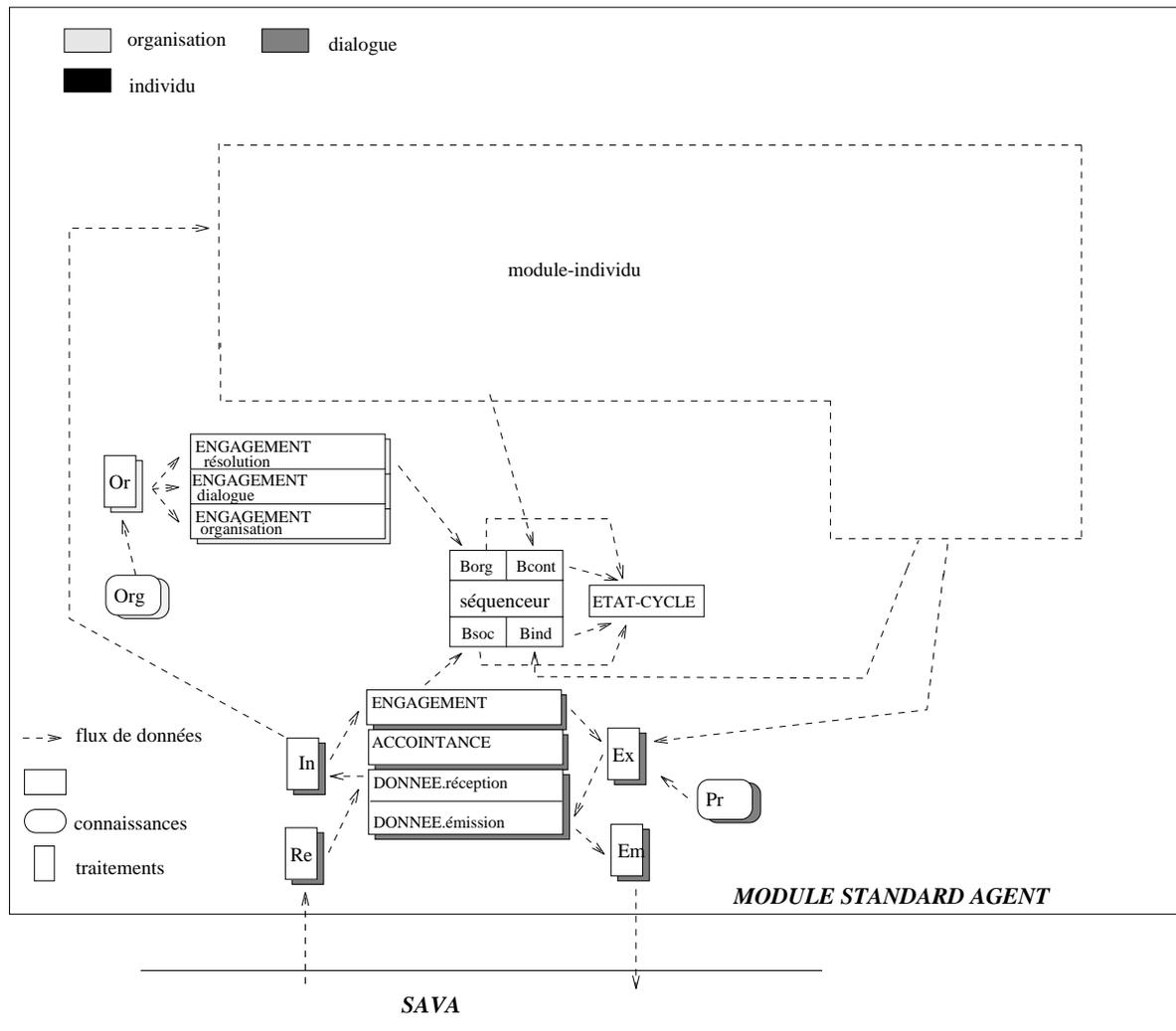


FIG. 6.5 - Architecture des capacités sociales du Module Standard pour un Agent d'un Système Intégré de Vision. Les symboles suivants correspondent aux fonctions : *Or* fonction d'organisation, *In* fonction d'interprétation dialogue, *Ex* fonction d'exécution dialogue, *Re* fonction récepteur, *Em* fonction émetteur, *Borg* bascule d'organisation, *Bcont* bascule inter-couche de contrôle, *Bind* bascule individu, *Bsoc* bascule social. L'état d'engagement est représenté par les *engagements* et *critères*. Les états de perception et d'exécution sont représentés par les *données*. *Pr* est la bibliothèque de protocoles, *Org* est la bibliothèque d'organisations, *L* est l'ensemble des langages d'interaction.

Un sous-ensemble des termes du langage d'interaction est utilisé pour la définition des organisations (rôles), des protocoles d'interaction et des interactions. C'est notamment le cas de la force et de la nature telles que nous les avons définies dans le cadre de notre modèle de contrôle social.

### Langage de communication

La nature et la force sont exprimées par les valeurs suivantes : La valeur *hypothese*

	valeurs possibles
<b>nature</b>	hypothese, action, plan, but
<b>force</b>	nulle, warning, informing, <i>bargaining</i> , <i>persuading</i> , <i>commanding</i>

TAB. 6.1 - Valeurs permises pour la définition de la nature et de la force. Les valeurs possibles pour la force apparaissant en italique et en gras sont celles appartenant à l'intervalle [*non-directive..directive*], les autres sont des forces de type informative.

symbolise la nature hypothèse d'une interaction, la valeur *action* désigne une interaction dont la nature est une action, la valeur *plan* désigne une interaction dont la nature est un plan, la valeur *but* désigne une interaction dont la nature est un but.

Le langage de communication est constitué des termes pour la définition de la *destination*, du *mode-communication*, du *type*, du *lien*.

- La *destination* peut prendre les valeurs *diffusion* ou *ciblée*. Pour cette dernière valeur, la fonction d'exécution-dialogue inscrira l'identité d'un agent.
- Le *mode-communication* peut prendre les valeurs *synchrone* ou *asynchrone*,
- le *type* peut prendre les valeurs *request*, *inform* ou *answer*,
- le *lien* est défini à partir de la force et de la nature tel que nous l'avons vu ci-dessus.

L'ensemble de ces primitives sont à la base de la définition du langage d'expression des protocoles d'interaction.

### Protocoles d'interaction

Nous représentons un protocole par un réseau augmenté de transitions (cf. figure 6.8).

- Un *état de conversation* est la configuration de l'échange entre les agents à la réception ou à l'émission d'une interaction.
- Une *transition* entre deux états exprime les contraintes sur les valeurs de certains champs d'une interaction : *destination*, *mode de communication*, *type*,

ou *lien* et des contraintes imposées sur le *contenu*. Ces dernières utilisent un sous ensemble du langage de représentation des connaissances. Nous en donnons un exemple dans la dernière section.

Du point de vue de l'implantation, les protocoles sont regroupés dans un ensemble que nous appelons *lois*. Une loi définit le comportement de communication de l'ensemble des agents du système. Dans cet ensemble de lois, un protocole est représenté déclarativement selon le formalisme suivant :

$$\text{protocole} = \{ \{ \text{etat} \{ [condition] \text{interaction} \rightarrow \text{etat}, \\ [OR [condition] \text{interaction} \rightarrow \text{etat}]^* \} \}^* \}$$

Le symbole \* exprime la répétition possible de l'expression à l'intérieur des crochets ou des accolades. Les expressions entre les crochets sont optionnelles.

- **état**: est un référent au sein du protocole de l'état de conversation. Le symbole particulier *end* est utilisé pour marquer la fin du dialogue.
- **condition**: L'expression de la condition est dépendante de la résolution mise en place par le protocole. Elle exprime la condition de transition. Elle utilise un sous ensemble de référents au langage de communication mais aussi surtout au langage de représentation.
- **interaction**: désigne la transition possible d'état. Elle est exprimée à partir du langage de communication et d'éléments du langage de représentation.

Chaque agent dispose localement, d'un exemplaire des lois de la société. Cette solution est préférée par rapport à une centralisation des lois pour limiter les communications nécessaires et introduire la possibilité d'une modification locale des lois par les agents eux-mêmes, c.-à-d. modification des protocoles en ajoutant ou supprimant des étapes d'interactions par exemple. L'autre atout est la diversification possible des types de loi avec des lois locales qui ne s'appliquent qu'à un groupe restreint d'agents et des loi globales qui s'appliquent à l'ensemble de ceux-ci. Les fonctions de création et de consultation des protocoles ont été réalisées pour l'expression des protocoles de présentation dans [Pop 92].

## b. Etat de perception, Etat d'exécution

L'état de perception *PS* représente les messages envoyés par les autres agents. L'état d'exécution *ES* représente ceux à envoyer aux autres agents. Ces deux états sont représentés par la même structure *DONNEE* (cf. figure 6.5), constituée des attributs suivants :

- *DONNEE*
  - *cycle*: numéro de cycle de création

- *id*: identificateur de la structure
- *niveau*: *emission* ou *reception* selon que la DONNEE appartient à l'état d'exécution ou à celui de perception.
- *statut*: *non-traite* à sa création et *traite* lorsqu'elle a été traitée par la fonction d'interprétation-dialogue ou par la fonction émetteur.
- *type*: sémantique des valeurs exprimées dans l'attribut *valeur*. (ex: la valeur *hypothese informing FIND IMAGE* dans l'attribut *type* de l'exemple 6.7 indique que le message est une hypothèse d'image relative aux buts envoyés (FIND IMAGE). La valeur de l'attribut *valeur* est l'identificateur de l'image *image-0*

Le lien *realise-par* relie la DONNEE à l'ENGAGEMENT de dialogue qui l'a créée (*eng-10*).

---

```
(donnee
 (module dialogue) (niveau reception)
 (cycle 1) (id data-12) (statut non-traite)
 (type hypothese informing FIND IMAGE)
 (realise-par eng-10)
 (valeur image-0)
 )
```

Exemple 6.7: **Exemple de donnee.** Donnée reçue par l'AGENT-DESCRIPTION-2D dont le contenu est relatif à une image. Elle est créée par l'interaction dont l'engagement correspondant est eng-10

---

### c. accointances

Une *accointance* rassemble la *description externe* des autres agents du système ainsi que celle de l'agent lui-même. Elles sont utilisées pour déterminer les agents destinataires des messages. Une accointance (cf. exemple 6.8), est représentée par la structure ACCOINTANCE. En plus de l'attribut *statut* géré de la même manière que précédemment, elle a les attributs suivants :

– ACCOINTANCE

- *agent-id*: identificateur de l'agent décrit par cette accointance. Au lancement du système, un identificateur est affecté à chaque agent. Celui-ci est utilisé dans toutes les interactions entre les agents. La valeur *mine* est utilisée pour désigner la propre description externe de l'agent.

- Les attributs, *role*, *actions-base*, *structure*, *exercices* et *usage* expriment la description externe telle qu'elle est introduite dans [Ber 92]. Dans le cadre de ce travail, seuls le *role* et les *actions-base* sont utilisés. Dans le cadre de nos expériences sur le contrôle, la valeur de *role* exprime le nom d'un niveau de représentation du système de vision sur lequel travaille l'agent : *DESCRIPTION\_2D*, *INTERPRETATION* par exemple. La valeur de *actions-base* est constituée des différents types de modèles que l'agent peut fournir.

Un seul *role* est exprimé par *accountance*. Si l'agent a d'autres rôles, une nouvelle *accountance* est créée.

---

```
(accountance
  (cycle 12) (id accountance-89) (statut integre)
  (agent-id mine)
  (role DESCRIPTION_2D)
  (actions-base SEGMENT JUNC PROTO PARALLEL OVERLAPPED ALIGNED))
```

Exemple 6.8: **Exemple d'accountance.** Exemple de description externe de l'*AGENT-DESCRIPTION-2D*.

---

#### d. Etat d'engagement dialogue

Une structure d'*ENGAGEMENT* représente une interaction en provenance d'un autre agent ou à destination d'un autre agent. Elle est utilisée pour la gestion des enchaînements d'actes de langage : à la différence des opérations, une interaction se déroule sur plusieurs cycles de l'agent. Il est donc nécessaire de conserver une trace de cette exécution entre son début et sa fin. La structure d'*ENGAGEMENT* de dialogue est similaire à celle des *ENGAGEMENT* individuel et social. Elle possède ainsi les attributs suivants :

– *ENGAGEMENT*

- *module: dialogue*
- *type: request, inform, answer* exprimant le type d'interaction entre les agents.
- *statut: choisi, attente, invalide, traite*. On retrouve la même gestion que pour les autres engagements.
- *valeur*: cet attribut est constitué de plusieurs champs correspondant respectivement à :
  - *social* ou *individu*, selon que l'interaction est en provenance de l'extérieur, ou de l'agent lui-même.

- *nature* de l'interaction
- *force* de l'interaction
- *sujet* de l'interaction, symboles exprimant la sémantique du contenu de l'interaction.

Différents liens sont utilisés :

- Lors de la réception, le lien *realise* relie l'ENGAGEMENT à la DONNEE dont il est issu. Ce même lien relie l'ENGAGEMENT à la DONNÉE lors de l'émission. Les éléments inscrits dans les états du module-individu et qui découlent d'une interaction avec les autres agents, sont reliés à cet ENGAGEMENT par le lien *realise*. Les modifications de ces éléments sont testés pour la suite du dialogue entre les agents.
- Le lien *a-context* (inverse *contexte-de*) relie l'engagement vers la CONVERSATION à laquelle est attachée l'interaction.

La structure CONVERSATION représente l'état de conversation courant de l'agent avec un autre agent. Cette structure est créée pour la gestion des engagements de dialogue entre les agents et permet un suivi des enchaînements d'acte de langage entre les agents par rapport à un sujet. A chaque nouvelle conversation une nouvelle structure est générée par l'agent à l'initiative de l'échange. La CONVERSATION est conservée tant que le dialogue n'est pas terminé et est mise à jour au fur et à mesure des interactions. Elle est constituée des attributs suivants :

- CONVERSATION
  - *id*: identificateur de la conversation
  - *etat-conversation*: ensemble des informations utilisées pour la progression du protocole: *version*, nom de la *loi*, nom du *protocole*, nom de l'*état*.
    - *version* est une date de version de la loi. Elle est utilisée en cas de modification de la loi, pour que les agents puissent utiliser la même version de schéma de dialogue.
    - *loi*, *protocole* et *état* font référence à l'expression des protocoles au sein des lois.
  - *agent-id*: nom de l'agent avec lequel se déroule la conversation.

Le contenu des attributs *id*, et *etat-conversation* de CONVERSATION sont inscrits dans l'interaction communiquée à l'agent récepteur. Une telle gestion même si elle alourdit le contenu des échanges, permet de faciliter le travail de suivi de déroulement du protocole. Pour toute interaction, il est possible de connaître l'état de conversation dans lequel l'agent récepteur sera.

## e. Mécanismes de traitement

### Récepteur

La réception se décompose en deux fonctions relatives à l'échange d'informations au sein d'un *agent-focus* (fonction de *récepteur-focus*) et à l'échange d'informations au sein d'un *agent-niveau* (fonction de *récepteur-niveau*). Ces fonctions produisent : des DONNEES, des ENGAGEMENTS correspondant aux interactions reçues. La fonction *consulte* fait appel aux fonctions de communication de SAVA que nous utilisons pour la communication entre les agents. Elle rend le premier message de la boîte aux lettres. Si celle-ci est vide, la valeur rendue est null. La fonction *recherche-conv* recherche la CONVERSATION correspondant aux informations inscrites dans le message. Si aucune conversation n'est trouvée, une nouvelle conversation est créée.

**fonction** recepteur()

```

tant que ((message = consulte()) ≠ null) faire
    engagement = creer-engagement(message);
    engagement.valeur = social;
    engagement.statut = choisi;
    conversation = recherche-conv(message);
    creer-lien("a-contexte",engagement, conversation);
    donnee = creer-donnee(message);
    donnee.niveau = reception;
    donnee.statut = non-traite;
    creer-lien("realise-par",donnee, engagement);

```

**fin tant que**

**fin fonction**

### Interprétation-dialogue

La fonction d'*interprétation-dialogue* traduit une donnée inscrite dans l'état de *perception* en éléments qui s'inscrivent dans un des deux états de l'agent : état de *raisonnement* et état de *décision* (*ensemble de choix* et *ensemble de possibilités*) en fonction du résultat de la fonction de *bascule-social* testant l'organisation de résolution relative aux interactions que l'agent peut recevoir. La fonction *interprete* crée une HYPOTHESE, un BUT, un PLAN ou une INTENTION en fonction de la valeur de l'attribut *type* de la DONNEE. La fonction *trouve* recherche les éléments relié à l'élément en paramètre par le lien fournit aussi en paramètre.

**fonction** interpretation-dialogue()

```

pour tout X : X est une DONNEE && X.statut = non traite &&
    X.niveau = reception faire
    element = interprete(X);
    engagement = trouve("realise-par",X);
    creer-lien("realise", element, engagement);

```

```

        X.statut = traite;
    finpour
fin fonction

```

### Exécution-dialogue

Pour chacune des CONVERSATIONS reliées à un engagement de dialogue en attente en provenance des autres agents ou du module-individu, la fonction d'*exécution-dialogue* consulte les protocoles en testant les conditions de transition à partir de l'état inscrit dans l'attribut *etat-conversation* de la CONVERSATION. Le test des conditions de transition est réalisé sur le contenu des états de raisonnement et de décision. Le comportement de cette fonction est régi par les *rôles de dialogue* qui définissent les protocoles utilisables par l'agent étant donné le sujet de conversation (champ de l'attribut *valeur* de la structure CONVERSATION).

### Emetteur

L'émission se décompose aussi en deux fonctions qui sont: *émetteur-niveau* et *émetteur-focus*. La fonction *envoyer* fait appel aux fonctions d'émission de SAVA.

```

fonction emetteur()
    tant que X : X est une DONNEE && X.niveau = emission &&
        X.statut = non-traite faire
        envoyer(X);
        engagement = trouve("realise",X);
        engagement.statut = attente;
    fin tant que
fin fonction

```

#### 6.1.5 Module-organisation

Le module-organisation est constitué d'une fonction d'*organisation* et d'une *bascule-organisation*. La fonction d'*organisation* est implantée actuellement par une fonction d'appel à l'utilisateur pour redéfinir une nouvelle organisation en cas d'invalidation de l'organisation précédente. La fonction de *bascule-organisation* teste la condition de validité des engagements sociaux dont la valeur du *type* est soit *role-resolution*, *role-organisation*, *role-dialogue*.

#### 6.1.6 Bascule social

La fonction *bascule-social* provoque le passage du module-dialogue au module-individu en tenant compte des interactions provenant du module-dialogue (engagements dont l'attribut *module* a pour valeur *dialogue* et dont l'attribut *valeur* contient la valeur *social*) et des engagements représentés dans  $CS_c.soc[resolution][social]$ : engagements dont l'attribut *module* a pour valeur *organisation*, l'attribut *type* est

*resolution-role*, l'attribut *valeur* contient la valeur *social*. Les autres valeurs de ce dernier attribut expriment la condition de rejet d'une interaction. Cette fonction fait appel pour chacune des interactions à la fonction *test-interaction*. La fonction *tester* teste le contenu de l'attribut *valeur* d'une interaction au contenu de l'attribut *valeur* des engagements sociaux courants. La fonction *modifier* inscrit le contenu de l'interaction dans les états correspondant à la nature et à la force exprimée dans celle-ci. La fonction *retirer* les enlève.

```

fonction test-interaction(interaction)
  /** interaction = engagement du module-dialogue **/
  element = trouve("realise",interaction);
  si (tester(interaction.valeur = false) alors
    interaction.statut = attente;
    modifier(element,interaction);
    activer-module("individu");
  sinon
    interaction.statut = invalide;
    retirer(element);
  fsi
fin fonction

```

### 6.1.7 Bascule individu

La fonction *bascule-individu* teste l'expression des rôles de résolution individu (engagements dont l'attribut *module* a pour valeur *organisation*, l'attribut *type* a pour valeur *resolution-role*, l'attribut *valeur* contient la valeur *individu*) au sein de chacun des états du module-individu. La fonction *test-engagement* dont l'algorithme est donné ci-dessous applique la fonction *verifie* à tout engagement de ce type sur les états du module-individu. La fonction *creer-engagement-dialogue* suit l'algorithme suivant : si l'engagement concerne une nouvelle conversation, création d'une CONVERSATION sinon il est relié à la conversation dépendant de ses informations. Par exemple, un engagement de dialogue consistant en une réponse à un but issu de l'extérieur sera relié à la CONVERSATION ayant conduit à la création de ce but à résoudre dans l'agent. Les champs force, nature de l'engagement sont remplis en fonction du sujet et des rôles de résolution.

```

fonction test-engagement(engagement)
  si (verifie(engagement.valeur) = vrai) alors
    engagement-dial = creer-engagement-dialogue(engagement);
    activer-module("dialogue");
  fsi
fin fonction

```

## 6.2 Contrôle dans un Système Intégré de Vision : VAP

Deux modules utilisés dans le cadre d'expérimentation lié au contrôle au sein du système VAP (cf. chapitre 4) ont été reformulés dans le Module Standard d'Agent : module de description bidimensionnelle en AGENT-DESCRIPTION-2D et module d'interprétation en AGENT-INTERPRETATION (cf. figure 6.1).

- l'AGENT-DESCRIPTION-2D construit la description 2D de la scène en terme de segments et de groupements perceptuels.
- l'AGENT-INTERPRETATION associe des modèles symboliques à la description de la scène construite par l'agent précédent et obtient une description en terme de formes géométriques simples. Dans cette réalisation, les capacités de cet agent ne sont pas très développées d'un point de vue interprétation symbolique. Elles sont, cependant, suffisantes pour la validation des modèles précédents sur un type d'agent qui, a priori, utilise des méthodes de raisonnement différentes de celles de l'AGENT-DESCRIPTION-2D.

Le cadre d'expérimentation (cf. figure 6.6) est constitué de ces deux agents et de deux autres modules préexistants encapsulés par les capacités sociales du module standard : CAMERA et SUPERVISEUR. A partir de ce nouveau cadre d'expérimentation, nous étudions le contrôle individuel au sein de l'AGENT-DESCRIPTION-2D et de l'AGENT-INTERPRETATION. Nous présentons l'utilisation des couches commande, adaptation et décision au sein de chacun d'eux. Nos expériences ont porté plus spécialement sur l'utilisation des mécanismes de bascule entre les couches de contrôle et sur l'expression des conditions de validité des engagements. Ces expériences constituent une validation pratique du modèle de contrôle individuel.

Le modèle de contrôle social a été validé au sein de la société constituée de tels agents, par :

- des protocoles d'interaction mettant en place des fonctionnements différents. Les protocoles testés permettent soit un mode d'*interaction ascendant*, soit un mode d'*interaction descendant* (cf. chapitre précédent, le Modèle d'Agent pour un Système Intégré de Vision).
- l'expression et l'utilisation d'organisations différentes : hiérarchie de contrôle existant dans le système VAP et hétéarchie dans laquelle la hiérarchie de contrôle précédente peut être réactivée ponctuellement selon les nécessités de la résolution au travers de l'expression de conditions de validité de cette organisation. Cet aspect, bien que testé et ayant produit quelques résultats, fait encore partie de la prospective.

Les modules CAMERA et SUPERVISEUR ayant une intégration incomplète, les fonctions *bascule-social* et *bascule-individu* ont été testées dans l'AGENT-DESCRIPTION-2D et l'AGENT-INTERPRETATION.

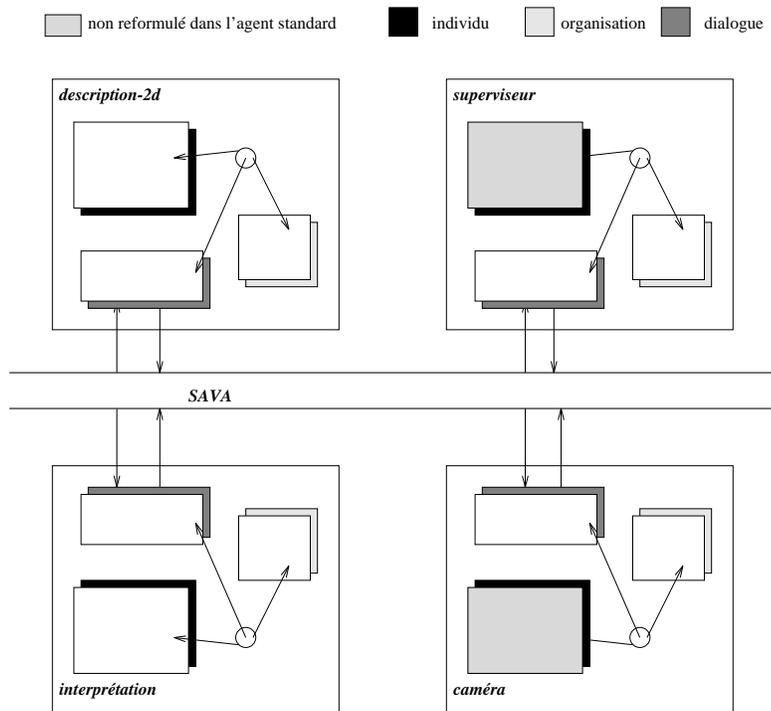


FIG. 6.6 - Système pour l'étude du contrôle individuel et social dans un Système Intégré de Vision.

## 6.3 Contrôle dans l'AGENT-DESCRIPTION-2D

Notre objectif est de donner un exemple de fonctionnement du Module Standard d'Agent dans un cas concret et de montrer l'intérêt d'installer les couches de contrôle au sein de l'AGENT-DESCRIPTION-2D. Nous montrons plus particulièrement la mise en place et l'utilisation des bascules dans cet agent : passage entre la couche *adaptation* et la couche *commande* pour modifier le critère de contrôle : région de l'image (ROI) dans laquelle effectuer les traitements.

### 6.3.1 AGENT-DESCRIPTION-2D

Cet agent est en interaction avec la CAMERA qui lui fournit à la demande des images d'intensité. L'AGENT-INTERPRETATION envoie des requêtes de buts à satisfaire visant, par exemple, à rechercher des segments dans une région de l'image ou demandant de rechercher une région dans laquelle se trouvent différents segments. L'agent met en œuvre ses connaissances et ses traitements afin de satisfaire ces buts et communique à l'AGENT-INTERPRETATION les résultats. Son fonctionnement organisé autour des trois couches de contrôle est le suivant :

- couche décision : A partir des buts choisis, définition des critères de contrôle fixant le type de modèle à rechercher *MOI* (segment, groupement par exemple) et la région d'intérêt maximale *ROI* dans laquelle réaliser les traitements.
- couche adaptation : Cet agent utilise une pyramide multi-résolutions pour l'extraction d'indices dans l'image. La couche *adaptation* a donc pour objectif d'affiner la définition de la région d'intérêt en précisant les niveaux de cette pyramide sur lesquels concentrer les traitements. Si ceci est nécessaire elle restreint la région. Elle définit aussi les critères contrôlant le fonctionnement des procédures de traitement exprimées dans la partie action des compétences.
- couche commande : Cette couche sélectionne et exécute les compétences applicables sur la description courante de la scène. Les compétences choisies sont exécutées dans la région et avec les paramètres de fonctionnement inscrits dans l'engagement courant et ce jusqu'à ses conditions de validité ne soient pas satisfaites ou que ses conditions de terminaison soient satisfaites. Dans cet agent, la fonction d'évaluation effectue un suivi temporel sur les segments produits à partir des images reçues de la CAMERA.

Nous explicitons dans la section suivante les éléments inscrits dans l'état de raisonnement ainsi que les connaissances de cet agent. L'état de décision n'étant pas, a priori, dépendant de l'application, la structure des INTENTIONS n'est pas modifiée. L'expression des ENGAGEMENTS n'est pas modifiée. Seul leur contenu varie.

### a. Etat de raisonnement

La description de la scène est représentée en termes d'HYPOTHESE. Elle est constituée des types d'hypothèses suivantes :

- Les images sont représentées par des HYPOTHESE dont les attributs sont :
  - HYPOTHESE
    - *type*: **IMAGE**
    - *valeur*: pointeur sur la structure C stockant les valeurs d'intensités de l'image.
  
- les segments sont représentés par des HYPOTHESES avec les attributs suivants :
  - HYPOTHESE
    - *type*: **SEGMENT**.
    - *valeur*: (*niveau* *xm* *ym* *c* *d*  $\theta$  *L* *x1* *y1* *x2* *y2* *dc* *dd* *d $\theta$*  *dL* *varc* *vard* *var $\theta$*  *varL*)  
 où *niveau* est le niveau de la pyramide de résolution sur lequel le segment a été trouvé, (*xm*, *ym*) coordonnées du point milieu, (*c*,*d*) coordonnées perpendiculaires et parallèles du point milieu,  $\theta$  angle du segment avec l'axe x du repère, *L* demi-longueur, (*x1*,*y1*) et (*x2*,*y2*) coordonnées cartésiennes des points extrémités <sup>3</sup>; *dc*, *dd*, *d $\theta$* , *dL*: variations temporelles de *c*, *d*,  $\theta$  et *L* <sup>4</sup>; *varc*, *vard*, *var $\theta$* , *varL*: incertitude sur ces paramètres.
  
- Les groupements perceptuels sont représentés par des HYPOTHESES dont les attributs sont :
  - HYPOTHESE
    - *type*: type de groupement perceptuel. La valeur est par exemple **JUNC** pour les groupements perceptuels relatifs aux connexions, **PARALLEL** pour les groupements perceptuels relatifs aux segments parallèles, ...
    - *valeur*: information propre au groupement. Cette valeur pour un groupement de type jonction (JUNC) est : (*niveau*, *x*, *y*,  $\theta$ , *Q*, *id1*, *id2*)  
 où *niveau* est le niveau de la pyramide; *x*, *y*: coordonnées de l'intersection des deux segments;  $\theta$  angle entre les deux segments; *Q* qualité; *id1*, *id2*: identificateurs des segments qui le composent.

---

<sup>3</sup>Ces représentations redondantes évitent des calculs coûteux. Cette option a été privilégiée du fait des contraintes temporelles imposées par le projet VAP.

<sup>4</sup>valeurs déduites du suivi du segment dans l'image

Les BUTS expriment une configuration de la description à obtenir. Leurs attributs sont les suivants :

- BUT
  - *type*: valeur exprimée à l'aide des tâches exprimées dans le chapitre précédent : FIND, TRACK, IDENTIFY, EXPLORE, WATCH, DESCRIBE, RELATE, VERIFY; auxquels est ajoutée l'expression symbolique du type de groupement demandé ou du segment : SEGMENT, JUNC, PROTO, ou par le terme ROI lorsqu'une région d'intérêt est spécifiée dans le but.
  - *valeur*: spécification numérique du but.

## b. Connaissances

### Compétences

La plupart des compétences utilisées dans cet agent ont une partie action qui fait appel à des procédures écrites en C. La partie condition teste la valeur des hypothèses pour activer ces procédures.

#### Compétences de transformation

Les opérations de transformation sont les fonctions d'extraction de segment. Elles sont de la forme suivante :

- CONNAISSANCE
  - *couche*: *commande*
  - *module*: *individu*
  - *condition*: Présence d'HYPOTHESE de *type IMAGE* dont le *statut* est *integre*,
  - *action*: calcul-pyramide(image), calcul-gradient(image), extraction-segment(image), création d'HYPOTHESES de type **SEGMENT**. Générer un lien *support* de l'image vers les segments extraits.

Les fonctions :

- calcul-pyramide : construction de la pyramide Multi-résolutions. Celle-ci est obtenue par une cascade de convolutions de l'image avec des filtres binomiaux [1 2 1] suivies d'un rééchantillonnage [Cro 84].(cf. figure 4.3).
- calcul-gradient : calcul du gradient sur les niveaux de la pyramide.
- extraction-segment : seuillage par hystérésis (seuil-haut et seuil-bas) appliqué sur les valeurs du gradient et chainage.

### Compétences d'enrichissement

Un ensemble de compétences est utilisé pour la production des groupements perceptuels. Ces compétences sont de la forme :

– CONNAISSANCE

- *couche*: *commande*
- *module*: *individu*
- *condition*: présence d'HYPOTHESE de type **SEGMENT** dont l'attribut *statut* a la valeur *integre* et n'ayant pas de lien *support* en direction d'une HYPOTHESE du type de groupement cherché.
- *action*: calcul du groupement sur cet ensemble de segments.

Le calcul du groupement est une fonction écrite en C travaillant sur l'ensemble des segments répondant à la partie condition. Les procédures de groupement sont appliquées avec des paramètres de contrôle définissant les caractéristiques du groupement. Ainsi par exemple pour une procédure recherchant les connexions aura son fonctionnement fixé par la valeur de l'angle  $\theta$  entre les deux segments, la variation permise sur cet angle  $\delta\theta$ , la distance maximale permis entre les extrémités au point de la jonction  $\epsilon$ . Ces paramètres sont définis par la couche adaptation.

D'autres compétences d'enrichissement expriment des actions de communication vers l'extérieur lorsqu'un élément de la description n'est pas présent localement. Une de ces connaissances est ainsi par exemple, la compétence consistant en une création d'interaction dès qu'une image a été traitée. Elle s'exprime de la manière suivante :

– CONNAISSANCE

- *couche*: *commande*
- *module*: *individu*
- *condition*: image traitée
- *action*: communication sujet:image  
où communication sujet:image fait appel à une action de communication au sujet d'une image. Cette action de communication, une fois choisie, sera prise en charge par la fonction d'exécution-dialogue qui mettra en œuvre le protocole adéquat pour interagir avec l'agent capable de fournir une telle image. Cet agent est déterminé par la consultation des accointances.

### Stratégies

Les stratégies expriment le contenu du but en termes d'engagement de type **ROI** et **MOI**. Elles sont distinguées selon le type de tâche qu'elles concernent. Une des stratégies que nous avons implanté consiste en une traduction du but en ENGAGEMENTS

dont les critères sont exprimés à partir de la valeur du *type* du BUT et du contenu de l'attribut *valeur*.

– CONNAISSANCE

- *couche: décision*
- *module: individu*
- *condition*: présence d'un BUT dont la valeur de l'attribut *statut* est *integre* et n'étant pas source d'un lien *realise*.( la présence d'un tel lien indique que le but a été déjà traduit en une intention )
- *action*: création d'ENGAGEMENTS dont la valeur du *type* est **ROI**, et **MOI**.

Une autre stratégie consiste à redéfinir les engagements en cas d'invalidation de l'engagement.

## Tactiques

Différentes tactiques ont été utilisées. Tout comme les stratégies, nous les décomposons en :

- tactique définissant les ENGAGEMENTS à partir des ENGAGEMENTS et structures inscrites dans l'état de raisonnement.
- tactique permettant de redéfinir un ENGAGEMENT invalidé.

Dans ce dernier type de tactique, nous trouvons par exemple, la tactique *tactique-centre-gravité* visant à définir la position de la région d'intérêt afin de garder le maximum de segments dans son centre :

– CONNAISSANCE

- *couche: adaptation*
- *module: individu*
- *condition*: ENGAGEMENT de type **ROI** remis en cause,
- *action*: calcul du centre de gravité des segments invalidant la ROI. Définir le centre de la région exprimée dans l'ENGAGEMENT de *type ROI* à partir de ce centre de gravité

Nos expériences nous ont amenés à définir d'autres tactiques, visant par exemple à déplacer la région d'intérêt au sein des niveaux de la pyramide Multi-résolution à partir des segments rencontrés sur un niveau de la pyramide. D'autres tactiques envisageables pourraient consister à modifier les paramètres des procédures afin d'augmenter la qualité des groupements ou segments.

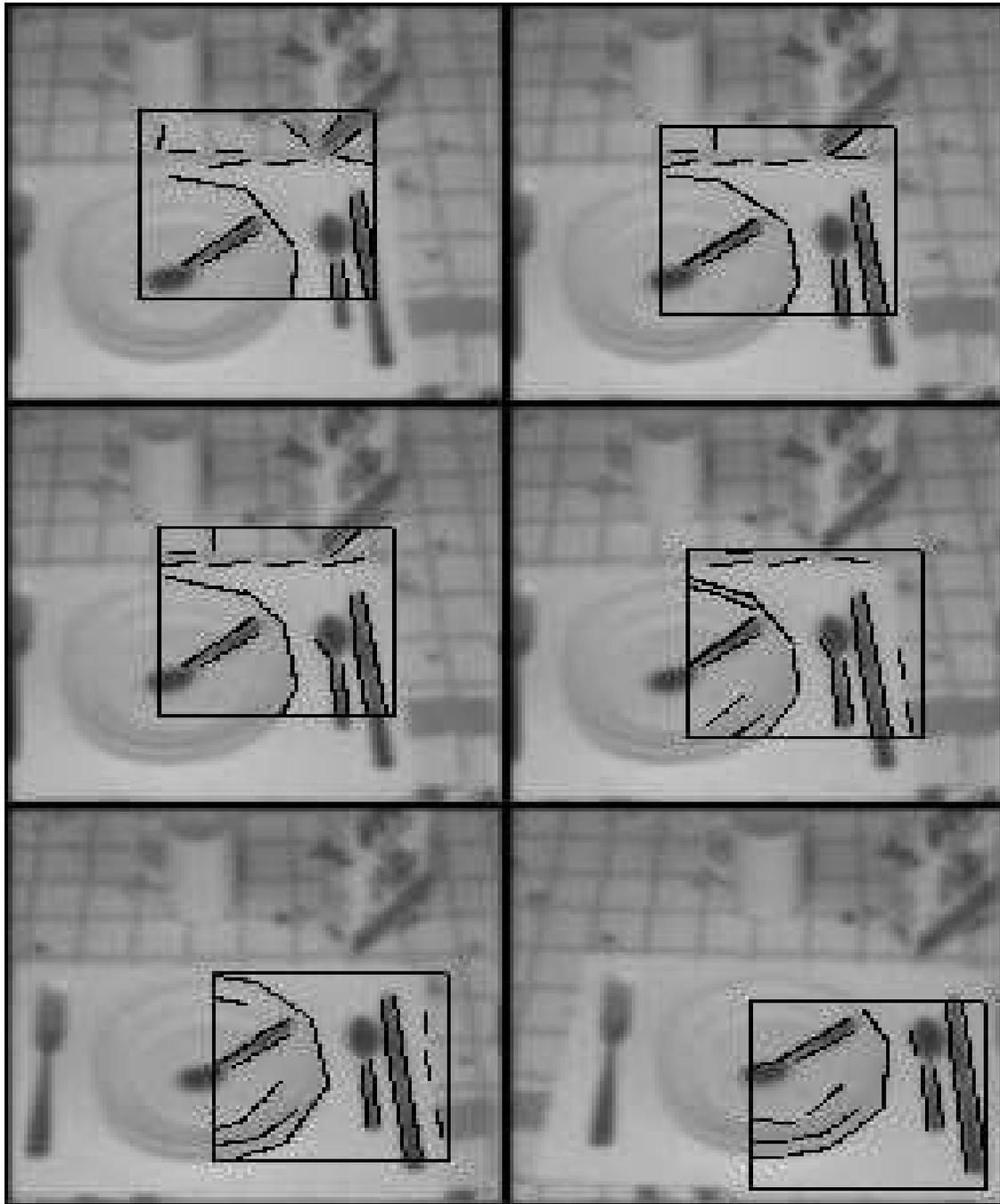


FIG. 6.7 - **Expériences de contrôle au sein de l'AGENT-DESCRIPTION-2D.** Modification de la région d'intérêt en cours de traitement d'une séquence d'image. Les images correspondent, en partant de la gauche, en haut et en allant vers la droite en descendant, aux instants 0, 1, 2, 4, 7 et 12. Lorsque la distance du centre de gravité avec le centre de la région d'intérêt est supérieure à 4 pixels celle-ci est déplacée (image 1, 4, 7, 12)

### 6.3.2 Contrôle

Ayant présenté l'architecture de cet agent, nous allons donner un exemple de réalisation utilisant les mécanismes de traitement inscrits sur les différentes couches de contrôle (cf. figure 6.7).

#### Traitements sur la couche décision

1. Fonction d'évaluation : La fonction d'évaluation inscrit le but dont le type est **FIND SEGMENT** avec la valeur *integre* dans la base de faits.
2. Fonction de raisonnement : Une intention relative à une stratégie propre à la tâche FIND est créée. La valeur de son *statut* est *possible*.
3. Fonction de décision : La fonction de décision choisit cette intention<sup>5</sup>.
4. Fonction d'engagement : La fonction d'engagement évaluant la partie *action* de cette stratégie dans le contexte d'activation, crée deux ENGAGEMENTS sur la couche *adaptation*. Ces engagements sont de deux types **ROI** et **MOI**:

– ENGAGEMENT de *type ROI*: définition de la Région d'intérêt dans laquelle effectuer le traitement.

– ENGAGEMENT

- *type: ROI*
- *condition-terminaison*: satisfaction du but.
- *condition-validite*

(inp (engagement (couche adaptation) (type ROI) (statut choisi))  
(engagement (couche decision) (type ROI) (statut choisi)))

exprimant que la ROI définie par la couche adaptation doit rester incluse dans la ROI définie par la couche décision.

Création du CRITERE rattaché à cet engagement :

– CRITERE

- *type: focus*
- *valeur*: (0, 0, 100, 100), coordonnées du centre de la région et demi-longueurs en x et y des côtés.

– ENGAGEMENT de *type MOI*:

– ENGAGEMENT

- *condition-validite*: (qualite SEGMENT > 10), critère de qualité minimal relatif aux hypothèses devant satisfaire l'expression du modèle.

---

<sup>5</sup>Nous n'avons pas encore actuellement défini de critères de contrôle permettant de sélectionner les différents buts que doit satisfaire l'agent. La procédure actuelle consiste à les satisfaire les uns après les autres

- *condition-terminaison*: (qualite SEGMENT > 50), critère de qualité minimal relatif aux hypothèses devant satisfaire l'expression du modèle.

Le CRITERE défini est :

- CRITERE
  - *type*: *focus*.
  - *valeur*: **FIND SEGMENT** 50. où 50 exprime la qualité des segments recherchés.

Un ENGAGEMENT ayant été défini sur la couche adaptation, le cycle de contrôle active la couche adaptation.

### Traitements sur la couche adaptation

1. Fonction d'évaluation : Exécution de la fonction évaluation sur les éléments de l'état de raisonnement
2. Fonction de raisonnement : Examen des tactiques possibles étant donné l'état de raisonnement et les engagements définis
3. Fonction de décision : Choix de l'INTENTION dont l'attribut *ressource* indique que les éléments fournis sont de type **FIND SEGMENT** (correspondant à la valeur du CRITERE défini par la couche supérieure). Cette tactique définit les critères de contrôle pour la couche commande.
4. Fonction d'engagement : Définition des ENGAGEMENTS. La tactique choisie définit les engagements suivants :

- ENGAGEMENT de *type* **MOI**,
  - ENGAGEMENT
    - *type*: **MOI**
    - *condition-terminaison*: satisfaction du but.
    - *condition-validite*: (qualite SEGMENT > 30). Cette condition de validité est supérieure à celle inscrite dans l'ENGAGEMENT provenant de la couche supérieure dans la mesure où les tactiques de la couche adaptation recherchent à s'approcher de la qualité en modifiant les paramètres des procédures d'extraction des segments.

Le CRITERE créé est le suivant :

- CRITERE
  - *type*: *focus*,
  - *valeur*: **FIND SEGMENT** 10 15 4 où 10 est la valeur du seuil-bas, 15 la valeur du seuil-haut pour le seuillage par hysteresis, 4 la longueur minimale d'extraction. Ces valeurs ont été obtenues par

traduction du seuil de qualité exprimé dans le critère focus issu de la couche supérieure.

- ENGAGEMENT de *type* ROI,
  - ENGAGEMENT
    - *type*: ROI
    - *condition-validite*: centre de gravité des segments présents dans la ROI.
    - *condition-terminaison*: but satisfait.

Le CRITERE créé est :

- CRITERE
  - *type*: *focus*
  - *valeur*: (0, 0, 3, 30, 30, 1) Expression de la ROI en terme d'une région de l'image (centrée en (0,0) de demi-longueur en x et y de 30, située le 3ème niveau de la pyramide, et prenant en compte 1 seul niveau).

Un ENGAGEMENT ayant été créé, et les conditions de validité de l'engagement de la couche décision étant vérifiées, passage de la couche adaptation à la couche commande.

### Traitements sur la couche commande

1. Fonction d'évaluation : Application de la fonction d'évaluation sur l'état de raisonnement.
2. Fonction de raisonnement : Examen des compétences activables sur l'état de raisonnement. Dans celui-ci une image étant disponible, la compétence de transformation relative à l'extraction de segments est activée.
3. Fonction de décision : Sélection des intentions par rapport au modèle qu'elles produisent, c-à-d. segment ou l'un des groupements. Le focus exprimant une valeur de type **FIND SEGMENT**, sélection de l'intention d'extraction.
4. Fonction d'engagement : Les segments sont extraits dans l'image dans la région d'intérêt définie par la couche adaptation. Les segments produits se situent ainsi sur le 3ème niveau de la pyramide multi-résolution. La procédure d'extraction utilise les paramètres définis dans le critère relatif à l'ENGAGEMENT dont le *type* est **MOI**.
5. Fonction bascule : Application de la bascule sur les segments extraits. Ceux-ci ont leur centre de gravité situé en dehors du centre de la région, la condition de validité étant insatisfaisante, les segments en cause sont reliés à l'ENGAGEMENT de *type* **ROI** par le lien *invalide*. Passage à la couche supérieure.

Sur la couche adaptation, la *tactique-centre-gravité* est exécutée. Elle redéfinit un nouveau CRITERE : nouvelle région calculée à partir du centre de gravité des segments. Si la nouvelle région est toujours dans la région définie par la couche décision (conditionnalité de l'ENGAGEMENT situé sur la couche adaptation, dont l'attribut *type* a pour valeur **ROI**), un engagement ayant été défini, les traitements de la couche commande sont réactivés.

Les traitements continuent ainsi à osciller entre les couches adaptation et commande. Les traitements s'arrêteront une fois que la qualité requise pour les segments sera obtenue.

### 6.3.3 Evaluation

Cette formulation du module de description 2D réalise les traitements fondamentaux existants dans le cadre de nos premières expérimentations dans le cadre de VAP : construction de la description en termes de segments à partir de l'image fournie (cf. chapitre 4). La formulation de cet agent dans le module standard a permis :

- la structuration des informations et des modes de fonctionnement selon les couches de contrôle.
- d'accroissement des compétences de cet agent en lui ajoutant les procédures de groupement perceptuel. Dans la précédente formulation, ces traitements n'étaient utilisés que pour l'accès à la description dans le cadre de la fonction de communication. L'introduction de ces traitements au sein de l'agent a fait apparaître l'obligation d'un choix entre les traitements.
- l'introduction de tactiques et de stratégies afin que cet agent puisse lui-même définir ses engagements. Ces tactiques et stratégies permettent, d'une part l'expression de connaissances sur le réglage des différents paramètres des procédures d'extraction, de groupement et, d'autre part le réglage de la région d'intérêt dans laquelle travailler.

Le fonctionnement du module standard permet ainsi :

- de modifier les critères de contrôle indépendamment les uns des autres.
- d'améliorer au sein de l'agent, les paramètres des procédures de segmentation, de groupements, de suivi

Nous avons essentiellement testé les critères de contrôle relatifs à la *sélection*, et à la *validité*. Nous devons poursuivre nos recherches sur l'utilisation des autres critères. L'intérêt du critère d'incohérence apparaît ici nettement dans la mesure où la fonction d'évaluation effectue un suivi d'indice. Celui-ci consistant en l'application d'un filtre de kalman pour la prédiction des nouvelles valeurs des indices, et en l'utilisation de la distance de Mahalanobis pour la mise en correspondance, peut voir son fonctionnement modifié par ce critère en modifiant les paramètres contrôlant ces deux fonctions.

## 6.4 Contrôle dans l'AGENT-INTERPRETATION

La reformulation de cet agent dans le Module Standard d'Agent a pour objectif de tester la capacité de ce Module à exprimer d'autres critères de sélection au sein d'un agent ayant a priori beaucoup plus de compétences que celui de l'AGENT-DESCRIPTION-2D. Notre objectif est donc :

- d'explicitier les couches de contrôle.
- de montrer comment un fonctionnement de suivi (utilisation prioritaire d'une couche de contrôle sans la remise en cause des engagements qui le contrôlent) peut être mis en place dans cet agent tout en inscrivant un fonctionnement de résolution (création et tests d'hypothèses permettant de modifier dynamiquement les critères de sélection, d'où un changement continu entre les couches de contrôle). En montrant l'intégration de ces deux types de fonctionnement au sein du module standard, nous montrons ainsi qu'un fonctionnement répétitif peut être combiné avec un fonctionnement de recherche.
- d'étudier la dynamique de passage d'une couche à l'autre.

### 6.4.1 AGENT-INTEPRETATION

Cet agent est constitué de la reformulation d'une partie des éléments exprimés dans le cadre de nos expériences dans VAP et d'expériences réalisées dans [Tho 92]. Il construit sa description à partir des segments et groupements fournis par l'AGENT-DESCRIPTION-2D. Ceux-ci sont indexés par les niveaux de la pyramide multi-resolution utilisée dans l'agent description 2D. Dans le cadre de nos expériences, nous avons fixé ce niveau au troisième niveau de la pyramide. La description construite par cet agent vise à répondre aux buts communiqués par le SUPERVISEUR. La description de la scène s'exprime en termes de formes géométriques simples.

- couche décision : sélection des buts à satisfaire, traduits en ENGAGEMENT pour la couche adaptation.
- couche adaptation : construction et sélection des plans d'interprétation. Ceux-ci sont traduits en ENGAGEMENT pour la couche commande.
- couche commande : choix de compétences d'interprétation. Celles-ci construisent les hypothèses sur la description de la scène en fonction des prédictions réalisées sur les formes présentes.

Nous explicitons dans la section suivante les éléments inscrits dans l'état de raisonnement ainsi que les connaissances de cet agent. L'état de décision n'étant pas, a priori, dépendant de l'application, la structure de ses éléments n'est pas modifiée. L'expression des ENGAGEMENTS n'est pas modifiée. Seul leur contenu varie.

### a. Etat de raisonnement

Les éléments constituant l'état de raisonnement sont la description de la scène en construction. Celle-ci s'exprime en terme de segments, groupements et formes géométriques simples. Les types d'HYPOTHESES suivants la représente :

- les segments sont représentés par des HYPOTHESE dont les attributs sont :
  - HYPOTHESE
    - *type*: **SEGMENT**
    - *valeur*: *niveau*, *id*, *x1*, *x2*, *x3*, *x4*, où *niveau* est le niveau de la pyramide multi-résolution, *id* est l'identificateur du segment dans la description de l'AGENT-DESCRIPTION-2D, *x1*, *y1*, *x2*, *y2* sont les coordonnées cartésiennes de ses extrémités. Cette représentation est moins lourde que dans l'AGENT-DESCRIPTION-2D dans la mesure où les segments lui sont communiqués par celui-ci.
- les groupements sont représentés par le même type d'HYPOTHESE que dans l'AGENT-DESCRIPTION-2D et par les mêmes attributs.
- les formes géométriques simples que nous considérons sont représentées par des HYPOTHESE dont les attributs sont les suivants :
  - HYPOTHESE
    - *type*: **AJ2** pour des alignements de connexion, **1/2PLAN** pour les demi-plans, **PLAN** pour les plans, **3/4PLAN** pour les trois quart de plan, **CUBE** pour les cubes, etc.
    - *valeur*: *niveau*, *ids*, *Q*  
où *niveau* est le niveau de la pyramide considéré, *ids* la liste des identificateurs des segments prenant part à cette forme, *Q* qualité.

Les BUTS sont similaires à ceux de l'agent précédent. Seuls les modèles utilisables pour compléter la spécification de la tâche dans l'attribut *type* sont modifiés dans la mesure où cet agent travaille sur un autre niveau de représentation. Ainsi l'attribut *type* de ces buts est par exemple : **FIND CUBE**, **FIND CYLINDRE**, ...

Cet agent manipule des PLANS d'interprétation. Nous les avons exprimés comme des arbres ET/OU. Un nœud de l'arbre est exprimé par la structure ETAPE. Le lien *a-etape* relie ce plan à la première étape l'implantant. La structure ETAPE représente une étape avec les attributs suivants :

- ETAPE
  - *type*: type du plan auquel elle fait référence
  - *valeur*: ensemble de champs dont les valeurs dépendent de la valeur du premier champ. Si celui-ci est **NŒUD** (l'étape est un nœud du plan) spécification d'une action et des éléments nécessaires pour la traduction en

ENGAGEMENTS. Dans le cas, où la première valeur est **ET** ou **OU**, les autres champs sont nuls. L'ETAPE est un nœud de branchement.

- *crec-res* expression des types des hypothèses fournies.
- *res-nec* expression des types des hypothèses nécessaires fournies par l'exécution de cette étape.

Chacune des étapes est reliée l'une à l'autre par les liens de précédence et de succession.

## b. Connaissances

### Compétences

Les compétences relatives aux groupements perceptuels primitifs (alignement, connexions, ...) ont une action faisant appel à des procédures écrites en C. Les compétences créant des formes manipulent directement les HYPOTHESES et font appel aux fonctions de création, modification et ajout de faits de CLIPS. Différentes compétences d'enrichissement ont été implantées : recherche de groupements, recherche de formes, etc. Elles sont de la forme suivante :

– CONNAISSANCE

- *couche*: **commande**
- *module*: **individu**
- *condition*: test de la présence d'un groupement particulier
- *action*: ensemble de règles pour réaliser le groupement de ces groupements.

### Tactiques

Les tactiques définissent un plan d'interprétation à partir des buts générés. Ces plans d'interprétation sont ensuite traduits en ENGAGEMENTS. Au sein de ces tactiques, nous avons réalisé une tactique (*tactique-cube*) pour la reconnaissance d'un cube dans une description constituée de segments. Son expression est la suivante :

– CONNAISSANCE

- *couche*: **adaptation**
- *module*: **individu**
- *condition*: présence de BUT dont la valeur du *type* est **FIND CUBE**,
- *action*: définition du plan de reconnaissance de ce cube dans la scène : (SEGMENT (AND ALIGNED JUNC PARALLEL) AJ2C (AND 1/2PLAN 3/4PLAN) PLAN JOIN-PLAN CUBE)

La signification de ce plan est la suivante : rechercher les segments, rechercher les groupements alignés, connexions, parallèles, rechercher les connexions alignées en forme de U, rechercher les demi-plans et trois-quart-plans, les plans, les plans se joignant, le cube.

## Stratégies

Tout comme dans l'AGENT-DESCRIPTION-2D, les stratégies utilisées sont distinguées selon le type de tâche qu'elles concernent. Dans le cadre de l'exemple ci-dessous, nous utilisons la stratégie suivante :

– CONNAISSANCE

- *couche*: *décision*
- *module*: *individu*
- *condition*: si BUT a *FIND CUBE* pour valeur de l'attribut *type*
- *action*: créer un engagement de type *MOI* dont la valeur du critère focus est *FIND CUBE*. créer un engagement de type *ROI* dont la valeur du critère focus est la région d'intérêt par défaut.

### 6.4.2 Contrôle

De manière similaire à ce que nous avons fait pour l'AGENT-DESCRIPTION-2D nous allons présenter un exemple de réalisation utilisant les mécanismes de traitement sur les différentes couches.

#### Traitements sur la couche décision

1. Fonction d'évaluation : Inscription du but dont le type est *FIND CUBE* avec la valeur *integre* dans la base de faits.
2. Fonction de raisonnement : Intention relative à la stratégie propre à la tâche FIND est créée. La valeur de son *statut* est *possible*.
3. Fonction de décision : Sélection de l'intention.
4. Fonction d'engagement : Evaluation de la partie *action* de la stratégie, création de deux ENGAGEMENTS sur la couche *adaptation*. Comme dans l'exemple présenté ci-dessus, ces engagements sont de *type ROI* et *MOI*. La valeur du critère correspondant à l'ENGAGEMENT de *type MOI*, est : *FIND CUBE*.

#### Traitements sur la couche adaptation

1. Fonction d'évaluation : Evaluation de l'état de raisonnement
2. Fonction de raisonnement : Examen des tactiques possibles étant donné l'état de raisonnement et les engagements. Un but de type *FIND CUBE* étant inscrit dans l'état de raisonnement, la tactique-cube est possible.
3. Fonction de décision : Le focus étant *FIND CUBE*, sélection de l'intention possible.

4. Fonction d'engagement : A partir de l'exécution de la partie action de la tactique-cube, création d'un plan constitué des étapes relatives aux formes devant être présentes pour la construction d'un cube. Ce plan est traduit en un ensemble d'ENGAGEMENT reliés les uns aux autres. Ainsi un engagement relatif à ce plan s'exprime :

– ENGAGEMENT

- *type*: **MOI**
- *condition-terminaison*: trouvé OVERLAPPED (des hypothèses de segment se recouvrant existent)
- *condition-validité*: des hypothèses de cylindre ne sont pas présentes dans la même région.

Le lien *prochain* le relie à l'engagement de type **AJ2C**. Le lien *precedent* le relie à l'engagement de *type* **SEGMENT**.

### Traitements sur la couche commande

1. Fonction d'évaluation : Evaluation de l'état de raisonnement
2. Fonction de raisonnement : Examen des parties *condition* des compétences sur l'état de raisonnement.
3. Fonction de décision : Sélection des intentions. Le critère de sélection est inscrit dans l'ENGAGEMENT de type **MOI** de cette couche. Ainsi la première intention sélectionnée sera relative à la recherche de segments dans la description.
4. Fonction d'engagement : La fonction d'engagement exécute les compétences choisies et évalue les bascules sur l'état de raisonnement. Une fois les engagements définis, passage à la couche commande : Examen des conditions de satisfaction des engagements face à l'état de raisonnement. Selon le résultat de cette activation, l'ENGAGEMENT courant est *traite* ou non.

Si les engagements restent valides, et non-satisfait, le plan relatif au cube peut donc s'exécuter en séquence et se répéter. L'agent travaille uniquement sur la couche commande. Afin de concilier un fonctionnement dirigé par les buts et par les données, une condition de validité peut être rajoutée aux engagements agissant sur la couche commande dans lequel l'apparition d'un nouveau but dans l'état de raisonnement conduit à une invalidation des engagements courants.

Dans le cadre d'un fonctionnement de suivi, nous pouvons retrouver le même type de fonctionnement de l'agent exécutant le plan de manière continue sur la couche commande à chaque fois que de nouveaux segments proviennent de l'AGENT-DESCRIPTION-2D. La remise en cause du plan s'exprimant alors par de nouvelles conditions de validités du type disparition de segment, baisse de la qualité des groupements...

### 6.4.3 Evaluation

La formulation de cet agent au sein du module standard montre la possibilité d'exprimer des méthodes de raisonnement plus élaborées avec notamment l'expression de plans d'interprétation. Comme dans l'agent précédent, la structuration selon les couches de contrôle met en lumière les différentes connaissances de contrôle à utiliser. Le mécanisme de bascule rend possible la définition des conditions de validité d'un critère de contrôle et de concilier un fonctionnement répétitif avec un fonctionnement de recherche.

La formulation de cet agent est encore à enrichir notamment en terme de tactiques utilisées au sein de la couche adaptation, définissant l'exploration des modèles. L'expression de la description de l'agent par rapport aux niveaux de la pyramide manipulés par l'AGENT-DESCRIPTION-2D, limite l'indépendance de ces deux agents. Ainsi, une évolution future consistera à reformuler la description de l'AGENT-DESCRIPTION-2D comme un ensemble de segments sans faire apparaître les niveaux de la pyramide. La gestion de la pyramide Multi-résolution pourra ainsi être faite plus naturellement uniquement par l'AGENT-DESCRIPTION-2D.

## 6.5 Contrôle Social dans un système intégré de vision : VAP

Dans cette section, nous reformulons le contrôle trouvé dans VAP entre chacun des modules, dans le cadre du modèle de société développé au chapitre précédent. Pour ce faire, nous utilisons le *module-organisation*, de la *module-dialogue* et les protocoles et organisations définis. Nous utilisons aussi les fonctions *bascule-social* et *bascule-individu* présentées dans le Module Standard d'Agent. La société est constituée de l'AGENT-DESCRIPTION-2D, de l'AGENT-INTERPRETATION, de la CAMERA et du SUPERVISEUR.

### 6.5.1 Capacités sociales

Nousinstancions les capacités sociales présentées dans le Module Standard au domaine d'application.

#### a. Accointances

Actuellement, chaque agent a une représentation des autres et de soi, établit a priori au lancement du système. Notre objectif à plus ou moins long terme est d'établir un protocole de présentation entre eux afin que ces structures soient définies explicitement et dynamiquement [Ber 92]. Les accointances expriment l'ensemble des types de modèles que l'agent peut traiter localement. Au sein de l'AGENT-DESCRIPTION-2D, étant

donnée la société définie ci-dessus, les accointances sont :

- ACCOINTANCE
  - *agent-id*: **agent-camera-1**
  - *role*: **IMAGE**
  - *actions-base*: **IMAGE**

L'ACCOINTANCE relative à la description externe de l'agent lui-même est :

- ACCOINTANCE
  - *agent-id*: **mine** L'accointance représentant l'AGENT-DESCRIPTION-2D au sein de l'AGENT-INTERPRETATION est similaire à celle-ci. Seule la valeur de l'attribut *agent-id* est modifiée. Cet attribut reçoit l'identificateur de l'AGENT-DESCRIPTION-2D.
  - *role*: **DESCRIPTION-2D**
  - *actions-base*: **SEGMENT JUNC PROTO PARALLEL, ...** Les valeurs de l'attribut *actions-base* expriment le type de modèle que cet agent peut fournir.

L'accointance représentant l'AGENT-INTERPRETATION au sein de l'AGENT-DESCRIPTION-2D est la suivante :

- ACCOINTANCE
  - *agent-id*: **agent-interpretation-3** (identificateur de l'AGENT-INTERPRETATION définie au lancement du système)
  - *role*: **INTERPRETATION**
  - *actions-base*: **CUBE, AJ2C, ...**

Ces accointances sont utilisées par la fonction *exécution-dialogue* pour réaliser le choix d'agent à partir des intentions relatives à des actions de type communication exprimées par le module-individu. La recherche des agents potentiels est réalisée par mise en correspondance des valeurs de modèles exprimées dans l'attribut *ressource* de l'INTENTION avec les valeurs inscrites dans l'attribut *actions-base* des ACCOINTANCES.

## b. Langage de représentation des connaissances

A partir de l'application à un système de vision, nous définissons le sujet de l'interaction ou celui relatif à la fondation de l'organisation comme étant composé de quatre champs.

$\langle \text{sujet} \rangle ::= \langle \text{niveau} \rangle \langle \text{action} \rangle \langle \text{classe} \rangle \langle \text{type} \rangle$

	valeurs possibles
<b>niveau</b>	null, image, desc_2D, desc_3D, interpretation
<b>action</b>	find, track, identify, explore, watch, describe, relate, verify
<b>classe</b>	nulle, ext_desc, MOI, ROI, POI
<b>type</b>	spécifique aux agents.

TAB. 6.2 - Valeurs permises pour le sujet.

Les termes utilisés pour l'expression du sujet sont aussi utilisés pour l'expression du *contenu* des interactions.

- **niveau** (cf. table 6.2) spécifie le niveau de représentation concerné par le sujet. *nul* désigne les sujets concernant tous les agents. Un sujet relatif à un **niveau** n'est accessible qu'aux agents travaillant sur ce niveau.
- **action** (cf. table 6.2): tâches entrant dans la définition des buts échangeables entre les agents.
- **classe** structure les types d'échanges. Une fois celle-ci définie le *type* vient particulariser l'échange. Les classes (cf. table 6.2) définissant le sujet sont les suivantes :
  - nulle : pas de classe
  - ext\_desc : informations relatives à la constitution des accointances au sein de l'agent.
  - MOI : modèle.
  - ROI : région d'intérêt.
  - POI : période temporelle d'intérêt.
- **type** : valeurs spécifiques à chacun des agents qui ne rentrent pas en compte à priori dans la spécification des protocoles et des organisations. Les valeurs correspondent aux différents modèles.

## Protocole

Les protocoles sont définis en utilisant les termes du langage de communication et les valeurs des *niveau*, *action* et *classe* entrant dans la définition du sujet. Ces derniers contraignent le contenu des échanges entre les agents. Nous avons représenté dans la figure 6.8 le protocole *recognize* utilisé dans le cadre de nos expériences pour gérer les interactions entre les agents. Ce protocole est représenté sous forme d'un réseau de transition.

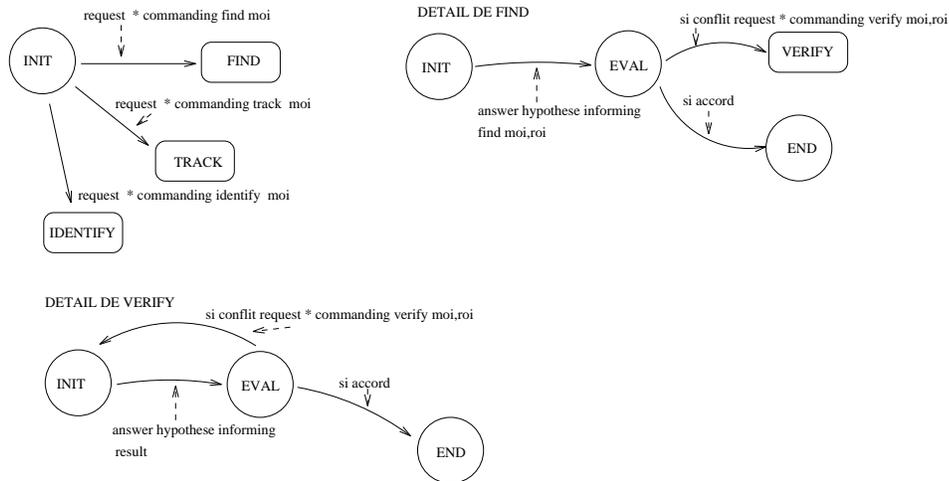


FIG. 6.8 - **Protocole d'interaction descendant.** Présentation de quelques états et transitions du protocole d'interaction. Les transitions sont inscrites sur les arcs reliant les états de conversation. Une interaction est une suite de valeurs relatives au type, nature, force, sujet. Une \* signifie que les valeurs s'inscrivant dans le champ de l'interaction ne sont pas limitées par le protocole. A l'initialisation de ce protocole selon l'action exprimée dans l'interaction, la conversation peut donner lieu à un sous ensemble d'interactions **FIND**, **IDENTIFY**, etc. Si une interaction dont l'action est **find** est envoyée, la conversation se déroule par un échange de request et d'answer qui peuvent conduire à l'utilisation du sous ensemble de transitions **VERIFY**.

### c. Organisation

L'organisation est exprimée par les engagements sociaux dont nous avons donné des exemples dans le cadre du module standard. L'expression du sujet entrant dans la définition du contenu de l'attribut *valeur* des engagements de type *role-resolution* ou *role-dialogue* sont définis à partir de l'expression précédente : *niveau action classe*.

Les rôles de résolution établissent par exemple (cf. exemple 6.9, 6.10) un lien d'interaction possible dont l'agent est la cible. Ce lien débouche sur la couche adaptation (nature *plan*) dans l'état *DS.choix* (force *commanding*) pour un sujet relatif à *interpretation* et *ROI*. Les rôles de dialogue incluent eux aussi l'expression du sujet (cf.

```
(engagement
 (module organisation) (couche commande)
 (type role-resolution) (statut choisi) (id eng-89) (cycle 12)
 (valeur social plan commanding interpretation * ROI))
```

Exemple 6.9: **Exemple de rôle de résolution social.** Dans cette expression le rôle exprime le fait que l'agent peut être contrôlé par tout agent, *social*, travaillant sur le niveau de représentation *interpretation*. Cette interaction de contrôle possible est *directive* et adresse un échange de région d'intérêt sur la couche *adaptation* pour n'importe quelle action (\*).

```
(engagement
 (module organisation) (couche commande)
 (type role-resolution) (statut choisi) (id eng-34) (cycle 13)
 (valeur individu plan informing interpretation * ROI))
```

Exemple 6.10: **Exemple de rôle de résolution individu** Dans cette expression le rôle exprime le fait que l'agent (*individu*) ne peut pas modifier la ROI, sur la couche adaptation. L'interaction doit être envoyée au niveau de représentation supérieur.

exemple 6.11)

#### 6.5.2 Exemple de contrôle social

L'expérience que nous avons réalisée met en place une organisation hiérarchique entre les agents : toute interaction en provenance du niveau de représentation supérieur est acceptée. Elle est exprimée au travers des rôles de résolution précédents. Le protocole utilisé est un protocole d'interaction descendant (*recognize*) illustré par la figure 6.8.

---

```
(engagement
  (module dialogue) (couche commande)
  (type dialogue-role) (statut choisi) (id eng-5) (cycle 1)
  (valeur recognize interpretation identify ROI))
```

Exemple 6.11: **Exemple de rôle de dialogue.** RECOGNIZE est le nom du protocole que l'agent peut utiliser pour des interactions avec les agents de niveau *interprétation* dont le sujet est *identify ROI*

---

Nous illustrons le type de contrôle pouvant se mettre en place dans la société par l'exemple suivant :

L'AGENT-INTERPRETATION décrit précédemment, reçoit le message suivant de la part du SUPERVISEUR :

- DONNEE
  - type : request
  - lien : BUT COMMANDING
  - sujet : INTERPRETATION FIND MOI CUBE

BUT est la nature, COMMANDING est la force, INTERPRETATION FIND MOI CUBE est le sujet de l'interaction. Le niveau de représentation utilisé est INTERPRETATION, l'action est FIND, la classe est MOI. Cette valeur est instanciée par le symbole CUBE.

Nous détaillons successivement les traitements se déroulant au sein de chacun des agents. Les traitements individuels ne sont pas détaillés dans la mesure où ils ont été explicités dans les sections relatives au contrôle.

### Traitements au sein de l'AGENT-INTERPRETATION

Le déroulement suivant se passe au sein de cet agent (cf. figure 6.9 étape 1) :

#### 1. Récepteur :

- création d'un engagement de dialogue :
  - ENGAGEMENT
    - *module dialogue*
    - *statut choisi*
    - *valeur* social BUT COMMANDING INTERPRETATION FIND MOI CUBE

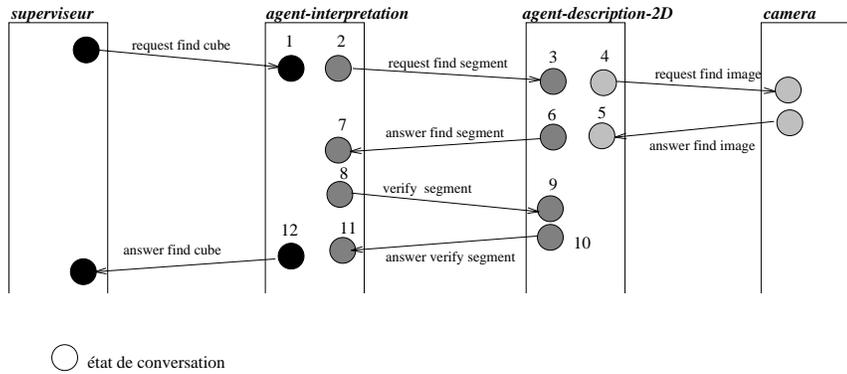


FIG. 6.9 - Exemple de contrôle social

- création de la CONVERSATION puisque cet agent n'a pas encore eu de conversation avec le superviseur à ce sujet. Elle est reliée à cet ENGAGEMENT.
- CONVERSATION
  - *id sup-10*, valeur de l'identificateur exprimé dans le contenu de l'interaction
  - *etat-conversation recognize find init*, noms de la loi, du protocole et de l'état de conversation dans lequel le superviseur est entré.
  - *agent-id* agent-superviseur-1
- Création d'une DONNEE contenant la spécification du but.

## 2. Interprétation-dialogue

- Traduction de la donnée et création d'un but de *statut* de valeur *non-integre*. Un lien *realise* relie le BUT et l'ENGAGEMENT de dialogue.

## 3. Bascule-social

- Comparaison de la force (COMMANDING), de la nature (BUT) et du sujet de l'ENGAGEMENT de dialogue avec les ENGAGEMENT sociaux exprimant les rôles de résolution. L'organisation permet un tel lien d'interaction pour ce sujet.
- création d'une INTENTION choisie par rapport au but, dans la mesure où celui-ci est émis avec une force *commanding* au sein du module-individu.
- ENGAGEMENT.*statut* = *attente*. Le traitement de cet engagement sera poursuivi lorsque le but qu'il a généré dans l'agent sera satisfait.
- Passage au module-individu.

## 4. Module-individu

- Une INTENTION relative à un but ayant été créé, activation de la couche décision. Création des ENGAGEMENT par rapport au but.

La suite de la résolution au sein de l'agent individu, se déroule selon l'exemple présenté dans la section précédente sur le contrôle au sein de l'AGENT-INTERPRETATION. Le module-individu a choisi l'INTENTION relative à une action de communication du type FIND SEGMENT.

## 5. Bascule-individu : création d'un ENGAGEMENT de dialogue en attente :

- ENGAGEMENT
  - *module*: *dialogue*
  - *type*: non spécifié
  - *valeur*: *individu PLAN COMMANDING DESC\_2D FIND MOI ROI SEGMENT* Le lien d'interaction pour ce sujet permet des interactions sur la couche adaptation.

## 6. Exécution-dialogue

- consultation des diverses accointances pour connaître les agents ayant dans leurs actions de base le modèle SEGMENT.
- choix de l'agent *agent-description-2D-1*.
- Aucune conversation par rapport à ce sujet et cet agent n'étant en attente, création d'une CONVERSATION (cf. figure 6.9 étape 2). La valeur du contenu de l'*etat de conversation* est *recognize init init*.
- Consultation du protocole d'interaction *recognize*. l'action exprimée par l'engagement étant *find* l'interaction possible est de type *request* et doit être relative à un sujet *find moi*. Le type de l'engagement est ainsi inscrit dans son attribut *type*.
- Création d'une DONNÉE constituée des valeurs spécifiant la ROI et les paramètres pour les segments.

## 7. Emetteur

- émission
  - type: request
  - lien: PLAN COMMANDING
  - sujet: DESC\_2D FIND MOI ROI SEGMENT
  - dest: agent-description-2d-1
  - contenu: 0 0 3 50 50 1 50, spécification de la région d'intérêt (0 0 3 50 50 1) et spécification de la qualité des segments (50).
- ENGAGEMENT.*statut* = *attente*

L'AGENT-INTERPRETATION continue ensuite ses traitements. Une fois que la réponse à cet engagement sera parvenu, il reprendra le traitement de la conversation.

### Traitements au sein de l'AGENT-DESCRIPTION-2D

Pour la suite des descriptions, nous ne faisons apparaître que les traitements nouveaux par rapport à la description du fonctionnement des interactions entre les agents.

1. Module-dialogue : Réception du message émis par l'AGENT-INTERPRETATION, création d'un ENGAGEMENT correspondant à cette interaction, création d'une CONVERSATION (cf. figure 6.9 étape 3) dont le contenu est déduit de l'interaction, création d'une DONNEE, interprétation de la DONNEE conduisant à la création d'un PLAN de type **ROI** dont la valeur est **0 0 3 50 50 1** et d'un plan de type **FIND SEGMENT** dont la valeur est **50**.
2. Bascule-social : Création des INTENTION relatives à ces deux éléments en marquant leur statut à **choisi**. Mise en attente de l'ENGAGEMENT. Passage au module-individu.
3. Module-individu : Recherche des segments pour l'agent. Le traitement est similaire à celui décrit dans le cadre du contrôle dans l'AGENT-DESCRIPTION-2D à la différence que le traitement est initialisé sur la couche adaptation. La recherche des segments peut donner lieu elle aussi à des interactions avec la caméra en vue de recevoir des images (cf. figure 6.9 étapes 4 et 5). Une fois les segments trouvés dans la région, le plan créé à la suite de l'engagement étant satisfait, activation du module dialogue.
4. Module-dialogue : Le plan lié à l'engagement en attente étant satisfait, la fonction d'exécution consulte le protocole à partir de l'état inscrit dans l'attribut *etat-conversation* de la CONVERSATION reliée à l'ENGAGEMENT. (cf. figure 6.9 étape 6) Émission du résultat en direction de l'AGENT-INTERPRETATION :
  - type : answer
  - dest : agent-interpretation-1
  - lien : HYPOTHESE INFORMING
  - sujet : DESC\_2D FIND MOI ROI SEGMENT
  - contenu : valeurs des segments

### Traitements au sein de l'AGENT-INTERPRETATION

1. Module-dialogue :
  - réception des segments trouvés.
  - mise à jour de l'état de conversation (cf. figure 6.9 étape 7),

2. Bascule-social : Inscription des hypothèses de segment avec un statut *non-integre*, mise en attente de l'ENGAGEMENT issu de l'interaction.
3. Module-individu :
  - évaluation des hypothèses pour détecter les incohérences possibles. Au cas où un conflit est créé conduisant à une interaction, la conversation peut être poursuivie avec l'AGENT DESCRIPTION-2D, (activation du sous ensemble d'interaction verify dans le protocole recognize). (cf. figure 6.9 étapes 8, 9, 10, 11)
  - Poursuite du traitement, satisfaction du but (lien d'hypothèses vers le but) : création d'une HYPOTHESE dont le *type* est CUBE ROI (ROI sont les coordonnées de l'hypothèse).
  - action de communication déclenchée créant une INTENTION de communication relative à : Hypothese CUBE ROI.
4. Module-dialogue : Consultation du protocole pour les réponses possibles par rapport à l'ENGAGEMENT en provenance du SUPERVISEUR qui était en attente. Choix d'une réponse, mise à jour de l'état de conversation de cet ENGAGEMENT.

### traitements au sein de l'agent *SUPERVISEUR*

L'agent SUPERVISEUR reçoit le but. Il peut évaluer localement la réponse et poursuivre la conversation au cas où un conflit est détecté. Dans le cas contraire il l'arrête.

#### 6.5.3 Evaluation

Le déroulement de ce protocole illustre les mécanismes d'interaction entre les agents. Ceux-ci s'ajoutent à l'expression de l'organisation. Du fait des modèles élaborés et de l'implantation réalisé, le module-individu n'est pas concerné par la modification des protocoles et des organisations. Notre travail se démarque ainsi de notre premier cadre d'expérimentation dans la mesure où le module standard de VAP ne faisait aucune distinction au sein des agents entre actions de communication et action de résolution. La stratégie d'interaction était liée à la stratégie de résolution au sein de l'agent.

Les fonctions *bascule-individu* et *bascule-social* offrent un mécanisme pour limiter les liens d'interaction avec les agents. Du fait de notre définition des liens d'interaction, nous avons la possibilité d'utiliser et d'exprimer différents degrés de contrôle entre les agents.

Nous pouvons envisager la définition et l'utilisation d'autres protocoles d'interaction du type ascendant par exemple, et la mise en place d'autres organisations. Le mécanisme de bascule organisation permettra de plus de tester et de mettre en évidence les conditions de validité d'organisation.

## 6.6 Conclusion

Cette réalisation nous a conduit à développer un prototype de module standard bâti sur les modèles de contrôle social et individuel pour un système intégré de vision du chapitre précédent. Elle nous a permis de valider l'architecture de contrôle, les mécanismes de bascule entre les couches de contrôle, le mécanisme de bascule social et de bascule individu. Cependant, ces dernières doivent être améliorées afin de fournir une meilleure conceptualisation des notions de rôle. Nous avons testé différents modes de fonctionnement entre les agents au travers des protocoles et des organisations. L'expression des engagements sociaux et des protocoles par rapport au sujet d'interaction les rend indépendants des agents présents dans le système. Les accointances font le lien entre ceux-ci et le sujet. Cet élément est important si l'on veut obtenir un système ouvert.

L'utilisation satisfaisante du module de contrôle individuel nous fait entrevoir de nouvelles perspectives dans notre démarche d'explicitation du contrôle. Ainsi, nous projetons de réaliser les modules de dialogue et d'organisation sur une architecture similaire à celle du module-individu, en utilisant le même type de mécanisme de bascule pour gérer le passage du dialogue à l'organisation et inversement. Nous devons cependant auparavant augmenter le langage d'interaction pour intégrer les termes nécessaires relatifs à la construction d'une organisation tel par exemple l'ajout de force exprimant l'engagement, la négociation.

La reformulation des modules de VAP dans le module standard d'agent nous a permis de mettre en évidence des perspectives nouvelles par rapport au fonctionnement de l'AGENT-DESCRIPTION-2D et de l'AGENT-INTERPRETATION. Les expérimentations concernant les interactions entre les agents ont permis de valider les éléments du modèle. De nouvelles perspectives sont ouvertes en ce qui concerne la définition et le test de nouveaux protocoles d'interaction et d'organisation. Cette phase d'étude pourra se dérouler sans que les module-individus des agents soient modifiés.

# Chapitre 7

## Conclusion

### 7.1 Problèmes abordés

#### Problèmes de contrôle

Dans cette thèse, nous avons abordé l'étude du contrôle dans un Système Intégré de Vision par Ordinateur. Nous avons proposé une *structure de contrôle* s'appuyant sur une étude de la notion de contrôle dans les domaines de l'Automatique, de la théorie des systèmes hiérarchiques, de l'intelligence artificielle et des systèmes multi-agents. Nous appuyant sur la définition du contrôle comme l'activité visant à instaurer et à modifier le fonctionnement d'un système en appliquant des contraintes dépendant de l'objectif à poursuivre, de son état et de celui de l'environnement, nous avons défini le problème du contrôle au sein d'un système comme étant constitué des quatre sous-problèmes suivants :

- problème de *décision*, relatif à la détermination de l'objectif à satisfaire,
- problème d'*adaptation*, concernant la définition des critères de sélection des traitements en fonction de cet objectif,
- problème de *commande* concerne le choix du traitement en appliquant les critères de sélection,

- problème d'*observation*, définissant les informations relatives à l'évolution de l'environnement ou du raisonnement nécessaires à la résolution des trois problèmes précédents.

A partir de ces problèmes, nous avons isolé et défini au sein de la structure de contrôle du système les mécanismes correspondant au *processus* et ceux relatifs au *contrôleur*. Au sein de ce dernier, les mécanismes sont organisés en une hiérarchie de contrôle constituée de trois couches relatives, respectivement à la résolution des problèmes de décision, d'adaptation et de commande. Chacune de ces couches définit la commande contraignant le fonctionnement de la couche inférieure à partir des observations sur son fonctionnement.

### Contrôle dans un Système Intégré de Vision

Analysant le contrôle dans les systèmes intégrés de vision selon cette grille, nous avons dégagé la problématique suivante concernant la définition de la structure de contrôle d'un tel système :

- Les *niveaux de représentation* sont un élément déterminant pour la définition des modules en termes de contrôleurs et des processus au sein du système. Les interactions qui s'établissent entre eux sont régies par le mécanisme de rétroaction entre contrôleur et processus : *prédiction-vérification*.
- Les modules n'ont pas la même organisation hiérarchique de leur structure. Leur structure possède un nombre de couches de contrôle spécifique à un type de fonctionnement global du système et s'avère peu propice pour un autre type de fonctionnement.
- Selon les niveaux de représentation et les contrôleurs définis, les critères de sélection utilisés pour contraindre le fonctionnement ont des expressions et des gestions différentes d'un module à l'autre.

Une des difficultés de l'étude du contrôle au sein d'un tel système est en partie due à la diversité des critères de définition des modules de traitement. Afin de définir les outils permettant d'étudier une telle problématique, nous avons décomposé la structure de contrôle d'un tel système selon deux dimensions conduisant à la mise en place d'entités de base correspondant à l'intersection des niveaux de représentation et de centre d'intérêts privilégiés se retrouvant sur chacun des niveaux : les méthodes d'inférence de forme en sont un exemple. Nous avons ensuite défini deux modèles :

- modèle de contrôle individuel : définition d'un modèle d'agent dans lequel chacun des modules est réexprimé. Il intègre les trois couches de contrôle. La définition de ce modèle permet de concentrer l'étude du contrôle sur la construction de la description à un niveau de représentation donné. Le mécanisme de bascule gérant le passage d'une couche de contrôle à l'autre résoud, en partie, le problème d'observation, en s'appuyant sur les conditions de validité des commandes régissant le fonctionnement de la couche de contrôle.

- modèle de contrôle social : définition d'un modèle de société précisant les liens de contrôle existant entre les agents ainsi que les informations échangées. Cette définition fait abstraction du contrôle individuel lié à la description de la scène au sein d'un niveau de représentation. La mise en place d'un outil de consultation et de création de protocoles nous permet de tester différentes politiques d'interaction.

Les protocoles résolvent le problème d'observation entre un contrôleur et un processus définis dans deux modules différents en contraignant explicitement les types d'échanges nécessaires pour la définition d'une commande par le contrôleur.

La définition des couples (contrôleur, processus) est réalisée explicitement et est modifiable au travers de l'expression de l'*organisation*. Celle-ci définit pour chaque agent quelles couches de contrôle ils peuvent utiliser et quels agents ils peuvent contrôler.

A partir de ces deux modèles, nous avons défini et implanté un module standard d'agent. Celui-ci nous a permis de valider notre approche d'un point de vue contrôle. Nous avons pu ainsi tester au sein d'une même architecture un fonctionnement de planification et un fonctionnement de réaction. Du point de vue du système, nous avons testé différents modes de fonctionnement du système avec, d'une part les protocoles d'interaction et d'autre part les organisations.

## 7.2 Perspectives

### 7.2.1 Intégration et Contrôle dans un système intégré de vision

#### Intégration

La structure de contrôle développée est incomplète par rapport à la résolution du problème d'intégration. Nous n'avons abordé que partiellement la notion d'ouverture au cours de notre travail en exprimant par exemple l'ensemble des protocoles, organisation et connaissances par rapport à des sujets d'interaction et non pas par rapport aux agents eux-mêmes. Les accointances fournissent le lien entre les agents et les sujets d'interaction. Une amélioration envisageable est d'utiliser notamment des protocoles de présentation [Ber 92] pour permettre l'intégration de nouveaux modules dynamiquement dans le système en offrant la possibilité à un agent de définir lui même ses accointances en fonction de ses intérêts propres. Dans l'optique de contrôle dans laquelle nous nous sommes placés, un langage de description externe des éléments relatifs au contrôle au sein d'un agent devrait être développé.

Dans le cadre de notre expérimentation nous nous sommes placés dans le cadre d'agents de base. Cependant, le modèle d'agent que nous proposons permet l'expression des trois types d'agent précédent : *agent-niveau*, *agent-focus* ou *agent-base*. Des études restent donc encore à mener pour pouvoir aborder complètement ces types d'agent.

## Contrôle

Du fait du cadre d'expérimentation lié au système *VAP*, nous avons essentiellement étudié le contrôle inter-niveau de représentation. Du fait des outils et de l'expression des protocoles indépendante du fonctionnement de l'agent, nous pouvons envisager l'étude des interactions intra-niveaux de représentation par le développement de protocoles dédiés sans remettre en cause la structure de contrôle développée. Nous pouvons ainsi envisager l'utilisation des modèles de contrôle individuel et de contrôle social dans le cadre du projet de recherche *SATURNE*.

D'autre part, les modèles de contrôle individuel et de contrôle social élaborés ne sont pas spécifiques à notre domaine d'application. Nous pouvons ainsi envisager leur utilisation dans d'autres contextes tels par exemple la compréhension de la parole, ou un environnement C.I.M. Le modèle de contrôle social doit être encore développé pour pouvoir expliciter le contrôle sur les couches adaptation et décision, c.-à-d. la définition du module d'organisation ainsi que les protocoles et rôles d'organisation. Les activités de dialogue et d'organisation mises en œuvre par les modules de dialogue et d'organisation sont similaires aux activités de l'agent individuel. Nous pouvons donc envisager l'explicitation de ces deux modules selon la même hiérarchie de contrôle dédiée respectivement au dialogue et à l'organisation (cf. figure 7.1 où seul le module dialogue est explicité). Cependant, tout notre travail s'est essentiellement concentré sur la structure de contrôle d'un système. Nous avons mis en évidence quelques stratégies de contrôle, des protocoles et des organisations. Ce travail doit être poursuivi afin d'augmenter les connaissances de contrôle utilisables.

### 7.2.2 Systèmes Multi-Agents

Lors de notre étude du contrôle nous avons été amené à élaborer un modèle d'agent et un modèle de société. Ils seront certainement à améliorer. Cependant, nous pouvons d'ores et déjà en retirer des enseignements quant à certains problèmes actuels dans les systèmes Multi-Agents.

#### Modèle d'agent : intégration Réactif Délibératif

Dans le modèle de contrôle individuel proposé, un mécanisme de bascule explicite les conditions de passage d'une couche de contrôle à l'autre. Par ce mécanisme, ce modèle peut ainsi être considéré comme un terrain d'étude pour l'intégration des propriétés de comportement *réactif* et *délibératif* au sein de la structure de contrôle. Nous recherchons à étudier essentiellement au niveau de cette intégration les différents types d'événements qui permettent de faire passer un fonctionnement d'une couche à une autre.

#### Modèle de société : étude du contrôle social

Nous avons mis en évidence les différentes relations de contrôle pouvant exister entre les agents d'un système. Nous avons défini au sein du modèle de contrôle social

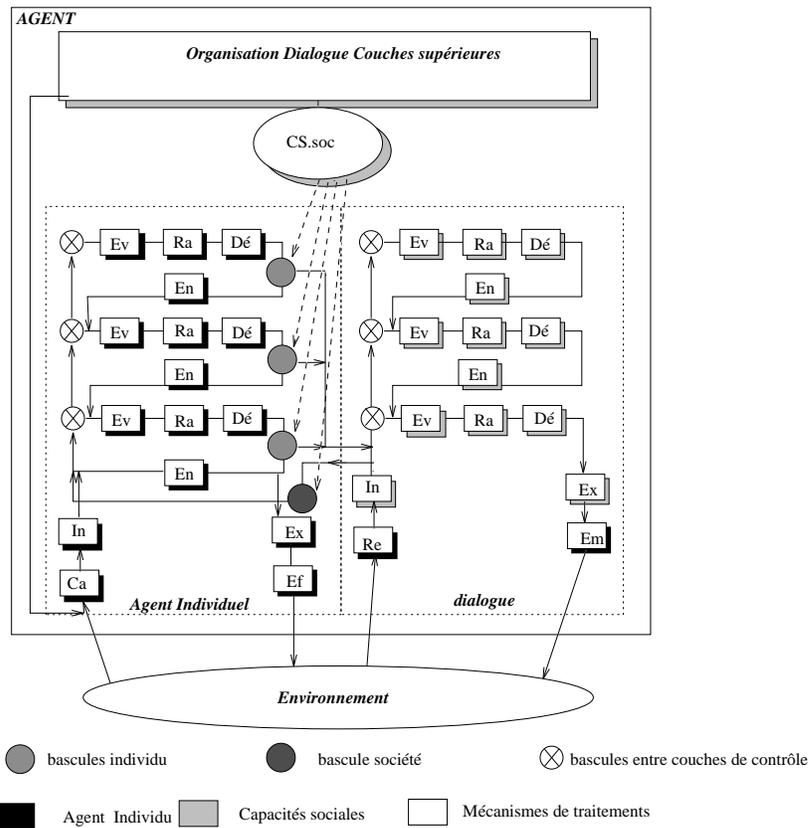


FIG. 7.1 - **Perspective du modèle de contrôle social** Chacune des couches supérieures peuvent être envisagées selon la même structure que la couche commande. *Ev* évaluation, *Ra* raisonnement, *Dé* décision, *En* engagement, *Ca* capteur, *In* interprétation, *Ef* effecteur, *Ex* exécution, *Em* émetteur, *Re* récepteur.

les éléments nécessaires à leur expression. Nous avons laissé de côté l'ensemble des études liées au fondement des relations de contrôle. Une des perspectives serait donc ainsi d'utiliser la structure que nous avons développé pour étudier les fondements du contrôle entre plusieurs agents. Cette étude vise à caractériser la relation de contrôle que l'on veut mettre entre deux agents par les études sur la notion de pouvoir que l'on trouve en sociologie ou à rechercher à expliciter le fondement de ces relations de contrôle comme le fait par exemple Castelfranchi [Cas 90].

Un mécanisme similaire à celui du mécanisme de bascule inter-couches de contrôle peut être envisagé dans le modèle de contrôle social pour remettre en cause des organisations. Il serait ainsi possible d'étudier la dynamique de passage entre un fonctionnement de *résolution* que l'on trouve dans l'approche résolution distribuée de problème et un fonctionnement de *simulation* beaucoup plus présent dans les systèmes multi-agents.

# Bibliographie

- [Agr 87] P.E. Agre et D. Chapman. PENG1: An implementation of a theory of activity. *AAAI Conference*, pages 268–272, 1987.
- [Alo 86] J.Y. Aloimonos. *Computing intrinsic images*. Thèse de Doctorat, University of Rochester, August 1986.
- [Alo 87] J.Y. Aloimonos, I. Weiss, et A. Bandyopadhyay. Active vision. *International Conference on Computer Vision*, pages 35–54, 1987.
- [Alo 90] J.Y. Aloimonos. Purposive and qualitative vision. *ECCV Esprit Workshop on Control of Perception*, 1990.
- [Ama 68] S. Amarel. On representations of problems of reasoning about actions. *Machine Intelligence*, 3:131–170, 1968.
- [Baj 85] R. Bajcsy. Active perception vs passive perception. *Proceedings of the IEEE Workshop on Computer Vision Representation and Control*, pages 55–62, Belleaire mich., October 1985. IEEE.
- [Baj 88] R. Bajcsy. Active perception. *IEEE Proceedings*, 76(8):996–1005, 1988.
- [Baj 91] R. Bajcsy et T. Sobh. Observing a moving agent. Ms-cis-91-01 grasp lab 247, Dpt of Computer and Information Science, University of Pennsylvania, January 1991.
- [Bal 78] D.H. Ballard. An approach to knowledge-directed image analysis. A.R.Hanson et E.M. Riseman, éditeurs, *Computer Vision Systems*, pages 271–281. Academic Press, 1978.
- [Bal 91] D.H. Ballard. Animate vision. *Artificial Intelligence*, 48:57–86, 1991.

- [Bar 78] H.G. Barrow et J.M. Tenenbaum. Recovering intrinsic scene characteristics from images. A.R.Hanson et E.M. Riseman, éditeurs, *Computer Vision Systems*, pages 3–26. Academic Press, 1978.
- [Bau 92] O. Baujard. *Un environnement de programmation Multi-Agents et son Application à la Vision par Ordinateur*. Thèse de doctorat, Université Joseph Fourier, 1992.
- [Ben 90] O. Benard et F. Charton. Squelette d’application de vision active (SAVA). Projet de troisième année, ENSIMAG, 1990.
- [Ber 91] S. Berthet et V. Caviggia. SAVA 2.0. Projet troisième année, ENSIMAG, Juin 1991.
- [Ber 92] S. Berthet, Y. Demazeau, et O. Boissier. Knowing each other better. *12th International Workshop on DAI*, Glen Arbor Michigan, February 1992.
- [Bie 81] I. Biederman. *Perceptual Organization*, chapitre On the semantics of a Scene at a glance. Hillsdale, N. J., Erlbaum, 1981.
- [Bie 85] I. Biederman. Human image understanding: Recent research and a theory. *Computer Vision, Graphics, and Image Processing*, 32:29–73, 1985.
- [Bin 82] T.O. Binford. Survey of model-based image analysis systems. *The International Journal of Robotics Research*, 1(1):18–64, spring 1982.
- [Boi 90] O. Boissier. La coopération entre systèmes à base de connaissances. Rapport de Recherche no. RR811-I-IMAG-96-LIFIA, LIFIA, IMAG, France, February 1990.
- [Boi 91] O. Boissier et Y. Demazeau. A distributed AI view on general purpose vision systems. *3rd Modeling Autonomous Agent in Multi-Agent World Workshop*, Kaiserslautern, 1991.
- [Bon 90] A.H. Bond. Projects: A normative model of collaboration in organizations. *10th International Workshop on DAI*, Bandera, Texas, October 1990.
- [Bou 91] T. Bouron, J. Ferber, et F. Samuel. MAGES: a multi-agent testbed for heterogeneous agents. Y. Demazeau et J.P. Müller, éditeurs, *Decentralized Artificial Intelligence 2*, pages 195–214. Elsevier Science Publishers B.V. (North Holland), 1991.
- [Bou 92] T. Bouron. *Structures de communication et d’organisation pour la coopération dans un univers Multi-Agents*. Thèse de doctorat, Université Pierre et Marie Curie Paris 6, 1992.
- [Bro 81] R.A. Brooks. Symbolic reasoning among 3D objects and 2D models. *Artificial Intelligence*, 17:285–348, 1981.

- [Bro 85] R.A. Brooks. A robust layered control system for a mobile robot. AI Memo no. 864, MIT AI Lab, Cambridge, MA: Massachusetts Institute of Technology, Artificial Intelligence Laboratory, september 1985.
- [Bur 91] B. Burmeister et K. Sundermeyer. Cooperative problem solving guided by intentions and perception. *3rd Modeling Autonomous Agent in Multi-Agent World Workshop*, Kaiserslautern, 1991.
- [Cal 90] A. Califano, R. Kjeldsen, et R.M. Bolle. Data and model driven foveation. *Computer Vision Pattern Recognition*, pages 1–7, 1990.
- [Cal 91] G. Calvary et B. Zoppis. Groupement d'indices visuels. Projet de troisième année, ENSIMAG, Juin 1991.
- [Cam 83] S. Cammarata, D. McArthur, et R. Steeb. Strategies of cooperation in distributed problem solving. *IJCAI Conference*, pages 767–770, 1983.
- [Cam 90] J.A. Campbell et M.P. D'Inverno. Knowledge interchange protocols. Y. Demazeau & J.P. Müller, éditeur, *Decentralized Artificial Intelligence*, pages 63–80. Elsevier Science Publishers B.V. (North Holland), 1990.
- [Cas 90] C. Castelfranchi. Social power. A point missed in Multi Agent, DAI and HCI. Y. Demazeau et J.P. Müller, éditeurs, *Decentralized Artificial Intelligence*, pages 49–62. Elsevier Science Publishers B.V. (North Holland), 1990.
- [Cha 90] D. Chapman. *Visual Attention, Search, and Routines*. Thèse de Doctorat, MIT, 1990.
- [Cha 91] M.K. Chang et C.C. Woo. SANP : A communication level protocol for negotiations. *3rd Modeling Autonomous Agent in Multi-Agent World Workshop*, Kaiserslautern, 1991.
- [Cla 88] J.L. Clark et N.J. Ferrier. Modal control of an attentive vision system. *ICCV*, pages 514–523, Tampa Floride, 1988.
- [Cla 89] J.L. Clark et N.J. Ferrier. Control of visual attention in mobile robots. *Robotics and Automation*, 1989.
- [Coh 79] P.R. Cohen et C.R. Perrault. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3(3), 1979.
- [Coh 90] P.R. Cohen et H.J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:213–261, 1990.
- [Col 88] A. Collinot. *Le problème du contrôle dans un système flexible d'ordonnement*. Thèse de doctorat, Université Paris 6, Novembre 1988.

- [Col 91] A. Collinot et B. Hayes-Roth. Real-time performance of intelligent autonomous agents. *3rd Modeling Autonomous Agent in Multi-Agent World Workshop*, Kaiserslautern, 1991.
- [Con 86] S.E. Conry, R.A. Meyer, et V.R. Lesser. Multistage negotiation in distributed planning. Coins technical report 86-67, Coins Amherst MA, December 1986.
- [Cor 83a] D.D. Corkill et V.R. Lesser. The distributed vehicle monitoring testbed : a tool for investigating distributed problem solving network. *Artificial Intelligence Magazine*, 4(3):15–33, Fall 1983.
- [Cor 83b] D.D. Corkill et V.R. Lesser. The use of meta-level control for coordination in a distributed problem solving network. *IJCAI Conference*, pages 748–756, 1983.
- [Cor 88] D.D. Corkill et K.Q. Gallagher. Tuning a blackboard-based application : A case study using gbb. *AAAI Conference*, pages 671–676, 1988.
- [Cro 84] J.L. Crowley et A.C. Parker. A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, march 1984.
- [Cro 89] J.L. Crowley, A. Chehikian, J. Kittler, J. Illingworth, J.O. Eklundh, G. Granlund, J. Wiklund, E. Granum, et H.I. Christensen. Technical annex for ESPRIT basic research action 3038 : Vision As Process. Rapport, Aalborg, 1989.
- [Cro 90] J.L. Crowley. Knowledge, symbolic reasoning and perception. *Intelligent Autonomous Systems, Amsterdam*, 1990.
- [Cro 92] J.L. Crowley, P. Bobet, et M. Mesrabi. Gaze control for a binocular camera head. G. Sandini, éditeur, *European Conference on Computer Vision*, pages 588–596. Springer Verlag, 1992.
- [Dav 80a] R. Davis. Content reference : Reasoning about rules. *Artificial Intelligence*, 15:223–239, 1980.
- [Dav 80b] R. Davis. Meta-rules : Reasoning about control. *Artificial Intelligence*, 15:179–222, 1980.
- [Dea 91] T.L. Dean et M.P. Wellman. *Planning and Control*. Morgan Kaufmann, 1991.
- [Dek 89] E. Dekneuvel, M. Ghallab, et P. Grandjean. Multi-sensory model-based object recognition through fusion. *1st Workshop on Multi-sensor fusion and environment modeling*, Toulouse, October 1989. IARP.
- [Dem 86] Y. Demazeau. *Niveaux de représentation pour la vision par ordinateur. Indices d'image et indices de scène*. Thèse de doctorat, INP Grenoble, 1986.

- [Dem 90a] Y. Demazeau. A distributed artificial intelligence approach to integration of visual modules. Heraklion, september 1990. European working week on vision, Workshop on integration of visual modules.
- [Dem 90b] Y. Demazeau et J.P. Müller. *Decentralized Artificial Intelligence*. Elsevier Science Publishers B.V. (North Holland), 1990.
- [Dem 90c] Y. Demazeau et J.P. Müller. Decentralized Artificial Intelligence. Y. Demazeau et J.P. Müller, éditeurs, *Decentralized Artificial Intelligence*, pages 3–13. Elsevier Science Publishers B.V. (North Holland), 1990.
- [Dem 91a] Y. Demazeau. Coordination patterns in multi-agent worlds application to computer vision and robotics. *IEE Colloquium on intelligent agents*, 25 February 1991.
- [Dem 91b] Y. Demazeau et J.P. Müller. *Decentralized Artificial Intelligence 2*. Elsevier Science Publishers B.V. (North Holland), 1991.
- [Dis 90] C. Discours. *Analyse du mouvement par mise en correspondance d'indices visuels*. Thèse de doctorat, INP Grenoble, Février 1990.
- [Dra 87] B.A. Draper, R.T. Collins, J. Brolio, J. Griffith, A.R. Hanson, et E.M. Riseman. Tools and experiments in the knowledge directed interpretation of road scenes. *DARPA Image Understanding Workshop*, pages 178–193, 1987.
- [Dra 89] B.A. Draper, R.T. Collins, J. Brolio, A.R. Hanson, et E.M. Riseman. The SCHEMA system. *International Journal of Computer Vision*, 2:209–250, 1989.
- [Dra 90] B.A. Draper et E.M. Riseman. Learning 3D recognition strategies. IEEE, éditeur, *International Conference on Computer Vision*, pages 320–324, 1990.
- [Dro 91] A. Drogoul et C. Dubreuil. Eco-problem-solving model: Results of the n-puzzle. *3rd Modeling Autonomous Agent in Multi-Agent World Workshop*, Kaiserslautern, 1991.
- [Dur 87a] E.H. Durfee, V.R. Lesser, et D.D. Corkill. Coherent cooperation among communicating problem solvers. *IEEE Transactions on Computers*, 36(11):1275–1291, november 1987.
- [Dur 87b] E.H. Durfee, V.R. Lesser, et D.D. Corkill. Cooperation through communication in a distributed solving network. M.N. Huhns, éditeur, *Distributed Artificial Intelligence. Research Notes in Artificial Intelligence*, pages 29–58. Pitman Morgan Kaufman, 1987.
- [Eng 87] R.S. Englemore et A.J. Morgan. *Blackboard Systems*. Addison-Wesley, 1987.

- [Erc 91] J. Erceau et J. Ferber. L'intelligence artificielle distribuée. *La Recherche*, 22(233):750–758, 1991.
- [Fer 91a] J. Ferber et L. Gasser. Intelligence Artificielle Distribuée. J.C. Rault, éditeur, *Expert Systems and their Applications 11th Workshop Tutorial Avignon*. EC2, 1991.
- [Fer 91b] J. Ferber et E. Jacopin. The framework of ECO-problem solving. Y. Demazeau et J.P. Müller, éditeurs, *Decentralized Artificial Intelligence 2*, pages 181–195. Elsevier Science Publishers B.V. (North Holland), 1991.
- [Fer 91c] I.A. Ferguson. Towards an architecture for adaptative rational mobile agents. *3rd Modeling Autonomous Agent in Multi-Agent World Workshop*, Kaiserslautern, 1991.
- [Fik 71] R.E. Fikes et N.J. Nilsson. Strips: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4), 1971.
- [Gal 90] J.R. Galliers. The positive role of conflicts in cooperative multi-agent systems. Y. Demazeau et J.P. Müller, éditeurs, *Decentralized Artificial Intelligence*, pages 33–46. Elsevier Science Publishers B.V. (North Holland), 1990.
- [Gar 76] T.D. Garvey. Perceptual strategies for purposive vision. Rapport no. 117, AI Center SRI, September 1976.
- [Gar 89] P. Garnesson, G. Giraudon, et P. Montesinos. MESSIE: Un système multi spécialistes en vision. application à l'interprétation en imagerie aérienne. Rapport de recherche no. 1012, INRIA, Avril 1989.
- [Gas 87] L. Gasser, C. Braganza, et N. Herman. MACE: a flexible testbed for distributed ai research. M.N. Huhns, éditeur, *Distributed Artificial Intelligence. Research Notes in Artificial Intelligence*, pages 119–175. Pitman Morgan Kaufman, 1987.
- [Gas 88] L. Gasser et A.H. Bond. *Readings in Distributed Artificial Intelligence*. Morgan Kaufman, 1988.
- [Gas 89] L. Gasser, N.F. Rouquette, R.W. Hill, et J. Lieb. Representing and using organizational knowledge in distributed AI systems. L. Gasser & M.N. Huhns, éditeur, *Distributed Artificial Intelligence Volume II. Research Notes in Artificial Intelligence*, pages 55–78. Pitman Morgan Kaufman, 1989.
- [Gen 83] M.R. Genesereth. An overview of meta-level architecture. *AAAI Conference*, pages 119–124, 1983.
- [Gen 87] M.R. Genesereth et N.J. Nilsson. *Logical Foundations of Artificial Intelligence*. Morgan Kaufman publishers, 1987.

- [Geo 87] M.P. Georgeff et A.L. Lansky. Procedural knowledge. Rapport no. note 411, SRI, january 1987.
- [Geo 89] M.P. Georgeff et F.F. Ingrand. Decision-making in an embedded reasoning system. *IJCAI Conference*, pages 972–978, 1989.
- [Guz 68] A. Guzman. Computer recognition of 3D objects in visual scenes. technical report no. MAC-TR-59, AI Laboratory, MIT-Cambridge Mass, 1968.
- [Han 78a] A.R. Hanson et E.M. Riseman. *Computer Vision Systems*. Academic Press, 1978.
- [Han 78b] A.R. Hanson et E.M. Riseman. VISIONS : a computer system for interpreting scenes. A.R. Hanson et E.M. Riseman, éditeurs, *Computer Vision Systems*, pages 303–333. Academic Press, 1978.
- [Hew 73] C. Hewitt. The actor formalism. *IJCAI Conference*, pages 235–245, 1973.
- [Hor 86] B.K.P. Horn. *Robot Vision*. McGrawHill, 1986.
- [Hr 85] B. Hayes-roth. A blackboard architecture for control. *Artificial Intelligence*, 26:251–361, 1985.
- [Hr 87] B. Hayes-roth. A multi-processor interrupt-driven for adaptative intelligent systems. KSL no. 87-31, KSL Computer Science Dpt Stanford University, June 1987.
- [IEE 89] IEEE. Proceedings of the IEEE, January 1989. vol 77 (1).
- [Kae 86] L.P. Kaelbling. An architecture for intelligent reactive systems. Rapport no. 400, SRI, October 1986.
- [Kah 90a] P. Kahn. *Proceedings of the Workshop on Qualitative Vision*. AAAI, 1990.
- [Kah 90b] P. Kahn, A. Winkler, et C.Y. Chong. Qualitative evaluation of perceptual groups. *AAAI Workshop on Qualitative Vision*, pages 157–161, 1990.
- [Kis 91] G. Kiss. Variable coupling of agents to their environment : combining situated and symbolic automata. *3rd Modeling Autonomous Agent in Multi-Agent World Workshop*, Kaiserslautern, 1991.
- [Koh 87] C.A. Kohl, A.R. Hanson, et E.M. Riseman. Goal-directed control of low-level processes for image interpretation. *DARPA Image Understanding Workshop*, pages 538–551, 1987.
- [Koh 91] W. Kohn. Declarative control architecture. *Communications of the ACM*, 34(8):65–78, August 1991.
- [Kon 80] K. Konolige et N.J. Nilsson. Multiple agent planning systems. *AAAI Conference*, pages 138–142, 1980.

- [Kre 91] T. Kreifelts et F. Von Martial. A negotiation framework for autonomous agents. Y. Demazeau et J.P. Müller, éditeurs, *Decentralized Artificial Intelligence 2*, pages 71–88. Elsevier Science Publishers B.V. (North Holland), 1991.
- [Lac 92] S. Lacroix, P. Grandjean, et M. Ghallab. Perception planning for a multi-sensory interpretation machine. *IEEE Robotics and Automation*, pages 1818–1824, Nice, May 1992.
- [Lai 83] J.E. Laird et A. Newell. A universal weak method : Summary of results. *IJCAI Conference*, pages 771–773, 1983.
- [Lai 87] J.E. Laird, A. Newell, et P.S. Rosenbloom. SOAR : An architecture for general intelligence. *Artificial Intelligence*, (33):1–64, 1987.
- [Lan 89] D.M. Lane, M.J. Chantler, E.W. Robertson, et A.G. McFadzean. A distributed problem solving architecture for knowledge based vision. L. Gasser & M.N. Huhns, éditeur, *Distributed Artificial Intelligence Volume II. Research Notes in Artificial Intelligence*, pages 433–462. Pitman Morgan Kaufman, 1989.
- [Les 81] V.R. Lesser et D.D. Corkill. Functionally accurate cooperative distributed systems. *IEEE Transactions on Systems, Man and Cybernetics*, 11(1):81–96, 1981.
- [Lev 90] P. Levi. Architectures of individual and distributed autonomous agents. *Intelligent Autonomous Systems, Amsterdam*, pages 315–324, 1990.
- [Lor 90] K. Lorenz. *L'envers du miroir*. Champs Flammarion, 1990.
- [Low 84] D.G. Lowe. *Perceptual organization and visual recognition*. Thèse de Doctorat, Computer science department, Stanford University, September 1984.
- [Lux 85] A. Lux. Algorithmique et contrôle en vision par ordinateur. Thèse d'état, INP Grenoble, septembre 1985.
- [Mac 78] A. Mackworth. Vision Research, Strategy : black magic, metaphors, mechanisms, miniworlds and maps. A.R. Hanson et E.M. Riseman, éditeurs, *Computer Vision Systems*, pages 53–60. Academic Press, 1978.
- [Mae 87] P. Maes. Computational reflection. technical Report no. 87-2, Vrije Universiteit Brussel Artificial Intelligence Laboratory, Brussels, Belgium, 1987.
- [Mae 89] P. Maes. How to do the right thing. *Connection Science Journal Special Issue on Hybrid Systems*, spring 1989.
- [Mae 90] P. Maes. Situated agents can have goals. *Robotics and Autonomous Systems*, 6:49–70, 1990.

- [Man 84] F. Manucci, J. Gallagher, M. Swabey, et C. Fisher. Expert system builder. *ESPRIT-84: Status Report of Ongoing Work*, pages 183–202. Elsevier North Holland, 1984.
- [Mar 82] D. Marr. *Vision*. Freeman San Francisco, 1982.
- [Mar 90] T. Maruichi, M. Ichikawa, et M. Tokoro. Modeling autonomous agents and their groups. Y. Demazeau et J.P. Müller, éditeurs, *Decentralized Artificial Intelligence*, pages 215–233. Elsevier Science Publishers B.V. (North Holland), 1990.
- [Mat 85] T. Matsuyama et V. Hwang. SIGMA : A framework for image understanding - integration of bottom-up and top-down analyses. *IJCAI Conference*, pages 908–915, 1985.
- [Mer 90] P. Merialdo, P.C. Pecollo, C.S. Regazzoni, G. Vernazza, et R. Zunino. Integration of territorial maps in the vision system of an autonomous land vehicle. *Intelligent Autonomous Systems, Amsterdam*, pages 694–704, 1990.
- [Mes 70] M.D. Mesarovic, D. Macko, et Y. Takahara. *Theory of hierarchical, multilevel systems*. Academic Press New York San Francisco London, 1970.
- [Min 75] M. Minsky. *Psychologie of Computer Vision*, chapitre A Framework for Representing Knowledge. McGraw-Hill, New York, 1975.
- [Mon 92] T.A. Montgomery, J. Lee, D.J. Musliner, E.H. Durfee, D. Damouth, et Y. So. MICE users guide. Rapport, University of Michigan, Ann Arbor MI, 1992.
- [Nag 79] M. Nagao, T. Matsuyama, et H. Mori. Structured analysis of complex aerial photographs. *IJCAI Conference*, pages 610–616, 1979.
- [Nel 91] R.C. Nelson. Introduction : Vision as intelligent behavior-an introduction to machine vision research at the university of rochester. *International Journal of Computer Vision*, 7(1):5–9, 1991.
- [Neu 90] B. Neumann. Niveaux de représentation en vision par ordinateur. communication personnelle, 1990.
- [Nii 86a] H.P. Nii. Part one. blackboard systems: The blackboard model of problem solving and the evolution of blackboard architectures. *Artificial Intelligence Magazine*, pages 38–53, Summer 1986.
- [Nii 86b] H.P. Nii. Part two. blackboard systems: The blackboard model of problem solving and the evolution of blackboard architectures. *Artificial Intelligence Magazine*, pages 82–106, August 1986.
- [Nil 82] N.J. Nilsson. *Principles of Artificial Intelligence*. Springer-Verlag, 1982.

- [Nil 87] N.J. Nilsson. Intelligent communicating agents. Project proposal, Stanford University, January 1987.
- [Num 91] C. Numaoka. Organizational activity through conversation in open systems. *3rd Modeling Autonomous Agent in Multi-Agent World Workshop*, Kaiserslautern, 1991.
- [Pah 92] K. Pahlavan, T. Uhlin, et J.O. Eklundh. Integrating primary ocular processes. G. Sandini, éditeur, *European Conference on Computer Vision*, pages 526–541. Springer Verlag, 1992.
- [pap 88] C. Le pape. Des systèmes d’ordonnancement flexibles et opportunistes, première partie. Thèse de doctorat, Univ. Paris Sud Orsay, Avril 1988.
- [Par 90] H. Van Dyke Parunak. Toward a formal model of inter-agent control. *10th International Workshop on DAI*, Bandera Texas, October 1990.
- [Pat 87] H.E. Pattison, D.D. Corkill, et V.R. Lesser. Instantiating description of organizational structures. M.N. Huhns, éditeur, *Distributed Artificial Intelligence. Research Notes in Artificial Intelligence*, pages 59–96. Morgan Kaufman Pitman, 1987.
- [PLE 92] PLEIAD. Vers une taxinomie du vocabulaire en IAD. *Journées systèmes Multi-Agents du PRC-IA*, Nancy, December 1992.
- [Pog 88] T. Poggio, J. Little, E. Gamble, et all. The MIT Vision Machine. *DARPA Conference*, 1988.
- [Pop 92] P. Populaire. Validation de protocoles de communication en systèmes multi-agents. Travail de fin d’étude, Faculté Polytechnique de Mons, Juin 1992.
- [PT 86] J. Pertin-Troccaz. *Modélisation du raisonnement géométrique pour la programmation des robots*. Thèse de doctorat, INP Grenoble, 1986.
- [Rao 91a] A.S. Rao et M.P. Georgeff. Modeling rational agents within a BDI-architecture. Technical Note no. 14, the Australian Artificial Intelligence Institute, 1 Grattant Street Carlton Victoria 3053 Australia, February 1991.
- [Rao 91b] A.S. Rao, M.P. Georgeff, et E.A. Sonenberg. Social plans: A preliminary report. *3rd Modeling Autonomous Agent in Multi-Agent World Workshop*, Kaiserslautern, 1991.
- [Ric 90] J.F. Richard. *Les activités mentales, comprendre, raisonner, trouver des solutions*. Armand Colin, 1990.
- [Rim 92a] R.D. Rimey et C.M. Brown. *Task-Oriented Vision with Multiple Bayes Nets in Active Vision*, chapitre 13. MIT press, 1992.

- [Rim 92b] R.D. Rimey et C.M. Brown. Where to look next using a bayes net: Incorporating geometric relations. G. Sandini, éditeur, *European Conference on Computer Vision*, pages 542–550. Springer Verlag, 1992.
- [Rob 65] L. Roberts. Machine perception of three-dimensional solids. J. Tippett, éditeur, *Optical and electro-optical information processing*, pages 159–197. MIT Press, Cambridge, 1965.
- [Ros 82] J.S. Rosenschein. Synchronization of multi-agents plans. *AAAI Conference*, pages 115–119, 1982.
- [Ros 84] J.S. Rosenschein. Rational interaction: Cooperation among intelligent agents. Report no. STAN-CS-851081, Stanford University, 1984.
- [Ros 86] S.J. Rosenschein et L.P. Kaelbling. The synthesis of digital machines with provable epistemic properties. J.Y. Halpern, éditeur, *Theoretical Aspects of Reasoning about Knowledge, Proceedings of the 1986 Conference*, pages 83–98. Morgan Kauffman, 1986.
- [Rus 85] S.J. Russel. The compleat guide to MRS. technical Report no. STAN-CS-85-1080, Dept of Computer Science, Stanford University, Stanford, CA 94305, 1985.
- [Rus 89] S.J. Russel. Execution architectures and compilation. *IJCAI Conference*, pages 15–20, 1989.
- [Sea 69] J.R. Searle. *Speech Acts*. Cambridge University Press, 1969.
- [Shi 91] K.G. Shin et X. Cui. Design of a knowledge-based controller for intelligent control systems. *IEEE Transactions on Systems, Man and Cybernetics*, 21(2):368–375, March/April 1991.
- [Sho 90] Y. Shoham. Agent Oriented Programming. Rapport no. STAN-CS-90-1335, Robotics Laboratory, Comp. Science Dpt, Stanford, 1990.
- [Sho 91a] R. Shoureshi. Learning and decision making for intelligent control systems. *IEEE control systems*, pages 34–37, January 1991.
- [Sho 91b] R. Shoureshi. The mystique of intelligent control. *IEEE control systems*, page 33, January 1991.
- [Sia 91] S.S. Sian. Adaptation based on cooperative learning in multi-agent systems. Y. Demazeau & J.P. Müller, éditeur, *Decentralized Artificial Intelligence 2*, pages 257–270. Elsevier Science Publishers B.V. (North Holland), 1991.
- [Sim 83] H.A. Simon. Search and reasoning in problem solving. *Artificial Intelligence*, 21:7–29, 1983.

- [Sin 91a] M.P. Singh. Intentions for multiagent systems. Rapport, Center for Cognitive Science and Dept of Computer Science University of Texas, University of Texas Austin TX 78712, 1991.
- [Sin 91b] M.P. Singh. Towards a formal theory of communication for multiagent systems. *IJCAI Conference*, 1991.
- [Sj 90] M. Sjölin. Description d'images dans un système de vision active. Projet troisième année, ENSIMAG, 1990.
- [Smi 81] R.G. Smith et R. Davis. Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11(1):61–70, 1981.
- [Smi 86a] D.E. Smith. *Controlling inference*. Thèse de Doctorat, Stanford University, Avril 1986.
- [Smi 86b] G.B. Smith et T.M. Strat. A knowledge-based architecture for organizing sensory data. Technical note no. 399, Artificial intelligence Center SRI, december 1986.
- [Ste 90] L. Steels. Cooperation between distributed agents through self-organisation. Y. Demazeau et J.P. Müller, éditeurs, *Decentralized Artificial Intelligence*, pages 215–234. Elsevier Science Publishers B.V. (North Holland), 1990.
- [Swa 91] M.J. Swain et D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.
- [Syc 89] K. Sycara. Multiagent compromise via negotiation. L. Gasser & M.N. Huhns, éditeur, *Distributed Artificial Intelligence Volume II. Research Notes in Artificial Intelligence*, pages 119–137. Pitman Morgan Kaufman, 1989.
- [Tho 92] S. Thouvenot et S. Figard. Interprétation de scènes dans un système de vision active. Projet troisième année, ENSIMAG, Juin 1992.
- [Tib 86] G. Tiberghien. Psychologie cognitive, science de la cognition et technologie de la connaissance. J.L. Le Moigne, éditeur, *Intelligence des mécanismes, mécanismes de l'intelligence*, pages 173–192. Fondation Diderot Fayard, 1986.
- [Tre 86] A. Treisman. Features and objects in visual processing. *Scientific American*, pages 106–115, june 1986.
- [Tso 87] J. K. Tsotsos. Image Understanding. S. Shapiro et D. Eckroth, éditeurs, *The Encyclopedia of Artificial Intelligence*. Wiley and Sons, 1987.
- [Tso 89] J.K. Tsotsos. The complexity of perceptual search tasks. *IJCAI Conference*, pages 1571–1577, 1989.

- [Tso 90] J.K. Tsotsos. Active vs. passive visual search : which is more efficient. Rapport, University of Toronto, 1990.
- [Ull 84] S. Ullman. Visual routines. *Cognition*, 18:97–106, 1984.
- [Wee 87] C.C. Weems, S.P. Levitan, A.R. Hanson, et E.M. Riseman. The image understanding architecture. *DARPA Image Understanding Workshop*, pages 483–496, 1987.
- [Wer 89] E. Werner. Cooperating agents : A unified theory of communication and social structure. *Distributed Artificial Intelligence Volume II. Research Notes in Artificial Intelligence*, pages 3–36, 1989.
- [Wey 80] R. Weyhrauch. Prolegomena to a Theory of Mechanized Formal Reasoning. *Artificial Intelligence*, 13(1-2), 1980.
- [Whi 92] S.D. Whitehead. Reinforcement learning for the adaptative control of perception and action. Rapport no. 406, University of Rochester Computer Science, February 1992.

# Index

- état d'engagement, 115, 154
- état d'exécution, 112
- état de décision, 114, 154
- état de perception, 111, 152
- état de raisonnement, 112, 152
- état du processus, 26
- évaluation, 130
  
- accointance, 112
- action, 18, 110
- agent, 91
- agent délibératif, 95
- agent de base, 97
- agent focus, 97
- agent mixte, 95
- agent niveau, 97
- agent réactif, 94
- attentif, 67
- attention visuelle, 60
- autonomie, 92
  
- bascule, 131, 157
- bascule individu, 143
- bascule individu-société, 143
- bascule inter-couches, 131
- bascule société-individu, 141
- bascule social, 141
- but, 18, 92, 113, 152
  
- capacités d'action, 110
- capacités de décision, 111
- capacités de raisonnement, 110
- centre d'intérêt, 69
- centre d'intérêt sémantique, 69
- centre d'intérêt spatial, 69
- centre d'intérêt temporel, 69
- commande, 12, 27
- compétence, 109, 150
- conflit, 113
- connaissance, 92
- connaissances de contrôle, 33
- contrôle individuel, 21
- contrôle externe, 60
- contrôle individuel, 93
- contrôle inter-niveaux, 62
- contrôle interne, 60
- contrôle intra-niveaux, 62
- contrôle social, 23, 93
- contrôleur, 13, 26
- couche d'adaptation, 31
- couche de commande, 31
- couche de décision, 31
- cycle de contrôle, 32, 125
  
- décision, 130
- donnée, 36, 92
  
- engagement, 130

enrichissement, 98  
 environnement d'un agent, 92  
 fait, 109, 150  
 focalisation, 148  
 focus, 69  
 focus sémantique, 69  
 focus spatial, 69  
 focus temporel, 69  
 fonction émetteur, 118, 155  
 fonction capteur, 115  
 fonction d'évaluation, 116, 156  
 fonction d'engagement, 117, 157  
 fonction d'exécution, 117  
 fonction d'exécution-dialogue, 117, 156  
 fonction d'interprétation, 116  
 fonction d'interprétation-dialogue, 116, 156  
 fonction d'organisation, 118  
 fonction de décision, 117, 157  
 fonction de raisonnement, 116, 157  
 fonction effecteur, 117  
 fonction récepteur, 115, 155  
 fonctionnement de reconnaissance, 38, 68  
 fonctionnement de reconstruction, 38, 68  
 fonctionnement teleonomique, 38  
 force, 137  
 hétérogénéité, 94  
 hypothèse, 92, 112, 152  
 identification, 69  
 image, 36  
 indice d'image, 37  
 indice visuel, 36, 148  
 inférences de formes, 37  
 intention, 114  
 interaction, 121  
 interaction de contrôle, 138  
 interaction de données, 138  
 interprétation, 37  
 langage d'interaction, 93, 120, 144  
 langage de communication, 144  
 langage de représentation des connaissances, 145  
 lien d'observation, 138  
 lien de contrôle, 138  
 localisation, 69  
 loi de commande, 27  
 mécanisme de décision de la société, 93  
 mécanisme de raisonnement de la société, 93  
 mécanismes de décision d'un agent, 92  
 mécanismes de raisonnement d'un agent, 92  
 mécanismes de traitement, 115  
 mécanismes de traitement d'un agent, 92  
 mécanismes de traitement de la société, 93  
 modèle, 36, 148  
 mode ascendant, 63  
 mode descendant, 63  
 modes de fonctionnement, 63  
 multi-agents, 96  
 nature, 136  
 niveau de représentation, 146  
 objectif, 13, 27  
 observation, 13, 27  
 opération, 110  
 opération d'enrichissement, 150  
 opération de fusion, 150  
 organisation, 93, 121  
 perturbation, 12  
 plan, 18, 113, 154  
 prédiction-vérification, 62  
 preattentif, 67  
 problème d'adaptation, 28  
 problème d'observation, 29  
 problème de commande, 28  
 problème de décision, 28  
 problème du contrôle, 29

processus, 12, 26  
protocole d'interaction, 121  
protocole d'organisation, 93, 121  
protocole de résolution, 93, 121  
protocoles d'interaction, 93

région d'intérêt, 78  
réseau d'interactions, 93  
résolution distribuée de problèmes, 96  
rôle, 121  
rôle d'organisation, 122  
rôle de dialogue, 122  
rôle de résolution, 122, 140  
raisonnement, 130  
reconnaissance, 68  
reconstruction, 68

scène, 36  
segmentation, 37  
société, 92  
stratégie, 111, 151  
structure d'un agent, 92  
structure d'une société, 93  
structure de contrôle, 32  
système indépendant, 40  
système intégré de vision, 40  
système ouvert, 40

tactique, 110, 151  
teleonomique, 68  
transformation, 97

vecteur d'état, 12  
vision active, 44  
vision passive, 44



## Résumé

La thèse concerne l'étude du contrôle dans un système intégré de vision par ordinateur. La structure d'un tel système permet l'ajout et le retrait dynamique de traitements et de représentations selon les types de scènes envisagées. Nous décomposons le problème du contrôle en quatre sous-problèmes définissant l'objectif à satisfaire (décision), les critères de sélection qui en découlent (adaptation), l'action à exécuter en appliquant ces critères de sélection (commande) et les observations nécessaires pour la résolution des sous-problèmes précédents (observation). Afin d'offrir une architecture d'intégration ouverte et flexible, nous utilisons une approche Multi-agents, répercutant ainsi les problèmes de contrôle selon deux dimensions : contrôle individuel (interne à un agent) et contrôle social (gestion des interactions et de la résolution des agents). La structure de contrôle de chaque agent est organisée selon trois couches dédiées respectivement à la résolution des problèmes de décision, d'adaptation et de commande. Le contrôle social restreint les interactions possibles et les résolutions entreprises par chacun des agents. Des protocoles définissent l'enchaînement des interactions. Ces deux notions s'inscrivent dans un modèle de société dans lequel sont aussi explicitées les trois couches résolvant les sous-problèmes de contrôle. L'architecture proposée permet d'étudier le contrôle nécessaire pour l'expression de modes de fonctionnement délibératifs et réactifs au sein de chacun des agents et, d'autre part, d'intégrer dynamiquement les modes de reconstruction et de reconnaissance au travers de la gestion des interactions. L'intérêt de cette approche est illustrée au travers de réalisations issues du système Vision As Process (VAP) développé dans le cadre du projet Européen BRA3038.

MOTS-CLES : VISION PAR ORDINATEUR, CONTROLE, INTELLIGENCE ARTIFICIELLE DISTRIBUEE, SYSTEMES MULTI-AGENTS, INTEGRATION.

## Abstract

This thesis is concerned with the study of control aspects in an Integrated Computer Vision System. Such a system has an architecture which enables to add or retract dynamically both processing methods and representations schemes according to the type of the scenes to be treated. First, we have decomposed the control problem in four different subproblems concerning the definition of the goal to be satisfied (decision), the selection criteria implied by this goal (adaptation), the action to be executed when applying these criteria (command), and the observations which must be collected for solving the former subproblems. We have adopted a Multi-Agent approach in order to design an open and flexible integration scheme. The control problems are represented in this architecture along two dimensions : individual control (internal control of an agent) and social control (interaction control). The control structure within each agent is organized in three different layers, which correspond to the resolution of the decision, adaptation and command problems respectively. The social control restricts the possible interactions and resolution methods which can be used by each of the agents. Protocols define the sequence of interactions between them. These two notions take part into our society model where three different control layers are responsible for solving each of the former control subproblems. The proposed architecture enables the study of control policies for deliberatif and reactive behaviour. On the other hand, it enables the dynamic integration of reconstruction and recognition processing modes into interaction management policies. The interest of this approach is illustrated by exemples from the Vision As Process system (VAP) developed in the framework of th European Project BRA3038.

KEYWORDS : COMPUTER VISION, CONTROL, DISTRIBUTED ARTIFICIAL INTELLIGENCE, MULTI-AGENTS SYSTEM, INTEGRATION.