TELECOM PARIS

École Nationale Supérieure des Télécommunications

Thèse

présentée pour obtenir le grade de docteur de l'Ecole nationale supérieure des télécommunications

Spécialité : Informatique et réseaux

Didier Verna

Télé-opération et réalité virtuelle : assistance à l'opérateur par modélisation cognitive de ses intentions

ENST 00 E 007

Soutenue le 16 Février 2000 devant le jury composé de

Jean-Paul Papin	Président
Philippe Coiffet Jean-François Richard	Rapporteurs
Alain Grumbach	Directeur de thèse

Ecole Nationale Supérieure des Télécommunications

Thèse présentée pour obtenir le grade de docteur de l'École Nationale Supérieure des Télécommunications de Paris

Télé-Opération et Réalité Virtuelle : assistance à l'opérateur par modélisation cognitive de ses intentions

Didier Verna
E.N.S.T.
verna@inf.enst.fr
http ://www.infres.enst.fr/~verna

Étude réalisée sous la responsabilité de :
Alain Grumbach
E.N.S.T.
grumbach@enst.fr

version <final.2> du 13 avril 2000

E.N.S.T
Département Informatique et Réseaux
46, rue Barrault
75013 Paris

Remerciements

Ce document n'aurait jamais vu le jour sans la présence et l'appui d'un certain nombre de personnes qui, de près ou de loin, ont contribué à l'aboutissement du travail qu'il représente. La toute première personne concernée est bien entendu mon directeur de thèse, Alain Grumbach.

Je suis redevable aux membres de mon jury : Jean-Paul Papin qui a bien voulu présider la soutenance, Jean-François Richard, qui a bien voulu être rapporteur de mon travail, et tout particulièrement Philippe Coiffet, également rapporteur, avec qui j'ai eu plaisir à discuter à plusieurs reprises, et dont les réflexions furent riches en enseignement.

La DRET / DGA (Direction Régionale des Études et Techniques, Direction Générale de l'Armement), puis l'association Arecom m'ont apporté le soutien financier sans lequel ce travail n'aurait put être mené à son terme. L'URA 820 du CNRS m'a permis de présenter des communications à l'étranger dans de multiples occasions.

La direction scientifique de l'ENST a bien voulu m'accueillir en thèse au sein de l'école et à plein temps. Le département informatique et réseaux de l'école m'a permi de bénéficier de conditions matérielles exceptionnelles, tant sur le plan informatique que logistique.

Je voudrais également exprimer toute ma reconnaissance aux personnes qui m'ont aidé dans les parties les plus ingrates de mon travail : corrections d'articles, relecture du rapport, préparation de la soutenance. Par égard pour ces personnes, je tairai leur identité.

Enfin, durant toute la période de ma thèse, et quelques fois depuis bien plus longtemps que cela, certaines personnes ont été amenées à me supporter quotidiennement ou presque, dans tous mes états, égayés comme égarés. Ces personnes sont douées d'un sens psychologique rare, d'une abnégation et d'une propension au sacrifice personnel hors du commun. Je dois aujourd'hui leur rendre un hommage particulier, et je les soulagerai sûrement en leur assurant que plus jamais, non, plus jamais ils ne revivront ça…

Merci à **Omid** *«Mon didou. Comment tu vas ? Moi ? Ben là, j'ai le nez ex–plo–sé.»* **Kohneh-Chahri**, **Mila** *«Galopin ! ! Descends du lit ! ! Y'a Omid dedans ! !»* **Kriloff** et **Galopin** *«Miâââooouuuu»* **Chahriloff**, **Laurent** *«Ch'peux pas, j'ai yoga.»* **Mugnier**, **Manuela** *«Ââââââââââh, mais vous êtes abominââââbles ! !»* **Santos** et **Black & Decker** *«Agaaah»* **Santonier**, **Le colonel** *«Ah, Verna, heureusement qu'on vous a. Sans ça, je ne sais fichtre pas ce qu'on aurait fait avec cet ordineur.»* **de Presle**, **Luisa** *«Non, mais vraiment, vous devriez vous abonner à Wanadoo, c'est les meilleurs. Et je dis pas ça passque j'y travaille.»* **Molina**, **Benoît** *«Non, mais vraiment, vous devriez prendre un Nokia. Sans blague, c'est les meilleurs. Et je dis pas ça passque j'y travaille.»* **Loth** et **Bruno** *«Agaaah»* **Moliloth**, **Papa** *«Oh, vous savez, les glomus carotidiens, c'est très surfait.»* **Verna**, **Maman** *«Vraiment, ta femme, je sais pas c'qu'elle te trouve.»* **Verna** et **Frangine** *«Chépa c'que j'ai en ce moment, j'ai mal au poignet.»* **Verna**, **Yves** *«J'ai décidé de monter une école de Jazz. Mais, le genre spécial, tu vois : spécialisé pour les iguanes. Y en a qui sont assez doués, si si.»* **Carbonne**, **Katia** *«Tu veux parler à Laurent ? Pas de problème : tu composes le message sur mon Palm, ensuite, je le transmets à mon portable par le port infrarouge, et hop ! Bon, sinon, tu peux lui parler directement aussi… il est dans la pièce à côté.»* **Hilal**, **Laurent** *«Elle est bien gentille, Katia, mais moi, j'ai pas de port infrarouge. Alors je tapes comme un mulet sur mon téléphone pour écrire trois mots.»* **Gros** et **Cloé** *«Agaaah»* **Grozillal**, **Muriel** *«Ah non, là, j'peux pas, je bosse. Ben oui, je sais qu'il est 23h45 et qu'on est samedi.»* **Bailet**, **Roman** *«'toute manière, le monde est pourri. D'ailleurs les gens sont tous pourris, la vie est pourrie. Et moi, je suis le plus pourri de tous.»* **Bressan-Carvallo** et **Jules** *«Agaaah»* **Carbailet**, **Stéphanie** *«Dis, si j'vais en boîte avec ta femme ce soir, ça t'embête ?»* **Polselli**, **Jolie-maman** *«Ah bon, elle est pas encore rentrée, laurence ? Mais tu sais où elle est ?»* **Loupiac** et **Joli-papa** *«Non mais attends, je vais te la faire tenir, moi, ton armoire. T'as de la ficelle ?»* **Loupiac**, **Cécile** *«Un jour, quand j'serai grande, j'irais ouvrir un resto au fond de l'afrique, avec un grand black à la cuisine.»* **Lemétayer**, **Nathalie** *«Non, mais ce mec, il était pas pour moi de toute façon. Enfin, quand même, ça fout un peu les boules. À trente piges, quoi… »* **Germain**, **Riton** *«Mais non, je fais pas de la musicologie. Vous autres en France, faut toujours que vous fassiez dans la litérature. Je fais de la mu–si–que.»* **Oppenheim**, **Nadine** *«Oh la la… c'est pas vrai… pfffffff… j'en ai mare, ce matin ! !»* **Richard** et **Samuel** *«Comment ? ! Tu n'utilises pas la version 12.28 beta 17 release 8 patchlevel 3 ? !»* **Tardieu**, **Isabelle** *«Bidou, ya Laurence qui voudrait changer la date de ses billets de train.»* **du boulot**, **Bidou** *«Tu pourras lui dire qu'à partir du quatorzième changement de date, ça va plus être possible ?»* **d'Isabelle du boulot** et **Nilou** *«Miâââooouuuu»* **Chat**, **Dominique** *«Bon, quand t'auras fini ta thèse, t'oublies pas de me garder le premier exemplaire, passque moi, je fais collec. Cela dit, c'est clair que je vais rien comprendre.»* **Poudevigne**, **Diane** *«Bon, quand t'auras fini ta thèse, t'oublies pas de lui garder le premier exemplaire, passque lui, y fait collec. Cela dit, c'est clair qu'il va rien comprendre, vu que de toute manière, il la lira pas.»* **Susmann** et **Marius** *«Agaaah»* **Poudemann**, **La Troupe** *«Ouais, mais là, vous êtes trop nombreux, je peux pas tous vous y mettre, sans ça mes remerciements, y tiendront plus sur une seule page.»* **de La Grange**, **Karine** *«Bon alors, pour notre liste de mariage, on a pensé à des baudriers en argent massif, deux paires de ski en marbre, et deux combinaisons de ski en soie pour aller avec, et »* **Lozach** et **Laurent** *«Didou, si tu veux des photos de ta lolo, j'en ai fait une ou deux. Ah oui, il faut que je te dise, si on voit pas bien son visage, c'est que j'ai cadré trop bas. Forcément, j'ai eu le regard attiré par autre chose… »* **Cochet**, **Akim** *«Eh, tu sais quoi ? Je suis co-mainteneur de Bison. Et puis je m'occupe d'Autoconf, aussi. Ah, et il va falloir que je sorte a2ps 4.15, et… »* **Demaille** et **Erika** *«Eh, tu sais quoi ? Je fais de l'Aïkido. Et puis je fais de la photo aussi. Ah, et il va falloir que j'apprenne à dessiner les kanjis, et… »* **Valencia**, **Jacques** *«Putain, mais quel con ! Mais quel con ! Mais pourquoi j'ai dissou ? !»* **Chirac**, **Théo** *«Que ça soit clair. Dis-toi qu'une pauvre idée, ben, déjà, c'est cool. En revanche, l'idée, c'est de se dire que j'prendrais bien une SB.»* **Dule**, **Sandrine** *«Alors dans ce film, si tu veux, l'histoire est ainsi. C'est une transcendance méta-lirique d'un moi parvenu, survenu, et même survécu, mais qui reste cependant accessible.»* **Rose** et **Laurent** *«Ouais, non mais c'est cool, quoi. Enfin, tu vois c'que j'veux dire ? Faut pas s'énerver, quoi.»* **Rouault**, **Pat** *«Didier, I think you're the best guitar player of the last decade.»* **Metheny**. **Deborah** *«Bon. Alors c'est bien. Je veux dire, vraiment. Mais il faut rester tour–né vers le pu–blic.»* **Dunn** et **Jeff** *«Mais non, Laurence, là, je crois que t'as rien compris, c'est pas de l'inceste : c'est juste les instincts primaires de tous les mecs de la planète.»* **Guermonprez**, **Laurence** *«Didou, je voudrais te parler. Pour l'an 2036, j'ai pensé à un super voyage. Tiens, d'ailleurs, j'ai acheté le guide du routard et j'ai récupéré 78 prospectus… »* **Loupiac**, **François** *«Non, mais la législation actuelle, si tu veux, pose le problème, suivant, c'est que… »* **Yvon** et **Adèle** *«vieux sac poubelle qui pue»* **Yvon**, **Catherine** *«Quoi ? ! Qu'est-ce qu'y z'ont mes yeux ? ! Mais non, je vous ai pas agressé avec mes yeux ! !»* **Millet**, **Christian** *«Non mais t'es sûr que tu veux encore jouer à Othello, tu va te faire encore te prendre une branlée monumentale.»* **Millet** et **Pauline et Charlotte** *«D'accord, on fait un bonhomme de neige. Mais je sais bien comment ça va finir. C'est Papilou qui va s'amuser avec.»* **Millet**, **Philippe** *«Rh, j'ai pas l'temps. Bon, attends… alors… vi cw metheny esc j j dd :x. Bon, allez, ça y est.»* **Dax**, **Ciara** *«Hey, techno dude, watch your vocabulary ! We don't use that word in Ireland !»* **Quill**.

Résumé

Compte tenu de l'état de l'art en matière de technologie, les systèmes actuels de réalité virtuelle doivent établir un compromis entre efficacité et utilisabilité : un écran d'ordinateur est d'emploi facile mais ne permet pas une bonne restitution du relief. Réciproquement, un casque de visualisation permet de fournir une vision stéréoscopique, mais constitue un périphérique encombrant. Pour résoudre ce type de dilemme, deux approches sont envisageables : une première approche, plus appliquée, consiste à améliorer la *qualité* de l'interaction Homme-Machine en perfectionnant les interfaces. Une deuxième approche, plus théorique, consiste à *assister* l'humain dans sa tâche en diminuant sa charge de travail et donc la quantité d'interaction nécessaire.

Notre travail porte sur la notion d'assistance et se situe par conséquent dans la deuxième approche : nous analysons sous un angle **cognitif** la notion d'assistance à l'humain dans un contexte de réalité virtuelle, et nous montrons en particulier l'intérêt du concept de **détection d'intentions** comme source d'assistance intelligente. Le travail est organisé en trois grandes parties : l'élaboration d'un point de vue cognitif sur la notion de réalité virtuelle, une étude théorique sur la notion d'assistance dans les systèmes de réalité virtuelle, et une approche fondée sur la détection d'intentions.

La première partie est consacrée à l'élaboration théorique d'une caractérisation **cognitive** du **virtuel**. Plutôt que de définir le virtuel en tant que tel, nous montrons qu'il est plus intéressant de définir les **processus de virtualisation**, c'est à dire les mécanismes cognitifs qui font qu'à partir d'une situation réelle, l'humain entre progressivement dans le virtuel. Par exemple, à partir de la situation réelle où un humain est devant un écran d'ordinateur, face à un monde de synthèse, on parlera de «virtualisation» au moment où celui-ci *oublie* qu'il s'agit d'un écran et *pense* qu'en se déplaçant, son point de vue sera modifié.

Dans un premier temps, un modèle cognitif décrivant l'interaction entre un humain et son environnement est proposé, le modèle $\boldsymbol{M_RIC}$. Les processus de virtualisation sont ensuite définis et nous permettent d'obtenir un certain nombre de variantes de ce modèle, décrivant les grandes catégories de situations virtualisées : situations de type CAO (avec environnement virtuel), situations dites «immersives» (avec représentation virtuelle de l'humain) etc.

La deuxième partie est consacrée à une étude théorique de la notion d'**assistance**. Notre objectif est d'étudier de quelle manière l'assistance à l'humain peut être mise en œuvre dans les systèmes de réalité virtuelle. Nous commençons par élaborer un modèle décrivant les situations d'interaction entre un humain et de tels systèmes, le modèle $\boldsymbol{L_iSA}$. Ce modèle fait intervenir un certain nombre de composantes (opérateur, interface, système de réalité virtuelle …) qui interagissent

en échangeant de l'information. L'apparition d'un processus d'assistance à l'humain dans une situation donnée a pour effet de modifier l'information véhiculée entre les différentes composantes du modèle.

Cette particularité nous permet d'analyser des exemples d'assistance caractéristiques des applications actuelles selon deux axes : quelle(s) composante(s) du modèle contiennent ces processus d'assistance, et comment ceux-ci modifient-ils l'information (enrichissement, filtrage, altération…).

La troisième partie décrit une approche fondée sur la **détection d'intentions** . L'idée générale est de concevoir un système qui, en fonction des actions de l'opérateur et de l'état de l'environnement, tente de déterminer l'opération que celui-ci est en train d'exécuter, et le cas échéant, prend le contrôle de l'exécution. Dans cette optique, nous commençons par présenter un modèle arborescent de représentation des connaissances servant à décrire les actions susceptibles d'êtres entreprises par l'opérateur. Puis, nous établissons les critères permettant au système d'évaluer la probabilité pour qu'une action dont il a connaissance soit effectivement celle entreprise par l'opérateur, puis nous proposons un jeu de processus permettant d'évaluer ces critères à tout instant.

Le mécanisme de détection d'intentions proposé, ainsi que son extension à des fonctionalités d'**exécution** et de **simulation** des opérations qu'il est initialement capable de détecter font l'objet d'une illustration à travers un démonstrateur informatique, le système T_OAS_t.

Abstract

Given the state-of-the-art in technology, current virtual reality systems must establish a compromise between efficiency and usability : computer screens, although they are easy to use, cannot correctly provide images in three-dimensional form. On the other hand, head mounted displays provide a good stereoscopic view, but these however are cumbersome devices. In order to find a solution to this sort of dilemma, two approaches are possible : the first, rather applicative, approach would be to improve the *quality* of human-machine interaction by making interfaces more user-friendly. The second, rather theoretical, approach would be to *assist* the person in his work by diminishing his workload and thereby reducing the quantity of interaction necessary.

Our work deals with the idea of assistance and is therefore of relevance to the second approach mentioned above. From a **cognitive** point of view, we analyze the idea of assisting the human being within a virtual reality context, and we particularly show the benefit of such a concept of **intention detection** as a possible source of intelligent assistance. This work is composed of three parts ; firstly, the elaboration of a cognitive view on virtual reality ; second, a theoretical study of the notion of assistance within virtual reality systems, and finally, an approach which is based on intention detection.

The first part is dedicated the theoretical elaboration of a **cognitive** characterization of virtual reality. Rather than defining virtual reality as such, we show that it is more interesting to define the **virtualisation processes**, that is the cognitive mechanisms by which a human being progressively goes from the real situation into virtuality. For instance, in a real situation whereby a human being is sitting down in front of a computer screen, facing a synthetic world, we speak of the idea of "virtualisation" when this person *forgets* that he is looking at a computer monitor, and ends up *believing* that his point of view will be affected if he himself moves.

In a first step, a cognitive model that describes the interaction between a human being and his environment is provided, known as the $\boldsymbol{M_RIC}$ model. We then define the virtualisation processes which allow us to obtain several variants of the initial model, describing the major categories of virtualized situations : CAD situations (with a virtual environment), so-called "immersive" situations (with a virtual representation of the human being) etc.

The second part is dedicated to a theoretical study of the notion of **assistance**. Our aim is to analyze the different ways assistance can be implemented in virtual reality systems. We begin by producing a model describing the situations whereby a human being interacts with such a system, known as the $\boldsymbol{L_iSA}$ model. This model comprises several components (including an operator, an interface, a virtual reality system…) which interact by exchanging information. When an assistance process is introduced to the situation, this has the effect of modifying the information

conveyed within the model components.

This allows us to analyze cases of assistance that currently exist in two ways : which component of the model contain the assistance processes, and how do they modify the conveyed information (i.e. by filtering, enrichment, alteration etc.).

The third and final part describes an approach which is based on the concept of **intention detection** . The idea behind this is to design a system which, bearing in mind the actions performed by the operator at present and the current state of the environment, attempts to determine the global operation that the operator is in the process of executing, and eventually takes control of the same process. First of all, we produce a knowledge representation model used to describe the actions an operator may undertake. Then, we establish the criteria that allow the system to evaluate the probability of a particular action which it knows will be the one undertaken by the operator. In the end we put forward a set of algorithms which may be used to compute these criteria in real time.

The proposed intention detection mechanism, in addition to its extension to **execution** and **simulation** capabilities, are illustrated through a computer demonstrator, known as the T_{OAS_t} system.

Table des matières

Avant Propos 1
Avertissements 1
Guides de lecture 2
Introduction Générale 5
Motivations 5
Apports 7
Organisation 8

I RÉALITÉ VIRTUELLE 11

Comment définir le Virtuel ? Le modèle MRIC

1 Introduction 13
2 Historique 15
2.1 Les premiers pas 15
2.2 Vers la synthèse d'image 16
2.3 Les interfaces immersives 16
2.4 Le marché de la Réalité Virtuelle 17
2.5 Situation actuelle 17
3 Définitions Actuelles 19
3.1 Temps-Lieu-Interaction 19
3.2 Interaction-Immersion-Imagination 20
3.3 Perception-Proprioception 21
3.4 Réalités réalistes et artificielles 22
3.5 Réalités réalistes et altérées 22
3.6 L'approche philosophique 22
3.7 Confrontation 23
3.7.1 Importance de l'interaction 23
3.7.2 La composante immersive 23
3.7.3 La composante infographique 23
3.7.4 La composante cognitive 24
3.7.5 Catégorisation des applications 24

3.7.6 Cas de la télé-opération 24
4 Le Modèle M_RIC 25
4.1 Opérateur *vs.* Environnement 25
4.2 Mental *vs.* Physique 26
4.3 Explicite *vs.* Implicite 27
4.4 Terminologie 29
4.5 Modèle complet 30
4.5.1 Chemins d'information 30
4.5.2 Modules terminaux 31
4.5.3 Traitement de l'information 31
4.5.4 Niveau conceptuel de l'information 33
4.5.5 Interaction : les quatre «I» du réel 34
4.6 Notion de «réalité» 34
4.6.1 Classes et instances du réel 35
4.6.2 Importance de l'interaction 35
4.6.3 Notion de prédiction 35
5 La Réalité Virtuelle 37
5.1 Le virtuel : une méta-définition 37
5.1.1 Processus de virtualisation 37
5.1.2 Types de virtualisation 38
5.1.3 Importance de l'interaction 39
5.2 Virtualisation de l'environnement 40
5.2.1 Description 40
5.2.2 Virtualisation de l'action 41
5.2.3 Virtualisation de la perception 42
5.2.4 Types d'environnement virtuels 42
5.2.5 Résumé 43
5.3 Virtualisation de l'agent physique 43
5.3.1 Description 43
5.3.2 Virtualisation de l'action 45
5.3.3 Virtualisation de la perception 45
5.3.4 Types de mondes virtuels 46
5.3.5 Résumé 46
5.4 Imbrication de la réalité virtuelle 47
5.5 Virtualité des environnements réels 47
5.5.1 Réalité virtuelle, modèle M_RIC et triangle sémiotique 48
5.5.2 Immersion réelle 49
5.5.3 Pseudo-immersion 50
5.6 La réalité augmentée : un cas litigieux 50
5.6.1 Altération de l'agent externe 51
5.6.2 Altération de l'agent opératoire 52
5.7 Réalité virtuelle et télé-opération 53
5.7.1 Description 53
5.7.2 Catégories de télé-opération 53
5.7.3 Télé-opération et réalité augmentée 54
5.7.4 Transmutation virtuelle 54

5.8 Détection d'intentions 55
5.9 Problèmes liés à l'immersion 55
5.9.1 Gestion des interactions 56
5.9.2 Gestion de la proprioception 56
5.9.3 Les «paradoxes» prédictifs 57
5.9.4 Problèmes issus des cas réels 58
5.10 Réalité virtuelle et langage 59
6 Conclusion 61

II ASSISTANCE 63

Une théorie de l'Assistance : Sémantique et Localisation

1 Introduction 65
2 Le Modèle *LiSA* 67
2.1 Côté opérateur : l'interface 67
2.2 Côté environnement : le manipulateur 68
2.3 Jonction : le système 69
2.4 Modèle complet 70
2.4.1 Chemins d'information 70
2.4.2 Traitement de l'information 71
3 Étude de Cas 73
3.1 Génération de perception 73
3.1.1 Exemples 73
3.1.2 Sémantique 74
3.1.3 Localisation 74
3.2 Altération de la perception 75
3.2.1 Exemples 75
3.2.2 Sémantique 75
3.2.3 Localisation 76
3.3 Transmodalisation de la perception 76
3.3.1 Exemples 76
3.3.2 Sémantique 77
3.3.3 Localisation 77
3.4 Reconstitution de perception 78
3.4.1 Exemples 78
3.4.2 Sémantique 79
3.4.3 Localisation 79
3.5 Filtrage de l'action 80
3.5.1 Exemples 80
3.5.2 Sémantique 80
3.5.3 Localisation 81
3.6 Commande paramétrique 81
3.6.1 Exemples 81
3.6.2 Sémantique 82
3.6.3 Localisation 82

3.7 Exécution virtuelle ... 83
3.7.1 Exemples ... 83
3.7.2 Sémantique ... 83
3.7.3 Localisation ... 83
3.8 Résumé ... 84
4 Symétrisation des Exemples ... 87
4.1 Méthodologie ... 87
4.1.1 Symétrie du modèle *LiSA* ... 87
4.1.2 Inversion des exemples ... 88
4.2 Génération d'action ... 88
4.2.1 Sémantique et localisation ... 88
4.2.2 Application ... 89
4.3 Altération de l'action ... 89
4.3.1 Sémantique et localisation ... 90
4.3.2 Application ... 90
4.4 Transmodalisation de l'action ... 91
4.4.1 Sémantique et localisation ... 91
4.4.2 Application ... 92
4.5 Reconstitution d'action ... 92
4.5.1 Sémantique et localisation ... 92
4.5.2 Application ... 93
4.6 Exécution paramétrique ... 94
4.6.1 Sémantique et localisation ... 94
4.6.2 Application ... 95
4.7 Filtrage de la perception ... 95
4.7.1 Sémantique et localisation ... 95
4.7.2 Application ... 95
4.8 Commande virtuelle ... 96
4.8.1 Sémantique et localisation ... 96
4.8.2 Application ... 97
4.9 Résultats ... 98
4.9.1 Classes sémantiques d'assistance ... 98
4.9.2 La réalité augmentée vue en sens inverse ... 99
5 Niveau Conceptuel de l'Assistance ... 101
5.1 Analogies entre *MRIC* et *LiSA* ... 101
5.2 Télé-Opération Immersive ... 102
5.3 Télé-Opération «brute» ... 103
5.4 Télé-Opération semi-autonome ... 105
5.5 Télé-Opération et Détection d'Intentions ... 106
5.6 Résumé ... 107
6 Conclusion ... 111

III DÉTECTION D'INTENTIONS 113

Assistance intelligente à la Télé-Opération : Détection d'Intentions

1 Introduction 115
2 Généralités 117
2.1 Capacités d'un système de détection d'intentions 117
2.1.1 Le compromis asservissement / autonomie 117
2.1.2 Répartition des compétences 118
2.1.3 Niveau conceptuel et complexité exécutoire 118
2.1.4 Application à $TAO_t AS_t$ 119
2.2 Détecter, ou *ne pas* détecter ? 119
2.2.1 Hiérarchie des actions 120
2.2.2 Multiplicité de la détection 120
2.3 Résumé 121
3 Représentation des Connaissances 123
3.1 Expérimentation 123
3.1.1 L'expérience 123
3.1.2 Données recueillies 124
3.2 Antécédents 125
3.2.1 Composantes d'une action 126
3.2.2 Formalisme de représentation 126
3.3 Structure Générique 127
3.3.1 Représentation arborescente des actions 127
3.3.2 Formalisation syntaxique préliminaire 128
3.3.3 Formalisation sémantique préliminaire 129
3.3.4 Exemple : saisie d'un objet 130
3.3.5 Comparaison avec les réseaux procéduraux 132
3.4 Catégories d'action 133
3.4.1 Problématique 133
3.4.2 Affinements du modèle 134
3.4.3 Formalisation syntaxique affinée 135
3.4.4 Formalisation sémantique affinée 135
3.4.5 Exemple : saisie d'un objet 138
3.5 Application à $TAO_t AS_t$ 140
3.5.1 Notion d'action terminale 140
3.5.2 Les actions terminales de $TAO_t AS_t$ 140
3.5.3 Action de saisie 142
3.6 Formalisation 142
3.6.1 Les actions et leur mode 143
3.6.2 Les branches et leur qualité 143
3.6.3 Syntaxe formelle du modèle 143
3.6.4 Dynamique du modèle 144
3.7 Caractéristiques du modèle 146
3.7.1 Mutuelle référence 146
3.7.2 Réutilisabilité des actions 147
3.7.3 Représentation implicite des buts 147
3.7.4 Opérationalisation 148
3.7.5 Actions (terminales) et schèmes 148
3.7.6 Correction *vs.* anticipation 149

3.7.7 Validation syntaxique et sémantique des opérations 149
3.7.8 Configurations particulières 150
3.7.9 Équivalence sémantique d'actions séquentielles 150
4 Détection d'Intentions 153
4.1 Taux de vraisemblance d'une opération 153
4.1.1 Composition des taux de vraisemblance 153
4.1.2 Pondération des actions 154
4.1.3 Formalisation 156
4.2 Algorithmes de détection 157
4.2.1 Algorithme préliminaire 157
4.2.2 Généralisation du seuil de confiance 158
4.2.3 Détection d'une action parallèle 158
4.2.4 Détection d'une action séquentielle 159
4.2.5 Actions englobantes 160
4.3 Cas des actions terminales 161
4.3.1 Taux de vraisemblance 161
4.3.2 Pondération 163
4.3.3 Algorithme de détection 163
4.3.4 Formalisation 163
4.4 Algorithme système 164
5 Exécution et corollaires 169
5.1 Algorithmes d'exécution 169
5.1.1 Relation entre détection et exécution 169
5.1.2 Exécution d'une action parallèle 170
5.1.3 Exécution d'une action séquentielle 171
5.2 Cas des actions terminales 172
5.2.1 Algorithme d'exécution 172
5.2.2 Application à $\mathcal{TOAS}_t$ 172
5.3 Corollaires 173
5.3.1 Simulation 173
5.3.2 Détection d'erreur 173
5.4 Limitation 175
5.4.1 Proprioception étendue 175
5.4.2 Conséquence 176
6 Un exemple de scénario 179
7 Conclusion 189

CONCLUSION 191

Résultats 192
Définir cognitivement le virtuel 192
Assistance en réalité virtuelle 192
Détection d'intentions 193
Perspectives 193
Mise en œuvre 194

Discussion 195

ANNEXES 197

A Description de TAS_t 199
B Preuve du théorème 207
C Table des figures 209
D Liste des définitions 213
E Glossaire 215
F Index 219
G Bibliographie 223

Avant Propos

Avertissements

En préalable à la lecture de ce document, il est conseillé de prendre connaissance et de garder à l'esprit les points qui vont suivre. Ces avertissements ont entre autres pour objet d'éviter toute ambiguïté ou mauvaise interprétation du texte qui suit.

Notion de «cognition»

La notion de «cognition» ou d'approche «cognitive» d'un problème peut revêtir de nombreuses formes, et contient par conséquent une grande part d'ambiguïté. Dans la mesure où cette notion est omniprésente dans le travail que ce document s'attache à détailler, il nous semble important d'apporter des précisions quant à l'acception que nous avons de cette notion.

Les «sciences cognitives» s'attachent à étudier les processus mentaux de l'humain. Ces études portent sur donc sur les mécanismes de perception, d'acquisition de connaissance, de réflexion, de mémoire... L'intelligence artificielle, de son côté, tente de modéliser les processus mentaux de l'humain afin d'en reproduire les fonctions. Quand nous parlons d'«approche cognitive» d'un problème en intelligence artificielle, nous pouvons faire référence à deux interprétations bien connues du concept :

- En se plaçant à un haut niveau d'abstraction, un système d'intelligence artificielle peut tenter de reproduire les *capacités* humaines de raisonnement, de manipulation des connaissances, d'intelligence *etc*. Il s'agit dans ce cas d'une «obligation de résultat» : les domaines de l'intelligence artificielle concernés sont par exemple le traitement du langage naturel et les systèmes experts.
- En se plaçant à un faible niveau d'abstraction, un système d'intelligence artificielle peut tenter de reproduire la *manière* dont ces capacités sont mises en œuvre, en termes de représentation des connaissances, perception, mémoire *etc*. Il s'agit dans ce cas d'une «obligation de moyens» : les domaines de l'intelligence artificielle concernés sont par exemple les réseaux neuro-mimétiques.

Cette distinction est cruciale dans la mesure où un système d'intelligence artificielle du premier type, c'est à dire qui s'attache à reproduire l'intelligence humaine, n'a nul besoin de reproduire la manière dont cette intelligence est mise en œuvre chez l'humain pour parvenir à ses fins. Par exemple, les systèmes experts ou de traitement du langage naturel n'utilisent en général pas de réseaux de neurones, mais des techniques de traitement symbolique d'information.

Dans le travail que ce document relate, nous avons adopté une approche cognitive du premier type. Nous nous sommes par conséquent intéressés à la modélisation de certains aspects de l'intelligence et du raisonnement humain sans pour cela éprouver le besoin d'ancrer cette modélisation dans une approche du deuxième type.

Divers...

Terminologie Comme dans toute discipline scientifique éprouvant le besoin de formaliser la connaissance qu'elle manipule, nous sommes amenés à mettre en place une terminologie spécifique à notre travail. Le lecteur devra donc être attentif au fait qu'un certain nombre de mots du vocabulaire courant sont définis ou employés ici dans un sens très spécifique. Ceci constitue un artefact inévitable de toute verbalisation de la pensée scientifique. Cependant, par soucis de clarté nous nous sommes efforcés de choisir des termes dont le sens commun reste proche de celui que nous adoptons, de manière à ce que l'emploi de ces termes reste, dans une certaine mesure, intuitif.

Redondance Une lecture attentive et séquentielle de ce document révélera par endroits une certaine redondance dans l'information dispensée. Cette redondance est volontaire, le lecteur voudra bien nous en excuser. Nous pensons que d'une part, une idée est plus facile à assimiler quand elle est formulée de plusieurs manières, et que d'autre part, les redites servent l'autonomie de chaque section du document. Le lecteur pourra donc, dans une certaine mesure, avoir une lecture non séquentielle du document, les points essentiels à la compréhension étant rappelés au moment opportun.

Progressivité La manière traditionnelle d'exposer un contenu scientifique consiste le plus souvent à exposer une théorie dans sa version finale, et débattre de ses qualités dans un deuxième temps. Il nous semble cependant que l'assimilation d'une théorie est plus naturelle et conviviale lorsque le lecteur a la possibilité de suivre pas à pas son élaboration. Bien qu'elle soit moins aisée à mettre en œuvre, c'est l'approche que nous avons décidé de suivre dans ce document : les théories, modèles et autres algorithmes présentés dans cet ouvrage font tous l'objet d'une élaboration progressive. Chaque étape de l'élaboration introduit une problématique que l'étape suivante résoud, jusqu'à la version finale. La contrepartie de cette progressivité est que le lecteur peut être amené à se poser des questions auxquelles une réponse ne sera fournie qu'ultérieurement. Dans un tel cas, le lecteur est invité à garder sa question en tête, poursuivre la lecture, et se faire rembourser si aucune réponse satisfaisante ne lui est fournie dans les vingt pages suivantes...

Guides de lecture

Afin d'aider le lecteur à parcourir cet ouvrage, un soin tout particulier a été apporté à son organisation ainsi qu'aux différents indices textuels et graphiques qu'il propose. Les éléments suivants sont destinés à faciliter la navigation en son sein.

Guides textuels

Le lecteur trouvera en fin de document une annexe contenant entre autres un certain nombre de listes, un glossaire, un index et une bibliographie.

- Les listes sont regroupées selon trois thèmes : liste des figures, liste des définitions / points de vocabulaire / formulations / reformulations, et liste des algorithmes. Chaque élément d'une liste renvoi à la page concernée dans le document.
- Le glossaire contient des précisions sur certains termes et acronymes utilisés courament dans ce document. Le lecteur est invité à le consulter lors de la première apparition dans le corps du texte de chaque terme y figurant.
- L'index contient des entrées par mot clé et par thème. Chaque citation bibliographique y est également présente et accompagnée de la liste des pages où elle apparaît.
- Enfin, chaque entrée bibliographique est commentée. En général, un court résumé de la référence concernée est fourni, ce qui permettra au lecteur de juger de l'intérêt de celle-ci.

Enfin, il est à noter que ce document peut être visionné interactivement dans sa version PDF en couleurs, auquel cas toutes les références internes (citations, figures, renvois, entrées de l'index et des listes. . .) et externes (liens Internet) sont pourvus d'hyperliens cliquables.

Indices visuels

Ce symbole est utilisé en marge d'une section. Il indique que le contenu textuel associé est à dominante bibliographique, et ne constitue donc pas un apport direct du travail.

Ce symbole est utilisé en marge d'une section, d'un paragraphe ou d'un élément formel (définition, vocabulaire, formulation. . .). Il indique que le contenu textuel associé constitue un point important devant rester à l'esprit du lecteur, et dont le reste du document est susceptible de dépendre.

Ce symbole est utilisé en marge d'une section, d'un paragraphe ou d'un élément formel (algorithme). Il indique que le contenu textuel associé est à composante applicative.

Ce ***claviuscript*** *a été réalisé exclusivement à l'aide de logiciels libres :* XEmacs *(http ://www.xemacs.org) pour l'édition,* LaTeX 2ε *(http ://www.tug.org) pour la typographie et* a2ps *(http ://www.infres.enst.fr/~demaille/a2ps) pour l'impression.*

Introduction Générale

Grâce en particulier aux rapides progrès de la technologie, la notion de réalité virtuelle a connu récemment un surcroît d'intérêt, de la part du grand public autant que de la communauté scientifique. Les applications sont désormais nombreuses : simulation, conception assistée par ordinateur, aide à la chirurgie, environnements virtuels coopératifs, distribués, jeux vidéos... Les domaines de recherche le sont tout autant : infographie, logiciels et matériels de réalité virtuelle, modélisation physique, capteurs, périphériques, systèmes distribués, temps-réel...

Si la réalité virtuelle n'avait pour but que de donner l'*illusion de la réalité*, on pourrait presque dire qu'elle est née avec l'Homme. Les peintures préhistoriques représentant des scènes de chasse vont par exemple déjà dans ce sens. Nous verrons cependant que la réalité virtuelle inclut également une dimension d'*interaction* avec cette illusion. Cette préoccupation est quant à elle beaucoup plus récente, puisqu'elle date approximativement des années 60, et qu'il fallut attendre encore bien longtemps avant que la technologie permette à la pensée de prendre racine. De ce point de vue, la réalité virtuelle n'en est encore qu'à ses débuts...

Motivations

Dans le foisonnement scientifique actuel, la réalité virtuelle, en tant que discipline à part entière, est caractérisée par trois aspects qui constituent nos principales sources de motivation pour ce travail.

Réalité virtuelle et cognition

La réalité virtuelle est en premier lieu une discipline résolument tournée vers l'Homme. Selon le sens commun qui lui est accordé, et sur lequel nous reviendrons abondament, un système de réalité virtuelle a pour but essentiel de fournir *à l'humain* une illusion de réalité. Un tel système est par essence construit pour être utilisé par l'Homme. Celui-ci en est donc une composante à part entière : la réalité virtuelle «s'adresse à l'Homme», par définition. On perçoit donc dès à présent l'importance capitale de la composante cognitive dans l'évolution de cette discipline.

Face à cette caractéristique de la réalité virtuelle, trois remarques s'imposent :

- Paradoxalement, l'axe cognitif n'est pas, loin s'en faut, l'optique la plus présente dans les travaux scientifiques actuels : le récent regain d'intérêt pour le domaine provient essentiellement des avancées technologiques de ces dernières années, et c'est selon ce même axe que l'on trouve l'activité scientifique la plus importante.
- Les quelques études cognitives menées à l'heure actuelle sont essentiellement axées vers la psychologie expérimentale. Le but fondamental de ces études reste en général d'étudier l'efficacité et les effets, au niveau perceptif, des périphériques matériels utilisés dans les systèmes de réalité virtuelle, de manière à améliorer l'interaction avec ceux-ci, notamment en matière d'immersion.
- D'autre part, avec l'expression «réalité virtuelle » sont apparu de nombreux vocables («réalité augmentée », «réalité artificielle»…) dont le sens reste encore flou et fréquemment propre aux auteurs qui les utilisent ou les proposent. Il semble donc que la notion de «réalité virtuelle » elle-même ne soit pas encore clairement définie, autrement que par la prise en compte des moyens technologiques qui sont mis en œuvre.

Dans ce contexte, la question d'une définition de la réalité virtuelle d'un point de vue cognitif, et non pas technologique, apparaît comme un enjeu crucial pour l'avenir de cette discipline. Qu'est-ce vraiment que la réalité virtuelle pour un humain qui en fait l'expérience ?

Réalité virtuelle et assistance

Le fait que la réalité virtuelle soit une discipline tournée vers l'humain lui confère un aspect foncièrement transversal : hormis son intérêt propre, la création et l'utilisation d'environnements virtuels pour eux-mêmes, la réalité virtuelle est potentiellement utile à tout domaine d'activité nécessitant d'établir une interface entre l'humain et un système artificiel, puisque c'est exactement sa place. Ceci est particulièrement vrai dans des domaines tels que la télé-opération ou la conception assistée par ordinateur, où son utilisation déjà importante est toujours croissante.

Chaque fois que la réalité virtuelle est utilisée dans une discipline jusque là indépendante, il existe une volonté commune, sous-jacente à son utilisation. Cette volonté s'exprime au moyen d'un simple mot : «assistance». En effet, l'utilisation de la réalité virtuelle dans d'autres disciplines scientifiques a toujours pour objectif d'assister l'humain dans son interaction avec le système artificiel qu'il manipule. La conception assistée par ordinateur est grandement facilitée quand l'utilisateur du système perçoit le relief. Il en va de même pour la télé-opération.

Cependant, cette volonté d'assistance à l'humain ne date pas de l'apparition de la réalité virtuelle dans les applications. La plupart des avancées en matière d'informatique et d'ergonomie (les interfaces graphiques par exemple) allaient déjà dans ce sens. Si le besoin d'utiliser les techniques de réalité virtuelle en matière d'assistance à l'humain s'est fait sentir, c'est donc vraissemblablement que cette discipline a un apport qui lui est propre.

Partant de cette constatation, nous comprenons qu'il est également capital de bien percevoir la spécificité de la réalité virtuelle en matière d'assistance à l'humain : quel bénéfice tire t'on de son utilisation, qu'est-ce qui relève de la réalité virtuelle, ou qu'est-ce qui n'en relève pas dans une procédure d'assistance particulière, comment cette discipline s'articule t'elle avec celles qui l'utilise ?

Réalité virtuelle : fidélité *vs.* intelligence

Dans un cadre applicatif tel que celui de la télé-opération, la réalité virtuelle est essentiellement utilisée à des fins immersives : le but est de mettre l'utilisateur du système en contact direct avec l'environnement de travail, comme si celui-ci était physiquement présent dans cet environnement. Dans cette optique, nous voyons que le soucis majeur d'un tel système est de transmettre le plus *fidèlement* possible les informations perceptives (à destination de l'humain) ou actionnelles (en sa provenance).

Cette notion de fidélité s'avère être omniprésente dans les travaux actuels sur la réalité virtuelle, en particulier du point de vue technologique : l'infographie progresse afin de fournir des images toujours plus réalistes a une cadence satisfaisante pour l'œil humain, de nouveaux périphériques d'immersion apparaissent, restituant toujours plus vite et plus fidèlement la vision en relief, le son spatialisé, les sensations tactiles…

Quand un système de télé-opération utilise la réalité virtuelle à des fins d'immersion, le concept sous-jacent est par conséquent que la *fidélité* de l'information véhiculée au travers du système est une source d'assistance à l'humain. Cependant, avant l'arrivée de la réalité virtuelle dans ses applications orthogonales, d'autres disciplines envisageaient l'*intelligence* plutôt que la fidélité comme une source assistance. Des notions telles que l'ergonomie des interfaces, les aides contextuelles, les systèmes d'aide à la décision ainsi que de nombreux autres domaines de l'intelligence artificielle et de l'interaction Homme-Machine vont dans ce sens.

Un troisième enjeu de la réalité virtuelle prend donc toute son ampleur ici même : la réalité virtuelle ne doit-elle être considérée que comme un transporteur précis et fiable d'information, ou peut-on y voir autre chose ? En particulier, est-il possible d'interpréter ce concept en termes d'assistance *intelligente* à l'humain ?

Apports

Le travail présenté dans ce document s'est attaché à fournir des éléments de réponse aux trois problématiques évoquées précédemment. Les apports de celui-ci de déclinent donc également selon trois axes.

Une vision cognitive du virtuel

Le premier apport de ce travail se situe sur un plan théorique. Nous proposons une définition possible de la réalité virtuelle sous un angle cognitif et non pas technologique. Il ne s'agit en fait pas d'une définition statique de la réalité virtuelle, au sens traditionnel du terme, mais plutôt d'une méta-définition, dynamique, de celle-ci : la réalité virtuelle est définie, à un instant donné, en fonction de la situation considérée et de la manière dont celle-ci est appréhendée par l'humain qui en fait l'expérience. Notre définition dépend par conséquent de deux paramètres que sont la situation et l'humain qui la vit. Il en découle qu'une situation donnée *est* virtuelle pour un individu si celui-ci la *ressent* comme telle, ce qui n'est pas forcément le cas pour d'autres individus.

Cette réflexion nous permet tout d'abord de replacer l'approche cognitive au niveau qui lui revient dans la discipline. D'autre part, notre démarche repositionne l'ensemble des point de vues actuels sur la notion autour d'un axe commun, et étend le champ d'application du concept de réalité virtuelle aux environnements réels et pas seulement aux mondes de synthèse.

Assistance en réalité virtuelle

Le deuxième apport de ce travail se situe sur un autre plan théorique. Nous proposons une réflexion mettant en évidence la manière selon laquelle le concept de réalité virtuelle se décline en termes d'assistance à l'humain. Cette réflexion prend appui sur un cadre applicatif tel que celui de la télé-opération. Au travers d'exemples d'assistance déjà opérationnels, nous montrons comment le concept de réalité virtuelle est utilisé en télé-opération selon deux axes : un axe de *sémantique* tout d'abord, c'est à dire quel type de traitement de l'information est mis en œuvre, puis un axe de *localisation*, c'est à dire à quel endroit du système le processus d'assistance intervient.

La théorie de l'assistance ainsi proposée a tout d'abord l'avantage d'unifier conceptuellement les domaines de la réalité virtuelle et de la télé-opération : certains types d'assistance *a priori* découplés relèvent en fait des mêmes concepts. D'autre part, elle permet de mettre en évidence un certain déséquilibre dans l'interprétation actuelle de la notion d'assistance, et fournit une méthode systématique pour corriger ce déséquilibre.

Détection d'intentions

Le troisième apport de ce travail se situe sur un plan applicatif. Les deux réflexions précédentes ont pour corollaire de faire apparaître, dans le cadre de notre étude, le concept de détection d'intentions comme un important degré de virtualisation d'une situation d'une part, et comme un important degré d'intelligence d'un système d'assistance à l'humain d'autre part. Nous proposons donc une mise en œuvre pratique de ce concept. La détection d'intentions, dans un contexte de réalité virtuelle, trouve son intérêt dans des systèmes capables de communiquer avec l'humain à un niveau conceptuel élevé, sans pour cela être doté de capacités de raisonnement aussi importantes.

La mise en œuvre du concept de détection d'intentions proposée ici passe par l'élaboration d'un modèle générique de représentation des connaissances, ainsi que par la conception d'un ensemble d'algorithmes utilisant les connaissances ainsi inculquées au système. Ces éléments ont été appliqués dans l'élaboration d'un démonstrateur appelé $TOAST$.

Organisation

Ce document décrit avec précision la démarche suivie pour répondre aux trois problématiques décrites précédemment, et se décompose par conséquent en trois parties, chacune d'elle sensiblement dédiée à l'une des motivations de départ.

Première partie

La première partie de ce document s'attache à élaborer une définition cognitive de la réalité virtuelle. Le premier chapitre propose un rapide historique du développement de cette discipline. Un certain nombre de définitions déjà proposées sont ensuite examinées, et les points particulièrement intéressants sont notés. Finalement, une définition de la réalité virtuelle est proposée. Cette définition est élaborée en plusieurs étapes :

- Premièrement, nous modélisons une situation d'interaction dite «réelle» entre un humain et son environnement. Ce modèle ne tente pas de *définir* le réel, mais en fixe une vision cognitive selon l'axe principal de l'interaction. Cette vision du réel sert de base à l'élaboration d'une définition du virtuel.
- Ensuite, nous définissons la réalité virtuelle non comme un ensemble de critères auxquels telle ou telle situation doit répondre, mais plutôt comme un processus de transformation, ou de «virtualisation» de la situation réelle décrite précédemment. Ce processus de virtualisation doit cependant répondre à un ensemble de règles qui sont détaillées.

Deuxième partie

La deuxième partie de ce document a pour objet d'éclaircir les liens qui existent entre la notion de réalité virtuelle et celle d'assistance à l'humain. Dans un premier temps, nous proposons l'éla-

boration d'un modèle permettant de décrire les situations de télé-opération, réalité augmentée ou réalité virtuelle. Le modèle proposé ici se focalise sur les aspects fonctionnels de l'interaction : il décrit les composantes matérielles impliquées dans les situations considérées.

Dans un deuxième temps, nous étudions un certain nombre de cas d'assistance à l'opérateur représentatifs des grandes classes d'applications actuelles. Ces exemples sont étudiés en référence à notre modèle selon deux axes : d'une part en termes de *localisation* des processus de traitement de l'information qu'ils mettent en œuvre, et d'autre part en termes de *sémantique* du traitement de l'information qu'ils opèrent.

Cette étude nous permet ensuite, par un procédé systématique de symétrisation, d'extrapoler les exemples étudiés à de nouveaux cas intéressants, et de montrer en quoi certaines notions comme celle de réalité augmentée on été jusqu'ici envisagées de manière incomplète.

Finalement, nous montrons que la présence de fonctionalités d'assistance à l'opérateur dans une situation donnée entraîne une augmentation du niveau conceptuel de l'information véhiculée au travers du modèle, comparativement à la même situation non assistée.

Troisième partie

La troisième et dernière partie de ce document est dédiée à l'étude de la détection d'intentions dans ses aspects théoriques et opérationnels. Dans un premier temps, nous introduisons le thème de la détection d'intentions d'un point de vue général. Nous déterminons dans quel cadre l'application du concept de détection d'intentions peut s'avérer utile, et quelles sont les fonctionalités importantes d'un tel système.

Dans un deuxième temps, nous présentons un modèle de représentation des connaissances servant à décrire les actions susceptibles d'êtres entreprises par l'opérateur. Les actions ainsi décrites sont connues du système, et peuvent par conséquent faire l'objet d'une détection.

Dans un troisième temps, nous élaborons un mécanisme de détection d'intentions exploitant le modèle de représentation des connaissances précédemment décrit. Le mécanisme proposé s'exprime sous la forme d'un ensemble de critères de détection se rapportant à la connaissance du système, ainsi que d'un jeu d'algorithmes mettant en œuvre ces critères.

Enfin, nous montrons que d'une part, grâce à la généricité du modèle de représentation des connaissances proposé, et d'autre part, grâce à l'élaboration du mécanisme de détection d'intentions précédent, nous pouvons, à un coût très faible, étendre les capacités de notre système à des fonctionalités telles que l'exécution ou la simulation des opérations que celui-ci est initialement capable de détecter.

À ce jour, le travail décrit dans cet ouvrage a fait l'objet d'un certain nombre de publications, toutes référencées dans la bibliographie en fin de document : Grumbach et Verna (1996), Verna (1999), Verna et Grumbach (1999), Verna et Grumbach (1998b,a), Verna (1997a,b)

Comment définir le Virtuel ? Le modèle M_RIC

1 Introduction **13**
2 Historique **15**
2.1 Les premiers pas 15
2.2 Vers la synthèse d'image 16
2.3 Les interfaces immersives 16
2.4 Le marché de la Réalité Virtuelle 17
2.5 Situation actuelle 17
3 Définitions Actuelles **19**
3.1 Temps-Lieu-Interaction 19
3.2 Interaction-Immersion-Imagination 20
3.3 Perception-Proprioception 21
3.4 Réalités réalistes et artificielles 22
3.5 Réalités réalistes et altérées 22
3.6 L'approche philosophique 22
3.7 Confrontation 23
4 Le Modèle M_RIC **25**
4.1 Opérateur *vs.* Environnement 25
4.2 Mental *vs.* Physique 26
4.3 Explicite *vs.* Implicite 27
4.4 Terminologie 29
4.5 Modèle complet 30
4.6 Notion de «réalité» 34
5 La Réalité Virtuelle **37**
5.1 Le virtuel : une méta-définition 37
5.2 Virtualisation de l'environnement 40
5.3 Virtualisation de l'agent physique 43
5.4 Imbrication de la réalité virtuelle 47
5.5 Virtualité des environnements réels 47
5.6 La réalité augmentée : un cas litigieux 50
5.7 Réalité virtuelle et télé-opération 53
5.8 Détection d'intentions 55
5.9 Problèmes liés à l'immersion 55
5.10 Réalité virtuelle et langage 59
6 Conclusion **61**

Chapitre 1

Introduction

«La réalité virtuelle corrompt, la réalité absolue corrompt absolument.»
Roy Ascott

Comme nous l'avons suggéré dans l'introduction générale, notre étude se focalise sur les relations entre la notion d'assistance et celle de réalité virtuelle. Notre projet ne se réduit donc pas à étudier la notion d'assistance dans tel ou tel système de réalité virtuelle (qui serait alors admis comme tel), mais plutôt comment la notion même de réalité virtuelle peut modifier notre vision de l'assistance. Dans ces conditions, il est primordial de définir clairement le concept de réalité virtuelle, cette définition ayant un impact sur le tout le reste de l'étude. Ceci constitue l'objet de cette première partie.

Face à cette nécessité d'une définition précise, ne pourrions-nous pas simplement prendre une définition existante comme point de départ ? Cela semble impossible pour plusieurs raisons :

- La récente prolifération des domaines de recherche en réalité virtuelle s'est agrémentée d'une extension de la terminologie employée : réalité ou mondes virtuels, réalité artificielle, environnements de synthèse, réalité augmentée, téléprésence… À l'heure actuelle, cette terminologie reste floue et le sens qui lui est attribué fréquemment propre aux auteurs qui l'utilisent ou la proposent. Nous ne pouvons donc mener à bien notre projet sans avoir auparavant clarifié l'ensemble des termes auxquels nous auront l'occasion de faire référence.
- Comme nous le verrons dans le deuxième chapitre, les définitions existantes présentent chacune des aspects intéressants mais que l'on ne retrouve pas systématiquement. Afin que notre étude soit la plus exhaustive possible, il nous est donc impossible de favoriser une définition plutôt qu'une autre, ni même de «regrouper» ces définitions en une seule, dans la mesure ou certaines d'entre elles présentent des aspects contradictoires.

Nous voyons donc maintenant l'intérêt de proposer une définition propre, tant sur le plan de l'unification des idées existantes que sur celui de l'adéquation de cette définition à notre étude. Dans cette optique, quelle direction donner à notre approche ? Bien que tout système de réalité virtuelle soit indissociable des utilisateurs à destination desquels il est conçu, on constate que la recherche technologique en infographie, matériels d'immersion etc. est beaucoup plus développée que l'étude des comportements humains dans de tels systèmes. Il apparaît donc important de favoriser la recherche *cognitive*, ainsi que les études plus abstraites sur l'*interaction* entre l'humain et ces systèmes. Ceci constitue la direction majeure que prendra notre définition.

Le premier chapitre propose un rapide historique du développement de la réalité virtuelle. Nous examinons ensuite un certain nombre de définitions déjà proposées, en notant les points qui nous intéressent particulièrement. Finalement, une définition de la réalité virtuelle est proposée. Cette définition est élaborée en plusieurs étapes :

- Premièrement, nous modélisons une situation d'interaction dite «réelle» entre un humain et son environnement. Ce modèle ne tente pas de *définir* le réel, mais en fixe une vision cognitive selon l'axe principal de l'interaction. Cette vision du réel sert de base à l'élaboration d'une définition du virtuel.
- Ensuite, nous définissons la réalité virtuelle non comme un ensemble de critères auxquels telle ou telle situation doit répondre, mais plutôt comme un processus de transformation, ou de «virtualisation» de la situation réelle décrite précédemment. Ce processus de virtualisation doit cependant répondre à un ensemble de règles qui sont détaillées.

Chapitre 2

Historique

Ce chapitre présente dans les grandes lignes les principaux événements ayant marqué le développement de l'activité en réalité virtuelle au sein de la communauté scientifique. Le lecteur intéressé par un historique plus complet pourra se reporter à Pimentel et Teixeira (1994) et Burdea et Coiffet (1993). Bien qu'il soit en général impossible de déterminer de manière ponctuelle le début d'une activité scientifique quelle qu'elle soit, l'ensemble de la communauté scientifique s'intéressant au domaine de la réalité virtuelle s'accorde pour situer le début de l'aventure aux années soixante.

2.1 Les premiers pas

1962 - Morton Heilig crée le *Sensorama*.

Le Sensorama constitue le premier système de réalité virtuelle digne de ce nom. Ce dispositif mono-utilisateur était doté d'un système vidéo en trois dimensions donnant des images en relief et en couleurs, d'un siège mobile capable de produire des vibrations, ainsi que de ventilateurs et de diffuseurs d'odeurs. La première simulation proposée fut celle d'un trajet en cyclomoteur dans les rues de New York. Grâce à ce dispositif, l'utilisateur avait une bonne impression de relief, ressentait les secousses dues à la route, le vent dû à la vitesse, et pouvait même sentir des odeurs de nourriture en passant à côté d'une épicerie.

Malgré le succès que Morton Heilig imaginait recevoir, il dû faire face à d'énormes difficultés pour pouvoir réaliser son système. Le Sensorama était en fait une version simplifiée d'un projet de beaucoup plus grande envergure, le «Théâtre Expérimental»* (Heilig, 1992), qu'il espérait pouvoir mettre en œuvre depuis deux ans déjà. Malheureusement pour lui, le désintéressement total des communautés scientifique et industrielle pour son projet le contraignit à modérer son ambition.

☛ Il est très étonnant de constater ce désintéressement général, quand on sait l'avance technologique aussi bien que conceptuelle dont son projet témoignait, d'autant qu'il fallut encore attendre plusieurs dizaines d'années avant que son idée soit reprise et développée.

2.2 Vers la synthèse d'image

1966 - L'ancêtre des visiocasques.

Parallèlement au développement du Sensorama, Heilig travaillait sur une idée de système de visualisation portable, monté sur la tête des utilisateurs (Heilig, 1960). Il s'agit là de l'ancêtre des visiocasques (Cf. glossaire en page 216) actuels : le casque était doté de deux visionneurs (un pour chaque œuil) permettant la restitution de diapositives en relief. Des dispositifs sonores et odorifiques étaient également prévus. À cette époque, Heilig est rejoint dans ses travaux par Sutherland, qui remplace en 1966 les diapositives par deux tubes cathodiques restituant des images filmées. C'est en travaillant sur ce projet que ce dernier a l'idée que les images pourraient être «calculées» par un ordinateur au lieu d'être filmées...

1973 - Le premier générateur de scène.

Evans et Sutherland reprennent l'idée de Morton Heilig et mettent au point un système permettant de visualiser des scènes composées de 200 à 400 polygones tracés en «fil de fer» (Cf. glossaire en page 215) et affichées à une cadence de 20 images par seconde. L'histoire des accélérateurs graphiques modernes ainsi que des techniques de restitution visuelle en trois dimensions commence à cet instant.

Un facteur motivant pour cette technologie naissante est que les militaires, déjà impliqués dans l'utilisation de systèmes plus anciens (pour les simulateurs de vol par exemple), y voient une avancée importante en matière de réutilisabilité : pour changer de scène, il suffit maintenant de reprogrammer la *même* machine, au lieu de fabriquer un nouveau système complet.

2.3 Les interfaces immersives

1981 - Pour la perception.

Ce sont les ingénieurs de la NASA qui mettent au point le premier casque de visualisation utilisant des écrans LCD (Cf. glossaire en page 216) comme moyen de visualisation. Ce visiocasque appelé *VIVED** s'intègre alors dans un projet à plus grande échelle, comportant non seulement le dispositif de visualisation, mais encore un capteur de position et un calculateur graphique installé dans un ordinateur prévu à cet effet. Le capteur détecte la position de la tête et l'ordinateur transmet ces données au calculateur graphique qui à son tour redessine les images à transmettre en fonction de ce nouveau point de vue.

☛ En plus du fait que ce système génère les images en temps réel, la nouveauté réside dans la manière dont est pris en compte le point de vue de l'utilisateur : pour la première fois, ce point de vue est calculé directement en fonction de la position naturelle de la tête. L'utilisateur n'a donc plus besoin de produire des commandes artificielles afin de modifier son point de vue. Ceci constitue un premier pas vers l'immersion perceptive.

1985 - Pour l'action.

Scott Fisher se joint alors au projet, et apporte un des premiers gants de données (Cf. glossaire en page 216) , périphérique capable de transmettre la position des doigts de la main au système.

**The Experience Theatre* est son nom d'origine.
*Acronyme de Virtual Visual Environment Display

☛ Il est important de noter ici que ce nouveau périphérique, contrairement à ses prédécesseurs, met l'accent sur la composante active (et non plus perceptive) de l'immersion. En effet, une des applications majeures de l'utilisation de gants de données réside dans la modélisation des opérations de saisie et de manipulation d'objets en monde virtuel.

1988 - Multimodalité des interfaces.

Fisher et Wenzel mettent au point le premier système capable de générer du son tridimensionnel, c'est à dire localisé dans une zone bien précise de l'espace, quelle que soit la position du sujet. Le réalisme de la simulation s'étend donc du champ visuel au champ sonore, suivant en cela l'idée qu'avait eu Heilig plus de vingt ans auparavant.

2.4 Le marché de la Réalité Virtuelle

1987 - Accessibilité au grand public.

Ce n'est que vers la fin des années 80 que l'esquisse d'une véritable commercialisation des outils de réalité virtuelle se fait sentir. On assiste alors à la naissance des premières sociétés dédiées à la fabrication de tels outils (comme VPL qui commercialise le fameux gant *Dataglove*) et à la mise sur le marché des premiers visiocasques et gants de données.

1991 - Commercialisation de systèmes complets.

Malgré ce développement rapide, les scientifiques ne disposent pas encore de systèmes de réalité virtuelle complètement intégrés, c'est-à-dire incorporant à la fois les outils d'interfaçage, les ordinateurs et les logiciels. La société britannique DIVISION introduit pour la première fois un système complet de réalité virtuelle offrant tous ces composants. À noter également l'apparition du maintenant célèbre «World Tool Kit» de Sense8*.

2.5 Situation actuelle

En cette fin de siècle, les outils et matériels de réalité virtuelle se développent rapidement. Il existe une multitude de compagnies commerciales fournissant de tels outils, et dans une gamme de prix assez étendue. On constate notamment l'apparition d'un nombre croissant d'outils abordables, de qualité moyenne ou minimale, fonctionnant sur des systèmes PC, ce qui constitue une volonté caractéristique de se tourner vers le grand public. Les systèmes de haute qualité restent cependant encore très chers.

Les domaines d'application de la réalité virtuelle les plus porteurs à l'heure actuelle sont probablement les jeux vidéos, la CAO (Cf. glossaire en page 215), les récentes évolutions en matière de réalité virtuelle distribuée (Calrsson et Jää-Aro, 1994), et les applications médicales (Vanderheiden et Mendenhall, 1994, Dubois et al., 1997). D'autre part, les applications militaires (Papin, 1999) sont également très demandeuses dans les cinq grands domaines que sont l'aide à la conception et à l'évaluation de systèmes, puis l'entraînement, la préparation et la réalisation de missions.

*http ://www.sense8.com

Chapitre 3

Définitions Actuelles

L'image la plus répandue de la notion de réalité virtuelle est sans doute celle d'un utilisateur doté d'un casque de visualisation et de gants de données, évoluant dans un environnement crée artificiellement par un ordinateur. Cette image ne reflète pourtant pas de manière exhaustive ce que l'on peut entendre par «réalité virtuelle». Un certain nombre de scientifiques travaillant dans des domaines reliés directement ou indirectement à cette notion ont proposé soit des définitions plus ou moins formelles de la réalité virtuelle, soit des catégorisations de ses applications. Dans ce chapitre, nous proposons d'examiner un certain nombre de ces points de vue et de mettre en exergue leurs caractéristiques.

3.1 Temps-Lieu-Interaction

Fuchs (1996) propose une définition selon trois axes : Temps, Lieu, Interaction. À partir d'une situation «réelle» (temps présent, environnement direct et interaction avec le monde réel), il envisage une situation de réalité virtuelle comme une situation dans laquelle un ou plusieurs de ces axes est modifié. La figure 3.1 en page suivante illustre cette définition.

La définition proposée par Fuchs contient implicitement certains éléments qu'il convient de souligner :

- Un «changement de lieu» s'exprime par l'évolution de l'utilisateur soit dans un environnement de synthèse simulant la réalité, soit dans un monde totalement imaginaire (par exemple un monde statistique). La réalité virtuelle est donc implicitement catégorisée selon deux classes : les simulations d'environnements réels, ou les mondes purement imaginaires.
- Un «changement d'interaction» est défini d'une manière similaire : il implique une simulation d'interaction avec un monde réel, ou une interaction purement imaginaire. Notons que d'une manière implicite, cet aspect de la définition implique qu'une forme d'interaction soit toujours présente dans une situation de réalité virtuelle.

☛ Selon cette approche, une situation de télé-opération ne constitue pas à proprement parler un cas de réalité virtuelle : bien que les mises en œuvre de ces situations soient très proches, en particulier car les deux domaines partagent un grand nombre d'interfaces, la télé-opération s'exécute dans un environnement réel ce qui est contraire à la définition proposée.

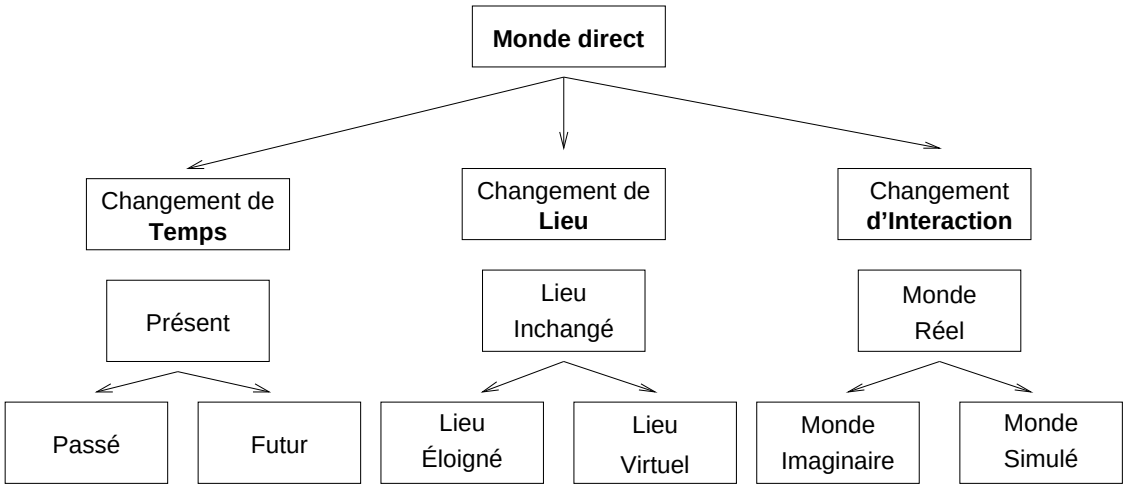

FIG. 3.1: Temps-Lieu-Interaction, d'après Fuchs (1996)

3.2 Interaction-Immersion-Imagination

Burdea et Coiffet (1993) décrivent un système de réalité virtuelle, en termes de fonctionalité, comme :

> [...] une simulation par ordinateur dans laquelle le graphisme est utilisé pour créer un monde qui semble réaliste. De plus, le monde synthétisé n'est pas statique mais répond aux ordres de l'utilisateur (gestes, paroles, etc.) [...]

La réalité virtuelle ne s'arrête pas au seul sens visuel, mais devrait permettre une interaction olfactive, tactile, plus généralement, selon tous les sens humains. C'est pourquoi Burdea (1993) propose la définition suivante :

> Un système de réalité virtuelle est une interface qui implique de la simulation en temps réel et des interactions via de multiples canaux sensoriels. Ces canaux sensoriels sont ceux de l'homme : vision, audition, toucher, odorat, goût.

De cette définition découlent plusieurs caractéristiques :

- La nécessité d'une «simulation par ordinateur» montre que l'environnement virtuel doit être un environnement de synthèse. Cet aspect était déjà présent dans la définition proposée par Fuchs (1996).
- Un système de réalité virtuelle doit fournir à l'utilisateur une interaction en temps réel avec le monde virtuel selon ses propres canaux de communication. L'accent est donc mis sur la notion d'interaction, point également rencontré précédemment.

La définition proposée par Burdea et Coiffet apporte cependant deux points nouveaux :

- Le fait de fournir une simulation «réaliste» doit aboutir à une sensation d'immersion de l'utilisateur dans l'environnement virtuel. La notion d'«immersion» peut être grossièrement décrite comme la sensation d'être physiquement présent dans l'environnement virtuel. Nous aurons l'occasion de revenir plus en détail sur cette notion.
- Enfin, une troisième composante est décrite comme essentielle au bon fonctionnement de l'application : l'imagination de l'opérateur, sans l'aide de laquelle le système ne serait pas aussi performant que souhaité. La définition de Burdea et Coiffet s'exprime donc au travers des «trois I» : Immersion-Interaction-Imagination, représentés sur la figure 3.2 ci-contre.

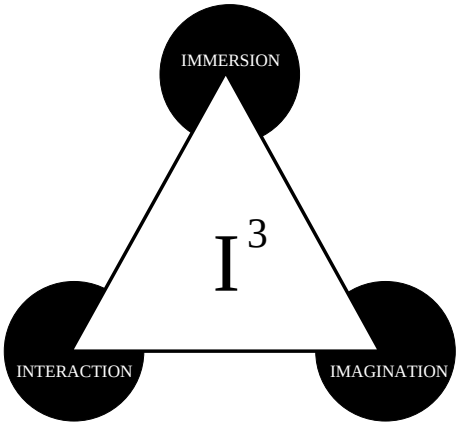

FIG. 3.2: Immersion-Interaction-Imagination, d'après Burdea (1993)

☛ La composante «immersive» dans cette définition est une caractéristique que l'on retrouve également chez Pimentel et Teixeira (1994), pour qui la réalité virtuelle «renvoie en général à une expérience d'immersion et d'interactivité générée par un ordinateur». Les notions d'«imagination» et d'«immersion» sont importantes car elles font intervenir une composante cognitive dans la définition : un utilisateur peut se *sentir* immergé dans un environnement virtuel ou pas. Cela dépend autant de l'environnement que de son utilisateur, avec l'aide de son imagination. Le fait qu'une situation donnée soit de la réalité virtuelle ou non peut ainsi dépendre de facteurs humains, notamment liés à l'utilisateur lui-même.

3.3 Perception-Proprioception

Quéau (1993) propose la définition suivante :

> Un monde virtuel est une base de données graphiques explorable et visualisable en temps réel, sous forme d'images de synthèse tridimensionelles, de façon à donner le sentiment d'une immersion dans l'image.

Nous retrouvons dans cette définition l'idée qu'un monde virtuel est nécessairement synthétisé par un ordinateur. La notion d'immersion y est également présente. Quéau ajoute que le plus souvent, cette sensation d'immersion est produite par la conjonction d'une perception visuelle avec la sensation proprioceptive (Cf. glossaire en page 216) équivalente. Par exemple, l'immersion sera efficace si un pas en avant réel, «ressenti» de manière interne, se traduit par une avancée dans le monde virtuel.

☛ Cette remarque est importante dans la mesure où elle traduit pour la première fois la sensation d'immersion non plus seulement en fonction des perceptions «externes» issues de l'environnement (la vision en particulier), mais aussi grâce aux sensations «internes» de l'humain. Selon cette définition, les sensations proprioceptives prennent donc également part au processus d'immersion.

☛ Notons également que la notion d'interaction apparaît dans une moindre mesure au travers de l'idée d'«exploration» de la base de données graphique, soit de navigation dans le monde virtuel. La navigation permet de percevoir le monde sous des angles différents de manière volontaire et non pas assujettie. Cependant, le but de la navigation reste essentiellement perceptuel. La navigation constitue t'elle une véritable interaction ? C'est une question à laquelle nous serons amenés à répondre ultérieurement.

3.4 Réalités réalistes et artificielles

Ceci relève plus d'une catégorisation des applications que d'une définition à proprement parler. Le terme de «réalité artificielle» est introduit par Krueger (1991) pour décrire la capacité des systèmes de réalité virtuelle à créer des mondes totalement libres, ne tentant en aucune manière de simuler des environnements réels :

> La promesse des réalités artificielles n'est pas de reproduire la réalité conventionelle ou d'agir sur le monde réel. Elle est précisément l'opportunité de créer des réalités synthétiques pour lesquelles il n'y a pas d'antécédents réels.

La réalité virtuelle selon cette optique, s'exprime soit en termes de *simulation* d'environnements réels, soit en termes de *création* d'environnements totalement imaginaires, et pourquoi pas, de proposer des environnements unifiant les deux aspects. C'est une catégorisation que nous avons déjà rencontrée auparavant.

3.5 Réalités réalistes et altérées

Dans une perspective d'assistance aux personnes handicapées, Vanderheiden et Mendenhall (1994) introduisent le terme de «réalité altérée», et sa mise en application dans des systèmes de *réalité virtuelle altérée*. Partant d'un environnement réel avec lequel l'interaction serait problématique pour une personne présentant des déficiences mentales ou physiques, ils décrivent comment les techniques et concepts de réalité virtuelle permettent d'améliorer cette interaction dans une version virtuelle de l'environnement, en *altérant* les propriétés de celui-ci. Par exemple, il est possible de «ralentir le temps» pour une personne handicapée moteur, agrandir des objets ou des parties d'objets pour des personnes amblyopes, utiliser la modalité sonore ou des textures tactiles pour reproduire des couleurs...

☛ Cette catégorisation est en fait assez proche de celle décrite dans la section précédente. Plus précisément, la notion de «réalité altérée » viendrait se positionner entre les réalités réalistes et les réalités artificielles. Une réalité réaliste est une simulation, la plus fidèle possible, du monde réel. Une réalité artificielle est un monde totalement imaginaire, sans aucun antécédent réel. Entre ces deux extrêmes se trouvent par conséquent les réalités altérées, basées sur des mondes réels, mais dont certaines propriétés sont modifiées.

3.6 L'approche philosophique

Les informaticiens, les roboticiens, mais aussi les philosophes se sont intéressés au concept de réalité virtuelle. Parmi ces derniers, citons Lévy (1995,p.10) :

> [...] le virtuel, rigoureusement défini, n'a que peu d'affinité avec le faux, l'illusoire ou l'imaginaire. Le virtuel n'est pas du tout l'opposé du réel. [...]

Le mot «virtuel» vient du latin *virtualis*, issu à son tour de *virtus*, qui signifie force, puissance. Il évoque donc une *potentialité*. Selon cette optique, le virtuel est un réel potentiel, dont on ne connaît pas la forme exacte. Lévy précise cette vision en positionnant la notion de virtuel dans un ensemble de quatre termes comme illustré en figure 3.3. Le «possible» représente un état en-

	Latent	**Manifeste**
Substance	Possible	Réel
Événement	Virtuel	Actuel

FIG. 3.3: Les quatre modes d'être, d'après Lévy (1995,p.135)

tièrement connu, différent du «réel» uniquement en ceci qu'il lui manque l'existence. Le possible devient réel sans que rien ne soit modifié dans sa détermination. Le «virtuel», lui, ne s'oppose pas au réel, mais à l'«actuel». Le virtuel est une potentialité dont l'instanciation n'est ni connue, ni fixe, mais dépendra du processus même d'instanciation (d'actualisation). «La graine est virtuellement un arbre» selon Lévy, dans la mesure où cet arbre n'est non seulement pas connu exactement à l'avance, mais sa forme, son instanciation finale dépend également du processus même de sa création.

L'approche philosophique, quoique relativement éloignée de l'objectif que nous tentons d'atteindre dans ce travail, présente certains aspects particulièrement intéressants :

- Cette approche place le réel et le virtuel sur des plans totalement différents. Il n'y a donc en particulier aucune opposition conceptuelle entre les deux notions. Le virtuel serait-il donc aussi réel que le réel ? Nous serons amenés, dans les chapitres qui suivent, à revenir sur ce point.
- Plutôt que d'étudier chaque situation d'une manière statique, Lévy (1995) préfère s'intéresser aux *processus* permettant de passer d'une situation à l'autre, et traite en particulier, des processus de virtualisation. C'est une démarche que nous adopterons également dans la suite de cette partie.

3.7 Confrontation

Au travers de ces définitions, nous voyons quelques directions communes s'établir, mais aussi quelques singularités sur lesquelles il convient de mettre l'accent. Au cours de l'élaboration de notre propre définition, nous aurons l'occasion de nous positionner par rapport aux points suivants :

3.7.1 Importance de l'interaction

L'interaction avec le monde virtuel, et non pas seulement la perception de celui-ci, est un point particulièrement mis en exergue dans deux des définitions précédentes (Fuchs, 1996, Burdea et Coiffet, 1993, Burdea, 1993) : un monde virtuel ne doit pas seulement être un environnement que l'on peut *percevoir*, mais il doit également «répondre aux commandes de l'utilisateur» (Burdea et Coiffet, 1993). Dans une moindre mesure, la possibilité d'exploration ou de navigation dans le monde considéré (Quéau, 1993), pourrait-elle être vue comme un début d'interaction ? Est-ce une condition suffisante pour que l'on puisse parler d'interaction ? Cette question sera débattue dans les chapitres suivants.

3.7.2 La composante immersive

Chez Pimentel et Teixeira (1994) et particulièrement chez Burdea (1993), Burdea et Coiffet (1993), la nécessité d'une forte interactivité entre l'utilisateur et le système de réalité virtuelle doit conduire à une sensation d'immersion dans le monde virtuel, c'est à dire à une sensation de présence physique dans l'environnement considéré. La composante immersive serait donc une part essentielle de la définition du virtuel, découlant directement des capacités d'interaction fournies par le système.

3.7.3 La composante infographique

Qu'il s'agisse d'un «lieu artificiel» pour Fuchs (1996), d'une «simulation par ordinateur» pour Burdea et Coiffet (1993), Burdea (1993) ou bien encore d'une «base de données infographique» pour Quéau (1993), on constate que la composante infographique est une caractéristique reprise dans la grande majorité des définitions proposées : que le monde virtuel soit complètement imaginaire ou tente d'imiter un environnement réel, celui-ci doit être généré par un ordinateur. A

l'inverse, l'approche philosophique de Lévy (1995) ne met pas en opposition les notions de réel et de virtuel, mais les place sur des plans différents. Cette particularité nous conduit à nous demander si un environnement réel peut être qualifié de virtuel, et si oui, dans quelles conditions. C'est un point important sur lequel nous reviendrons dans les chapitres suivants.

3.7.4 La composante cognitive

Burdea et Coiffet (1993), Burdea (1993) mettent explicitement l'accent sur un aspect cognitif de la réalité virtuelle : l'imagination de l'utilisateur. Leur idée est que la sensation d'immersion ne sera réellement efficace que si l'utilisateur participe activement, par l'imagination, à la simulation. Quel rôle l'imagination de l'utilisateur doit-elle jouer ? Notamment aider celui-ci à «oublier» les imperfections du système de réalité virtuelle et à se concentrer sur les aspects fonctionnels de l'interaction. La définition proposée par Quéau (1993) contient également une part de considération cognitive, dans le fait de considérer les aspect proprioceptifs comme une composante critique de l'efficacité des processus d'immersion.

3.7.5 Catégorisation des applications

Au travers des visions de la réalité virtuelle explorées dans les sections précédentes, il se détache un consensus général sur les manières de distinguer les différentes applications de celle-ci. Krueger (1991) place les deux extrémités en distinguant d'une part les systèmes destinés à reproduire les environnements réels de manière aussi fidèle que possible, et d'autre part les «réalités artificielles », purs fruits de l'imagination de leur concepteur, sans aucun antécédent réel. Vanderheiden et Mendenhall (1994) viennent ensuite nuancer cette catégorisation par l'introduction des «réalités altérées », étape intermédiaire dans laquelle les environnements réels sont modifiés, mais conservent malgré tout certains de leurs aspects réalistes.

3.7.6 Cas de la télé-opération

Malgré le consensus général sur l'aspect «artificiel» ou «synthétique» des environnements virtuels, Fuchs (1996) rapproche télé-opération et réalité virtuelle en notant la proximité de leurs mises en œuvre, notament en termes d'interaces utilisées. Il semblerait donc que la frontière entre réel et virtuel ne soit pas aussi catégorique que le critère de distinction infographique le laisse supposer. Ceci constitue une deuxième raison nous poussant à poser la question de la virtualité des environnements réels.

Chapitre 4

Le Modèle MRIC

Dans les chapitres qui suivent, nous élaborons une définition de la réalité virtuelle fondée essentiellement sur la notion d'interaction, et vue sous un angle principalement cognitif. Comme nous l'avons dit précédemment, l'approche cognitive nous pousse, en harmonie avec la démarche de Lévy (1995), à nous demander ce qui fait que telle ou telle situation est virtuelle pour l'humain qui l'expérimente. Dans la mesure où un système de réalité virtuelle est un système bel et bien réel, nous ne tentons pas de caractériser une situation donnée d'une manière absolue, mais plutôt d'étudier les processus cognitifs de virtualisation qui font que de la situation réelle, on passe au virtuel.

Selon cette optique, il est primordial d'avoir une vision claire sur la notion de réel avant de pouvoir s'interroger sur les manières de s'en abstraire. C'est ce que nous tentons de faire dans ce chapitre. Nous nous proposons de préciser ce que l'on entend par «réel» au travers d'un modèle cognitif représentant l'humain dans son environnement, le modèle ***MRIC*** (acronyme de **M**odèle **R**eprésentatif de l'**I**nteraction **C**ognitive).

4.1 Opérateur *vs.* Environnement

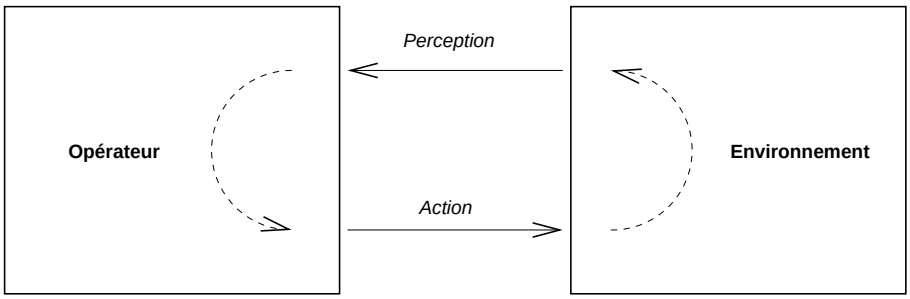

FIG. 4.1: Opérateur *vs.* Environnement

La première étape de modélisation, illustrée par la figure 4.1, consiste en la distinction humain *vs.* environnement. Comme nous l'avons déjà précisé, notre démarche se focalise sur l'interaction d'un humain *unique* avec son milieu. Ce présupposé nous conduit à définir les termes suivants :

Définition 1: Opérateur

Afin, de lever toute ambiguïté quand à la désignation de l'humain que nous étudions, et dans la mesure où celui-ci «opère» dans son milieu propre, l'unique humain considéré dans cette étude sera appelé ***opérateur****, par opposition aux autres humains pouvant être présent autour de celui-ci.* ■

Définition 2: Environnement

On désignera par le terme ***environnement*** *tout ce qui n'est pas l'opérateur, c'est à dire l'ensemble des objets physiques* ou autres personnes *au milieu desquels il est susceptible d'évoluer.* ■

Cette première étape de modélisation fait apparaître deux «chemins d'information» circulant entre l'opérateur et l'environnement, communément appelés **action** et **perception**. Ces chemins d'information ne sont pas indépendants l'un de l'autre : une action peut découler ou dépendre d'une perception, et réciproquement. Par exemple, le déplacement d'un objet (action) résulte en une modification de la position visuelle de celui-ci (perception). Symétriquement, pour saisir cet objet, la position visuelle de celui-ci (perception) conditionne le déplacement du bras de l'opérateur (action).

4.2 Mental *vs.* Physique

La deuxième étape de modélisation, illustrée par la figure 4.2, consiste en la distinction mental *vs.* physique au sein de l'opérateur. Il existe deux raisons qui nous poussent à intégrer cette distinction dans notre modèle :

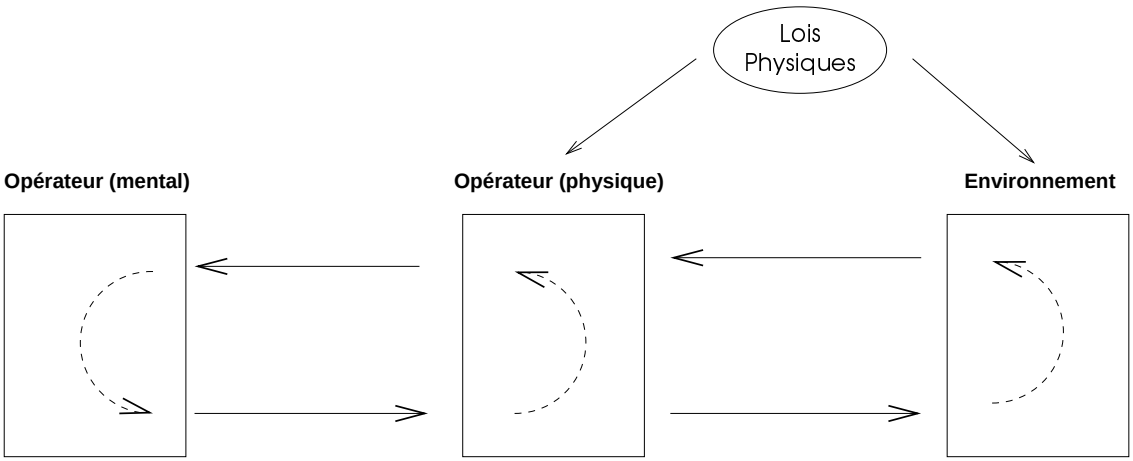

FIG. 4.2: Mental *vs.* Physique

➤ L'opérateur n'agit pas directement sur son environnement. Il ne peut agir que sur son propre corps et c'est seulement par cette indirection qu'une action sur l'environnement est possible. Par exemple, la saisie d'un objet n'est pas à proprement parler une action sur cet objet. La véritable action mise en œuvre est celle consistant à déplacer son bras, puis contracter les muscles de la main. Le fait que l'objet soit saisi est ainsi une conséquence directe des lois de la physique (ici, les frottements entre surfaces), mais seulement indirecte de l'action entreprise. Il est nécessaire de distinguer d'un côté les processus de *déclenchement*

d'une action (domaine du mental), et de l'autre l'*expression* de cette action (domaine physique). Strictement parlant, une action directe de l'opérateur sur l'environnement relèverait de la télékinésie.

➤ Le corps, en tant qu'objet matériel, est soumis aux mêmes lois physiques que n'importe quel autre objet de l'environnement. C'est d'ailleurs grâce à cette propriété que l'interaction avec l'environnement est possible. On peut donc dire que l'opérateur est partiellement «soumis» au monde dans lequel il évolue : il est maître de lui-même dans la limite de ce qu'imposent les lois de la physique. En conséquence, la seule véritable distinction entre le corps de l'humain et un objet quelconque est son rattachement à l'«esprit».

☛ Il est important de bien comprendre le sens que nous donnons à la distinction mental *vs.* physique : il ne s'agit pas de distinguer les domaines *corporels* reliés au mental et au physique, mais bien les domaines *conceptuels*, indépendamment du support physique concerné. Le cerveau, comme objet matériel, est considéré comme faisant partie de l'agent physique.

La caractéristique de «soumission» aux lois physiques, communes à l'environnement et à l'opérateur physique nous conduit à proposer la définition suivante :

Définition 3: Monde

Nous appellerons ***Monde*** *l'ensemble constitué de l'environnement, de l'opérateur physique ainsi que des lois de la physique qui les régissent.* ■

☛ Cette nouvelle définition, mise en rapport avec celle d'«environnement», constitue au niveau du vocabulaire un choix que nous faisons arbitrairement afin de lever toute ambiguïté sur les concepts sous-jacents. Ce choix s'efforce cependant de respecter le sens commun attribué aux termes concernés : la notion d'«environnement» désigne bien en général ce qui est *autour* de l'humain.

La figure 4.2 en page précédente nous permet également de préciser deux points :

➤ Tout d'abord, il n'existe aucune action directe de l'opérateur mental sur l'environnement. Réciproquement, toute perception issue de l'environnement passe tout d'abord par les organes sensoriels de l'opérateur, qui appartiennent au domaine physique.

➤ Enfin, ce niveau de modélisation fait apparaître une nouvelle boucle d'information. Il existe en effet un chemin d'information n'ayant pas d'effet sur l'environnement : le chemin proprioceptif, sur lequel Quéau (1993) mettait l'accent dans sa définition de la réalité virtuelle. Ce chemin illustre par exemple la capacité de l'opérateur à se positionner dans l'espace sans aucune information extérieure.

4.3 Explicite *vs.* Implicite

Notre modélisation ne serait pas complète sans une dernière étape, la distinction entre processus *explicites* et *explicites* au sein de l'opérateur mental. Cette distinction puise sa motivation dans la remarque suivante :

Nous avons précédemment noté que l'humain est d'une certaine manière «soumis» au monde physique dans lequel il évolue, puisque son corps est régi par les mêmes lois physiques que tout autre objet. Cette notion de soumission s'étend plus loin que la seule soumission physique : autrement dit, il existe une forme de soumission mentale. L'humain n'est en effet que partiellement maître de ses actions et de ses perceptions. Il existe entre le niveau de décision et le niveau physique, des mécanismes dont la mise en œuvre est indépendante de la volonté de l'opérateur, c'est à dire qui ne sont pas contrôlables. Trois exemples sont donnés à titre d'illustration :

Réflexe actionnel La respiration est un phénomène intéressant dans la mesure où elle peut jusqu'à un certain point être contrôlée par la volonté, alors que la plupart du temps, elle intervient sans que l'humain n'en ait conscience. Il s'agit dans ce cas d'une action exécutée hors de la volonté de l'opérateur, une action réflexe.

Illusion d'optique Une expérience décrite par Norman (1980) et restituée en figure 4.3 illustre bien l'absence de perceptif : dans cette expérience, l'humain ne peut s'empêcher de «voir» un carré blanc qui n'est pourtant pas présent de manière explicite dans l'image. Les processus d'interprétation de l'image, qui ne sont pas contrôlables de manière volontaire, lui imposent une version prétraitée de l'information.

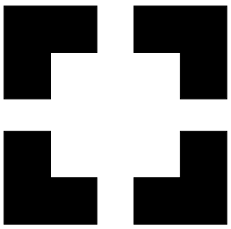

FIG. 4.3: Illusion d'optique

Réflexe sensori-moteur Alors qu'un humain normalement constitué marche sans difficulté, celui-ci éprouve une grande peine à décrire verbalement les mouvements occasionnés par le processus de la marche. Il n'a en fait pas une connaissance explicite de toutes les étapes nécessaires à l'accomplissement d'un pas. On peut donc dire que certains processus *réflexes* sont chargés de traduire la volonté de déplacement en séquences d'informations de plus en plus bas niveau, jusqu'à l'influx nerveux, séquences qui ne sont pas explicitement connues de l'opérateur. Contrairement aux deux exemples précédents, celui-ci décrit un phénomène impliquant à la fois des actions et des perceptions qui ne remontent pas jusqu'au niveau de la conscience.

Ces considérations nous amènent à scinder l'opérateur mental en deux nouvelles composantes, comme illustré en figure 4.4.

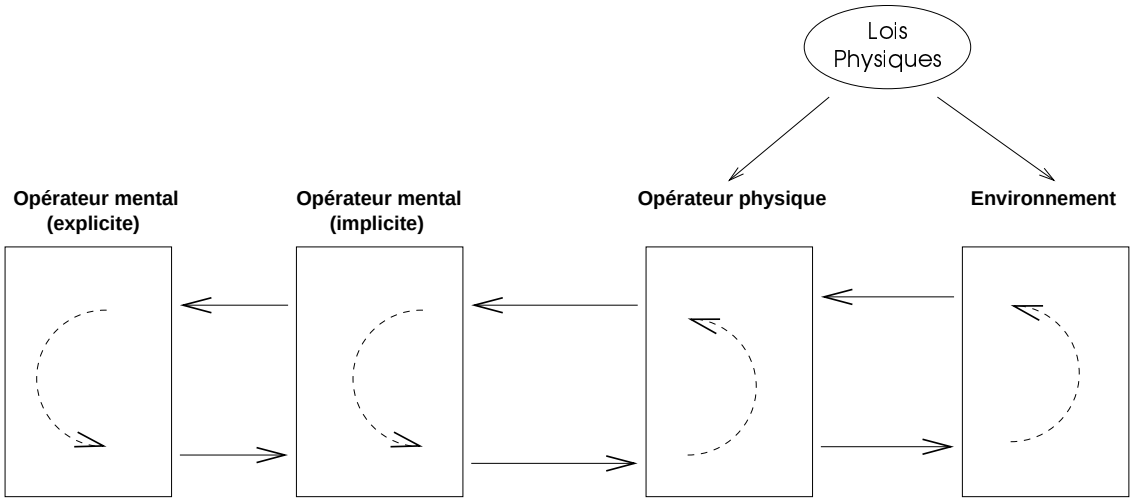

FIG. 4.4: Explicite *vs.* Réflexe

Cette figure fait également apparaître une nouvelle boucle dans les chemins d'information, au sein de l'opérateur mental implicite. Cette boucle illustre la propriété que des processus réflexes tels que la marche utilisent également des informations perceptives pour s'exécuter, sans pour cela remonter jusqu'au niveau de la volonté.

4.4 Terminologie

Les sections précédentes ont fait apparaître quatre composants distincts, jusqu'ici désignés, d'une manière hétérogène, comme opérateur mental (explicite et réflexe), opérateur physique, et environnement. Afin de fournir une terminologie parcimonieuse pour décrire ces composants, nous proposons une vision en termes d'«agents». Ainsi, les quatre définitions suivantes sont proposées :

Définition 4: Agent de Contrôle
*On appelle **Agent de Contrôle** le domaine mental de l'opérateur (précédemment qualifié d'«opérateur mental explicite») lié à l'intelligence, au raisonnement, à la connaissance explicite. Cet agent est notamment responsable des processus de décision.* ■

Définition 5: Agent de transcription
*On appelle **Agent de Transcription** la partie de l'opérateur rassemblant les processus d'action ou de perception ne relevant pas de la volonté où du contrôle de celui-ci.* ■

Définition 6: Agent Opératoire
*On appelle **Agent Opératoire** le domaine physique de l'opérateur (précédemment qualifié d'«opérateur physique»).* ■

Définition 7: Agent Externe
*On appelle **Agent Externe** le domaine physique extérieur à l'opérateur (précédemment qualifié d'«environnement»), c'est à dire l'ensemble des objets physiques et des autres humains de la situation.* ■

☛ Dans la suite de ce document, on préférera utiliser la terminologie «agent» pour toute discussion liée explicitement au modèle M_RIC. Cependant, les termes «opérateur», «environnement» et «monde» continueront d'être utilisés dans un contexte plus général.

Afin de parfaire notre ensemble de définitions, il convient de rendre explicite la distinction mental *vs.* physique effectuée précédemment, en regroupant nos agents selon les deux catégories définies ci-dessous :

Définition 8: Agent Mental
*On appelle **Agent Mental** l'ensemble (précédemment qualifié d'«opérateur mental») constitué de l'agent de contrôle et de l'agent de transcription. Cet agent rassemble les processus mentaux, explicites ou implicites, existant chez l'opérateur.* ■

Définition 9: Agent Physique
*On appelle **Agent Physique** l'ensemble constitué de l'agent opératoire, de l'agent externe et des lois physiques qui les gouvernent. Cet agent est désigné dans un contexte général par le terme de «monde».*

4.5 Modèle complet

Nous disposons maintenant de l'ensemble des éléments nécessaires à l'élaboration finale de notre modèle. Ce modèle est présenté dans sa globalité sur la figure 4.5. Le reste de cette section revient en détail sur chaque composante du modèle.

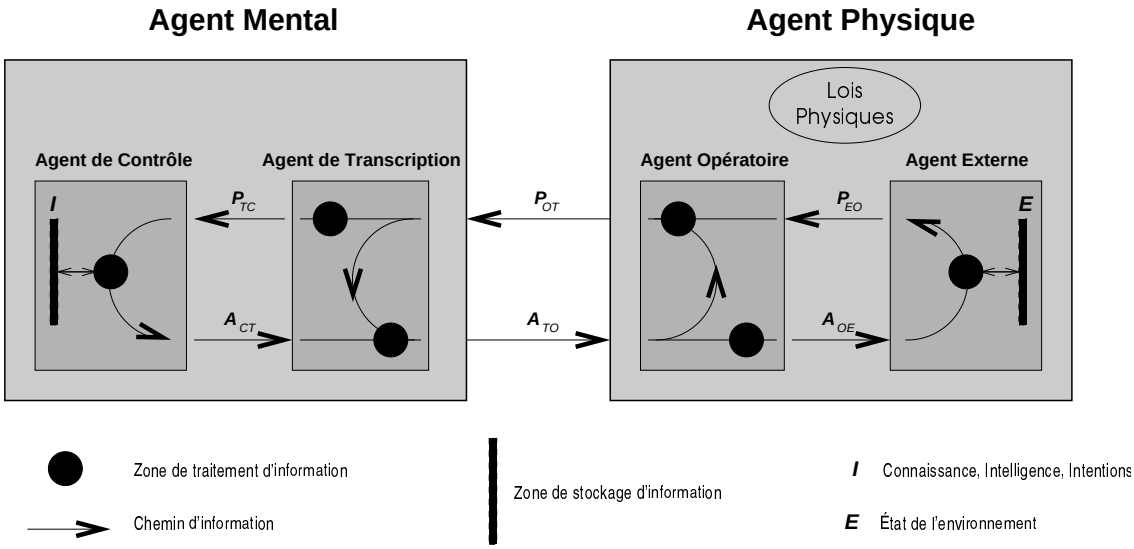

FIG. 4.5: Modèle M_R^{IC} complet

4.5.1 Chemins d'information

Le modèle proposé en figure 4.5 fait intervenir six chemins d'information. Ces chemins sont notés par un A ou un P suivant qu'ils sont dirigés dans le sens de l'Action ou de la Perception, et sont indicés par les initiales des agents qu'ils relient. Ainsi, une information transitant de l'agent de contrôle vers l'agent de transcription est notée A_{CT}, une information transitant de l'agent externe vers l'agent opératoire est notée P_{EO}. Les six chemins d'information sont donc les suivants :

A_{CT} L'information véhiculée par ce chemin est caractérisée par le fait qu'elle reste dans le domaine de l'explicite. L'intention de déplacement à pied dans une direction donnée est une information entrant dans cette catégorie.

A_{TO} Sur ce chemin, l'information véhiculée n'est plus du domaine de l'explicite, ce qui signifie que l'opérateur n'en a pas conscience, ne la contrôle pas, ne saurait la décrire, ou ne pourrait y parvenir qu'au prix d'une charge cognitive importante. Étant donnée l'intention précédente de déplacement, l'information décomposée du mouvement (positionnement des membres, déplacement des articulations...) en est un exemple.

A_{OE} Ce chemin recèle les informations d'action du plus bas niveau, relevant directement du domaine de la physique. En suivant toujours l'exemple d'une intention initiale de déplacement

de l'opérateur, le mouvement physique de la matière (déplacement des membres...) se situe à ce niveau.

$\mathbf{P_{EO}}$ Parallèlement à A_{OE}, ce chemin recèle les informations de perception du plus bas niveau, relevant directement du domaine de la physique. L'information lumineuse émanant d'un objet en est un exemple.

$\mathbf{P_{OT}}$ Ce chemin véhicule l'information résultant de la première étape de traitement de l'information perceptive précédente, cette information n'étant toujours pas de nature explicite. Consécutivement à l'information lumineuse issue de notre objet, celle directement issue des organes perceptifs de l'opérateur intervient à ce niveau.

$\mathbf{P_{TC}}$ L'information transportée à ce niveau est la première information perceptive complètement explicite, c'est à dire dont l'opérateur peut disposer au niveau de l'agent de contrôle, pour les processus de compréhension et de décision. À l'issue de la perception visuelle de notre objet, l'identification de cet objet en est un exemple.

4.5.2 Modules terminaux

Le modèle complet fait apparaître deux modules terminaux, notés I et E aux deux extrémités des chemins d'information. Ces modules représentent des «états statiques», des «mémoires» d'information, par opposition aux cheminements dynamiques que nous venons d'expliciter :

Module E Ce module représente l'état général de l'agent physique, par exemple la position des objets, les conditions de lumière etc. Les processus de perception peuvent utiliser des informations en provenance de E, de même que les processus d'action peuvent lui fournir de l'information. Par exemple, lorsque l'opérateur déplace un objet, il modifie l'état E de l'environnement. Réciproquement, une collision entre l'opérateur et un objet de l'environnement produit un retour perceptif qui dépend à la fois de l'effet induit par l'opérateur (son mouvement), et d'informations en provenance de E (la position des objets).

Module I D'une manière analogue, le module I représente l'état général de l'agent mental, et comprend en particulier les intentions de l'opérateur ainsi que tout ce qui relève de la connaissance dont il dispose. Cette connaissance s'étend de la connaissance de soi (agent physique), à celle de l'environnement, ainsi qu'aux relations entre les deux. Cette fois-ci, ce sont les processus d'action qui peuvent utiliser de l'information en provenance de I, et les processus de perception qui peuvent lui fournir de l'information. Par exemple, la saisie d'un objet fait appel à la connaissance relative à la manière de saisir l'objet en question, connaissance située dans I. Réciproquement, la perception visuelle de l'objet renseigne sur sa position, et donc modifie l'état de I.

4.5.3 Traitement de l'information

Les processus de traitement de l'information interviennent au sein des quatre agents élémentaires de notre modèle. Les disques noirs représentés sur la figure 4.5 en page précédente indiquent les zones où ces processus interviennent. Compte tenu des différentes boucles que nous avons mises en évidence, en particulier celles moins immédiates introduites par l'agent de traduction et l'agent physique, une information peut être engendrée à partir d'une ou plusieurs sources d'information précédentes. Nous obtenons donc une combinatoire de seize processus de traitement d'information : quatre par agent élémentaire. Afin de fixer les idées, nous proposons un court exemple pour chacun de ces processus :

Agent de contrôle

$\mathbf{I \Longrightarrow A_{CT}}$ Processus de prise de décision spontanée, de «matérialisation» des intentions. L'opérateur décide par exemple de saisir un objet. Bien entendu, une décision ou une intention n'existe jamais dans l'absolu : il existe toujours des raisons qui font que cette décision a été prise (par exemple, la soif impose la décision de saisir un verre). Cependant, ces raisons ne sont pas dynamiques, ce sont des «états perceptifs», et à ce titre, ils ne relèvent plus du chemin P_{TC} mais de I.

$\mathbf{P_{TC} + I \Longrightarrow A_{CT}}$ Processus impliquant à la fois la connaissance explicite de l'opérateur et la dynamique perceptive. Par exemple, l'opérateur entame la procédure de saisie de l'objet. Pour effectuer cette opération, l'opérateur doit mettre en œuvre sa connaissance explicite de l'objet (en particulier la manière de le saisir), mais aussi l'information perceptive (et dynamique) qui lui indique la position de celui-ci.

$\mathbf{P_{TC} \Longrightarrow A_{CT}}$ Processus de «réaction» explicite aux perceptions dynamiques. Par exemple, une fois l'opération de saisie entamée, cette opération tient compte dynamiquement de la position de l'objet, si celui-ci est en mouvement.

$\mathbf{P_{TC} \Longrightarrow I}$ Processus de perception explicite d'information. Par exemple, l'opérateur perçoit et identifie un objet en face de lui, sans pour autant faire quoi que ce soit de cette information.

Agent de transcription

$\mathbf{A_{CT} \Longrightarrow A_{TO}}$ Processus de transcription d'une information explicite en une séquence de plus bas niveau. Par exemple, la volonté de prononcer un mot entraîne une séquence réflexe d'informations à destination des muscles de la mâchoire.

$\mathbf{A_{CT} + P_{OT} \Longrightarrow A_{TO}}$ Même processus que le précédent, à ceci près qu'un retour perceptif est également appliqué. Par exemple, la séquence réflexe de mouvements corporels engendrée par une intention de saisie ou de déplacement nécessite en partie un retour proprioceptif à des fins de contrôle.

$\mathbf{P_{OT} \Longrightarrow A_{TO}}$ Processus de «réaction» immédiat et non contrôlé à partir d'une perception. Les réflexes humains, comme le sursaut en cas de peur, en sont les exemples les plus démonstratifs. Même si ces réactions parviennent en général jusqu'à la conscience de l'opérateur (et donc remontent jusqu'à I), ceci se produit après coup. La réaction est bien déclenchée en aval de l'agent de contrôle.

$\mathbf{P_{OT} \Longrightarrow P_{TC}}$ Processus de transcription d'une information perceptive de bas niveau vers une information explicite. Par exemple la perception visuelle d'un objet se traduit par une information explicite statuant sur l'existence de cet objet dans l'environnement.

Agent opératoire

$\mathbf{A_{TO} \Longrightarrow P_{OT}}$ Processus de retour direct des actions sur l'agent opératoire. Le sens proprioceptif et kinesthésique en particulier, c'est à dire la capacité de ressentir les mouvements de son propre corps, intervient à ce niveau.

$\mathbf{A_{TO} + P_{EO} \Longrightarrow P_{OT}}$ Même processus que précédemment, avec en plus la nécessité d'une information en provenance de l'agent externe. Par exemple, la sensation proprioceptive de contraction de la main sur un objet saisi requiert de l'information sur l'objet en question, information en provenance de l'agent externe.

$\mathbf{P_{EO}} \Longrightarrow \mathbf{P_{OT}}$ Processus de traduction d'une perception externe en perception de premier degré interne à l'opérateur. Par exemple, le rayonnement photonique émanant d'un objet déclenche une perception visuelle à travers les organes sensoriels de l'opérateur.

$\mathbf{A_{TO}} \Longrightarrow \mathbf{A_{OE}}$ Processus de traduction finale d'une action à destination de l'environnement. Par exemple, l'intention de saisie d'un objet, après avoir été traduite en une séquence décomposée d'influx nerveux à l'intention des muscles, se traduit finalement par un déplacement physique de la matière.

Agent externe

$\mathbf{E} \Longrightarrow \mathbf{P_{EO}}$ Processus de génération d'une perception à partir de l'environnement, indépendamment de l'opérateur. Les informations telles que la température ambiante ou la luminosité en sont des exemples.

$\mathbf{E} + \mathbf{A_{OE}} \Longrightarrow \mathbf{P_{EO}}$ Même processus que le précédent avec en plus une influence de l'agent physique. Par exemple, la collision entre l'opérateur et un objet de l'environnement engendre une perception dépendant à la fois des mouvements de l'opérateur et de la position de l'objet en question.

$\mathbf{A_{OE}} \Longrightarrow \mathbf{P_{EO}}$ Processus de retour perceptif indépendant de E. La perception visuelle de soi en est un exemple, lorsqu'aucun obstacle n'interfère avec le champ de perception.

$\mathbf{A_{OE}} \Longrightarrow \mathbf{E}$ Processus de modification de l'environnement à partir d'une action physique. Le déplacement physique de la main, une fois un objet saisi, abouti à une modification de la position de celui-ci.

4.5.4 Niveau conceptuel de l'information

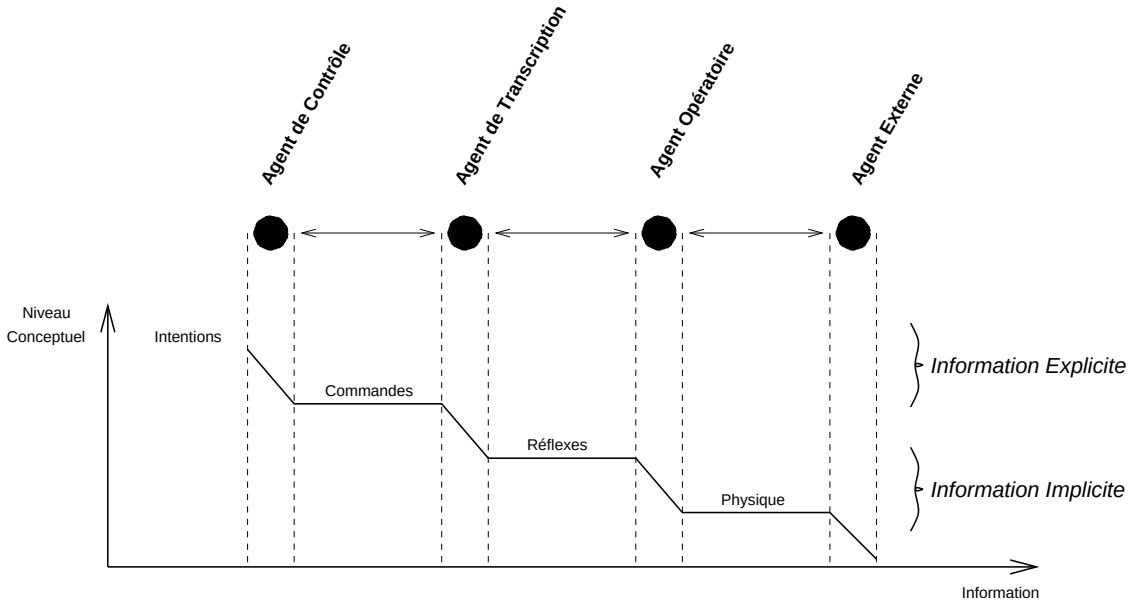

FIG. 4.6: Niveau conceptuel de l'information

Il est intéressant de constater que le niveau conceptuel de l'information véhiculée au travers des différents agents du modèle décroît quand on se rapproche de l'environnement. La figure 4.6 en page précédente illustre cette constatation. Les processus de traitement de l'information permettant de traverser les agents, c'est à dire à l'exclusion des phénomènes de bouclage, peuvent donc être considérés comme des «traducteurs conceptuels» dans le sens des actions (ils décomposent l'information jusqu'au plus bas niveau), et des «interpréteurs conceptuels» dans le sens des perceptions (ils recomposent l'information jusqu'au plus haut niveau).

4.5.5 Interaction : les quatre «I» du réel

Au travers du modèle ***M*$_R$*IC*** (figure 4.5 en page 30), nous avons constaté que la connectique de nos agents élémentaires établissait des boucles d'information, au nombre de quatre. Une boucle d'information est établie quand un agent a la propriété de retourner de l'information, après traitement, vers l'agent qui lui a envoyé. Autrement dit, une action génère une perception, une perception génère une action. Ceci nous fournit une manière très naturelle de définir la notion d'interaction :

Définition 10: Interaction

Deux agents élémentaires du modèle ***M*$_R$*IC*** *sont en* ***interaction*** *quand il existe une boucle d'information les reliant, telle que ces agents en soient les extrémités.* ■

Ainsi donc, tandis que Burdea et Coiffet (1993), Burdea (1993) proposaient les trois «I» du virtuel, nous proposons les quatre «I» du réel. Il existe en effet quatre combinaisons d'agents en interaction selon notre modèle. Ces combinaisons sont résumées sur la figure 4.7.

Agent de Contrôle	$\Longleftrightarrow$	Agent Opératoire
Agent de Contrôle	$\Longleftrightarrow$	Agent Externe
Agent de Transcription	$\Longleftrightarrow$	Agent Opératoire
Agent de Transcription	$\Longleftrightarrow$	Agent Externe

FIG. 4.7: Les quatre «I» du réel

☛ Précisons que cette définition de la notion d'interaction contient plus qu'il n'y paraît de prime abord : il ne suffit pas d'avoir une communication bidirectionnelle pour parler d'interaction. Par exemple, selon notre définition, les agents de contrôle et de transcription ne sont pas en interaction : en effet, bien qu'ils soient en communication l'un avec l'autre à la fois dans le sens des actions et dans celui des perceptions, l'agent de transcription n'est pas une extrémité de boucle avec l'agent de contrôle (le processus de traitement d'information $A_{CT} \Longrightarrow P_{TC}$ n'existe pas). En d'autres termes, il y a interaction quand les agents dont capables de se «répondre» l'un l'autre.

4.6 Notion de «réalité»

Comme nous l'avons dit précédemment, le but de ce chapitre était de clarifier ce que l'on peut attendre de la notion de réel. Dans cette optique notre but n'était en aucune manière de *définir* le réel. L'élaboration d'une telle définition est une tâche très ambitieuse, appartenant plutôt au domaine philosophique, et dépassant largement le cadre de nos préoccupations. Au contraire, l'élaboration du modèle ***M*$_R$*IC*** est une manière de *voir*, de *décrire* le réel, plutôt que de le définir.

☛ Dans tout ce qui suit, la notion de réalité, non définie, sera interprétée selon le sens qui lui est communément accordé, et vue au travers du modèle M_RIC. Par conséquent, nous statuons que le modèle M_RIC *est* un modèle cognitif décrivant le réel.

4.6.1 Classes et instances du réel

Au travers du modèle M_RIC, nous constatons qu'il peut être intéressant de décrire le réel grâce à un vocabulaire emprunté à l'orienté objet.

Chaque agent élémentaire du modèle constitue une classe particulière d'objets. Ces classes sont définies, par les entrées et sorties d'information dont elles disposent, ainsi que par les processus de traitement de l'information qu'elles implémentent. Ensuite, les agents mental et physique, puis le modèle M_RIC lui-même s'expriment comme de nouvelles classes «héritant» des quatre classes élémentaires.

Le modèle M_RIC traite de l'instanciation physique de l'humain (agent opératoire), mais aussi de son instanciation mentale. Il en découle qu'au travers de ce modèle, le terme de «réalité» décrit non seulement le monde dans lequel l'opérateur évolue, mais également la manière dont il se le représente. Par là même, la «réalité», telle que décrite par notre modèle, est nécessairement différente pour chaque opérateur. Chaque opérateur est donc caractérisé par son instanciation des trois classes d'agents élémentaires qui le composent, et cette instanciation est nécessairement différente pour chaque opérateur.

4.6.2 Importance de l'interaction

L'importance de l'interaction dans le réel ne s'arrête pas aux quatre «I» que nous avons définis plus haut. L'interaction est un phénomène capital pour l'existence de I dans notre modèle, c'est à dire pour tout ce qui relève de la connaissance de l'opérateur :

L'opérateur possède des modèles «à sens unique» de l'environnement : ces modèles sont relatifs à la (re)connaissance des objets, des sons… mais ils ne sont pas figés. Nous pouvons à tout moment faire face à un objet jusque ici inconnu, et «apprendre», c'est-à-dire étendre notre modèle de connaissance à cet objet. Cependant, si cet objet est inconnu, nous *savons* que c'est un objet. Cette connaissance là est beaucoup plus profonde ; elle est ancrée dans l'interaction. Le concept d'objet ne peut s'apprendre que par l'interaction avec l'environnement : variations de l'image en fonction du mouvement de la tête et des yeux, sensations tactiles… Les connaissances qui régissent notre comportement sont donc essentiellement des connaissances relatives à l'interaction. Ces connaissances sont en fait de quatre types, chaque type étant associé à un type d'interaction, telle que définie précédemment. Le concept d'interaction tient donc une place capitale dans notre vision du réel.

4.6.3 Notion de prédiction

La notion de «prédiction de comportement» est une autre notion importante qui découle des modèles d'interaction que nous venons de décrire. Un modèle d'interaction suffisamment précis permet de *prévoir* l'information reçue en retour d'une information envoyée. Au niveau implicite par exemple, notre connaissance de l'interaction visuelle nous indique que si l'on tourne la tête, tout objet du champ visuel se déplacera en direction opposée. À un niveau conceptuel plus élevé, nous savons qu'en dirigeant la main vers un objet, nous verrons la main bouger, puis un contact se produira. Nous pensons que l'humain en situation d'interaction est constamment (mais inconsciemment) en train de confronter les faits à la prédiction que lui fournissent ses modèles d'interaction. Une contradiction dans les résultats engendre un apprentissage cognitif. Quand ces modèles sont bien éprouvés, ce qui est le cas par exemple des modèles implicites pour un individu adulte, une discordance entre la prédiction et le résultat obtenu est source de malaise en même temps que d'adaptation (Kenedy et al., 1997).

Chapitre 5

La Réalité Virtuelle

Dans le chapitre précédent, nous avons élaboré un modèle, le modèle M_RIC, au travers duquel nous décrivons ce que le sens commun appelle «réalité». Dans ce chapitre, nous proposons de mettre à profit ce modèle afin d'élaborer une définition du virtuel homogène à notre vision du réel, c'est à dire d'un point de vue cognitif et selon l'axe de l'interaction. Cette définition s'exprime au moyen d'un ensemble de nouveaux modèles décrivant de nouvelles situations cognitives, et nous permet de clarifier un certain nombre de termes communément utilisés, tout autant que de nous positionner par rapport aux définitions existantes.

5.1 Le virtuel : une méta-définition

Comment définir le virtuel ? Comme nous l'avons déjà signalé, l'approche cognitive d'une définition du virtuel implique de raisonner non pas d'une manière statique (suis-je ici dans le virtuel ?) mais d'une manière dynamique (comment puis-je passer dans le virtuel ?). En effet, tout système de réalité virtuelle est avant tout réel : l'opérateur interagit avec un ordinateur, des périphériques qui sont bel et bien matériels. D'un point de vue cognitif, la réalité virtuelle est donc un état dans lequel l'opérateur arrive à «oublier» les contraintes matérielles auxquelles il est soumis, c'est à dire, à oublier le système, pour ne se concentrer que sur la résultante de celui-ci.

Pour cette raison, il n'est pas possible de définir le virtuel d'un point de vue cognitif. Le virtuel peut être différent pour chacun d'entre nous. Par contre, nous pouvons espérer définir les *processus de virtualisation*, c'est à dire les phénomènes cognitifs qui font qu'à un instant donné, l'opérateur entrera dans le virtuel. C'est ce que nous proposons de faire dans cette section.

5.1.1 Processus de virtualisation

Un processus de virtualisation est donc un processus cognitif par lequel l'opérateur «oublie» sa réalité et/ou celle de son environnement, et *accepte* une réalité différente. Dans la mesure où nous voyons le réel au travers du modèle M_RIC tel que décrit dans le chapitre précédent, il convient de définir ce qu'est un processus de virtualisation par rapport à ce même modèle. Selon cette optique, la définition recherchée vient en fait assez naturellement : le modèle M_RIC, comme nous l'avons vu, est structurellement fondé sur l'existence de quatre agents élémentaires. Si la virtualisation consiste en la proposition d'une *autre* réalité, par rapport au modèle M_RIC, la virtualisation consisterait en la proposition d'*autres* agents. Ceci nous conduit naturellement à la définition suivante :

Définition 11: Virtualisation

*Étant donnée une situation réelle décrite par le modèle M_RIC, on dira qu'il y a **virtualisation** lorsqu'un ou plusieurs des quatre agents élémentaires du modèle est remplacé par une instance différente de l'instance réelle.* ■

Cette définition nous permet maintenant préciser ce que l'on entend par «réalité virtuelle » :

Vocabulaire: Réalité Virtuelle

*On parlera de **réalité virtuelle** dès lors que pour* un *opérateur donné dans* une *situation donnée, l'ensemble étant décrit par le modèle M_RIC, un ou plusieurs processus cognitifs de virtualisation interviennent.* ●

- ☛ Notons bien que nous sommes en train de parler de processus *cognitifs* et non physiques de virtualisation. Dire qu'un agent du modèle est remplacé par une nouvelle instance non réelle ne signifie pas, d'un point de vue cognitif, qu'elle existe matériellement. Cela signifie que l'opérateur *croit* en cette nouvelle instance, autrement dit qu'il a l'illusion que cette instance est réelle.
- ☛ Nous avons fréquemment associé le terme d'«illusion» à la notion de réalité virtuelle. Comment les phénomènes d'illusion dans la réalité (par exemple les mirages ou les illusions d'optique) se positionnent-ils dans notre définition ? La différence fondamentale est que la réalité virtuelle, selon notre acception, est obtenue grâce à des moyens *artificiels*. Même si des phénomènes naturels d'illusion ou d'interprétation peuvent intervenir (par exemple, la perception du relief sur une image plate), une situation de réalité virtuelle passe nécessairement par l'emploi de périphériques ou de machines artificielles. Cet aspect est en harmonie avec les définitions proposées jusqu'ici.

5.1.2 Types de virtualisation

Étant donné qu'un processus de virtualisation a pour fonction de réinstancier un agent du modèle, quelle(s) forme(s) cette nouvelle instance peut-elle revêtir ? À quoi ressemble t'elle, ou ressemble t'elle seulement à quelque chose ? Pour répondre à cette question, nous proposons d'interpréter l'expression «réalité virtuelle » de deux manières duales, chacune aboutissant à un type particulier de virtualisation :

- ➤ La première idée est de considérer les termes séparément. Selon cette optique, le terme «réalité» fait exactement référence à la vision traditionnelle du concept : ce que nous en percevons chaque jour, ce que le sens commun définit comme tel. L'expression «réalité virtuelle » doit alors être interprétée comme signifiant «une autre version de la réalité», ce qui veut dire une *imitation* de la réalité, une réalité qui pourrait tout à fait être la nôtre. Par rapport au modèle M_RIC, nous aboutissons donc à réinstancier un agent d'une manière telle que cette nouvelle instance soit *réaliste*. Ceci constitue notre **premier type de virtualisation**.
- ➤ La démarche duale consiste à partir du principe que l'expression «réalité virtuelle » est insécable. Selon cette optique, il ne s'agit plus de fournir une version virtuelle mais réaliste de la réalité, mais une virtualité pure, sans aucun antécédent réel. Par rapport au modèle M_RIC, nous aboutissons donc à réinstancier un agent d'une manière totalement imaginaire. On peut par exemple trouver dans le monde artistique des environnements virtuels totalement imaginaires, comme des tableaux abstraits en trois dimensions constitués de couleurs et de sons. Ceci constitue notre **deuxième type de virtualisation**.

Cette distinction basée sur l'examen de l'expression «réalité virtuelle» rejoint la classification des applications de réalité virtuelle proposée par Krueger (1991) (Cf. section 3.4 en page 22). Ce que Krueger entend par «réalité artificielle » correspond à l'aboutissement de notre deuxième

type de virtualisation. Ici encore, cette distinction majeure entre réalisme et réalité artificielle est tempérée par l'introduction des «réalités altérées» proposées par Vanderheiden et Mendenhall (1994) (Cf. section 3.5 en page 22). Ces différentes expressions peuvent maintenant être positionnées par rapport à notre vision du virtuel, et donc en particulier par rapport au modèle M_RIC :

Vocabulaire: Réalité Virtuelle Artificielle

*Étant donnée une situation réelle décrite par le modèle M_RIC, on parlera de **réalité virtuelle artificielle**, selon l'expression proposée par Krueger (1991), dès lors qu'il y a virtualisation de la situation et que les processus de virtualisation impliqués sont du deuxième type.* ●

Vocabulaire: Réalité Virtuelle Altérée

*Étant donnée une situation réelle décrite par le modèle M_RIC, on parlera de **réalité virtuelle altérée**, selon l'expression proposée par Vanderheiden et Mendenhall (1994), dès lors qu'il y a virtualisation de la situation, que les processus de virtualisation impliqués sont majoritairement, mais non exclusivement du premier type.* ●

Notons que la création de réalités artificielles est une démarche bien souvent plus délicate que celle de création de réalités altérées. Il est en effet assez difficile d'imaginer des instances de réalité totalement imaginaires, ceci pour deux raisons :

- ➤ Tout d'abord, notre système cognitif est profondément ancré dans le monde réel. Toute notre connaissance, y compris la connaissance théorique, est fondée sur notre interaction avec l'environnement réel. Il est donc particulièrement délicat de s'en abstraire.
- ➤ Deuxièmement, comme nous l'avons déjà remarqué, le corps de l'opérateur est une partie intégrante du monde réel, en tant qu'objet physique. Imaginer par exemple un monde virtuel n'obéissant pas aux lois de la physique produirait implicitement une cassure cognitive dans la mesure où notre corps ne pourrait malgré tout pas s'abstraire des lois physiques réelles qui le gouvernent.

Il est donc beaucoup plus facile de concevoir des mondes virtuels «altérés», c'est à dire inspirés du réel, mais fonctionnant différemment.

- ☛ Comme la notion de réalité altérée semble le confirmer, les deux types majeurs de virtualisation ne sont pas mutuellement exclusifs : une même situation peut être virtualisée simultanément par des processus des deux types. Comme nous le constaterons d'ailleurs dans la deuxième partie de ce document, les situations mettant en œuvre simultanément de tels processus sont en fait les cas les plus intéressants.
- ☛ Certaines expérimentations aboutissent à des notions intéressantes au niveau du mélange réel / virtuel. À titre d'exemple, citons le concept de «réalité virtualisée» (Kanade et al., 1999) : une scène est filmée en temps réel par plusieurs caméras situées à des endroits différents. Si le nombre de caméras est suffisant, il est possible, par interpolation des diverses prises de vues, de reconstituer l'image correspondant à un point de vue arbitraire de la scène considérée. Ces images sont *réelles* dans le sens où elles sont issues d'images vidéo et restituent une scène réelle, mais elles sont également *virtuelles* dans le sens où aucune caméra ne les a réellement prises.

5.1.3 Importance de l'interaction

Nous avons défini les processus de virtualisation comme des processus de réinstanciation des agents du modèle. En se replacement précisément dans le domaine lexical de l'orienté objet, cette définition sous entend en particulier que les nouvelles instances de nos agents ont leur *classe* en

commun avec les agents initiaux. Or la classe de chaque agent était définie (Cf. section 4.6.1 en page 35) par l'architecture de ses entrées et sorties d'information ainsi que par ses processus de traitement d'information. Il en découle que quelle que soit la situation (virtualisée) considérée, la topographie de connexion entre les différents agents est conservée.

Ceci a notamment l'implication fondamentale que les quatre types d'interaction que nous avons définis dans le chapitre précédent sont toujours d'actualité, même si certains d'entre eux sont virtualisés. Autrement dit, la conservation de nos quatre types d'interaction est une condition nécessaire pour parler de réalité virtuelle. On constate donc à nouveau l'importance de l'interaction dans la notion de réalité virtuelle, et d'autre part que notre point vue sur cette notion reste en harmonie avec l'ensemble des définitions proposées jusqu'ici.

Pour obtenir un panorama exhaustif de ce que l'on peut appeler «réalité virtuelle », et compte tenu de ce qui précède, il faut et il suffit d'envisager tous les cas de virtualisation du modèle M_R^*IC, c'est à dire d'examiner la combinatoire de réinstanciation des agents du modèle. Ceci constitue l'objet des sections qui suivent.

5.2 Virtualisation de l'environnement

5.2.1 Description

La première étape de virtualisation qui vient naturellement à l'esprit est celle qui consiste à virtualiser l'environnement. Par rapport au modèle M_R^*IC, ce premier niveau de virtualisation consiste donc à réinstancier l'agent externe, ainsi qu'à «dévier» cognitivement les chemins d'information A_{OE} et P_{EO} vers cette nouvelle instance d'agent externe. La situation ainsi obtenue est représentée sur la figure 5.1, par une première variante du modèle M_R^*IC nommée M_R^*IC-*e*.

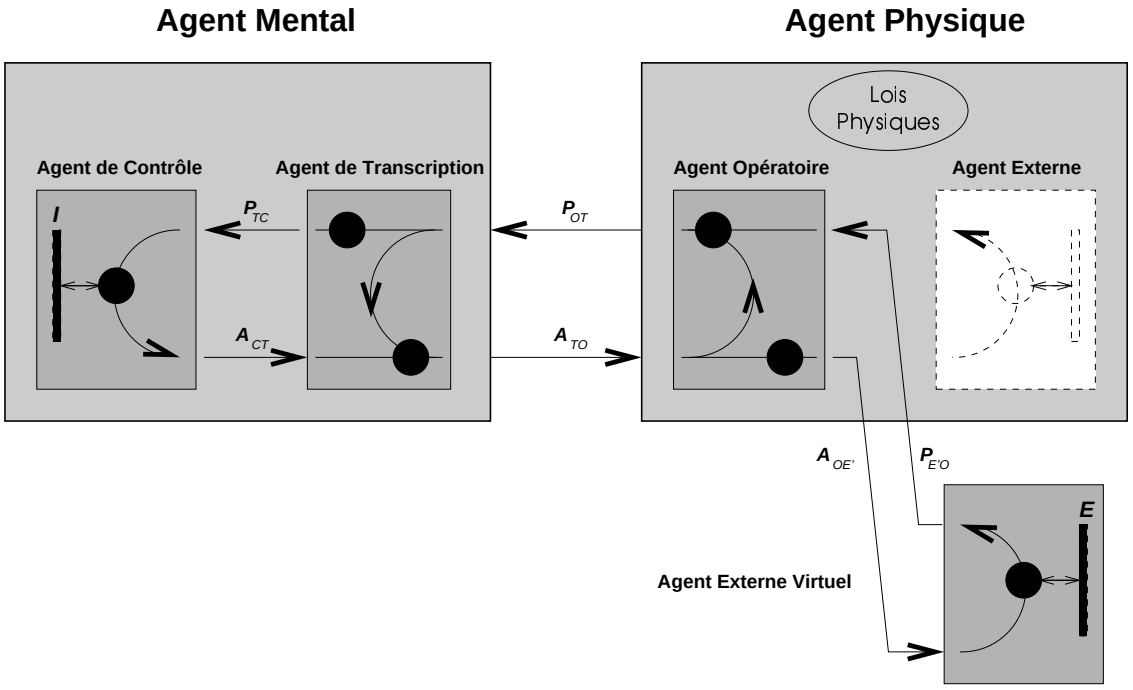

FIG. 5.1: M_R^*IC-*e* : virtualisation de l'agent externe.

La caractéristique principale de cette situation est que seul l'environnement (l'agent externe) est virtualisé. L'opérateur est donc censé garder pleine conscience de son corps, mais dans une

certaine mesure faire abstraction de son environnement réel afin de se concentrer sur la nouvelle instance, virtuelle, de celui-ci. Cette nouvelle modélisation nous permet également de préciser, toujours par rapport à notre modèle initial, le point de vocabulaire suivant :

Vocabulaire: Environnement Virtuel

Étant donnée une situation réelle décrite par le modèle ***M_R^*IC****, on parlera d'****environnement virtuel*** *dès lors qu'un processus de virtualisation de l'agent externe intervient.* ●

Nous devons maintenant nous demander si cette première étape de virtualisation correspond à quelque chose de vécu dans le domaine applicatif de la réalité virtuelle que nous connaissons aujourd'hui. Nous pensons que ce premier niveau de virtualisation est représentatif de ce que l'on pourrait appeler la «réalité virtuelle de bureau», c'est à dire des systèmes ne comprenant pas ou peu de périphériques spécialisés, en particulier aucun périphérique de type immersif. De tels systèmes existent à l'heure actuelle pour des applications de CAO utilisant des techniques de réalité virtuelle, et dans une certaine mesure, pour des applications dites de «navigation 3D» apparues récemment avec le langage VRML (Cf. glossaire en page 216).

Pour fixer les idées, prenons un exemple concret d'application, celui décrit par Duchon et Balet (1996) : il s'agit d'une application de CAO destinée à prototyper des modèles de satellites et à s'assurer de la cohérence de ces modèles avant de passer en phase de réalisation. L'opérateur dispose d'une station de travail graphique ainsi que de quelques périphériques de réalité virtuelle, tels un trackball (Cf. glossaire en page 216) et des lunettes stéréoscopiques. Un logiciel de CAO permet à l'opérateur de créer et d'examiner des modèles virtuels de satellites en assemblant des «briques de base», c'est à dire des pièces élémentaires que l'on retrouve sur tous les modèles.

Afin que l'on puisse effectivement parler d'environnement virtuel, nous devons montrer en quoi à la fois la perception (P_{EO}) et l'action (A_{OE}) sont virtualisés dans cette situation.

5.2.2 Virtualisation de l'action

La première chose à constater est qu'il y a effectivement action sur l'environnement : par l'intermédiaire du trackball et de mécanismes de sélection sur l'interface graphique, l'opérateur est capable de modifier l'environnement en ajoutant, supprimant ou déplaçant des pièces composant le satellite. L'opérateur est également capable de modifier son point de vue sur la scène afin de percevoir son modèle en cours d'élaboration sous des angles différents. À quel moment peut-on parler de «virtualisation des actions» ? En référence à la version ***M_R^*IC-e*** du modèle (figure 5.1 en page précédente), l'action est virtualisée quand $A_{OE'}$ prend le dessus sur A_{OE}, c'est à dire que l'opérateur «oublie» qu'il interagit avec son véritable environnement, et «ressent» l'interaction directe avec l'environnement virtuel.

Concrètement, ce phénomène peut être vu comme l'adaptation de l'opérateur aux outils d'interaction proposés par le système. Une particularité fondamentale de ce type de système est que l'interaction ne se produit pas de manière naturelle : l'opérateur ne se sert pas de ses mains comme il le ferait naturellement, mais doit *apprendre* à utiliser l'interface proposée par le système (souris, clavier, joystick, interface graphique etc). Tant que cet apprentissage n'est pas complet, l'opérateur doit encore fournir un travail cognitif non négligeable pour parvenir à ses fins. Par contre, une fois le système d'interaction maîtrisé, l'opérateur oubliera qu'il ne dispose en fait que d'une interface symbolique d'action sur l'environnement virtuel, et associera mentalement ses actions physiques directement à leur résultante sur cet environnement. Ce phénomène d'«accoutumance» à l'interface, caractéristique d'un processus de virtualisation de A_{OE} est particulièrement visible dans d'utilisation intensive de jeux vidéos, où les utilisateurs finissent par acquérir des *réflexes* qui ne sont pas du tout innés, dans la mesure où ils concernent l'action sur les touches d'un clavier ou les boutons d'une souris.

5.2.3 Virtualisation de la perception

De manière réciproque à l'action, constatons tout d'abord qu'il y a effectivement perception de l'environnement : par l'intermédiaire de l'écran de la station graphique, l'opérateur perçoit l'environnement de travail selon la modalité visuelle, critique dans une grande majorité des cas. L'utilisation de lunettes stéréoscopiques vient accroître le réalisme de la perception en y ajoutant une information de relief. À quel moment peut-on parler de «virtualisation de la perception» ? En référence à la version ***MIC-e*** du modèle (figure 5.1 en page 40), la perception est virtualisée quand $P_{E'O}$ prends le dessus sur P_{EO}, c'est à dire que l'opérateur «oublie» qu'il perçoit son véritable environnement, et «ressent» une perception directe de l'environnement virtuel.

Concrètement, un tel phénomène correspond par exemple à ne plus percevoir l'écran de la station graphique comme tel, mais à l'interpréter comme une «fenêtre» sur l'environnement virtuel. Abstraction est ainsi faite de la réalité matérielle des objets immédiats, et l'environnement virtuel est alors perçu comme étant «ici et maintenant». La perception est virtualisée. Pour faire à nouveau référence aux phénomènes cognitifs expérimentés dans les jeux vidéos, une expression caractéristique de ce phénomène de virtualisation est la position des joueurs devant l'écran. L'on verra par exemple ceux-ci se déplacer légèrement pour tenter de mieux voir à travers une fenêtre virtuelle, alors que ce déplacement n'a évidemment aucun effet dans la mesure où le point de vue du joueur ne dépend pas de sa position physique réelle. Cependant, ce comportement devant l'écran montre bien que la perception du joueur est «abusée» par l'environnement virtuel, et que P_{EO} est virtualisée.

5.2.4 Types d'environnement virtuels

Nous avons défini plus haut deux types opposés de virtualisation, ce qui nous a également permis de préciser les expressions «réalité altérée» et «réalité artificielle». Ces considérations s'appliquent en particulier aux environnements virtuels, tels que nous venons de les définir.

L'application ayant servi d'illustration à notre raisonnement (Duchon et Balet, 1996), ainsi que la plupart des applications de CAO, ont pour vocation principale de produire des environnements virtuels du premier type, c'est à dire des environnements simulant la réalité. La raison essentielle en est que ces environnements virtuels sont souvent destinés à être réalisés concrètement dans un futur proche.

À l'opposé de ces environnements virtuels du premier type, on trouve ceux du deuxième type, c'est à dire sans antécédent réel : des **environnements virtuels artificiels**, par extrapolation de l'expression proposée par Krueger (1991). De tels environnements se rencontrent essentiellement dans les applications artistiques (Kissileva, 1998) : environnements générant de la musique à partir du mouvement, tableaux tridimensionnels n'existant qu'en présence du spectateur...

Cela dit, dans une optique plus applicative et opérationnelle, la conception d'environnements hybrides, à la fois réalistes et artificiels, que l'on pourrait donc qualifier d'**environnements virtuels altérés** par extrapolation de l'expression proposée par Vanderheiden et Mendenhall (1994) revêt un intérêt tout particulier. Un des atouts majeurs de la réalité virtuelle est justement de pouvoir ne pas se cantonner à la simulation du réel, mais de fournir à l'utilisateur des fonctionalités dont il ne disposerait pas dans un véritable environnement. C'est ainsi que grâce aux techniques et concepts de réalité virtuelle, de nombreuses applications jusqu'ici impossible à mettre en œuvre ont pu voir le jour. Pour mémoire, citons quelques exemples :

- L'application de CAO décrite précédemment implémente en fait une version altérée d'environnement réel. En effet, l'opérateur est capable par un simple mouvement de souris, de manipuler virtuellement des pièces de satellite (comme des panneaux solaires) qui en réalité ne sont pas maniable aussi facilement. L'«altération» de la réalité consiste en fait ici à supprimer la gravité. L'opérateur obtient donc une fonctionnalité que seule la réalité virtuelle pouvait lui fournir. Un autre application bien connue de la même idée réside dans les systèmes d'aménagement de cuisine ou de bureaux (Fagot, 1998). L'utilisateur dispose de tous les éléments de base du mobilier, manipulables à loisir grâce à la souris, et peut

virtuellement aménager en quelques minutes l'espace dont il dispose, avant de se décider fermement sur l'achat des composants.

- Un autre type d'altération de la réalité est fréquemment rencontré dans les navigateurs VRML tels celui décrit par Balaguer et Gobetti (1995) : parmi les différents modes de navigation, le mode «survol» permet cette fois à l'utilisateur (et non plus à l'environnement) lui-même de s'abstraire de la gravité. Celui-ci peut alors survoler virtuellement des endroits comme des grandes villes, chose impossible à faire dans la réalité sans un équipement considérable.
- Un dernier exemple de mise à profit d'une réalité altérée consiste en l'implémentation d'un musée virtuel (Miller et al., 1992), dans lequel le spectateur peut non seulement admirer des toiles de maître, mais passer instantanément d'une pièce à l'autre, supprimant ainsi la contrainte temporelle du mouvement, ou encore faire fi de la matière en traversant directement les murs.

Tous ces exemples ont pour trait caractéristique qu'ils implémentent des environnements virtuels principalement réalistes, mais dont certaines composantes sont artificielles, se démarquent du comportement réel.

5.2.5 Résumé

Il nous semble maintenant opportun de résumer l'ensemble des points essentiels que nous venons d'expliciter dans cette section.

- Ce premier degré de virtualisation consiste en la réinstanciation de l'agent externe, c'est à dire la mise en interaction de l'opérateur avec un environnement virtuel. Pour ce faire, les chemins d'information A_{OE} et P_{EO} doivent faire l'objet d'une *déviation* cognitive, c'est à dire que l'opérateur doit arriver à s'abstraire des chemins naturels pour se focaliser sur les chemins virtualisés.
- Concrètement, les systèmes descriptibles par ce niveau de virtualisation correspondent aux applications de type «réalité virtuelle de bureau», c'est à dire aux systèmes de réalité virtuelle non immersifs. Ces applications, bien que devant nécessairement fournir un haut niveau d'interaction, ont pour particularité que l'action sur l'environnement virtuel ne se fait pas d'une manière naturelle à l'opérateur, mais au moyen d'une interface proposée par le système, et dont l'opérateur doit apprendre à se servir. Comme artefact de conception de tels systèmes, la représentation et la gestion d'un modèle de l'opérateur dans l'environnement virtuel considéré est en général extrêmement limitée, voire inexistante. La plupart du temps, cette représentation se borne à ne conserver que l'information relative au point de vue géographique de l'opérateur sur l'environnement.
- On trouve dans les applications de cette catégorie un panorama complet des types de virtualisation : environnement réalistes, comme la reproduction de monuments historiques ou de sites archéologiques, environnements totalement artificiels pour des applications à vocation artistique, mais un intérêt tout particulier est porté aux environnements altérés, c'est à dire majoritairement réalistes, mais dont certaines composantes virtuelles permettent la réalisation d'applications jusqu'ici non envisageables.

5.3 Virtualisation de l'agent physique

5.3.1 Description

Après avoir examiné un premier niveau de virtualisation, celui de l'agent externe, nous pouvons remonter d'un degré supplémentaire dans le modèle et envisager une virtualisation de l'agent physique tout entier. Ce processus de virtualisation consiste donc à réinstancier non seulement l'agent externe, mais aussi l'agent opératoire, ainsi qu'à «dévier» cognitivement les chemins

d'information A_{TO} et P_{OT} vers la nouvelle instance de l'agent physique. La situation ainsi obtenue est représentée sur la figure 5.2, par une deuxième variante du modèle M_R^*IC nommée M_R^*IC-*m*.

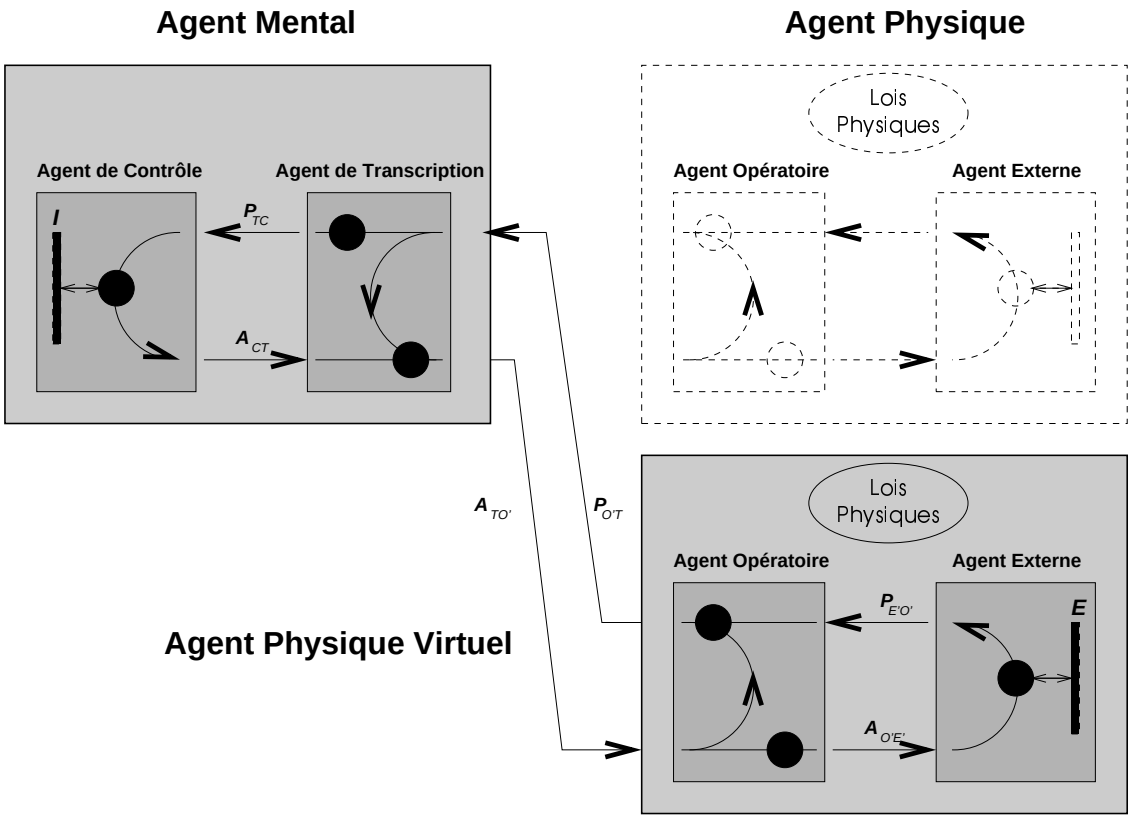

FIG. 5.2: M_R^*IC-*m* : virtualisation du monde.

Contrairement à une simple virtualisation de l'agent externe, cette situation est caractérisée par le fait que l'agent physique tout entier est virtualisé, c'est à dire que l'agent opératoire (le corps de l'opérateur) fait également l'objet d'une virtualisation. En se replaçant dans le contexte lexical Opérateur - Environnement - Monde, nous pouvons maintenant préciser le point de vocabulaire suivant :

Vocabulaire: Monde Virtuel

Étant donnée une situation réelle décrite par le modèle M_R^*IC, on parlera de ***monde virtuel*** *dès lors qu'un processus de virtualisation de l'agent physique intervient.* ●

Nous devons maintenant nous demander si cette deuxième étape de virtualisation correspond à quelque chose de vécu dans le domaine applicatif de la réalité virtuelle que nous connaissons aujourd'hui. Nous pensons que ce deuxième niveau de virtualisation est représentatif de ce que l'on a coutume d'appeler la «réalité virtuelle immersive», dont les applications et les implications sont extrêmement nombreuses à l'heure actuelle.

Selon Burdea (1993), Burdea et Coiffet (1993) l'immersion consiste à procurer à l'opérateur une sensation de «présence physique» dans le monde virtuel, c'est à dire que celui-ci doit oublier son existence dans l'environnement immédiat, et se *croire* dans le monde virtuel. Ce terme fut sans doute adopté pour sa forte connotation physique : l'accent est mis sur la sensation physique

que procure le milieu immersif sur le corps. Pimentel et Teixeira (1994) précisent d'ailleurs que d'un point de vue opérationnel, un système d'immersion idéal devrait être capable de stimuler tous les sens humains, c'est à dire aussi bien le toucher, l'odorat et même le goût, que la vue et l'ouïe, jusqu'ici mieux maîtrisés.

À l'heure actuelle, un système complet d'immersion requiert une station graphique pour générer le monde virtuel, et un ensemble important de périphériques immersifs, en particulier une combinaison de données, des gants de données et un visiocasque. L'ensemble de ces périphériques doivent permettre à l'opérateur d'agir sur (d'évoluer dans) le monde virtuel d'une manière naturelle, comme il le ferait dans son propre environnement.

☛ La notion de «périphérique immersif» que nous venons d'introduire ici correspond à ce que Fuchs (1996) appelle les «interfaces comportementales», c'est à dire les périphériques matériels permettant à l'humain d'agir naturellement et ne nécessitant par conséquent aucun apprentissage préalable.

Afin que l'on puisse effectivement parler de monde virtuel, nous devons montrer en quoi à la fois la perception (P_{TO}) et l'action (A_{OT}) sont virtualisées dans cette situation.

5.3.2 Virtualisation de l'action

La première chose à constater est qu'il y a effectivement action sur le monde virtuel : l'utilisation de combinaison et gants de données permet au système informatique de connaître en permanence la position de l'opérateur, jusqu'à la position des doigts pour les systèmes les plus performants, et de reproduire cet état physique au travers d'un avatar (Cf. glossaire en page 215) dans le monde virtuel. Il y a donc action sur l'environnement virtuel, mais en premier lieu sur l'agent opératoire virtuel, puisque celui-ci suit les attitudes de l'opérateur. Comme tout mouvement naturel de l'opérateur est traduit par un mouvement similaire de l'avatar, l'opérateur dispose en théorie de ses capacités d'interactions naturelles, notamment la saisie des objets. Dans des systèmes coûteux tels que le CAVE (Browning et al., 1993), l'opérateur a même la possibilité de se déplacer sur une surface plus ou moins étendue. À quel moment peut-on parler de «virtualisation des actions» ? En référence à la version ***M**R**I**C**-m* du modèle (figure 5.2 en page précédente), l'action est virtualisée quand $A_{TO'}$ prend le dessus sur A_{TO}, c'est à dire que l'opérateur «oublie» qu'il commande son propre corps, et «ressent» l'action directe sur son avatar virtuel.

Concrètement, une bonne virtualisation de A_{TO} intervient quand le système parvient à gérer un isomorphisme total entre l'opérateur son avatar, chose qui reste, comme nous le verrons plus loin, extrêmement complexe sur le plan technologique. À l'opposé des systèmes que nous avons qualifiés de «réalité virtuelle de bureau», le principe des systèmes immersifs est de laisser l'opérateur agir naturellement dans le monde virtuel. Cela se traduit par exemple par le fait que pour saisir et manipuler un objet virtuel, l'opérateur ne doit effectuer aucune commande spéciale (sur souris, trackball…), mais doit effectuer le geste *naturel* de saisie qu'il aurait effectué dans la réalité. Si ce mécanisme est correctement mis en œuvre, le processus de virtualisation de A_{TO} permettra à l'opérateur d'*accepter* cognitivement son avatar comme nouvel agent opératoire. Ce processus d'association de l'avatar à son propre agent opératoire, caractéristique de la virtualisation de A_{TO} est par exemple bien illustré dans un système de jeu de tennis virtuel (Thalmann, 1998) où l'on constate que les joueurs, malgré les contraintes matérielles qui leur sont imposées, arrivent à disputer des matchs avec un minimum d'habitude du système.

5.3.3 Virtualisation de la perception

De manière réciproque à l'action, constatons tout d'abord qu'il y a effectivement perception du monde virtuel : par l'utilisation d'un casque de visualisation, l'opérateur perçoit correctement l'environnement virtuel, avec une information de relief, ce qui constitue comme on l'a dit la modalité principale de perception à fournir dans la majorité des cas. Les casques de visualisation vont souvent de pair avec les casques audio, permettant quand à eux de restituer la modalité

sonore, souvent stéréophonique. Des études sont également menées sur la restitution spatiale du son, en particulier dans les laboratoires de l'IRCAM. Pour des modèles de gants de données haut de gamme, en particulier la catégorie des exosquelettes (Cf. glossaire en page 215), des fonctionalités de retour d'effort sont offertes, ce qui constitue un commencement de restitution de perception tactile. Des études sont également menées sur la possibilité de restituer le toucher par conduction de faible courant électrique. À quel moment peut-on parler de «virtualisation de la perception» ? En référence à la version ***MRIC-m*** du modèle (figure 5.2 en page 44), la perception est virtualisée quand $P_{O'T}$ prends le dessus sur P_{OT}, c'est à dire que l'opérateur «oublie» qu'il perçoit le harnachement des périphériques d'immersion dont il est revêtu, et «ressent» une perception directe de son avatar virtuel.

Concrètement, l'efficacité de tels systèmes dépend d'une manière cruciale du retour perceptif lié aux interactions avec l'environnement, et pas seulement des simples perceptions statiques : par exemple, l'utilisation d'un casque de visualisation plutôt que de lunettes stéréoscopiques devant un écran offre au système de réalité virtuelle la possibilité de modifier le point de vue de l'opérateur quand celui-ci tourne la tête, chose impossible à réaliser avec un écran d'ordinateur. La possibilité de voir son propre avatar (en baissant la tête par exemple) est une autre composante essentielle de l'état d'immersion. Ce phénomène qui pousse l'opérateur à faire abstraction du côté artificiel de la perception est caractéristique d'une virtualisation de P_{OT}, et trouve une illustration démonstrative dans les récentes expériences de «Cyber Sex», où deux personnes, réellement partenaires dans la vie déclaraient avoir ressenti des émotions aussi vives qu'elles auraient pu l'être dans la réalité.

5.3.4 Types de mondes virtuels

Les distinctions que nous avons établies dans la section précédente, sur les types d'environnements virtuels sont encore parfaitement d'actualité ici, dans la mesure où un monde virtuel est un agent physique virtuel au sens du modèle ***MRIC***, donc un ensemble environnement / avatar. L'apport de cette section réside en revanche dans les différents types de virtualisation de l'agent opératoire.

La très grande majorité des applications immersives à l'heure actuelle utilise des virtualisations d'agent opératoire du premier type (réalistes). En effet, ces applications ont pour but essentiel de transporter l'opérateur lui-même (c'est à dire le moins virtualisé possible) dans un autre monde. Une catégorie majeure d'applications immersives regroupe les systèmes d'entraînement en vue d'opération en milieu hostile ou en situation critique : il existe par exemple des systèmes immersifs reproduisant l'environnement d'une centrale nucléaire (Fertey et al., 1995) afin de fournir un entraînement réaliste à l'opérateur en vue d'éventuelles fuites de radiation. Le but de ces systèmes est d'immerger l'opérateur tel qu'il est dans le monde virtuel. On est donc encore dans le domaine d'une virtualisation du premier type. Une ébauche de virtualisation d'un type différent est rencontrée dans certains jeux vidéos où l'utilisateur a par exemple la possibilité d'endosser le rôle d'un personnage qu'il n'est pas, y compris d'un personnage de sexe différent. Ces exemples de virtualisation restent cependant de nature «réaliste».

Comme on le constate, l'idée de virtualisation du deuxième type pour l'agent opératoire, c'est à dire l'instanciation d'un agent opératoire totalement artificiel semble complètement absente du domaine applicatif que nous connaissons. Cette idée est donc soit effectivement inexploitée, soit exploitée d'une manière que nous n'avons pas encore identifiée. La section «Réalité Virtuelle et Télé-Opération» nous permettra de lever le voile sur cette ambiguïté.

5.3.5 Résumé

Il nous semble maintenant opportun de résumer l'ensemble des points essentiels que nous venons d'expliciter dans cette section.

➤ Ce deuxième degré de virtualisation consiste en la réinstanciation de l'agent physique tout entier, c'est à dire la mise en interaction de l'opérateur avec un monde virtuel (environ-

nement virtuel, plus avatar). Pour ce faire, les chemins d'information A_{TO} et P_{OT} doivent faire l'objet d'une *déviation* cognitive, c'est à dire que l'opérateur doit arriver à s'abstraire des chemins naturels pour se focaliser sur les chemins virtualisés.

- ➤ Concrètement, les systèmes descriptibles par ce niveau de virtualisation correspondent aux applications dites «immersives», c'est à dire dans lesquelles l'opérateur ressent physiquement sa présence dans le monde virtuel. Contrairement aux applications décrites dans la section précédente, les applications immersives ont pour particularité que l'interaction avec le monde virtuel doit se faire de la manière qui est naturelle à l'opérateur, sans aucun apprentissage d'une quelconque interface de contrôle. Comme démontré par le modèle ***M*$_R$*IC*-*m***, une immersion totale de l'opérateur nécessite une gestion complète des interactions naturelles entre l'agent opératoire et l'agent externe, ainsi que des informations proprioceptives.
- ➤ Contrairement aux applications décrites dans la section précédente, la virtualisation de l'agent opératoire consiste majoritairement en une virtualisation du premier type, c'est à dire à vocation réaliste. Le but recherché est principalement d'immerger l'opérateur lui-même dans le monde virtuel, idée que l'on retrouve par exemple dans les systèmes de simulation et d'entraînement par ordinateur. Les sections suivantes, ainsi que la deuxième partie relative à la notion d'assistance nous aideront à envisager d'autres types de virtualisation de l'agent opératoire.

5.4 Imbrication de la réalité virtuelle

Les deux sections précédentes nous ont permis d'identifier selon notre point de vue les deux grandes classes d'applications de réalité virtuelle : la «réalité virtuelle de bureau» d'une part, où l'opérateur, physiquement intègre, interagit avec un environnement virtuel au moyen de périphériques non immersifs (écran d'ordinateur, joystick. . .), et la réalité virtuelle immersive, d'autre part, dans laquelle l'opérateur est physiquement plongé dans un monde virtuel, et peut interagir avec lui d'une manière naturelle, grâce à des périphériques d'immersion (casques, gants de données. . .).

Par une simple combinaison logique des deux modèles correspondants (***M*$_R$*IC*-*e*** et ***M*$_R$*IC*-*m***), nous pouvons aboutir à une situation de réalité virtuelle «au carré», dans laquelle les deux grandes classes de situation sont imbriquées et interviennent simultanément. Considérons tout d'abord un système d'immersion traditionnel : l'opérateur est revêtu d'un visiocasque, d'une combinaison et de gants de données. Il est immergé dans un environnement de synthèse. Imaginons maintenant que dans cet environnement de synthèse se trouve un ordinateur (virtuel) doté par exemple d'un jeu vidéo (virtuel) ou d'un logiciel de CAO (virtuel). L'opérateur doit donc, au sein du premier monde virtuel jouer (virtuellement) à un jeu qui lui propose un deuxième environnement virtuel ! Cette situation de réalité virtuelle en cascade est illustrée par le modèle ***M*$_R$*IC*-*me*** sur la figure 5.3 en page suivante.

5.5 Virtualité des environnements réels

Dans les sections qui précèdent, les exemples d'application que nous avons fournit sont tous des exemples mettant en œuvre des environnements ou des mondes de *synthèse*, c'est à dire générés par ordinateur (systèmes de CAO, musées virtuels, simulation d'environnements hostiles. . .). Rappelons-nous également que les définitions proposées jusqu'ici s'accordaient toutes pour spécifier qu'un monde virtuel était avant tout une base de données infographique. Dans cette section, nous voudrions montrer que l'approche cognitive qui est la nôtre n'impose pas cette caractéristique, et que bien au contraire, elle peut dans certains cas faire de la réalité de bien meilleurs mondes virtuels que les mondes de synthèse.

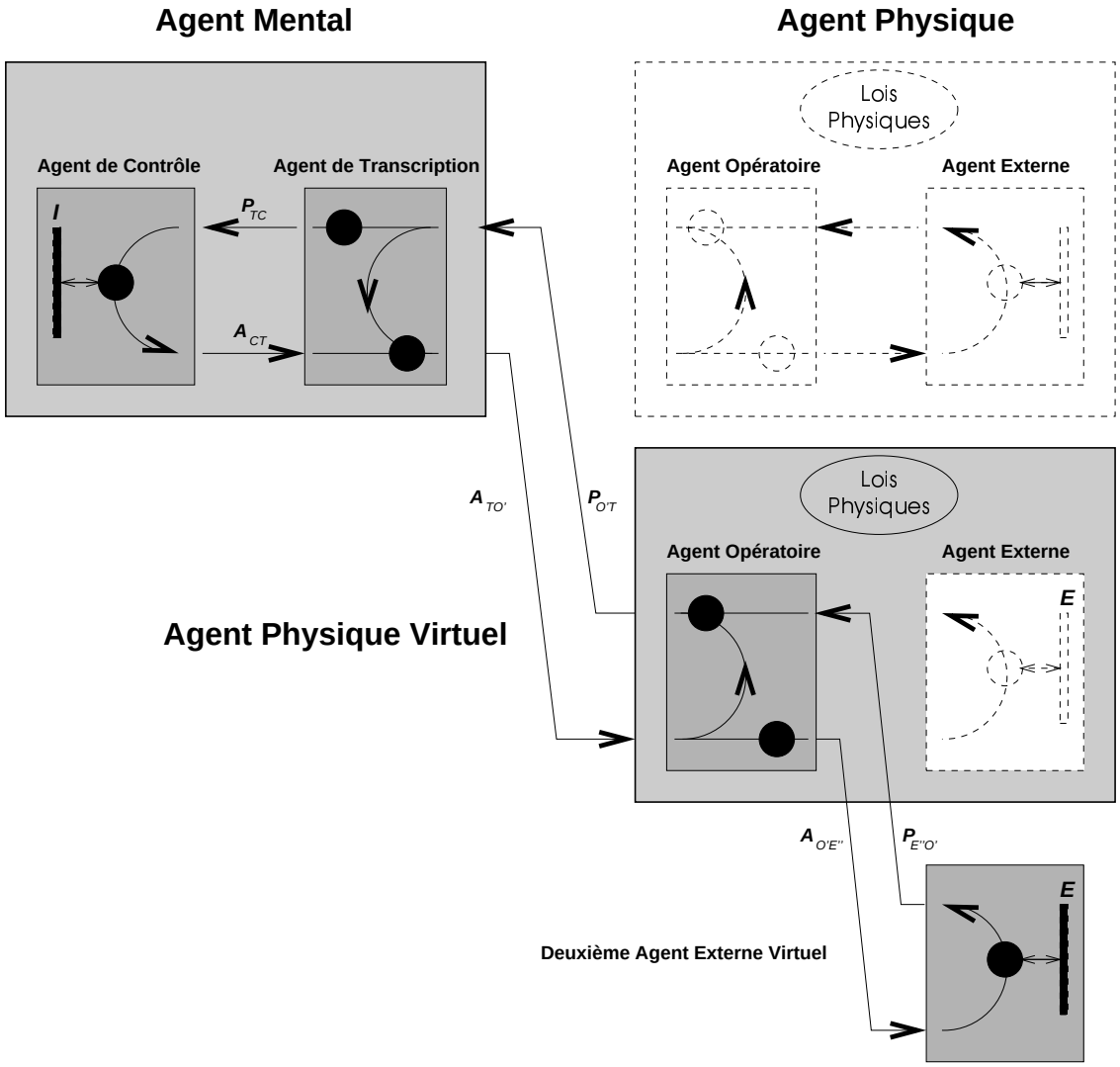

FIG. 5.3: M_RIC*-me* : Réalité virtuelle au deuxième degré

5.5.1 Réalité virtuelle, modèle M_RIC et triangle sémiotique

Comme nous l'avons déjà noté à plusieurs reprises, l'obtention d'une situation de réalité virtuelle passe par la virtualisation d'un (ou plus) agent du modèle M_RIC. Une conséquence importante de cette caractéristique est que l'opérateur n'est *jamais* en contact direct avec l'environnement virtuel concerné : il perçoit celui-ci au travers d'un écran ou grâce à un visiocasque, il agit sur celui-ci grâce à une interface de contrôle (souris, joystick…) ou grâce à des périphériques d'immersion (gants de données). Dès lors que le contact direct est impossible, comment faire la différence entre en environnement réel et un environnement de synthèse ? Les seules raisons actuelles qui permettent de faire une telle différence sont des raisons de nature technologique : piètre qualité des systèmes d'immersion, caractère visiblement artificiel des images de synthèse… Mais ces raisons ne constituent en rien un argument théorique valable. Il est par exemple envisageable que dans l'avenir, les moteurs graphiques puissent produire des images qui ne soient plus distinguables de la réalité filmée.

Tout ceci montre que d'un point de vue cognitif et compte tenu du fait que l'interaction avec

l'environnement virtuel est toujours indirecte, c'est à dire qu'elle passe par un système artificiel de transmission d'information, un environnement réel peut jouer le même rôle qu'un environnement de synthèse dans les situations décrites par M_RIC-*e* et M_RIC*m*. Cette idée peut être illustrée d'une manière intéressante par une adaptation du triangle sémiotique (Cf. glossaire en page 216) à la réalité virtuelle (Ogden et Richards, 1923, Grumbach, 1995, 1994).

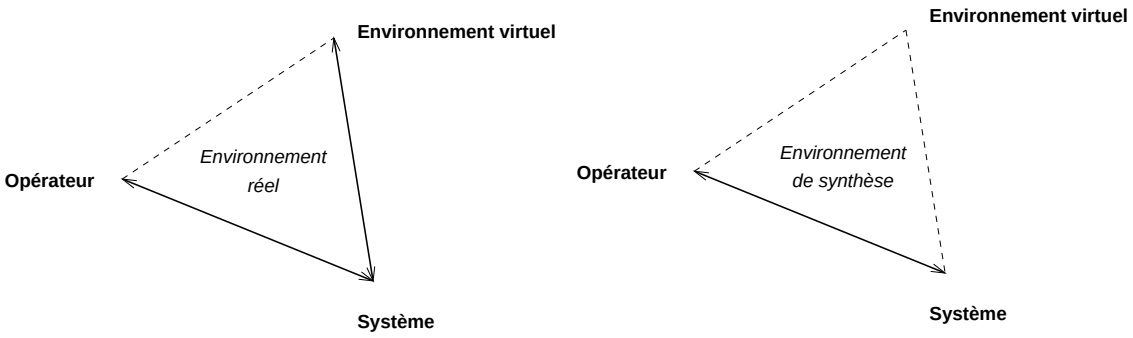

FIG. 5.4: Réalité Virtuelle et Triangle Sémiotique

Le triangle sémiotique, dans le cas du langage par exemple, met en scène un sujet, un objet et un locuteur qui prononce par exemple le nom de l'objet (signifiant). Le sujet peut accéder à l'objet directement, ou indirectement par son nom. Une adaptation du triangle sémiotique à la réalité virtuelle, décrite par Grumbach et Verna (1996) et illustrée en figure 5.4 associe le sujet à l'opérateur, l'objet au monde physique (s'il existe) et le locuteur au système de transmission (voire de traitement) de l'information. La situation devient virtuelle lorsque le chemin entre l'opérateur et l'environnement est rompu. Dès lors que l'opérateur doit nécessairement passer par le système pour accéder à l'environnement, il ne lui est plus possible de savoir si l'environnement en question existe ou pas.

☛ Bien que les auteurs n'y fassent pas explicitement référence, Fuchs et al. (1999) présentent un modèle triangulaire analogue, et donc proche du triangle sémiotique, mais dans un but légèrement différent. Ce modèle, reproduit sur la figure 5.5 en page suivante, a pour vocation de représenter les situations immersives : l'opérateur et l'interface sont liés par les actions et perceptions *réelles*, l'interface et le monde virtuel sont liés par les action et perceptions *programmées*, et le lien direct (inexistant) entre opérateur et le monde virtuel représente le comportement naturel de l'opérateur, que tout système d'immersion souhaite atteindre.

5.5.2 Immersion réelle

De même que les environnements réels peuvent être virtuels dans certaines situations, nous devons aussi nous poser la question du statut des *mondes* réels, ce qui correspond au cas où l'agent opératoire est réel lui aussi. Plaçons-nous dans une situation ou un robot totalement isomorphe à un corps humain (humanoïde) évolue dans un environnement réel distant. Un opérateur, totalement coupé de l'environnement du robot, est doté des mêmes systèmes d'interaction que dans le cas d'une immersion dans un monde de synthèse. Chaque mouvement de l'opérateur est ainsi retransmis au robot, et chaque perception du robot est réciproquement retournée à l'opérateur. Cette situation se modélise exactement de la même manière qu'une immersion dans un monde de synthèse, c'est à dire par le modèle M_RIC-*m*. La particularité d'une telle situation est que le «système» de réalité virtuelle n'a en charge que les processus de transmission de l'information entre le robot et l'humain. Les mécanismes de gestion de l'environnement et de son interaction avec le robot sont implicites, dans la mesure où la situation est réelle. Nous sommes ici dans une situation d'immersion, qui, réelle ou pas, rentre bien dans le cadre de la réalité virtuelle. De plus,

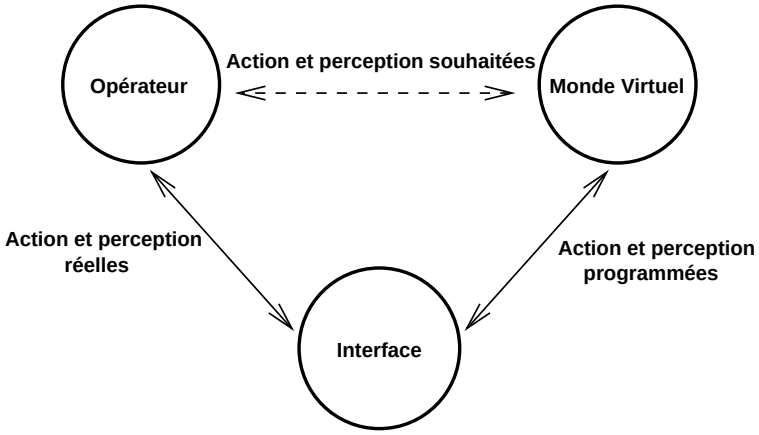

FIG. 5.5: Immersion et Triangle Sémiotique, d'après Fuchs et al. (1999)

fonctionellement parlant, un tel système d'immersion constitue le système le plus simple que l'on puisse imaginer, dans la mesure où il n'agit que comme un «convoyeur» d'information.

5.5.3 Pseudo-immersion

Que se passe t'il maintenant si notre robot humanoïde se trouve en présence de l'opérateur, c'est à dire dans le même environnement que lui ? Se trouve t'on encore dans un cas de réalité virtuelle ? D'après notre vision du concept, la réponse est oui, à condition que l'opérateur ne perçoive pas son environnement par d'autre intermédiaire que celui du robot. En effet, tel que nous l'avons défini précédemment, il y a virtualisation dès que l'un au moins des agents du modèle M_R^*IC est réinstancié. Dans le cas de l'immersion réelle décrite précédemment, à la fois l'agent opératoire et l'agent externe étaient réinstanciés. Cependant, rien ne nous empêche théoriquement de ne réinstancier que l'agent opératoire. C'est bien ce qui se produit dans cette situation, qui est traduite par le modèle M_R^*IC-*o* sur la figure 5.6 ci-contre.

L'isomorphisme total entre l'opérateur et le robot assure que l'opérateur «accepte» le robot comme nouvel agent opératoire. D'autre part, que l'environnement cible soit celui de l'opérateur ou non n'influe pas sur le processus cognitif de virtualisation, dans la mesure où l'opérateur ne le perçoit qu'à travers un système artificiel (casque, gants...) identique à celui qui aurait servi pour un environnement de synthèse.

Si cette situation ne présente vraissemblablement que peu d'intérêt sur un plan opérationnel, il était cependant intéressant de la signaler d'un point de vue théorique. On peut d'autre part s'interroger sur les effets cognitifs et psychologiques qu'auraient une telle situation sur des sujets volontaires. En effet, un aspect paradoxal d'une telle situation «pseudo-immersive» est que l'opérateur, après avoir cognitivement accepté le robot comme nouveau corps, garde pourtant la possibilité de se voir lui-même, mais de l'extérieur, puisque par l'intermédiaire du robot.

5.6 La réalité augmentée : un cas litigieux

Dans le sens le plus usité, la réalité augmentée consiste en l'apport d'objets ou d'information virtuels à une scène réelle, ou l'inverse, comme l'incrustation d'objets réels sur une image virtuelle. Simsarian et Akesson (1997) propose même dans ce cas le terme de «virtualité augmentée». Un périphérique typique de la réalité augmentée est la paire de lunettes transparentes qui permet de percevoir normalement son propre environnement, tout en ayant la possibilité d'obtenir de l'information supplémentaire par affichage sur les verres.

La réalité augmentée constitue pour nous un cas litigieux dans la mesure où elle existe par

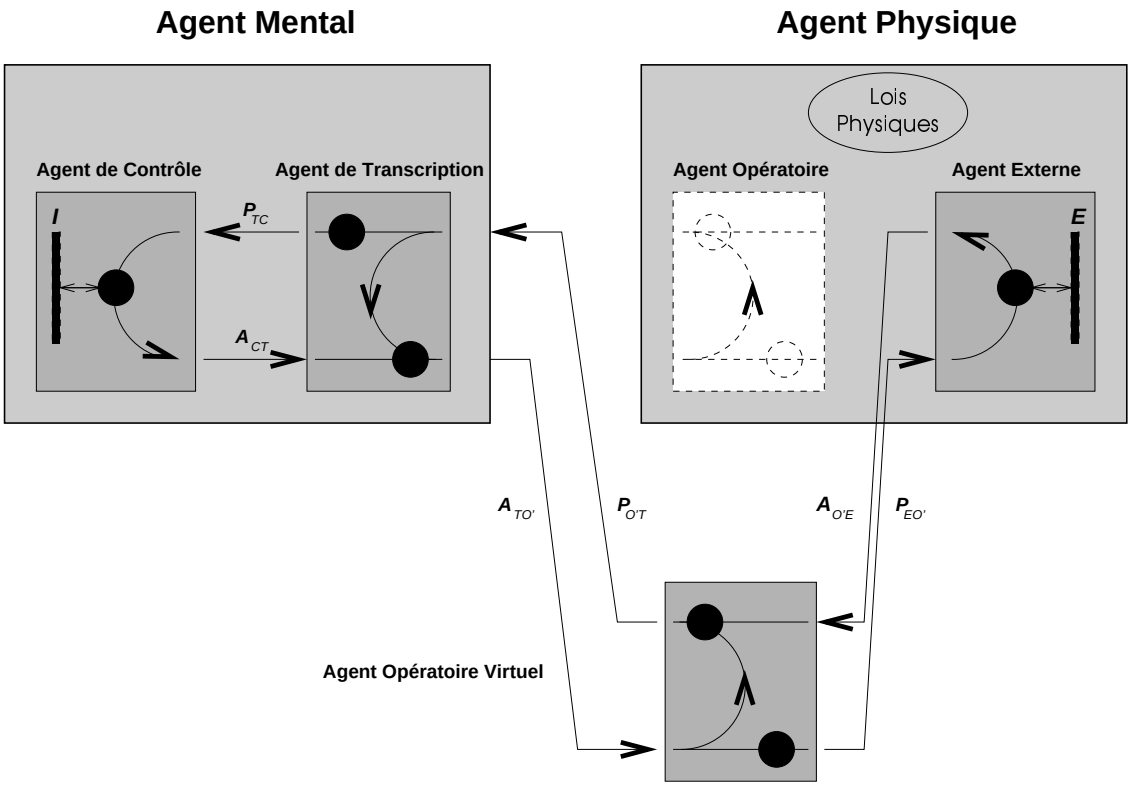

FIG. 5.6: M_RIC-*o* : pseudo-immersion.

l'utilisation de processus de virtualisation, ce dont nous discutons tout au long de cette partie, mais impose à l'opérateur de rester dans son propre environnement, ce qui est contraire à notre vision de la réalité virtuelle. Par conséquent, même si la réalité augmentée ne fait pas strictement partie de la réalité virtuelle, son essence reste la même, c'est à dire qu'elle s'attache à virtualiser les composantes du réel. Il peut donc être intéressant de replacer la réalité augmentée dans le cadre du modèle M_RIC.

5.6.1 Altération de l'agent externe

Par rapport au modèle M_RIC, le travail principal de la réalité augmentée consiste en l'altération de l'agent externe. La réalité augmentée touche en effet essentiellement l'environnement de l'opérateur, et d'une manière qui consiste soit à lui ajouter de nouveaux objets, virtuels mais réalistes (premier type de virtualisation), soit à modifier le comportement de ceux-ci (virtualisation hybride). Ce processus est décrit par le modèle M_RIC-*ae*, sur la figure 5.7 en page suivante (l'aspect polygonal de l'agent externe dénote son altération). L'opérateur est ici conduit à «accepter» pour réelles les altérations de son environnement.

Pour illustrer cette idée, citons les exemples décrits par Rekimoto (1997) : un opérateur est doté d'un système de visualisation permettant de superposer de l'information synthétique sur le champ de vision normal de celui-ci. Ainsi, le visiteur d'un «musée augmenté» peut voir s'afficher une description textuelle d'un tableau au moment où il s'en approche. Dans un deuxième exemple, la superposition de flèches virtuelles en trois dimensions dans les couloirs d'un bâtiment permet au visiteur de trouver son chemin.

Ces exemples ont pour trait commun d'altérer l'environnement réel en lui ajoutant de l'information ou des objets virtuels. Il existe aussi d'autres types d'application consistant plutôt en une

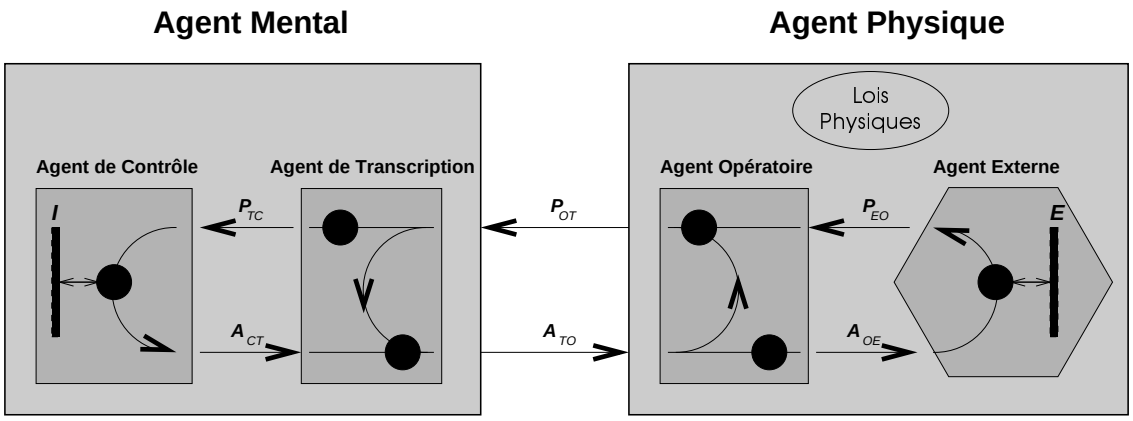

FIG. 5.7: M_RIC-*ae* : Réalité Augmentée par altération de l'agent externe

altération des propriétés d'objets déjà existant dans l'environnement. Citons par exemple certaines applications du domaine médical, comme celle décrite par Mellor (1995), où un chirurgien dispose naturellement d'une image de la tête du patient à opérer, mais sur laquelle un système de réalité augmentée vient ajouter un schéma représentant en grisé la zone de tissu endommagé, et par la même donne l'illusion de transparence des tissus du patient.

5.6.2 Altération de l'agent opératoire

En basant notre approche sur une altération systématique du modèle M_RIC, nous devons naturellement nous interroger sur l'éventualité d'une réalité augmentée par altération de l'agent opératoire, et non plus, ou plus seulement de l'agent externe. Une telle situation serait représentée par le modèle M_RIC-*ao* sur la figure 5.8.

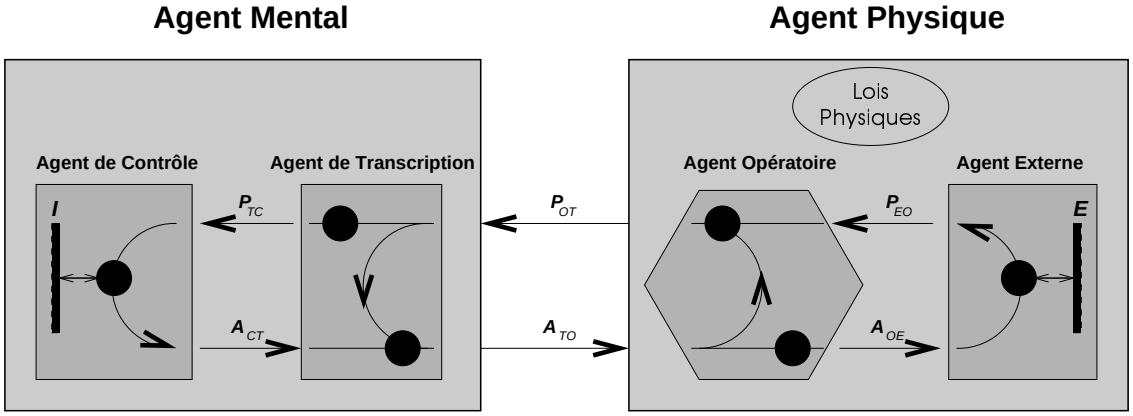

FIG. 5.8: M_RIC-*ao* : Réalité Augmentée par altération de l'agent opératoire

Cognitivement parlant, altérer l'agent opératoire revient à donner à l'opérateur l'illusion de fonctionnalités dont sont corps ne dispose pas, ou dont le fonctionnement est modifié. Il se trouve que de nombreuses applications de cette idée existent déjà depuis longtemps, sans pour autant que leur relation avec la notion de réalité augmentée ait été mise en évidence. Pour rester dans le domaine médical dont certains exemples ont été décrits précédemment, nous pensons que l'utilisation de prothèses artificielles pour compenser le handicap physique est un exemple très

démonstratif d'altération de l'agent opératoire, et donc de réalité augmentée selon la vision décrite par $M_R IC$-*ao* : quand un unijambiste utilise quotidiennement une jambe artificielle, quand un malentendant se sert en permanence d'une prothèse auditive, il y a accoutumance à ces éléments artificiels ; l'agent mental «accepte» cognitivement les altérations de l'agent opératoire. Il s'agit donc bien d'un processus de virtualisation.

☛ Le modèle $M_R IC$ a déjà montré son intérêt dans une approche de la réalité virtuelle aboutissant non seulement à une vision claire du concept, mais également à une classification des applications existantes. Nous voyons maintenant comment l'utilisation systématique de ce modèle nous permet d'établir un lien conceptuel entre certaines applications d'«assistance à l'humain», d'un domaine technologique déjà mûr, et la notion de réalité virtuelle, lien qui jusqu'ici n'avait pas été clairement explicité. Nous continuerons, dans les deux prochaines parties, de mettre en évidence les relations étroites entre la notion d'assistance et celle de réalité virtuelle.

5.7 Réalité virtuelle et télé-opération

Les sections précédentes nous ont permis de décrire les deux types majeurs de réalité virtuelle, et nous ont également permis de démontrer que les environnements réels ne sont pas à exclure du concept, bien au contraire. Une situation d'immersion idéale a d'ailleurs été décrite comme utilisant un «robot» isomorphe à l'opérateur. Nous nous interrogeons donc naturellement sur le statut de la télé-opération par rapport au concept de réalité virtuelle.

5.7.1 Description

On appelle en général «télé-opération» une situation dans laquelle un opérateur pilote à distance une machine dans une zone inaccessible. Une telle situation nécessite un système, mécanique, électrique ou électronique, permettant de transmettre les ordres au robot, ainsi qu'un système symétrique permettant d'acquérir un «retour» de l'environnement de travail. Une situation de télé-opération est donc une situation dans laquelle à la fois action et perception sont pris en compte par un système artificiel de communication. En cela, il n'existe a priori aucun argument contradictoire avec notre vision de la réalité virtuelle.

Certaines des situations sont à la limite de la télé-opération telle que décrite précédemment : la manipulation de produits toxiques derrière une vitre en est un exemple. Il ne s'agit pas à proprement parler de télé-opération puisque la zone de travail est effectivement inaccessible au niveau de l'action, mais que le retour perceptif est immédiat.

Dans la définition du terme «télé-opération», la notion de distance est à prendre au sens large : elle rassemble tout ce qui n'est pas à la portée directe de l'opérateur. Dans une situation de pilotage de robot sur une autre planète, la distance est à la fois spatiale et temporelle à cause des délais de transmission. Dans une situation de télé-opération dans une zone proche mais non visible (impliquant la présence par exemple d'une caméra et d'un moniteur), on dira également que l'environnement de travail est à distance.

5.7.2 Catégories de télé-opération

La télé-opération, selon ses mises en œuvre, entre en fait de manière identique dans nos deux catégories de réalité virtuelle :

- Les situations les plus classiques de télé-opération, celles où le télé-opérateur perçoit par exemple l'environnement au moyen d'un écran de contrôle, rentrent dans le cadre de ce que nous avons «réalité virtuelle de bureau».
- Il est d'autre part intéressant de constater que de nombreux chercheurs expérimentent aujourd'hui l'utilisation de périphériques de réalité virtuelle immersive au niveau de la perception aussi bien que de l'action : les téléopérateurs sont dotés de visiocasques leur per-

mettant de percevoir la scène distante en relief, et commandent des bras articulés au moyen d'exosquelettes à retour d'effort (Tachi et Yasuda, 1994). La télé-opération immersive constitue donc une branche active de la recherche actuelle.

5.7.3 Télé-opération et réalité augmentée

Cependant, l'utilisation de périphériques de réalité virtuelle en télé-opération ne s'arrête pas au seul souci d'immersion de l'opérateur. Les différentes expérimentations menées jusqu'ici ont montré l'utilité d'aller plus loin dans l'utilisation des techniques de réalité virtuelle. Dans le dessein de faciliter le travail du télé-opérateur ou de rendre celui-ci plus performant, l'accent a été mis sur les applications de réalité augmentée, non seulement au niveau de la perception mais aussi au niveau de l'action. Les principes de réalité augmentée pour la télé-opération sont les mêmes que ceux décrits dans la section précédente (ajout d'information ou d'objets virtuels…) avec un intérêt encore accru pour ce qui se rapporte directement aux aspects physiques (comme les problèmes de retour d'effort), puisque l'environnement de travail est réel.

D'autre part, la notion de robotique autonome est liée à un aspect de virtualisation qui n'a jamais été envisagé jusqu'ici. Les degrés de virtualisation que nous avons abordés dans les sections précédentes touchaient essentiellement l'agent externe, et dans une certaine mesure l'agent opératoire. En fournissant une autonomie (partielle) au robot, on lui attribue en fait certaines caractéristiques qui relèvent normalement de l'agent mental : ces caractéristiques commencent par exemple par des déplacements automatiques, qui sont surtout du domaine de l'agent réflexe, puis vont vers une intelligence artificielle (surveillance d'obstacles, résolution de problèmes…) normalement du domaine de l'agent de contrôle.

Par rapport au modèle M_{RIC}, l'autonomie opératoire fait intervenir un agent mental virtuel, qui entre en collaboration avec l'opérateur. Il s'agit là d'un troisième cas de réalité augmentée, ni par la perception, ni par l'action, mais par la réflexion.

Dans notre étude des processus de virtualisation, deux points que l'étude systématique nous impose d'envisager n'ont pas encore été abordés : d'une part, nous ne nous sommes pas interrogés sur la possibilité d'un agent opératoire virtualisé du deuxième type (c'est à dire totalement artificiel, non réaliste), et d'autre part nous n'avons pas envisagé le cas de la réalité virtuelle au troisième degré, c'est à dire avec virtualisation de l'agent de transcription. Le fait que la télé-opération soit un cas de réalité virtuelle selon notre approche nous permet, dans les deux sections suivantes, de combler ce manque.

5.7.4 Transmutation virtuelle

Jusqu'à présent, nous avons décrit deux grandes classes de réalité virtuelle : la «réalité virtuelle de bureau» où l'opérateur interagit avec un environnement virtuel grâce à une interface de contrôle spécialisée, et la réalité virtuelle immersive dans laquelle l'opérateur agit de manière naturelle, comme sur son environnement réel.

La logique de notre modèle nous a précédemment conduit à la notion d'immersion réelle, illustrée dans le domaine de la télé-opération par le contrôle d'un robot totalement isomorphe au corps humain (par exemple, un exosquelette appliqué au bras *de l'opérateur* est utilisé pour piloter un bras *robotique*). Cet isomorphisme, quoi que possible, n'est cependant pas nécessaire pour pouvoir parler d'immersion en environnement réel : si le robot n'est pas isomorphe au corps humain, le problème majeur à résoudre est celui de la facilité de contrôle : comment concevoir des interfaces de commande qui soient les plus ergonomiques et les plus intuitives possibles ? Comment concevoir des systèmes où l'opérateur ait un minimum d'apprentissage à subir ? De manière évidente, la situation idéale serait une situation immersive, où l'opérateur aurait le sentiment d'agir naturellement, bien que le robot ne soit pas humanoïde.

Prenons un exemple : un robot d'assemblage est capable de percer des trous, de visser, de souder des pièces entre elles… Du côté de l'opérateur, en immersion dans l'environnement de travail,

une représentation mélangeant les images réelles de la zone de travail et une vision synthétique de son propre corps lui est fournie. Par la suite, seule la sémantique des actions corporelles de l'opérateur est conservée. Par exemple, l'approche d'un doigt vers une cloison se traduit par une action de perçage. Visuellement, on fournirait à l'opérateur une représentation synthétique de sa main dotée d'une perceuse, superposée à la vision réelle de la zone de perçage. Simultanément, le robot aurait saisi l'outil réel et commencé l'opération. De même, la saisie (virtuelle) d'une vis et sa mise en position sur la cloison se traduit par une action de vissage et une représentation synthétique de l'action correspondante.

☛ Cette idée fait référence à ce que les roboticiens appellent les «problèmes de transposition». On en retrouve par exemple une illustration expérimentale avec la notion de «représentation fonctionnelle intermédiaire» et de «robot caché» développés par Kheddar et al. (1997, 1998). L'idée est de *cacher* la complexité opératoire d'une situation de télé-opération en fournissant une représentation naturelle et donc travestie de la réalité : un exemple est développé, dans lequel un robot télé-opéré peut déplacer des objets. L'opérateur n'a cependant pas une vision réelle de la scène, mais perçoit un environnement virtuel contenant des représentations virtuelles de l'objet et de sa propre main (un retour d'effort est fourni). Le robot n'est pas représenté puisque le but est de simuler une action manuelle. L'opérateur a donc la possibilité d'agir naturellement pour déplacer l'objet sans avoir conscience qu'en réalité, il contrôle un robot.

Par rapport au modèle ***M*$_R$*IC***, et du point de vue cognitif, cette situation se traduit logiquement par une virtualisation du deuxième type de l'agent opératoire. L'opérateur a l'illusion de travailler avec son propres corps ; il est donc dans une situation immersive. Cependant, il s'agit bien de virtualisation du deuxième type, dans la mesure où l'agent opératoire initial (le corps) est remplacé par une instance qui n'a plus rien à voir avec l'instance d'origine (robot non isomorphe au corps humain). Ce type de situation est illustré par la figure 5.9 en page suivante. Il s'agit de réalité virtuelle immersive avec la particularité que l'agent opératoire virtuel est totalement artificiel au lieu d'être réaliste, comme un avatar le serait.

5.8 Détection d'intentions

En poussant à l'extrême notre raisonnement, nous aboutissons à l'idée que la situation idéale d'interaction en réalité virtuelle (que ce soit avec un monde de synthèse, ou en télé-opération) serait celle où l'opérateur n'aurait plus à travailler physiquement. Asymptotiquement, il s'agirait alors d'une communication par transmission directe de pensée.

Concrètement, si l'on veut que l'opérateur garde le contrôle de la situation, le principe de la détection d'intentions est alors une manière satisfaisante de s'approcher du concept : le but est de détecter le plus rapidement possible les intentions opératoires de l'humain, sans que celui-ci ne les exprime explicitement, pour accomplir automatiquement la tâche visée et réduire au maximum le travail physique de l'opérateur.

Par rapport au modèle ***M*$_R$*IC*** et sur le plan cognitif, ce dernier exemple consiste en une virtualisation au troisième degré, c'est à dire où l'agent de transcription lui-même est virtualisé. L'opérateur a ainsi l'impression de n'avoir qu'à exprimer ses intentions (A_{CT}) pour que le travail soit effectué. Cette situation est illustrée par le modèle ***M*$_R$*IC*-*i*** sur la figure 5.10 en page 57. Ce niveau de virtualisation constitue le niveau maximal que l'on puisse atteindre dans le cadre de notre modèle, tout en gardant un opérateur dans la situation.

5.9 Problèmes liés à l'immersion

Comme nous l'avons vu dans le chapitre présentant quelques définitions actuelles de la réalité virtuelle, la composante immersive est souvent considérée comme une part essentielle de la définition (Burdea, 1993, Burdea et Coiffet, 1993, Pimentel et Teixeira, 1994). Cela dit, nous

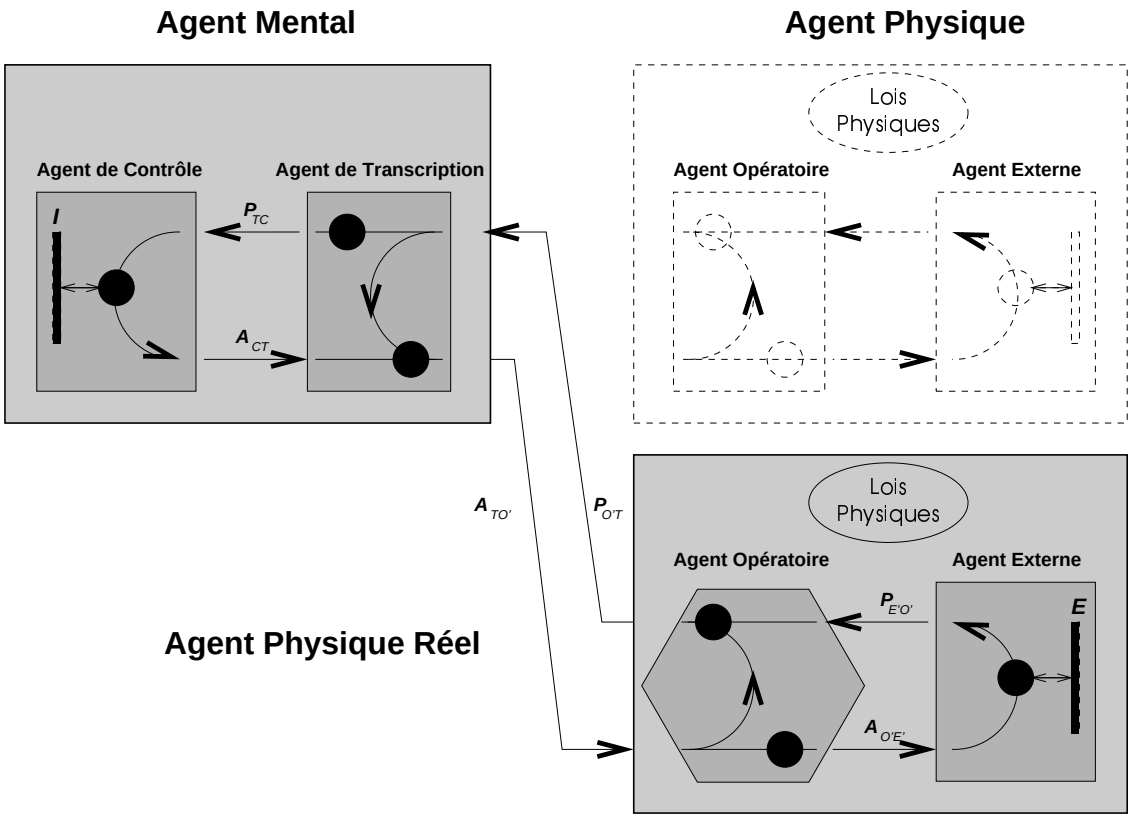

FIG. 5.9: Transmutation virtuelle

devons prendre conscience que la composante immersive dans la réalité virtuelle pose certains problèmes majeurs dans les systèmes actuels, problèmes qui vraisemblablement resteront entiers pendant encore de nombreuses années.

5.9.1 Gestion des interactions

Grâce au modèle ***MIRIC-m***, nous avons constaté qu'un système de réalité virtuelle immersive doit prendre en charge toute l'information circulant au sein de l'agent physique. Cela implique en particulier la gestion de l'interaction entre l'agent opératoire et l'environnement (virtuels), et nécessite par exemple la modélisation physique de l'avatar, de l'environnement et la gestion des collisions entre ceux-ci ($A_{PE} + E \Longrightarrow P_{EP}$). Or les systèmes actuels sont encore incapables de produire une coupure parfaite entre l'opérateur et son environnement réel. Par exemple, dans une situation de sport collectif virtuel telle que celle décrite par Mazeau et Bryche (1995), le joueur en immersion reste en contact avec le sol de son environnement *réel*. Cette situation n'est pas gênante tant que le sol virtuel est plat ou de même nature que le sol réel, comme dans le cas d'un terrain de football virtuel. La mécanique à mettre en œuvre pour simuler un sol bosselé ou un pan de montagne serait par contre extrêmement lourde voire impossible à réaliser. Il y a donc encore à l'heure actuelle un compromis à trouver entre étendue et qualité de la simulation.

5.9.2 Gestion de la proprioception

Toujours grâce au modèle ***MIRIC-m***, nous avons constaté que le chemin proprioceptif ($A_{TO} \Longrightarrow P_{OT}$) au sein même de l'agent opératoire est également censé être géré par le système d'immersion. À son tour, ce chemin d'information est extrêmement délicat, voire impossible à virtualiser à

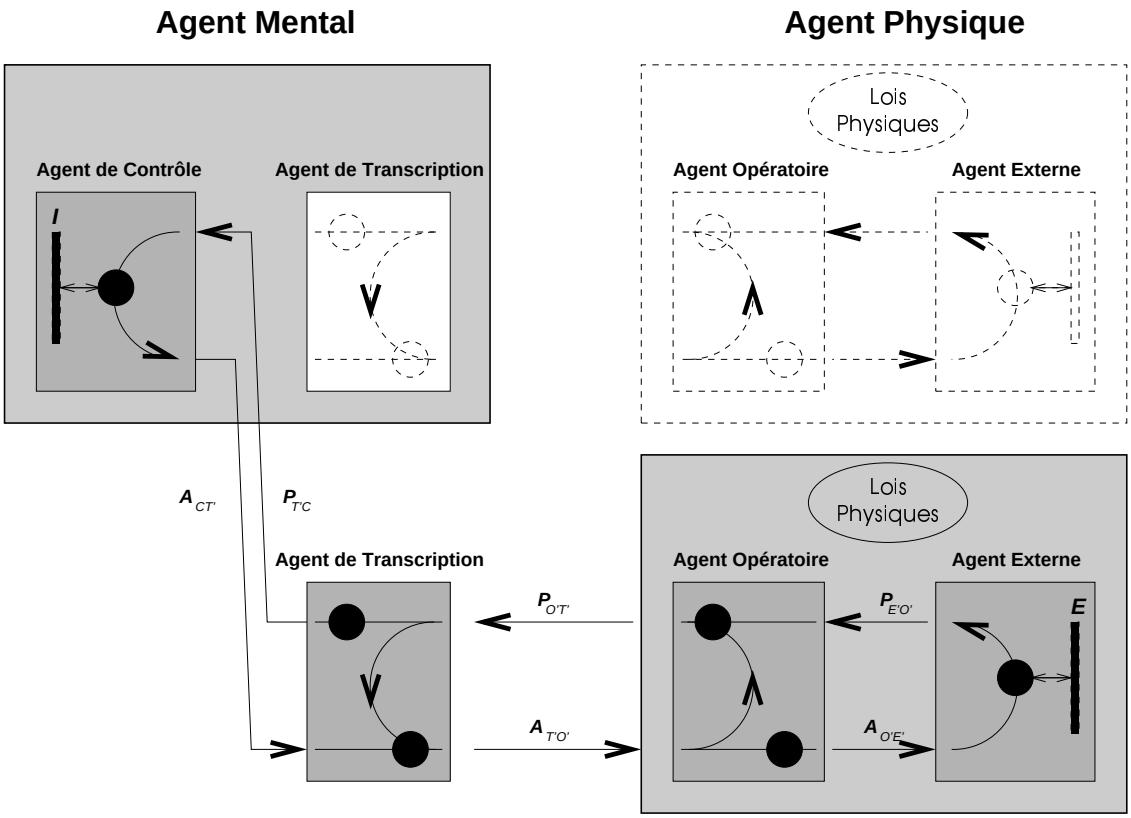

FIG. 5.10: $M_R IC$-*i* : détection d'intentions

l'heure actuelle. Il existe certaines tentatives d'une telle virtualisation : le «Cynaxe», par exemple, est un cinéma 3D dynamique dans lequel les spectateurs sont placés dans une cabine montée sur vérins hydrauliques. En fonction du déplacement dans l'image, la cabine bouge de manière à restituer les sensations (kinestésiques) d'accélération et (proprioceptives) d'orientation dans l'espace. Si le résultat est assez efficace, on perçoit très aisément la lourdeur de la mécanique mise en œuvre pour y parvenir.

☛ Rappelons que l'importance de la proprioception était particulièrement mise en exergue par Quéau (1993) dans sa définition du virtuel. Le modèle $M_R IC$-*m* a la vertu de représenter explicitement cette nécessité d'une virtualisation proprioceptive, et montre que notre point de vue sur le virtuel rejoint complètement le précité sur ce point.

5.9.3 Les «paradoxes» prédictifs

Dans le premier chapitre, nous parlions également de la notion de prédiction (Cf. section 4.6.3 en page 35), comme conséquence de l'apprentissage des interactions avec l'environnement. Cette notion resurgit également ici dans la mesure où une mauvaise qualité d'immersion vient contrecarrer les modèles implicites de prédiction dont nous disposons, et peut produire de véritables états de malaise physique des environnements virtuels* (Kenedy et al., 1997). Pour n'en citer que quelques uns, le cas d'un simulateur de vol statique où l'avion virtuel est en virage génère une sensation de malaise physique dans la mesure où la perception (virtuelle) de virage sur

*Le terme consacré d'origine anglaise est «cybersickness».

l'aile est en contradiction avec l'absence de force centrifuge correspondante dans la proprioception (réelle). Une sensation similaire est produite par des systèmes à casque de visualisation qui mettent trop longtemps à recalculer les images après mouvement de la tête.

5.9.4 Problèmes issus des cas réels

Dans les cas réels tels ceux rentrant dans le cadre de la télé-opération immersive, certains problèmes spécifiques deviennent plus apparents. Même si comme nous l'avons vu, la distinction entre environnements réels et synthétiques n'est pas la distinction principale à établir sur le plan cognitif, il n'en reste pas moins que le comportement de l'opérateur sera incontestablement affecté par la conscience de travailler dans du «vrai». Dans la réalité, les effets sont irrémédiables : faire s'écraser un avion réel n'est vraiment pas la même chose que faire s'écraser un simulateur de vol.

En conséquence, le critère de fonctionalité de l'interface homme-machine revêt un aspect tout particulier dès lors que l'on travaille en environnement réel : il devient crucial que la *fonction* recherchée soit correctement mise en œuvre, même si la qualité de l'immersion est moindre. Cette problématique est effectivement largement débattue à l'heure actuelle :

- Barfield et al. (1995) montrent que la technologie de l'immersion au niveau perceptif est encore très en retard sur les développements que l'on peut imaginer, et qu'il manque de nombreuses fonctionalités pour recouvrir l'ensemble du domaine perceptif humain, et donc arriver à une immersion optimale.
- Tachi et Yasuda (1994) évaluent l'efficacité d'un système de télé-opération de bras manipulateur, et montrent que l'efficacité optimale est obtenue pour une combinaison vision-binoculaire et mouvements naturels du bras et de la tête, combinaison qui constitue en effet une situation immersive. Cependant, et même si cet aspect n'est pas explicitement exposé dans leur article, le système étudié n'a pas pour vocation d'être un système d'immersion totale. L'opérateur, par exemple, reste assis pendant le déroulement de l'opération, et n'a donc pas la possibilité de se déplacer dans l'environnement de travail. Cette situation illustre le fait que les concepteurs de systèmes de télé-opération immersive ne cherchent pas nécessairement l'immersion totale, mais seulement les composantes essentielles à l'opération à effectuer.
- Pour relativiser encore plus l'utilité de l'immersion dans la télé-opération, Schloerb (1995) propose de découpler les aspects cognitifs (subjectifs) et objectifs de la téléprésence. Il exprime la «téléprésence subjective» comme la «sensation de présence physique dans l'environnement», ce qui correspond à notre vision cognitive de l'immersion, et la «téléprésence objective» comme le degré de causalité entre les actions de l'opérateur et le résultat sur l'environnement, ou la probabilité de succès d'une tâche spécifique. Il y a donc cette fois un découplage explicite entre le degré d'interaction opérateur / environnement et le degré d'immersion, au sens où nous l'entendons.
- Des idées similaires sont développées dans un article récent (Fuchs et al., 1999), dont le but est d'établir une méthode systématique de conception et d'évaluation pour des systèmes de réalité virtuelle : «L'erreur principale vient du fait que les concepteurs cherchent le plus haut degré de réalisme (limité en pratique par les aspects technologiques et économiques) sans se soucier de savoir exactement de quelles composantes réelles l'application a besoin»*. Les auteurs distinguent trois niveaux d'immersion : l'immersion «sensori-motrice» mise en œuvre par l'interface du système, l'immersion «mentale» correspondant à la «téléprésence subjective» de Schloerb et l'immersion «fonctionelle» correspondant à sa «téléprésence objective».
- L'apogée dans la nuance est obtenue avec Ellis (1996) qui remet explicitement en cause l'utilité de l'immersion dans certaines situations. Par exemple, il décrit une situation où un pilote obtient artificiellement une vision du trafic aérien avoisinant, et montre qu'en évitant

*Citation traduite de l'anglais.

d'utiliser un retour visuel de type immersif, le pilote peut profiter d'une vue d'ensemble du trafic en question bien plus confortable qu'un point de vue égocentrique.

Au travers de tous ces problèmes, nous comprenons que dès l'instant où l'agent opératoire doit explicitement être virtualisé, ce qui est le cas des «mondes virtuels», c'est à dire de la réalité virtuelle immersive, nous entrons dans une problématique relativement inextricable dans la mesure où le handicap technologique freine l'avancée conceptuelle et bride en quelque sorte le potentiel d'imagination que nous pourrions exploiter afin de bâtir de nouvelles applications. Mais bien plus que le seul handicap technologique, c'est l'utilité même de la sensation immersive qui est remise en question dans bien des situations. Ces considérations sont un argument de plus en faveur de notre distinction principale des cas de réalité virtuelle, selon qu'ils sont «de bureau» ou immersifs.

Terminons cette section sur une note futuriste. Puisqu'il est extrêmement difficile, voire impossible de virtualiser totalement l'agent opératoire, la seule véritable solution pour une immersion efficace consisterait à déconnecter *physiquement* l'opérateur de son propre corps, et agir directement sur le système nerveux. eXistenZ, le film de David Cronnenberg sorti en Avril 1999, traite justement de cette idée : l'héroïne, une informaticienne de haut niveau, vient d'inventer un jeu vidéo qui se branche directement sur la moelle épinière...

D'un point de vue technologique, le principe des «biocapteurs» va dans ce sens : ce sont des capteurs non-intrusifs détectant de faibles différences de potentiels à la surface de la peau, comme celles qui se produisent suite à une commande musculaire. De tels capteurs ont par exemple été utilisés dans l'expérimentation de prothèses : une main artificielle était connectée, via un ensemble de réeaux neuromimétiques, à des biocapteurs situés sur le bras du patient. En déclenchant volontairement des ordres d'ouverture et de fermeture de la main amputée (comme si celle-ci était encore présente), les artefacts de l'influx nerveux correspondant étaient captés, et l'on pouvait reproduire le mouvement sur la main artificielle.

5.10 Réalité virtuelle et langage

Pour terminer ce chapitre sur la réalité virtuelle, nous voudrions ouvrir le débat en faisant à nouveau référence au langage. Ce lien a été esquissé auparavant lorsque nous avons illustré le fait que les situations réelles peuvent être virtuelles d'un point de vue cognitif, et cela au moyen du triangle sémiotique.

Reprenons notre modèle initial de réalité, le modèle $\boldsymbol{M_R IC}$. On constate aisément que ce modèle fait montre d'une parfaite symétrie centrale : les agents de contrôle et externe, puis de transcription et opératoire jouent respectivement des rôles symétriques. Partant de cette constatation, nous pouvons imaginer une situation dans laquelle l'agent physique est remplacé par un deuxième agent mental, situation qui se modélise d'une manière structurellement identique, comme illustré par le modèle $\boldsymbol{M_R IC}$-*l* sur la figure 5.11 en page suivante.

Que signifierait une telle situation d'un point de vue cognitif ? Deux agents mentaux A et B en état d'interaction directe, non physique. Autrement dit, deux «esprits» capables d'interagir, c'est à dire de modifier réciproquement I_A et I_B sans pour cela que les relations entre agent opératoire et agent externe soient nécessaires. Le cas d'une conversation en langage naturel entre deux individus A et B est bien décrit par une telle modélisation. Deux individus, par le langage, sont en effet capables d'échanger de l'information symbolique sans qu'il y ait une quelconque interaction physique entre eux (d'un point de vue cognitif, bien entendu, ce qui signifie que les processus *mécaniques* de mise en œuvre du langage de nous concernent pas).

Cette extrapolation du modèle $\boldsymbol{M_R IC}$ nous permet donc de considérer que par le langage, nous créons non seulement des réalités virtuelles (le *sens* que porte notre discours), mais encore que nous mettons ces réalités virtuelles à disposition des autres. Deux agents mentaux en interaction langagière sont des *mondes virtuels* l'un pour l'autre.

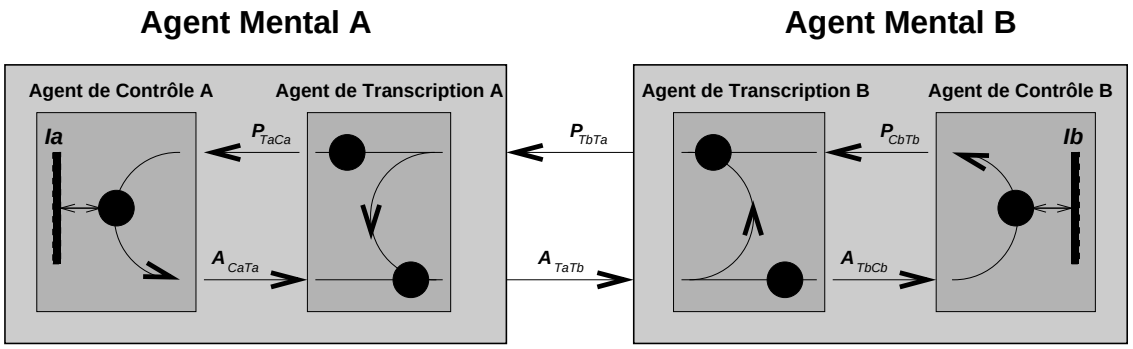

FIG. 5.11: M_R^IC-*l* : réalité virtuelle et langage

Chapitre 6

Conclusion

«Reality is that which, when you stop believing in it, doesn't go away»
Philipp K. Dick

Dans cette première partie, nous avons tenté d'éclaircir le sens que l'on peut attribuer à la notion de réalité virtuelle. Un panorama des propositions existantes a montré les points suivants :

- Les définitions proposées présentent chacune des aspects spécialement mis en exergue, que l'on ne retrouve pas nécessairement dans toutes les propositions, ou à des niveaux différents d'importance. L'ensemble des propositions étudiées s'accorde cependant sur deux points : l'aspect infographique des mondes virtuels (images de synthèse en trois dimensions) et l'importance de l'interaction. Dans une moindre mesure, l'aspect immersif est également considéré comme un facteur important.
- La terminologie utilisée est opulente, et souvent propre aux auteurs qui l'utilisent. Par ailleurs, cette terminologie ne fait pas toujours l'objet d'une définition ou d'un cadre d'utilisation précis.
- Au sein des applications de réalité virtuelle, une distinction est souvent faite entre les applications à vocation réaliste (simulation du réel) ou totalement artificielle.
- Enfin, la prise en compte des aspects cognitifs intervient plus ou moins explicitement dans quelques unes des propositions, mais ne constitue l'axe principal d'aucune des approches citées.

Face à cet état de l'art, nous avons choisi le point de vue cognitif comme axe principal d'étude. Cette optique nous a d'abord conduit à proposer un modèle de description du réel (ou plus précisément de description des interactions entre un humain et son environnement), le modèle ***M$_R$IC***. Nous avons ensuite montré qu'une tentative de définition cognitive de la réalité virtuelle était vaine, mais que la véritable question était de définir les «processus de virtualisation» qui font que l'humain accède au virtuel à partir du réel. Ces processus ont été définis en terme de réinstanciation des agents du modèle initial. L'apport de ce modèle réside principalement dans la capacité que nous avons eue de rassembler les propositions existantes et leurs caractéristiques en une seule et unique vision : l'approche cognitive. Grâce à cette approche, les différents aspects caractéristiques du virtuel ont pu être mis en évidence directement sur le modèle, et la terminologie actuelle a pu être précisée.

Par delà le bénéfice unificateur de notre démarche, l'approche cognitive développée dans cette première partie nous a permis, par l'analyse théorique, de retrouver deux conclusions jusqu'ici mises en évidence par des moyens plus empiriques :

- Alors que les deux grandes classes d'applications de réalité virtuelle majoritairement distinguées sont celles qui simulent des environnements réalistes d'une part, et celles qui produisent des réalités artificielles d'autre part, il est plus pertinent d'un point de vue cognitif de distinguer en premier lieu les applications «de bureau» ou l'interaction se fait par des périphériques de contrôle nécessitant un apprentissage minimum, et les applications «immersives» dans lesquelles l'interaction se fait d'une manière naturelle à l'opérateur. Ce point de vue semble d'autant plus justifié que le bénéfice véritable de l'immersion dans les situations réelles est actuellement quelque peu nuancé.
- L'utilisation du modèle ***MRIC*** a également montré que d'un point de vue cognitif, il n'existe pas de différence conceptuelle entre les environnements réels et les environnements de synthèse. En effet, même si l'opérateur réagira différemment dans les deux cas (on ne pilote pas un simulateur comme on pilote un véritable avion), les processus de virtualisation (c'est à dire de substitution d'un ou plusieurs agents aux agents réels) sont les mêmes.

Une théorie de l'Assistance : Sémantique et Localisation

1 Introduction 65
2 Le Modèle L_iSA 67
2.1 Côté opérateur : l'interface 67
2.2 Côté environnement : le manipulateur 68
2.3 Jonction : le système 69
2.4 Modèle complet 70
3 Étude de Cas 73
3.1 Génération de perception 73
3.2 Altération de la perception 75
3.3 Transmodalisation de la perception 76
3.4 Reconstitution de perception 78
3.5 Filtrage de l'action 80
3.6 Commande paramétrique 81
3.7 Exécution virtuelle 83
3.8 Résumé 84
4 Symétrisation des Exemples 87
4.1 Méthodologie 87
4.2 Génération d'action 88
4.3 Altération de l'action 89
4.4 Transmodalisation de l'action 91
4.5 Reconstitution d'action 92
4.6 Exécution paramétrique 94
4.7 Filtrage de la perception 95
4.8 Commande virtuelle 96
4.9 Résultats 98
5 Niveau Conceptuel de l'Assistance 101
5.1 Analogies entre M_RIC et L_iSA 101
5.2 Télé-Opération Immersive 102
5.3 Télé-Opération «brute» 103
5.4 Télé-Opération semi-autonome 105
5.5 Télé-Opération et Détection d'Intentions 106
5.6 Résumé 107
6 Conclusion 111

Chapitre 1

Introduction

«The only "intuitive" interface is the nipple. After that, it's all learned.»
Bruce Ediger

Outre la possibilité de définir le virtuel d'un point de vue cognitif, la première partie de ce document nous a permis de mettre en évidence l'une des problématiques majeures du virtuel au niveau applicatif : alors que la réalité virtuelle immersive constitue l'un des principaux centres d'intérêt actuels, elle doit cependant faire face à un important retard technologique : le manque de fiabilité ou d'utilisabilité, la lourdeur des interfaces immersives gênent l'opérateur (particulièrement le télé-opérateur) et nuisent à son efficacité. Une première raison pour s'intéresser à la notion d'*assistance* à l'opérateur s'impose donc d'elle-même : l'assistance comme palliatif temporaire au handicap technologique.

Parallèlement à cette problématique, force est de constater que l'efficacité d'un opérateur ne dépend pas seulement de la qualité de l'interface homme-machine dont il dispose, mais aussi du type même d'opération dont il a la charge : en chirurgie assistée par ordinateur, par exemple, il peut s'avérer que telle ou telle opération sur les tissus du patient requière une précision de loin supérieure à celle qu'une opération manuelle pourrait offrir. Si l'opérateur connaît mieux que personne l'opération à effectuer, la machine, par contre, pourrait être mieux placée que quiconque pour l'exécuter. Une deuxième raison de s'intéresser à la notion d'assistance à l'opérateur se détache donc ici : l'assistance comme substitutif aux limites des capacités de l'opérateur.

Il existe enfin une spécificité toute particulière aux environnements réels qui donne à la notion d'assistance une dimension incontournable : le cas particulier de la télé-opération est d'autant plus critique que certaines opérations en environnement réel sont en général irréversibles, et ne laissent aucun droit à l'erreur. On imagine sans peine les conséquences que pourraient avoir une mauvaise décision en télé-opération chirurgicale, ou dans une zone radioactive. Dans de telles situations, l'opérateur ne doit pas être considéré seulement comme un «décideur» dont on est encore loin de savoir reproduire l'intelligence ou la connaissance, mais il est nécessaire de prendre en compte les aspects émotionnels qui peuvent, dans des situations critiques, altérer son jugement. L'assistance à l'opérateur apparaît donc également comme un garde-fou.

Ces trois points montrent d'une part l'intérêt de se pencher sur la notion d'assistance à l'opérateur, et justifient d'autre part le fait que cette activité soit déjà largement développée par les roboticiens ou les concepteurs de systèmes de réalité augmentée. Cela dit, il n'existe à l'heure actuelle aucune théorie mettant explicitement en relation les notions d'assistance à l'opérateur, et de réalité virtuelle : la notion d'assistance est en général mise à profit au cas par cas, parfois grâce à des outils, des techniques ou des concepts de réalité virtuelle, sans nécessairement qu'un processus d'abstraction y donne suite. Cette deuxième partie tend à combler ce manque.

Dans un premier temps, nous proposons l'élaboration d'un modèle permettant de décrire les situations de télé-opération, réalité augmentée ou réalité virtuelle. Bien qu'il soit très proche et largement inspiré du modèle ***MRIC***, le modèle proposé ici ne se focalise pas sur les aspects cognitifs de l'interaction, mais sur les aspects **fonctionnels** de celle-ci. Il décrit donc plutôt les composantes matérielles impliquées dans les situations considérées.

Dans un deuxième temps, nous étudions un certain nombre de cas d'assistance à l'opérateur représentatifs des grandes classes d'applications actuelles. Ces exemples sont étudiés en référence à notre modèle selon deux axes : d'une part en termes de *localisation* des processus de traitement de l'information qu'ils mettent en œuvre, et d'autre part en termes de *sémantique* du traitement de l'information qu'ils opèrent.

Cette étude nous permet ensuite, par un procédé systématique de symétrisation, d'extrapoler les exemples étudiés à de nouveaux cas intéressants, et de montrer en quoi certaines notions comme celle de réalité augmentée on été jusqu'ici envisagées de manière incomplète.

Finalement, nous étudions l'influence de la présence de fonctionalités d'assistance à l'opérateur dans une situation donnée sur le niveau conceptuel de l'information véhiculée au travers du modèle, comparativement à la même situation non assistée.

Chapitre 2

Le Modèle LiSA

Afin de pouvoir convenablement définir ce que nous entendons par «assistance à l'opérateur» dans un contexte de télé-opération en réalité virtuelle, il est important de décrire en premier lieu le cadre précis des situations considérées. L'objet de ce chapitre est donc de proposer un modèle générique de situations dans lesquelles une assistance à l'opérateur est susceptible d'intervenir. Ce modèle, appelé ***LiSA*** (Acronyme de «**L**ocal**i**sation et **S**émantique de l'**A**ssistance»), s'inspire du modèle ***MRIC*** développé dans la partie précédente, et insiste cette fois non pas sur les aspects cognitifs mais sur les aspects fonctionnels du système considéré.

Les deux composantes principales d'une situation de télé-opération en réalité virtuelle sont bien entendu l'**opérateur** et l'**environnement** sur lequel il travaille. Ces deux éléments ont été définis précédemment, nous ne reviendrons donc pas dessus. Par contre, d'un point de vue fonctionnel, nous devons également faire apparaître entre les deux, le «système» artificiel qui permet leur mise en communication.

2.1 Côté opérateur : l'interface

La vision du virtuel que nous avons développée dans la partie précédente nous impose une coupure totale entre l'opérateur et l'environnement de travail : soit cet environnement est synthétique, auquel cas la coupure est implicite, soit cet environnement est réel, auquel cas l'opérateur ne doit avoir aucun contact direct avec lui. Dans les deux cas, l'opérateur n'interagit qu'indirectement avec l'environnement cible, par l'intermédiaire d'une *interface*.

- L'interface doit tout d'abord permettre à l'opérateur de percevoir l'environnement. Elle contient donc un ensemble de périphériques matériels prévus à cet effet : écran, casque de visualisation, haut parleurs, dispositifs de retour d'effort, et plus généralement, tout dispositif permettant de stimuler les sens humains.
- L'interface doit également permettre à l'opérateur de produire des actions sur l'environnement. Elle contient donc également un ensemble de périphériques matériels prévus à cet effet : joystick, trackball, gants de données, éventuellement un dispositif de reconnaissance vocale, et plus généralement, tout dispositif permettant de recevoir de l'information en provenance d'un humain.
- Ce que nous entendons par «interface» ne se résume cependant pas aux seuls périphériques matériels, mais comprend également les processus de traitement de l'information élémentaires associés. Par exemple, le traitement informatique consistant à générer une image de

synthèse à partir d'un modèle tridimensionnel est considéré comme faisant partie de l'interface. De même pour le traitement consistant à fabriquer une information de déplacement à partir de la manipulation d'un joystick. Bien que ces traitements puissent utiliser une information d'un niveau conceptuel non nul, comme la position courante du point de vue, ils restent cependant «élémentaires» dans le sens où ils peuvent être complètement câblés et ne requièrent aucune intelligence particulière.

Les éléments précédents peuvent être figés de la manière suivante :

Vocabulaire: Restitution

*On qualifiera de **restitution** les processus consistant à mettre une information à disposition de l'opérateur, selon ses modalités perceptives propres.* ●

Vocabulaire: Commande

*On qualifiera de **commande** les processus consistant à saisir une information en provenance de l'opérateur, selon ses modalités actionnelles propres.* ●

Définition 12: Interface

*On appelle **interface** l'ensemble des processus de restitution et de commande, ainsi que des périphériques matériels du côté de l'opérateur qui en sont les supports.* ■

☛ Remarquons que le caractère «élémentaire» des processus de restitution et de commande que nous venons de décrire ne présage en rien du niveau conceptuel de l'information traitée. Par exemple, l'interface peut manipuler une information de bas niveau comme le mouvement d'un joystick ou un retour d'effort particulier. Cependant, une interface de synthèse ou d'acquisition vocale manipulera de l'information en langage naturel, par conséquent d'un important niveau conceptuel.

2.2 Côté environnement : le manipulateur

Dans la mesure où l'opérateur n'est pas en contact direct avec l'environnement, les interactions physiques avec celui-ci se font indirectement et artificiellement, par l'intermédiaire d'un *manipulateur*.

- ➤ Le manipulateur doit tout d'abord être capable de saisir l'information issue de l'environnement (ou de lui-même), afin que celle-ci puisse être portée à la connaissance de l'opérateur. Il contient donc un ensemble de périphériques matériels prévus à cet effet : caméras, microphones, capteurs et détecteurs de toutes sortes, réceptifs à toute modalité utile selon laquelle l'environnement est susceptible de générer de l'information.
- ➤ Le manipulateur doit également permettre de produire effectivement les actions qui ont été commandées par l'opérateur. Il contient donc un ensemble de périphériques matériels prévus à cet effet : roues motrices, bras articulés, émetteurs de toutes sortes susceptibles de modifier l'état de l'environnement.
- ➤ Ce que nous entendons par «manipulateur» ne se résume cependant pas aux seuls périphériques matériels, mais comprend également les processus de traitement de l'information

élémentaires associés. Par exemple, le traitement informatique consistant à calculer la vitesse respective de chaque roue motrice afin d'obtenir un rayon de courbure particulier est considéré comme faisant partie du manipulateur. De même pour le traitement consistant à numériser une image acquise grâce à une caméra. Bien que ces traitements puissent utiliser une information d'un niveau conceptuel non nul, comme le rayon de courbure à obtenir, ils restent cependant «élémentaires» dans la mesure où ils peuvent être complètement câblés et ne requièrent aucune intelligence particulière.

Les éléments précédents peuvent être figés de la manière suivante :

Vocabulaire: Acquisition

*On qualifiera d'**acquisition** les processus consistant à collecter de l'information issue de l'environnement ou du manipulateur lui-même, selon toute modalité utile, et ceci en vue d'une éventuelle restitution.* ●

Vocabulaire: Exécution

*On qualifiera d'**exécution** les processus consistant à produire de l'information vers l'environnement ou le manipulateur lui-même, selon toute modalité utile, cette information provenant éventuellement d'une «commande», au sens précisé plus haut.* ●

Définition 13: Manipulateur

*On appelle **manipulateur** l'ensemble des processus d'acquisition et d'exécution, ainsi que des périphériques matériels, situés du côté de l'environnement, qui en sont les supports.* ■

☛ Notons qu'il est intéressant de considérer que certains processus d'exécution, comme le calcul des vitesses respectives de chaque moteur, font partie du manipulateur lui-même : prenons l'exemple d'un bras manipulateur articulé à la manière d'un bras humain. Si l'opérateur a la possibilité de commander la vitesse globale du bras, une telle vitesse, constante, implique une modification permanente des vitesses de rotation de chaque moteur d'articulation. En localisant au manipulateur les processus consistant à traduire la vitesse globale en vitesses moteurs, on limite implicitement le débit nécessaire à une éventuelle communication réseau.

2.3 Jonction : le système

Après avoir décrit les composantes intervenant du côté de l'opérateur d'une part et du côté de l'environnement d'autre part, nous devons finalement ajouter un dernier élément permettant de faire la jonction : le *système*.

- ➤ D'un point de vue matériel, le système comprend tous les dispositifs permettant de relier l'interface au manipulateur. Ceci peut par exemple faire intervenir des outils de communication à distance (réseau, radio…) dans le cas d'une situation de télé-opération.
- ➤ Ce que nous entendons par «système» ne se résume cependant pas aux seuls périphériques matériels : nous y incluons également l'ensemble des processus de traitement de l'information non inclus dans l'interface ou le manipulateur. Ces processus peuvent être inexistants, ou se résumer à de simples traitements destinés au transport de l'information (compression, encodage…). Cependant, dans l'optique d'une assistance «intelligente» à l'opérateur,

où des processus de traitement de l'information non élémentaires doivent intervenir, ceux-ci seront également considérés comme faisant partie du système.

Les éléments précédents peuvent être figés de la manière suivante :

Définition 14: Système

*On appelle **système** l'ensemble des processus de traitement de l'information non intégrés dans l'interface ou le manipulateur, ainsi que du matériel qui en est le support.* ■

2.4 Modèle complet

Les composantes décrites précédemment sont rassemblées pour former le modèle définitif, comme illustré sur la figure 2.1. Les liens entre ce nouveau modèle et le modèle M_RIC seront développés dans le chapitre 5 en page 101. Le reste de cette section revient en détail sur certains aspects du modèle élaboré.

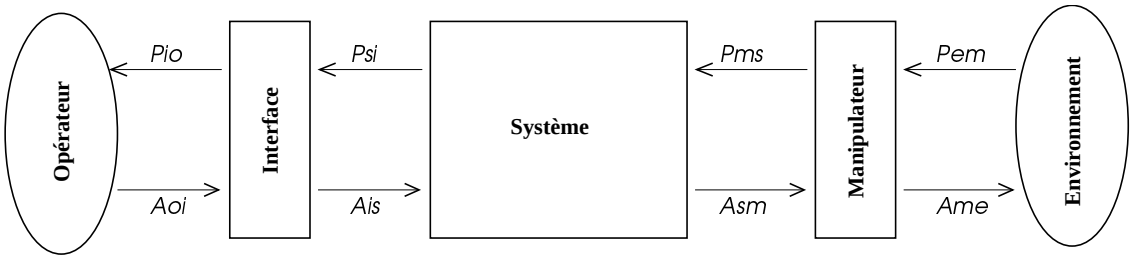

FIG. 2.1: Modèle L_iSA complet

☛ Nous retrouvons ici le modèle communément utilisé pour décrire les situations de télé-opération assistée par ordinateur. L'utilisation que nous en faisons est cependant plus large, dans la mesure où en application à la réalité virtuelle, ce modèle permet également de représenter des situations d'interaction avec des mondes de synthèse, dans lesquels l'environnement est virtuel et le «manipulateur» correspond à un avatar de l'opérateur.

2.4.1 Chemins d'information

Le modèle proposé en figure 2.1 fait intervenir huit *chemins d'information*. Ces chemins sont notés A ou P selon leur sens (action ou perception), et indexées par leur source et destination. Par exemple, l'information transitant de l'interface vers le système est notée A_{is}, et celle transitant de l'environnement vers le manipulateur est notée P_{em}.

Afin d'illustrer chacun des chemins d'information mis en évidence sur le modèle, prenons le cas d'un bras manipulateur capable de saisir des objets. L'opérateur dispose, comme interface, d'un écran lui restituant une vue de l'environnement de travail et d'un joystick permettant de déplacer le bras. Le manipulateur est donc constitué du bras lui-même ainsi que de la caméra filmant la scène. Dans ces circonstances, les huit chemins d'information peuvent être décrits par les exemples qui suivent :

A_{oi} L'opérateur entame un mouvement physique sur le joystick afin de commander une séquence de mouvement.

$\mathbf{A_{is}}$ La nouvelle position du joystick est traduite en une vitesse de déplacement à appliquer au bras manipulateur.

$\mathbf{A_{sm}}$ Le système se charge de transmettre cette information au manipulateur lui-même.

$\mathbf{A_{me}}$ Le bras manipulateur actionne ses moteurs, produisant ainsi le mouvement requis par l'opérateur.

$\mathbf{P_{em}}$ La caméra reçoit une information lumineuse émanant des objets présent dans l'environnement.

$\mathbf{P_{ms}}$ Cette information lumineuse est traduite en un signal vidéo représentant l'image acquise.

$\mathbf{P_{si}}$ Le système se charge de transmettre cette information à l'interface elle-même.

$\mathbf{P_{io}}$ L'écran restitue l'image à l'opérateur en générant une information lumineuse analogue à celle produite par l'environnement.

2.4.2 Traitement de l'information

Les sections précédentes ont fait apparaître six processus élémentaires de traitement de l'information. Ces processus peuvent maintenant être localisés sur le modèle ***L_iSA*** complet :

$\mathbf{A_{oi}} \Longrightarrow \mathbf{A_{is}}$ Commande
$\mathbf{A_{sm}} \Longrightarrow \mathbf{A_{me}}$ Exécution
$\mathbf{P_{em}} \Longrightarrow \mathbf{P_{ms}}$ Acquisition
$\mathbf{P_{si}} \Longrightarrow \mathbf{P_{io}}$ Restitution
$\mathbf{A_{is}} \Longrightarrow \mathbf{A_{sm}}$ Transport
$\mathbf{P_{ms}} \Longrightarrow \mathbf{P_{si}}$ Transport

Hormis ces processus élémentaires, il peut *a priori* exister d'autres traitements impliquant une combinatoire des chemins d'information analogue à celle présentée sur le modèle ***M_RIC***. Cependant, aucun de ces processus de traitement de l'information ne figure explicitement dans le modèle ***L_iSA***. L'étude de la notion d'assistance que nous menons dans cette partie consiste justement à les faire apparaître, quand ceux-ci présentent un intérêt particulier.

Chapitre 3

Étude de Cas

Que ce soit en réalité augmentée ou en télé-opération (Azuma, 1997), l'assistance à l'opérateur est une notion qui a déjà fait son chemin dans les esprits. Dans ce chapitre, nous proposons d'examiner comment les concepts issus de la réalité virtuelle sont mis en œuvre dans les applications actuelles pour assister l'opérateur dans sa tâche. Cette étude est menée au travers de quelques cas d'assistance déjà opérationnels et représentatifs de l'activité actuelle dans ce domaine.

Chaque cas d'assistance est étudié, en référence au modèle ***LiSA***, selon deux axes : un axe de *localisation* tout d'abord, c'est à dire à quel(s) endroit(s) du modèle les processus d'assistance considérés interviennent, puis selon un axe de *sémantique*, c'est à dire quel type de traitement est opéré sur l'information concernée.

3.1 Génération de perception

Le principe le plus répandu dans les applications de réalité augmentée est vraisemblablement celui consistant à mélanger des perceptions réelles et virtuelles, en particulier au niveau de l'image. Voici quelques exemples illustrant cette fonctionnalité.

3.1.1 Exemples

- Rekimoto (1995), Nagao et Rekimoto (1995) décrivent un système permettant de superposer des annotations textuelles au champ visuel normal : ce système est basé sur un écran de type PalmTop, au moyen duquel l'utilisateur perçoit son environnement direct, soit par transparence, soit par une image vidéo capturée à proximité de son regard. Une information de nature textuelle peut alors être affichée sur l'écran, en superposition à l'image réelle. Une application de ce système à l'acquisition d'information dans une bibliothèque est proposée : l'utilisateur positionne son écran vers un rayon ou un livre particulier. Si par exemple le point de vue se focalise sur un livre particulier, le système ajoute automatiquement de l'information sur le livre en question (nom de l'auteur, contenu, éditeur. . .).
- Fertey et al. (1995) pour une application de simulation en réalité virtuelle et Rekimoto (1997) pour un cas de réalité augmentée proposent des systèmes d'aide à la navigation par superposition d'information ou d'objets virtuels sur la scène d'origine. Par exemple, la navigation simulée dans une centrale nucléaire, ou réelle dans un immeuble de bureaux est assistée en superposant au moment opportun des flèches virtuelles indiquant la direction à prendre

(par exemple vers une salle de réunion). L'opérateur a ainsi l'illusion que ces flèches sont imprimées sur le sol, ou que ce sont des objets flottants apparaissant quand ils peuvent lui être utiles.

Le concept d'annotation textuelle se retrouve à l'heure actuelle dans de nombreuses autres applications de réalité augmentée, d'autant que les périphériques d'augmentation de la réalité commencent à être «portables». Ronald Azuma et Steve Freiner ont ainsi présenté en 1999 les premiers dispositifs de réalité augmentée portables, permettant par exemple de visiter une ville avec un visiocasque et de voir apparaître le nom des bâtiments ou des rues présents dans le champ de vision.

Selon un axe plus orienté vers la télé-opération cette fois, de nombreux chercheurs travaillent également à l'incrustation d'objets virtuels sur des scènes réelles (Ernadotte et al., 1997) ou l'inverse, par exemple l'incrustation de personnages réels dans des scènes virtuelles. Ce dernier cas revêt un intérêt particulier pour tout ce qui relève du travail coopératif ou de la téléconférence (Dugelay et al., 1998).

3.1.2 Sémantique

D'un point de vue sémantique, les processus d'assistance exploités dans ces exemples consistent à enrichir la perception normale de l'opérateur en y ajoutant de l'information supplémentaire. Ces processus sont caractérisés par les deux points suivants :

- L'information perceptive supplémentaire fournie à l'opérateur n'est pas initialement présente, sous aucune forme que ce soit, dans l'environnement : cette information est tirée d'une base de données bibliographiques dans le cas de la librairie, et géographiques dans celui de la navigation. Dans tous les cas, le système dispose de cette information par un moyen autre que l'acquisition, c'est à dire sans passer par le manipulateur. Du point de vue de l'opérateur, l'information perceptive est donc «générée» dans la mesure où elle ne provient pas de l'environnement.
- Cependant, le système peut dans certains cas avoir besoin de la perception normale (information acquise par le manipulateur) afin de générer l'information perceptive souhaitée au moment propice. Par exemple, pour afficher le nom de l'auteur du livre sur lequel le point de vue est focalisé, il faut déjà détecter de quel livre il s'agit. De même l'aide à la navigation dépendra de la position courante de l'opérateur dans l'immeuble. Dans ces exemples, la génération de perception supplémentaire s'effectue donc en partie grâce à une donnée extérieure.

3.1.3 Localisation

De tels processus d'assistance enrichissent l'information transmise à l'interface (P_{si}) en y ajoutant une information n'émanant pas du manipulateur (P_{ms}). Ces processus interviennent donc au sein du système, et n'affectent pas la manière dont cette perception supplémentaire sera mise à la disposition de l'opérateur ; ils n'ont donc pas d'incidence théorique sur la restitution. La figure 3.1 ci-contre localise ces processus en indiquant (en gras) le chemin d'information affecté.

Afin de figer une description synthétique de ce type d'assistance, la formulation suivante est proposée :

Formulation: Génération de Perception

*Dans un cas d'assistance à l'opérateur par **génération de perception**, le système génère artificiellement une information perceptive supplémentaire, non issue de l'environnement et non acquise par le manipulateur, afin de la restituer à l'opérateur.* ▲

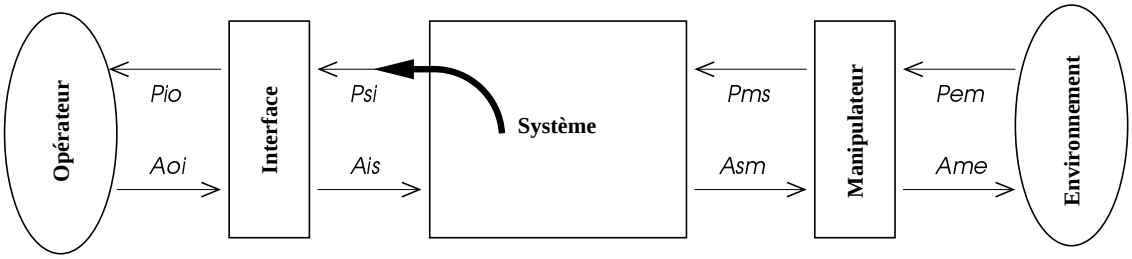

FIG. 3.1: Génération de perception

3.2 Altération de la perception

Les applications de la réalité augmentée ne se bornent pas à ajouter des perceptions originellement inexistantes. Elles permettent aussi de modifier les perceptions naturelles. Voici quelques exemples représentatifs de cette idée.

3.2.1 Exemples

- Berger et al. (1996) décrivent un exemple de réalité augmentée destiné à la simulation d'éclairage. À partir d'une vue réelle d'un pont de Paris, il est possible d'ajouter des sources de lumière virtuelles à n'importe quel endroit du pont. Ces sources de lumière virtuelles sont complètement configurables (intensité lumineuse, couleur etc.) et permettent de tester différentes conditions d'éclairement, reproduites dans la réalité par de véritables projecteurs. La résultante lumineuse de chaque configuration est présentée superposée à l'image réelle du pont, de manière à produire une photo réaliste de l'ensemble. Avec ce système, il n'est plus besoin de faire des tests réels d'éclairage avant de décider de l'emplacement définitif de chaque projecteur, ce qui constitue un gain de temps et d'argent.
- Bajura et al. (1992) décrivent un exemple de réalité augmentée appliquée au domaine médical et destiné à améliorer les systèmes échographiques. Une vue réelle (filmée) du ventre d'une femme enceinte est tout d'abord proposée au gynécologue sur un écran d'ordinateur. En plus de cette vue réelle, le ventre de la patiente est virtuellement «découpé» puis une image virtuelle du fœtus, reconstituée à partir d'une séquence d'échos ultrasonores, est superposée à l'image réelle. Le gynécologue obtient ainsi virtuellement la possibilité de «voir à travers» le ventre de la patiente, en trois dimensions et en temps réel, comme si celui-ci était transparent.

L'idée de transparence des matériaux est en fait particulièrement utilisée dans le domaine médical. Elle permet de visualiser des tissus endommagés ou des eudèmes cérébraux (Mellor, 1995), ou encore de percevoir dans l'espace la position d'outils chirurgicaux endoscopiques par rapport à la zone opératoire (State et al., 1996).

3.2.2 Sémantique

D'un point de vue sémantique, les processus d'assistance exploités dans ces exemples consistent à modifier l'information perceptive naturelle. Ces processus sont caractérisés par les deux points suivants :

- Contrairement au cas précédent, l'information perceptive fournie à l'opérateur et affectée par ce type d'assistance est issue de l'environnement. Cette information est donc acquise par le manipulateur avant d'être traitée : le ventre de la patiente existe dans l'environnement et le manipulateur l'acquiert au moyen d'une caméra. Le fœtus existe dans l'environnement et le manipulateur l'acquiert au moyen de capteurs à ultra-sons. De même, le pont,

ainsi que ses conditions d'éclairage initiales sont photographiés depuis l'environnement réel.

➤ Cependant, cette information perceptive naturelle n'est pas restituée telle quelle à l'opérateur : elle fait l'objet d'une modification, ou d'une *altération*. Le ventre de la mère est rendu transparent, ou plus exactement partiellement «effacé» sur l'image vidéo, de manière à laisser voir le fœtus. De même, les conditions initiales d'éclairage du pont sont altérées par la présence de projecteurs virtuels.

3.2.3 Localisation

De tels processus d'assistance modifient l'information perceptive acquise par le manipulateur (P_{ms}) pour en transmettre une version altérée à l'interface (P_{si}). De même que dans le cas précédent, ces processus n'ont pas d'incidence théorique sur la restitution de l'information modifiée, ni sur la manière dont elle a été saisie, puisqu'ils interviennent après la phase d'acquisition. Ils interviennent donc au niveau du système. La figure 3.2 localise ces processus en indiquant (en gras) le chemin d'information affecté.

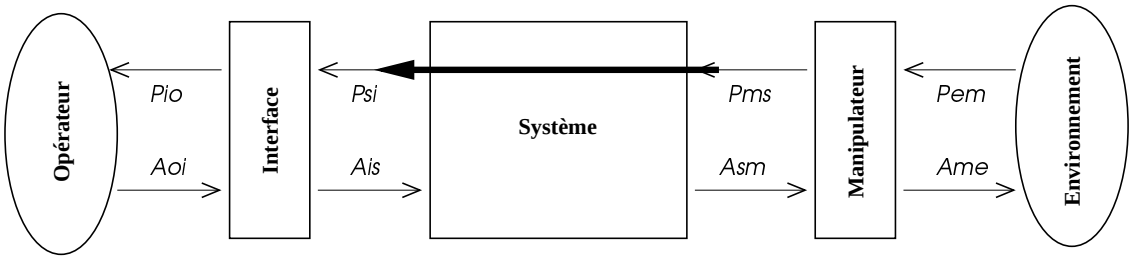

FIG. 3.2: Altération de la perception

Afin de figer une description synthétique de ces exemples, la formulation suivante est proposée :

Formulation: Altération de la Perception

*Dans un cas d'assistance à l'opérateur par **altération de la perception**, le système altère artificiellement l'information perceptive issue de l'environnement et acquise par le manipulateur, afin d'en restituer une version modifiée à l'opérateur.* ▲

3.3 Transmodalisation de la perception

Une autre manière de travailler sur une perception naturelle, sans pour autant en modifier le contenu, consiste à la restituer d'une manière différente de celle dont elle est acquise. Les exemples suivants illustrent cette idée.

3.3.1 Exemples

➤ Kitamura et al. (1998) présentent un système d'aide au placement relatif d'objets virtuels. L'opérateur ne dispose que d'un retour visuel sur écran, sans relief, de l'environnement. Aucun retour d'effort n'est non plus disponible. Le système décrit propose cependant une aide au placement des objets consistant à colorer en rouge les surfaces des objets en collision. L'opérateur dispose donc, grâce à la modalité visuelle, d'une information qui serait autrement inaccessible.

- Plus démonstratif est le système d'assistance à la navigation pour les personnes aveugles présenté par Loomis et al. (1998). L'idée de cette assistance est de leur fournir, sous une modalité sonore, de l'information sur la présence éventuelle d'obstacles et leur distance, l'orientation ou encore la proximité du point d'arrivé souhaité. Ainsi, la présence d'un obstacle ou simplement d'un objet est représentée par une source sonore virtuelle dont la position spatiale correspond à celle de l'obstacle. Le volume du son virtuel émis est d'autant plus important que l'objet est proche. Malgré sa cécité, la personne pourvue de ce système peut donc se représenter spatialement son environnement. Un signal audio synthétisant la parole peut également être utilisé pour donner les noms des rues quand la personne arrive à un carrefour.

Il existe de nombreuses autres utilisations de la modalité sonore en guise de restitution alternative d'information : celle-ci est par exemple également utilisée de manière redondante, ou bien en lieu et place du retour d'effort quand aucun périphérique de ce type n'est disponible (Richard et al., 1995). Il est alors possible de produire un son virtuel dont la hauteur représente l'intensité du retour d'effort qui aurait dû être perçu.

3.3.2 Sémantique

D'un point de vue sémantique, nous sommes ici en face de processus qui consistent à changer la manière de restituer une perception donnée. Ces processus sont caractérisés par deux aspects :

- De même que dans le cas précédent, l'information perceptive affectée par ce type de traitement est issue de l'environnement et est par conséquent acquise par le manipulateur. Dans les exemples précédents, cette information est normalement de nature visuelle (obstacles, panneaux indicateurs de rues…) ou tactile (retour d'effort, collision…). Cette information est *a priori* destinée à être transmise à l'opérateur afin que celui-ci en prenne connaissance.
- Contrairement au cas précédent, cette information perceptive n'a pas vocation à être altérée. Au contraire, le but est bien de pouvoir la restituer intégralement à l'opérateur, sans modification. Cependant, l'information perceptive en question ne lui est pas retransmise sous sa modalité d'origine : dans les exemples précédents, l'information visuelle comme la présence et la position d'un obstacle est restituée sous forme sonore, tandis que l'information tactile comme le retour d'effort ou la collision entre objets est transmise sous forme visuelle. Il s'agit donc d'un changement de modalité pour la restitution, soit d'une *transmodalisation* de l'information perceptive.

3.3.3 Localisation

La mise en place de tels processus d'assistance ne requiert un travail particulier qu'au niveau de l'interface : l'information perceptive est acquise normalement, et le travail de transmodalisation consiste à modifier la manière dont cette information sera restituée à l'opérateur ($P_{si} \Longrightarrow P_{io}$). La figure 3.3 en page suivante localise ces processus en indiquant (en gras) le chemin d'information affecté.

Les exemples que nous venons de décrire correspondent en fait à deux situations légèrement différentes :

- Dans un premier cas, comme celui du placement relatif d'objets virtuels, la modalité naturelle de l'information est accessible à l'opérateur, mais l'interface ne dispose pas du mécanisme de restitution nécessaire. Par exemple, l'opérateur peut naturellement acquérir une information (tactile) de collision entre les objets qu'il manipule, mais dans le cas de la manipulation virtuelle, l'interface n'est pas dotée de périphériques à retour d'effort. La modalité visuelle est donc utilisée.
- Dans un deuxième cas, comme celui de l'assistance aux personnes aveugles, la modalité naturelle de l'information n'est pas accessible à l'opérateur, indépendamment de la capacité

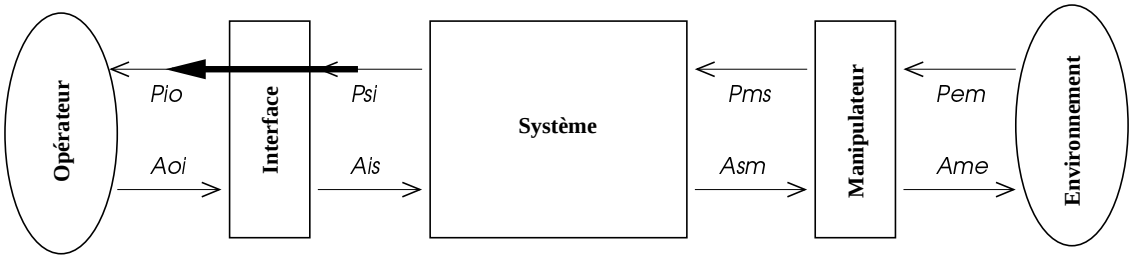

FIG. 3.3: Transmodalisation de la perception

de l'interface à la restituer. Dans notre exemple, la personne aveugle de peut accéder à une information visuelle. La modalité sonore est donc utilisée en remplacement.

Afin de figer une description synthétique de ces exemples, la formulation suivante est proposée :

Formulation: Transmodalisation de la perception

Dans un cas d'assistance à l'opérateur par ***transmodalisation de la perception****, une information perceptive dont la modalité d'origine est non gérable par l'interface ou inaccessible à l'opérateur, est restituée à celui-ci sous une modalité différente.* ▲

☛ Notons que dans l'exemple de l'assistance aux personnes aveugles, l'incapacité de l'opérateur à recevoir de l'information selon une modalité particulière est due à un handicap. Cependant, une information dont la modalité naturelle est extérieure au domaine de perception sensorielle de l'humain (infrarouges, ultrasons, champs magnétiques…) peut aussi bien faire l'objet d'un tel traitement.

3.4 Reconstitution de perception

Un quatrième cas d'assistance à la perception issu de la réalité augmentée réside dans l'idée de fabriquer artificiellement une perception manquante. Voici deux exemples illustrant cette idée.

3.4.1 Exemples

➤ Côté et Lavallée (1995) décrivent le cas d'un submersible téléguidé, chargé d'une mission de maintenance (détection de fissures) au pied d'un barrage hydro-électrique. La profondeur de l'eau rend la visibilité nulle dans la zone d'inspection. Cependant, la structure du barrage est connue et un modèle géométrique tridimensionnel en est disponible. Afin de remédier au manque de visibilité, des capteurs de position sont disposés sur le barrage et le submersible, ce qui permet de connaître la position exacte de celui-ci par rapport à la structure. Une fois cette information disponible, le modèle géométrique du barrage est utilisé pour reconstituer artificiellement (en images de synthèse) la vision du barrage que l'opérateur aurait eu en pleine lumière. Ce type d'utilisation des techniques de réalité virtuelle se retrouve plus généralement dans les applications de téléguidage de sous-marin (Hine et al., 1994) dans lesquelles l'environnement est reconstitué par images de synthèse.

➤ L'exemple de la visualisation d'un fœtus par transparence, décrit précédemment dans le cas d'une assistance par altération de perception, entre également dans cette catégorie : en plus

de l'altération de la perception, consistant à rendre transparent le ventre de la mère, l'image du fœtus n'est en fait pas disponible dans l'environnement. Elle est cependant reconstituée à partir d'une séquence d'information différente, provenant des capteurs d'ultrasons.

3.4.2 Sémantique

D'un point de vue sémantique, les processus d'assistance exploités dans ces exemples consistent cette fois-ci à fabriquer artificiellement une information perceptive manquante. Ces processus sont caractérisés par trois aspects :

- Tout d'abord, l'information perceptive souhaitée n'est pas initialement disponible dans l'environnement. Dans les exemples décrits plus haut, il manque à chaque fois, pour des raisons différentes, le retour visuel de l'objet considéré (le barrage ou le fœtus).
- Cependant, il est possible de fabriquer artificiellement cette information perceptive en utilisant une *autre* information perceptive, qui elle, existe naturellement. Il s'agit donc d'une *reconstitution de perception*. Dans le cas du barrage, la connaissance de la position relative du sous-marin par rapport au barrage permet de reconstituer la perception visuelle. Dans le cas de l'échographie, l'information issue des capteurs à ultrasons est utilisée dans le même dessein.
- Dans certains cas, il peut s'avérer nécessaire de disposer en plus d'une information non perceptive (non acquise par le manipulateur). Dans le cas du barrage par exemple, la seule information issue des capteurs de position ne suffit pas. Il est nécessaire de disposer en plus du modèle tridimensionnel du barrage.

3.4.3 Localisation

La mise en place de processus d'assistance de ce type requiert un travail supplémentaire d'acquisition ($P_{em} \Longrightarrow P_{ms}$), afin d'obtenir l'information perceptive de substitution qui n'aurait pas été nécessaire autrement. D'autre part, cette information doit être traitée afin de reconstituer l'information perceptive manquante (P_{si}). La figure 3.4 localise ces processus en indiquant (en gras) le chemin d'information affecté.

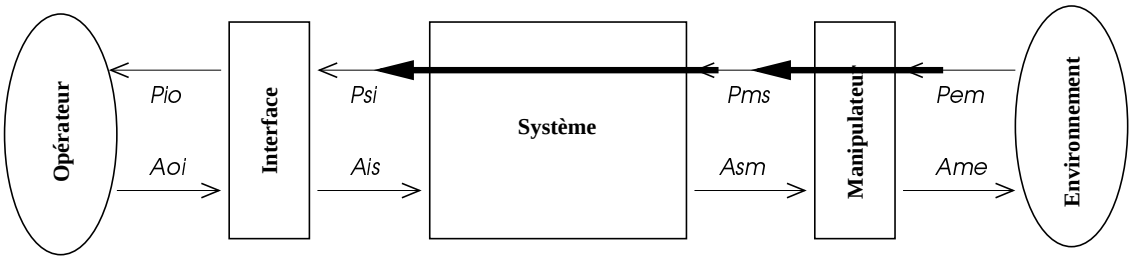

FIG. 3.4: Reconstitution de perception

Afin de figer une description synthétique de ces exemples, la formulation suivante est proposée :

Formulation: Reconstitution de Perception

*Dans un cas d'assistance à l'opérateur par **reconstitution de perception**, le système reconstitue une information perceptive manquante dans l'environnement, ou que le manipulateur ne peut pas acquérir, à partir d'une information perceptive de substitution.* ▲

3.5 Filtrage de l'action

Jusqu'à présent, les cas d'assistance étudiés ont exclusivement affecté la perception de l'opérateur. Les exemples suivants concernent cette fois plus spécifiquement les actions de celui-ci.

3.5.1 Exemples

- Kitamura et al. (1998) décrivent un système d'assistance à la manipulation et au placement relatif d'objets virtuels : le problème est de positionner correctement un certain nombre d'objets, par exemple un ensemble de cubes les uns contre ou sur les autres. Pour ce faire, l'opérateur dispose d'un périphérique à six degrés de liberté comme un trackball, qui lui permet de manipuler les objets dans tous les sens. Le seul retour disponible est un retour visuel sur écran. La difficulté du problème est que les objets virtuels ne subissent pas de collision physique réelle et peuvent donc se traverser. Il en résulte que l'opérateur ne peut compter sur celle-ci pour l'aider à positionner les faces les unes contre les autres. Pour remédier à ce manque, le système proposé détecte les collisions entre faces, et en déduit certains degrés de liberté contraints dans le mouvement des cubes. Par la suite, même si l'opérateur commande un mouvement sur ce(s) degré(s) de liberté contraint(s), le système n'en tient pas compte et assure ainsi que les faces restent en contact.
- L'exemple précédent se déroule dans un contexte de réalité virtuelle de synthèse, bien que l'on puisse aisément imaginer sa transposition à un cas de télé-opération. Il existe cependant déjà des cas de filtrage de l'action dans de véritables situations de télé-opération, en particulier dans les applications du concept de «lignes de tâche et guides de déplacement»* (Backes et al., 1998). Le concept de lignes de tâche et guides de déplacement implique deux phases dans la télé-opération d'un robot : dans une première phase, l'opérateur détermine un chemin que le manipulateur devra suivre. Il ne s'agit pas nécessairement d'une télé-opération de robot mobile : le chemin peut également définir un déplacement complexe pour un bras manipulateur doté de nombreuses articulations. Une fois ce chemin défini, l'opérateur ne contrôle le manipulateur qu'en termes de vitesse à une dimension (avant, arrière et arrêt) le long de ce chemin prédéfini. Dans l'exemple décrit par Backes et al. (1998), l'opérateur dispose d'un trackball pour les deux phases de la télé-opération. En conséquence, pendant la phase de contrôle de vitesse, un seul degré de liberté du trackball est utilisé pour représenter la vitesse requise. Les autres degrés de liberté sont simplement ignorés par le système, même si l'opérateur, volontairement ou non, joue dessus.

3.5.2 Sémantique

D'un point de vue sémantique, les processus d'assistance exploités dans ces exemples consistent à traiter les actions commandées par l'opérateur. Les point suivants caractérisent ces processus :

- Le traitement opéré sur l'information actionnelle consiste à n'en conserver qu'une certaine partie. Dans l'exemple précédent, le système ignore certains degrés de liberté du périphérique de commande et ne conserve que ceux qui l'intéressent. La partie de la requête qui est conservée ne fait l'objet d'aucune modification, elle est utilisée telle quelle. On peut donc parler de *filtrage de l'action*, dans la mesure ou le traitement consiste à ne conserver, sans modification, que l'information utile à un moment donné.
- Ce type d'assistance a également la particularité de traiter l'information actionnelle *après* la commande complète de celle-ci. Cela signifie en particulier que l'opérateur a la possibilité de spécifier des actions non autorisées car le traitement (ou la vérification) n'interviendra qu'après coup. Il en résulte que l'exécution véritable de la commande n'est pas conforme à la commande produite par l'opérateur. Kitamura et al. (1998) en sont d'ailleurs pleinement

*«Task Lines and Motion Guides» en anglais.

conscients et signalent que ce type de fonctionnalité doit être utilisé avec précaution, car cette non conformité est une source potentielle de gêne pour l'opérateur.

3.5.3 Localisation

La mise en œuvre de tels processus d'assistance implique un traitement de l'information actionnelle après commande (A_{is}) et avant transmission au manipulateur (A_{sm}). Ils sont donc localisés au sein du système. La figure 3.5 localise ces processus en indiquant (en gras) le chemin d'information affecté.

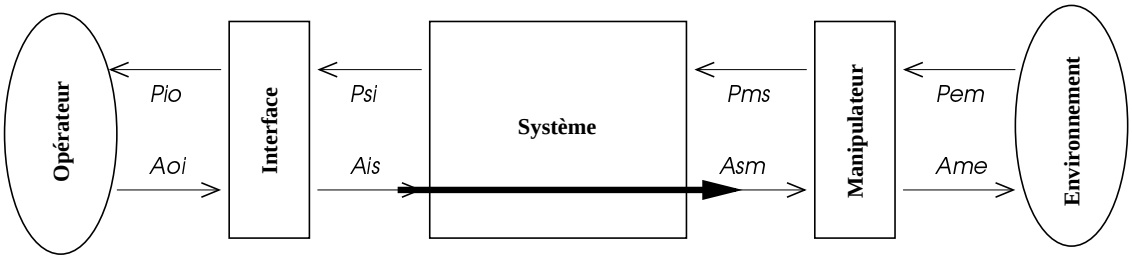

FIG. 3.5: Filtrage de l'action

Afin de figer une description synthétique de ces exemples, la formulation suivante est proposée :

Formulation: Filtrage de l'action

*Dans un cas d'assistance à l'opérateur par **filtrage de l'action**, le système ne conserve qu'une partie de l'information actionnelle spécifiée par l'opérateur, afin de faire exécuter cette partie, non modifiée, par le manipulateur.* ▲

3.6 Commande paramétrique

Après avoir rencontré plusieurs cas d'assistance à la perception et un cas d'assistance à l'action, nous détaillons maintenant, au travers de cette section et de la suivante, des cas d'assistance affectant l'information à la fois dans le sens de l'action et dans celui de la perception. Dans ce premier cas, des exemples d'assistance à la commande sont donnés.

3.6.1 Exemples

- Dubois et al. (1997) décrit une application de la réalité augmentée au domaine chirurgical de la ponction péricardique : le chirurgien doit manœuvrer une aiguille à l'intérieur du patient afin de l'amener à l'endroit requis puis opérer la ponction. L'outil endoscopique est accompagné d'une micro-caméra permettant au chirurgien de se repérer dans les ramifications du réseau vasculaire. Un système de réalité augmentée assiste le chirurgien, après que celui-ci lui ait indiqué le trajet optimal, en produisant un retour d'effort sur le dispositif de guidage de l'aiguille : le retour d'effort informe le chirurgien sur l'optimalité du déplacement qu'il spécifie, et bloque les mouvements dangereux restreignant ainsi la marge de commande aux déplacements sûrs et autorisés.
- Un deuxième exemple d'assistance à la commande existe dans pratiquement toutes les applications de CAO ainsi que dans les applications militaires : il s'agit du mécanisme de

présélection graphique. L'opérateur dispose par exemple d'une souris qui lui permet de sélectionner des objets ou des zones sur l'écran. Selon le contexte, certaines zones de l'image ou certains objets sont présélectionnés. La présélection est souvent indiquée par un cadre virtuel tracé en fil de fer autour de la zone ou de l'objet. En dehors de ces zones, toute action de l'opérateur est impossible, ce qui oblige celui-ci à ne produire que des commandes valides.

3.6.2 Sémantique

D'un point de vue sémantique, les processus exploités dans ces exemples assistent l'opérateur au niveau de la commande des actions. Ces processus sont caractérisés par les points suivants :

- Un tel type d'assistance consiste en fait à configurer certains paramètres de l'interface par laquelle l'opérateur spécifie les actions, d'où le terme de commande *paramétrique*. Par rapport à une situation non assistée où l'opérateur dispose de toute la marge de manœuvre permise par son interface de commande, ces processus d'assistance ont pour effet, selon les exemples précédents, de restreindre les positions autorisées du dispositif de guidage, ou les zones cliquables sur l'écran.
- Bien que poursuivant des objectifs voisins, il existe une différence fondamentale entre ce type d'assistance et l'assistance par filtrage de l'action, analysée précédemment. Dans le cas du filtrage, l'assistance intervient *après* la commande de l'action, ce qui implique que celle-ci puisse être non conforme à la commande exacte de l'opérateur. Dans le cas qui nous occupe ici, l'assistance intervient *au moment* de la commande. Il en résulte que l'opérateur ne peut que spécifier des commandes valides, car les commandes erronées ne sont même pas physiquement autorisées par l'interface. Par rapport à l'assistance par filtrage de l'action, l'opérateur est informé de la validité de ses commandes au moment même où elles sont effectuées, et que l'exécution d'une action donnée sera toujours conforme à sa commande.

Ce type d'assistance requiert en général un matériel plus lourd que celui requis pour une assistance par filtrage de l'action (par exemple, si un dispositif à retour d'effort s'avère nécessaire), mais évitera cependant la gène occasionnée par une non conformité de l'exécution à la commande correspondante.

3.6.3 Localisation

La mise en place de processus d'assistance de ce type s'effectue au moyen d'une boucle d'information entre l'opérateur et l'interface. En effet, l'action de l'opérateur sur l'interface elle-même (A_{oi}) peut varier d'un moment à l'autre, suivant le paramétrage des périphériques de commande, mais l'opérateur est constamment informé de ce paramétrage par un retour perceptif de l'interface (P_{io}). La figure 3.6 localise ces processus en indiquant (en gras) le chemin d'information affecté.

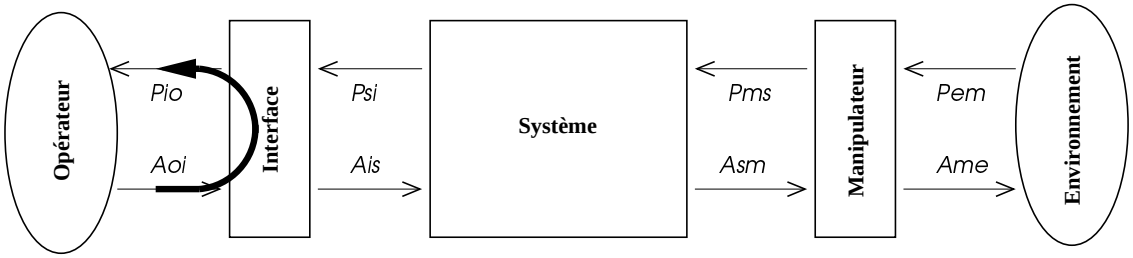

FIG. 3.6: Commande paramétrique

Afin de figer une description synthétique de ces exemples, la formulation suivante est proposée :

Formulation: Commande paramétrique

*Dans un cas d'assistance à l'opérateur par **commande paramétrique**, l'interface est configurée de manière à prédéfinir certains paramètres de la commande.* ▲

3.7 Exécution virtuelle

Un deuxième cas d'assistance impliquant à la fois l'action et la perception, consiste cette fois-ci à aider l'opérateur au niveau de l'exécution même d'une action, et non plus seulement au niveau de sa commande.

3.7.1 Exemples

Une fonctionalité particulièrement développée en robotique semi-autonome est la planification de trajectoire (Kim, 1996). La problématique est par exemple de faire évoluer un robot mobile dans un environnement distant avec lequel les délais de communication sont importants. Le pilotage en temps réel est donc impossible. Pour résoudre ce problème, de nombreuses applications ont été développées pour permettre à l'opérateur de tester la globalité de la commande avant de l'exécuter réellement : l'opérateur manipule, en temps réel cette fois-ci, une version virtuelle du robot. Une fois satisfait de son plan, il le transfère au manipulateur pour exécution.

Dans des applications de télé-opération, ce type d'assistance se rencontre aussi bien en réalité virtuelle pure, ou à la fois le manipulateur et l'environnement sont simulés, qu'en réalité augmentée. Dans ce dernier cas, il est fréquent de voir une version virtuelle du manipulateur s'afficher en mode fil de fer par dessus l'image réelle de l'environnement.

3.7.2 Sémantique

D'un point de vue sémantique, les processus d'assistance exploités dans ces exemples consistent à simuler une opération avant de l'effectuer réellement. Ces processus sont caractérisés par les points suivants :

- Dans une telle situation, l'opérateur est mis temporairement en interaction avec une instance virtuelle du couple manipulateur / environnement et ce jusqu'à ce qu'il soit satisfait de la manière dont l'opération concernée s'est déroulée. Nous sommes donc ici devant un cas d'assistance par *exécution virtuelle* d'une opération, dans la mesure ou l'opérateur peut interagir en temps réel avec la simulation, et de la même manière qu'il l'aurait fait dans la réalité si la communication avec le manipulateur avait était immédiate.
- Afin de mener à bien cette exécution virtuelle, il est nécessaire de disposer des connaissances suffisantes pour représenter non seulement le robot et son environnement, mais également les interactions entre ceux-ci. Dans l'exemple précédent de la planification de trajectoire, le but de l'assistance est justement de prévoir les collisions avec d'éventuels obstacles, afin de pouvoir les éviter en phase d'exécution réelle. La seule modélisation virtuelle du robot en donc en général insuffisante.

3.7.3 Localisation

Dans une situation de ce type, l'opérateur agit normalement sur l'interface, comme si l'action était exécutée réellement. En termes de commande, la situation est donc inchangée. Réciproquement, l'opérateur est censé percevoir une information analogue à celle qui lui serait retournée si l'action était vraiment exécutée. La phase de restitution est donc elle aussi identique. Il en résulte

que de tels processus d'assistance interviennent au sein du système, en coupant le chemin normal d'information ($A_{is} \Longrightarrow P_{si}$). La figure 3.7 localise ces processus en indiquant (en gras) le chemin d'information affecté.

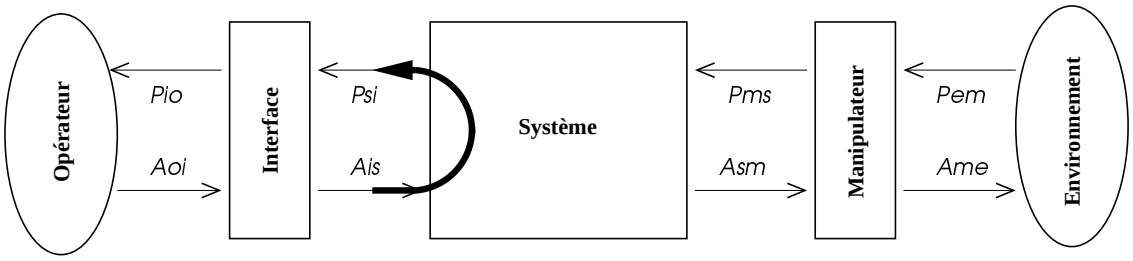

FIG. 3.7: Exécution virtuelle

Afin de figer une description synthétique de ces exemples, la formulation suivante est proposée :

Formulation: Exécution virtuelle

*Dans un cas d'assistance à l'opérateur par **exécution virtuelle**, le système simule les processus d'exécution et d'acquisition au lieu de faire exécuter réellement l'opération concernée par le manipulateur.* ▲

3.8 Résumé

La figure 3.8 résume l'ensemble des cas d'assistance que nous venons d'étudier. Ces exemples sont représentatif de l'utilisation des principes de réalité virtuelle pour assister l'opérateur, que ce soit en réalité augmentée ou en télé-opération.

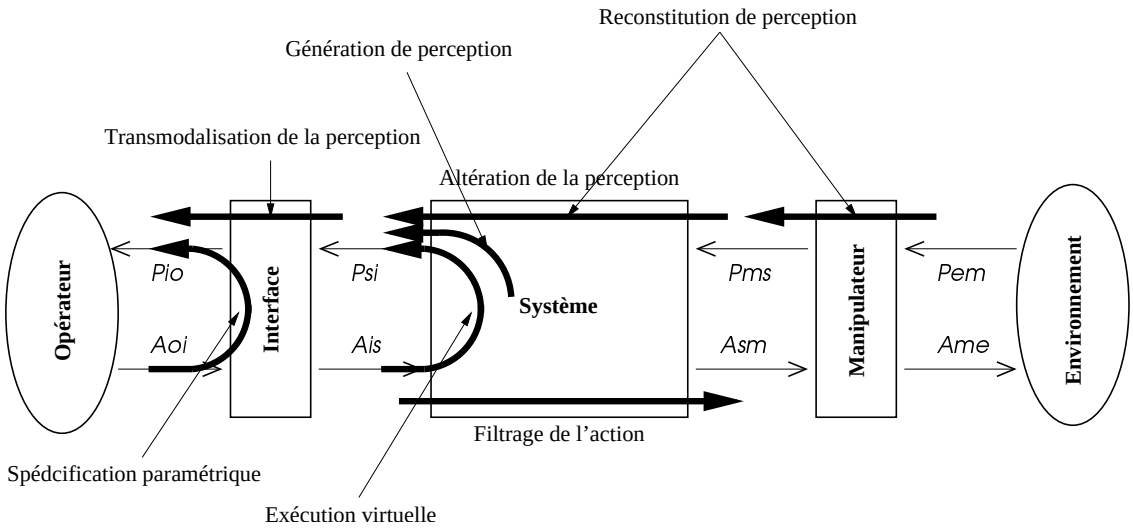

FIG. 3.8: Résumé des cas d'assistance

Un comptage rapide montre que sur les sept cas étudiés, quatre d'entre eux affectent uniquement la perception de l'opérateur, un seul affecte uniquement les actions de celui-ci, et deux d'entre eux affectent à la fois la perception et l'action. D'autre part, les deux cas affectant à la fois l'action et la perception sont «tournés» vers l'opérateur, c'est à dire qu'ils affectent l'interaction entre l'opérateur et l'interface et / ou le système (par l'intermédiaire de l'interface), sans aller plus loin dans le modèle. Il semble donc qu'il existe un déséquilibre de nature égocentrique dans l'utilisation des concepts de réalité virtuelle pour l'assistance à l'opérateur. D'où vient ce déséquilibre, et est-il possible de le corriger ? Le chapitre suivant s'attache à répondre à ces questions.

Chapitre 4

Symétrisation des Exemples

Le chapitre précédent nous a permis d'étudier comment les concepts issus de la réalité virtuelle sont utilisés pour concevoir des systèmes d'assistance à l'opérateur, en télé-opération comme en réalité augmentée. Outre la mise en évidence de plusieurs grandes classes sémantiques d'assistance, il est apparu un certain déséquilibre dans l'application de ces concepts : leur utilisation est principalement orientée dans le sens de la perception, et toujours à destination de l'opérateur.

Face à cette constatation, deux hypothèses sont possibles : soit il manque encore à l'heure actuelle un certain nombre d'applications de ces idées, soit ces applications existent, mais leur relation avec les idées développées dans le chapitre précédent n'ont pas jusqu'ici été mises à jour. L'objet de ce chapitre est d'examiner la validité de ces deux hypothèses, sachant qu'il n'y a *a priori* aucune raison pour qu'elles soient mutuellement exclusives.

4.1 Méthodologie

La question à laquelle nous devons répondre est donc la suivante : peut-on rencontrer ou imaginer des cas d'assistance à l'opérateur qui correspondent à l'une des sept classes sémantiques analysées dans le chapitre précédent, mais qui soient cette fois-ci orientés principalement dans le sens des actions et toujours à destination de l'environnement ?

Afin de répondre à cette question, nous proposons de tirer parti d'une caractéristique essentielle du modèle ***LiSA*** : sa symétrie.

4.1.1 Symétrie du modèle *LiSA*

Le modèle ***LiSA*** (figure 2.1 en page 70) est pourvu d'une grande symétrie. Celle qui nous intéresse particulièrement est sa symétrie centrale (le centre étant le système) qui reflète l'équivalence des propriétés de l'opérateur et du manipulateur.

- L'opérateur et l'environnement d'une part, puis l'interface et le manipulateur d'autre part, jouent des rôles symétriques au sein du modèle. Plus précisément, en termes de traitement de l'information, les processus symétriques sont l'acquisition et la commande d'une part, puis la restitution et l'exécution d'autre part. En effet, le processus d'acquisition consiste à saisir de l'information (perceptive) en provenance de l'environnement. Symétriquement, le processus de commande consiste à saisir de l'information (actionnelle) en provenance de

l'opérateur. De la même manière, le processus de restitution transmet une information (perceptive) à l'opérateur, et le processus d'exécution transmet une information (actionnelle) à l'environnement.

➤ Mis à part ces processus élémentaires de saisie et de transport d'information, les exemples précédents mettent en œuvre d'autres processus de traitement de l'information, dont la sémantique a été analysée dans chaque cas. La symétrie de notre modèle nous conduit à l'idée que chaque classe sémantique étudiée précédemment a sa contrepartie symétrique : quand un type de traitement s'applique à la perception, il est possible d'envisager le même type de traitement appliqué à l'action, et réciproquement. Ceci ne présage cependant pas de l'utilité du résultat obtenu, mais fournit un moyen systématique de découvrir de nouveaux cas d'assistance.

4.1.2 Inversion des exemples

Compte tenu des éléments précédents, nous proposons donc la démarche suivante :

➤ Pour chaque classe sémantique d'assistance envisagée dans le chapitre précédent, nous procédons systématiquement à son inversion en profitant de la symétrie centrale du modèle ***LiSA***.

➤ Cette inversion systématique, selon les deux axes d'analyse utilisés précédemment, nous apporte une nouvelle localisation des processus d'assistance, ainsi qu'une nouvelle formulation synthétique décrivant la sémantique de ces processus.

➤ Un fois ces deux axes redéfinis, nous tentons de faire le rapprochement avec des applications existantes, et le cas échéant, nous proposons de nouveaux exemples d'assistance entrant dans cette catégorie.

4.2 Génération d'action

Dans cette section, nous procédons à la symétrisation du cas «Génération de perception» décrit en page 73. La description synthétique de ce cas d'assistance peut être reformulée de la manière suivante :

Reformulation: Génération d'action

*Dans un cas d'assistance à l'opérateur par **génération d'action**, le système génère artificiellement une information actionnelle supplémentaire, non fournie par l'opérateur, et non spécifiée sur l'interface, afin qu'elle soit exécutée par le manipulateur.* ▼

4.2.1 Sémantique et localisation

D'un point de vue sémantique, les processus d'assistance décrits ici consistent à enrichir l'action normale de l'opérateur en y ajoutant de l'information supplémentaire. Ces processus sont caractérisés par les deux points suivants :

➤ L'information actionnelle supplémentaire n'est pas fournie par l'opérateur. En particulier, cette information n'est donc pas spécifiée au moyen de l'interface. Du point de vue de l'opérateur, l'information actionnelle est donc «générée».

➤ Cependant, le système peut, dans certains cas, avoir besoin de l'action fournie par l'opérateur afin de générer l'information actionnelle supplémentaire souhaitée au moment propice.

En termes de localisation, de tels processus d'assistance enrichissent l'information transmise au manipulateur (A_{sm}) en y ajoutant une information n'émanant pas de l'interface (A_{is}). Ces

processus interviennent donc au sein du système, et n'affectent pas la manière dont cette action supplémentaire sera gérée par le manipulateur ; ils n'ont donc pas d'incidence théorique sur l'exécution. La figure 4.1 localise ces processus en indiquant (en gras) le chemin d'information affecté.

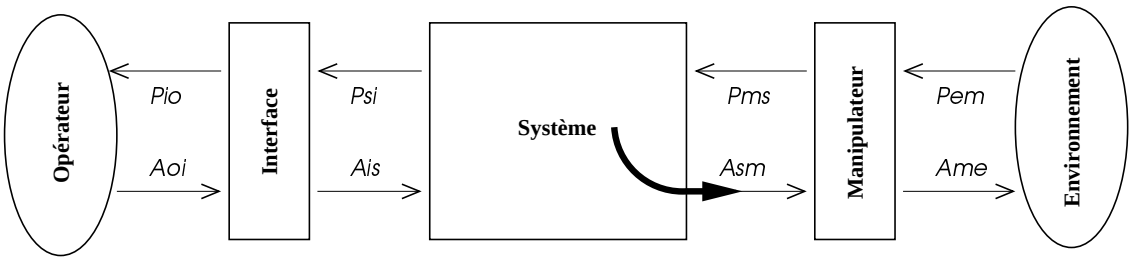

FIG. 4.1: Génération d'action

4.2.2 Application

Il existe des utilisations de bras manipulateurs téléguidés derrière une vitre et destinés à la manipulation de produits toxiques ou dangereux. L'opérateur doit par exemple manipuler différentes fioles pour étudier certaines réactions chimiques. Dans une version non assistée du système, l'opérateur aura donc la charge de positionner le bras à l'endroit souhaité (autour d'une fiole), saisir le récipient, puis le transporter en préservant sa verticalité.

Les opérations de positionnement et de transport sont similaires : il s'agit de déplacer le bras manipulateur. Cependant, l'opération de transport requiert un travail supplémentaire de conservation de la verticalité de l'objet. Afin d'éviter de fournir deux commandes distinctes à l'opérateur, on implémente un processus permettant de faire la différence entre les commandes de positionnement et de transport (le transport a lieu quand un objet est saisi). L'opérateur ne dispose par conséquent plus que d'une seule commande : celle du déplacement du bras. Le système, dans le cas du transport, *génère* automatiquement l'action supplémentaire consistant à assurer en permanence une orientation verticale constante de l'organe de préhension.

Notons que la mise en œuvre un tel cas d'assistance interdit toute communication directe entre l'opérateur et le manipulateur : si l'opérateur contrôlait directement le positionnement du bras manipulateur, le système ne pourrait que corriger *a posteriori* l'orientation de l'organe de préhension, en ayant constaté une variation parasite due aux commandes de positionnement générées par l'opérateur. Au contraire, pour que l'assistance soit efficace, il est nécessaire que le système applique sa correction *en même temps* qu'il transmet les commandes générées par l'opérateur.

Cet exemple d'assistance correspond bien à un processus de génération d'action, tel que nous l'avons décrit précédemment. En effet :

- L'information actionnelle supplémentaire (l'orientation de l'organe de préhension) n'est pas fournie par l'opérateur. C'est le système qui décide de la générer.
- Cependant, afin de générer une commande d'orientation correcte, le système a besoin de connaître les instructions de positionnement du bras émises par l'opérateur, information actionnelle de référence.

4.3 Altération de l'action

Dans cette section, nous procédons à la symétrisation du cas «Altération de la perception» décrit page 75. La description synthétique de ce cas d'assistance peut être reformulée de la manière suivante :

Reformulation: Altération de l'action

Dans un cas d'assistance à l'opérateur par ***altération de l'action****, le système altère artificiellement l'information actionnelle fournie par l'opérateur et spécifiée sur l'interface, afin d'en faire exécuter une version modifiée par le manipulateur.* ▼

4.3.1 Sémantique et localisation

D'un point de vue sémantique, les processus d'assistance décrits ici consistent à modifier l'information actionnelle naturelle. Ces processus sont caractérisés par les deux points suivants :

- ➤ Contrairement au cas précédent, l'information actionnelle fournie au manipulateur et affectée par ce type d'assistance provient de l'opérateur. Cette information est donc spécifiée au moyen de l'interface avant d'être traitée.
- ➤ Cependant, cette information actionnelle naturelle n'est pas exécutée telle quelle par le manipulateur : elle fait l'objet d'une modification, ou d'une *altération*.

En termes de localisation, de tels processus d'assistance modifient l'information actionnelle spécifiée par l'opérateur (A_{is}) pour en exécuter une version altérée transmise au manipulateur (A_{sm}). De même que dans le cas précédent, ces processus n'ont pas d'incidence théorique sur l'exécution de l'information modifiée, ni sur la manière dont elle a été saisie, puisqu'ils interviennent après la phase de commande. Ils interviennent donc au niveau du système. La figure 4.2 localise ces processus en indiquant (en gras) le chemin d'information affecté.

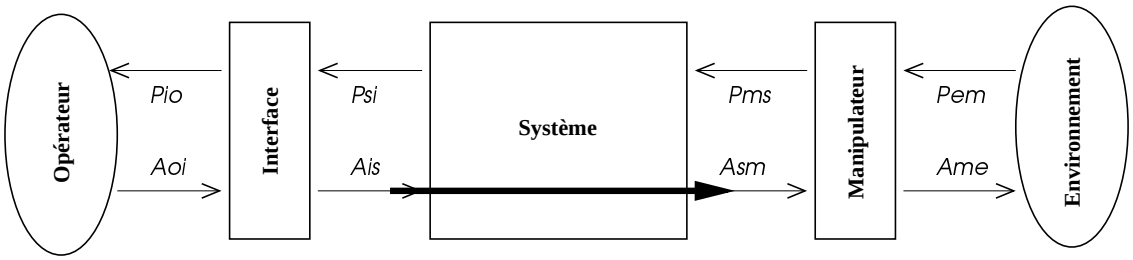

FIG. 4.2: Altération de l'action

4.3.2 Application

Plusieurs domaines d'application, à la fois en réalité virtuelle et en télé-opération utilisent des fonctionalités de ce type (Fernando et al., 1995). Un exemple parmi les plus connus est celui de la reconnaissance, puis de la manipulation tridimensionnelle d'objets avec contrainte, ce qui s'applique en particulier aux problèmes d'assemblage : les différentes pièces d'un robot (membres, articulations...) ou d'une voiture doivent être assemblées d'une manière précise, par exemple en les emboîtant dans un ordre et une direction donnée. Les systèmes de bras manipulateur permettant d'effectuer ces opérations d'assemblage sont capables de contraindre les mouvements que l'opérateur commande : la proximité de deux pièces dont l'une est déplacée par l'opérateur est reconnue comme le début d'une opération d'assemblage. Le mouvement commandé par l'opérateur est alors contraint de manière à ce que les pièces s'imbriquent correctement l'une dans l'autre, même si les gestes de l'opérateur ne sont pas suffisamment précis.

Notons qu'un tel cas d'assistance est bien différent de l'assistance par filtrage de l'action, décrite dans le chapitre précédent. Une assistance par filtrage de l'action consiste à ignorer certains paramètres de la commande (comme des degrés de liberté) et transmettre les autres sans modification. Dans le cas qui nous occupe ici, aucun degré de liberté de la commande (vecteur vitesse en trois dimensions) n'est ignoré, mais la valeur spécifiée pour chacun d'entre eux fait l'objet d'une modification.

Cet exemple d'assistance correspond bien à une processus d'altération de l'action, tel que nous l'avons décrit précédemment. En effet :

- Contrairement au cas précédent, l'information actionnelle fournie au manipulateur et affectée par ce type d'assistance provient de l'opérateur. Il s'agit de la commande de déplacement du bras manipulateur.
- Cependant, cette information actionnelle naturelle n'est pas exécutée telle quelle par le manipulateur : elle fait l'objet d'une modification, ou d'une *altération* de ses trois composantes spatiales.

4.4 Transmodalisation de l'action

Dans cette section, nous procédons à la symétrisation du cas «Transmodalisation de la perception» décrit en page 76. La description synthétique de ce cas d'assistance peut être reformulée de la manière suivante :

Reformulation: Transmodalisation de l'action

Dans un cas d'assistance à l'opérateur par ***transmodalisation de l'action****, une information actionnelle, dont la modalité d'origine est inaccessible à l'environnement ou non gérable par le manipulateur, est exécutée par celui-ci sous une modalité différente.* ▼

4.4.1 Sémantique et localisation

D'un point de vue sémantique, nous sommes ici en face de processus qui consistent à changer la manière d'exécuter une action donnée. Ces processus sont caractérisés par deux aspects :

- De même que dans le cas précédent, l'information actionnelle affectée par ce type de traitement est fournie par l'opérateur, et par conséquent spécifiée au moyen de l'interface. Cette information est *a priori* destinée à être transmise au manipulateur afin que celui-ci l'exécute.
- Contrairement au cas précédent, cette information n'a pas vocation à être altérée. Au contraire, le but est bien de pouvoir l'exécuter intégralement, sans modification. Cependant, l'information en question n'est pas exécutée sous sa modalité d'origine.

En termes de localisation, la mise en place de tels processus d'assistance ne requiert un travail particulier qu'au niveau du manipulateur : l'information actionnelle est spécifiée normalement, et le travail de transmodalisation consiste à modifier la manière dont cette information sera exécutée par le manipulateur ($A_{sm} \Longrightarrow A_{me}$). La figure 4.3 localise ces processus en indiquant (en gras) le chemin d'information affecté.

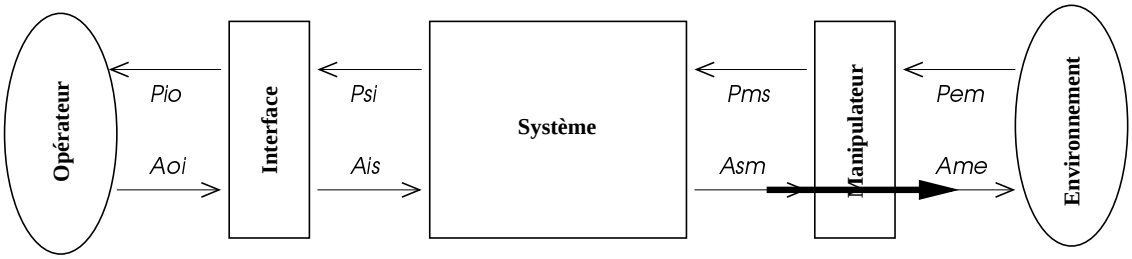

FIG. 4.3: Transmodalisation de l'action

4.4.2 Application

Dans la première partie de ce document, nous avons évoqué les problèmes de «transposition», bien connus en télé-opération : l'idée est par exemple que d'un côté, un robot effectue des opérations sur son environnement de travail, mais que de l'autre côté, le télé-opérateur à l'illusion de travailler manuellement et n'a pas conscience de commander un robot.

Dans une telle situation, il n'y a aucune raison *a priori* pour que le robot effectue les opérations de la même manière que l'opérateur. Dans le cas d'un bras manipulateur par exemple, la saisie d'un objet peut s'effectuer de manière «humaine», avec frottement physique entre l'objet et un organe de préhension, mais peut également être mise en œuvre grâce à un champ magnétique pour des objets métalliques (comme une grue manipulant des carcasses de voitures), ou par un système de dépression et de ventouse *etc*. Ce type d'assistance correspond bien à un cas de transmodalisation de l'action tel que nous l'avons décrit précédemment. En effet :

- De même que dans le cas précédent, l'information actionnelle affectée par ce type de traitement est fournie par l'opérateur : il s'agit de l'action de saisie, qui est spécifiée par un geste naturel grâce à un exosquelette couvrant le bras et la main.
- Contrairement au cas précédent, cette information n'a pas vocation à être altérée : le but est bien de saisir une carcasse donnée, comme l'a requis l'opérateur. Cependant, l'information en question n'est pas exécutée sous sa modalité d'origine. En effet, l'opérateur la spécifie en termes de saisie naturelle par frottements entre surfaces, mais son exécution passe par l'utilisation d'un champ magnétique et non d'un organe de préhension.

4.5 Reconstitution d'action

Dans cette section, nous procédons à la symétrisation du cas «Reconstitution de perception» décrit en page 78. La description synthétique de ce cas d'assistance peut être reformulée de la manière suivante :

Reformulation: Reconstitution d'action

Dans un cas d'assistance à l'opérateur par ***reconstitution d'action****, le système reconstitue une information actionnelle manquante chez l'opérateur, ou non spécifiable sur l'interface, à partir d'une information actionnelle de substitution.* ▼

4.5.1 Sémantique et localisation

D'un point de vue sémantique, les processus d'assistance décrits ici consistent cette fois-ci à fabriquer artificiellement une information actionnelle manquante. Ces processus sont caractérisés par trois aspects :

- Tout d'abord, l'information actionnelle souhaitée n'est pas initialement disponible. Soit l'opérateur ne peut la produire, soit l'interface ne peut la saisir.
- Cependant, il est possible de fabriquer artificiellement cette information actionnelle en utilisant une *autre* information actionnelle, qui elle, existe naturellement. Il s'agit donc d'une *reconstitution d'action*.
- Dans certains cas, il peut s'avérer nécessaire de disposer d'une information non actionnelle (non spécifiée par l'opérateur) en plus de l'information actionnelle de substitution.

En termes de localisation, la mise en place de processus d'assistance de ce type requiert un travail supplémentaire de commande ($A_{oi} \Longrightarrow A_{is}$), afin d'obtenir l'information actionnelle de substitution qui n'aurait pas été nécessaire autrement. D'autre part, cette information doit être traitée afin de reconstituer l'information actionnelle manquante (A_{sm}). La figure 4.4 ci-contre localise ces processus en indiquant (en gras) le chemin d'information affecté.

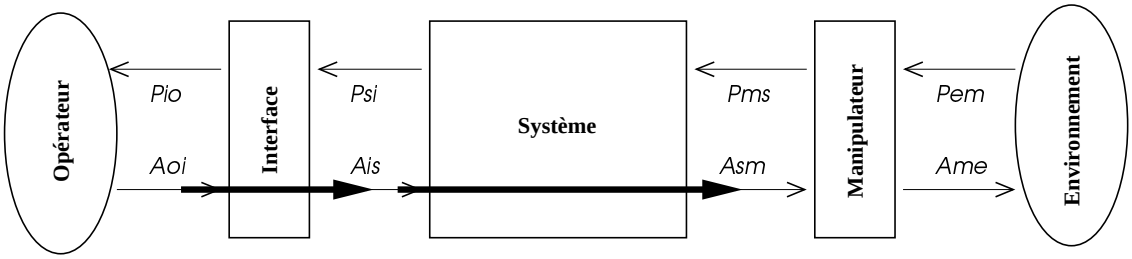

FIG. 4.4: Reconstitution de l'action

4.5.2 Application

Le cas d'assistance obtenu ici par symétrisation nous permet d'aboutir à un champ applicatif déjà largement développé : celui des systèmes utilisant des périphériques non immersifs pour le contrôle.

Pour commander le bras manipulateur de notre exemple, l'opérateur peut disposer de différents types de périphériques. L'utilisation d'un périphérique immersif comme un exosquelette ou un gant de données permet à l'opérateur d'agir de manière naturelle : pour produire un certain déplacement du bras manipulateur, l'opérateur exécute simplement le même mouvement avec son propre bras.

Dans une deuxième approche, l'opérateur peut également être dispensé du mouvement naturel et commander le déplacement du bras à partir d'un trackball. Dans ce cas, il spécifie plutôt un vecteur vitesse de déplacement pour l'extrémité du bras. Ce type de commande existe notamment en robotique depuis bien avant l'apparition des périphériques immersifs, et constitue en fait un cas d'assistance par reconstitution d'action, tel que nous l'avons décrit précédemment. En effet :

- L'information actionnelle souhaitée n'est pas initialement disponible. Dans notre exemple de commande par trackball, l'interface utilisée ne permet pas de saisir une telle information.
- Cependant, il est possible de fabriquer artificiellement cette information actionnelle en utilisant une *autre* information actionnelle de substitution : celle produite sur le trackball.
- Enfin, il est nécessaire dans ce cas de disposer d'une information non actionnelle en plus de l'information de substitution. Dans notre exemple, il est nécessaire de connaître la morphologie du bras manipulateur afin de reconstituer vitesse de déplacement de l'extrémité du bras par une séquence de commandes motrices sur les articulations.

Notons que réciproquement au cas de l'assistance par transmodalisation de la perception, l'information requise est manquante, soit par incapacité de l'interface à la saisir, soit par incapacité de l'opérateur à la spécifier. Dans notre exemple, il s'agit d'une incapacité de l'interface (le trackball) à saisir le mouvement naturel du bras. Il existe cependant de nombreux cas entrant dans la deuxième catégorie. Plus précisément, deux cas peuvent se présenter dans lesquels un opérateur sera dans l'incapacité de fournir l'information actionnelle souhaitée :

- Soit cette incapacité provient d'un handicap (Vanderheiden et Mendenhall, 1994). Par exemple, un handicapé moteur sera dans l'incapacité de contrôler naturellement un bras manipulateur. Ce premier cas est à mettre en parallèle avec le cas de l'assistance aux personnes aveugles, décrit dans la section «Transmodalisation de la perception» du chapitre précédent, en page 76.
- Soit le manipulateur peut ne pas être isomorphe au corps humain, ce qui rend la commande naturelle impossible. Ainsi, Sturman (1992) décrit l'exemple du pilotage d'une créature virtuelle qui a trop de degrés de liberté pour être contrôlée de manière naturelle. Le contrôle

de la locomotion de cette créature est effectué en mimant la marche avec les doigts de la main.

☛ Il est intéressant de constater que par une méthode systématique de symétrisation d'un cas d'assistance à l'opérateur utilisant des techniques de réalité virtuelle, nous retombons sur une notion déjà largement utilisée, mais dont le lien avec la réalité virtuelle n'avait pas pour autant été établi de manière explicite.

4.6 Exécution paramétrique

Dans cette section, nous procédons à la symétrisation du cas «Commande paramétrique» décrit en page 81. La description synthétique de ce cas d'assistance peut être reformulée de la manière suivante :

Reformulation: Exécution paramétrique

Dans un cas d'assistance à l'opérateur par ***exécution paramétrique****, le manipulateur est configuré de manière à prédéfinir certains paramètres de l'exécution.* ▼

4.6.1 Sémantique et localisation

D'un point de vue sémantique, les processus exploités dans ces exemples assistent l'opérateur au niveau de l'exécution des actions. Ces processus sont caractérisés par les points suivants :

- ➤ Un tel type d'assistance consiste en fait à configurer certains paramètres que le manipulateur utilisera pour l'action, d'où le terme d'*exécution paramétrique*.
- ➤ Dans le cas (réciproque) d'une assistance par commande paramétrique, certains paramètres de l'action sont positionnés avant d'être utilisés pour l'exécution. Symétriquement, une assistance par exécution paramétrique consiste donc à positionner certains paramètres pour l'exécution d'une action, avant de les restituer à l'opérateur.

En termes de localisation, la mise en place de processus d'assistance de ce type s'effectue au moyen d'une boucle d'information entre l'environnement et le manipulateur. La figure 4.5 localise ces processus en indiquant (en gras) le chemin d'information affecté.

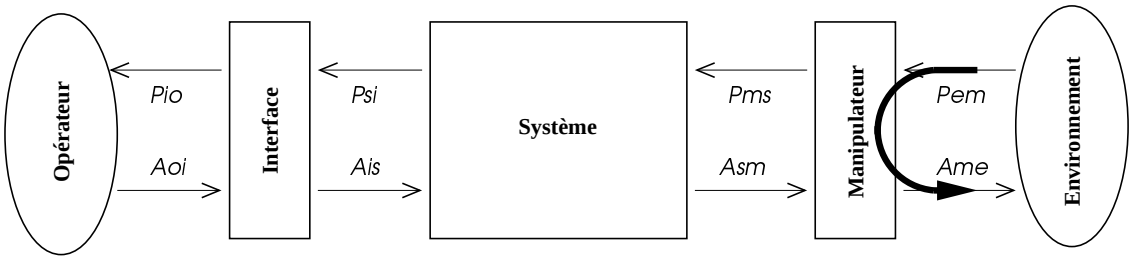

FIG. 4.5: Exécution paramétrique

4.6.2 Application

Parmi les cas d'assistance entrant dans cette catégorie, nous retrouvons par exemple la notion de contrôle d'effort : reprenons à nouveau l'exemple d'un bras manipulateur, chargé du déplacement d'objets, cette fois de différentes tailles et de différents poids.

Dans une version sans contrôle d'effort, l'opérateur devra lui-même gérer le dosage de la pression de saisie à appliquer, selon l'objet concerné. Cette situation impose donc l'utilisation d'un périphérique à retour d'effort. Dans une version avec contrôle d'effort, ce dosage s'effectue automatiquement, sans intervention de l'opérateur. Celui-ci ne dispose plus que d'une commande simple d'ouverture / fermeture de l'organe de préhension, l'ajustement de la pression de saisie étant à la charge du manipulateur.

Notons qu'un tel cas d'assistance est bien différent de l'assistance par génération d'action : un processus de génération d'action consiste à ajouter de l'information actionnelle ne provenant pas de l'opérateur. Ici, l'information actionnelle est la commande de saisie, qui est bien spécifiée par l'opérateur lui-même. Cet exemple d'assistance correspond bien à un processus d'exécution paramétrique, tel que nous l'avons décrit précédemment. En effet :

- Un tel type d'assistance consiste en fait à configurer un certain paramètre, la pression de saisie à appliquer, que le manipulateur utilisera pour l'action.
- Ce paramètre est configuré automatiquement (en mode adaptatif), et directement pour l'exécution de l'action, donc sans passer par une phase de commande par l'opérateur.

4.7 Filtrage de la perception

Dans cette section, nous procédons à la symétrisation du cas «Filtrage de l'action» décrit en page 80. La description synthétique de ce cas d'assistance peut être reformulée de la manière suivante :

Reformulation: Filtrage de la perception

*Dans un cas d'assistance à l'opérateur par **filtrage de la perception**, le système ne conserve qu'une partie de l'information perceptive acquise par le manipulateur, afin de restituer cette partie, non modifiée, à l'opérateur.* ▼

4.7.1 Sémantique et localisation

D'un point de vue sémantique, les processus d'assistance décrits ici consistent à traiter les perceptions acquises par le manipulateur. Les point suivants caractérisent ces processus :

- Le traitement opéré sur l'information perceptive consiste à n'en conserver qu'une certaine partie, sans modification de celle-ci. On peut donc parler de *filtrage de la perception*.
- Ce type d'assistance a également la particularité de traiter l'information perceptive *après* acquisition complète de celle-ci par le manipulateur : même si l'information concernée n'est finalement pas transmise à l'opérateur, elle a malgré tout déjà été acquise par le manipulateur.

En termes de localisation, la mise en œuvre de tels processus d'assistance implique un traitement de l'information perceptive après acquisition (P_{ms}) et avant restitution (P_{si}) par l'interface. Ils sont donc localisés au sein du système. La figure 4.6 en page suivante localise ces processus en indiquant (en gras) le chemin d'information affecté.

4.7.2 Application

Reprenons à nouveau le cas de la saisie avec contrôle d'effort décrite dans la section précédente. Dans cet exemple, nous voyons qu'il est inutile voire encombrant de restituer à l'opérateur,

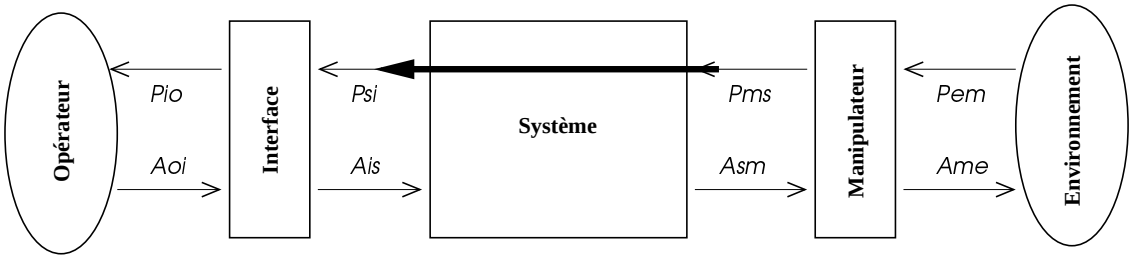

FIG. 4.6: Filtrage de la perception

par un retour d'effort, la pression appliquée à l'objet. En effet, en assistant l'opération de saisie par une configuration automatique de la pression, l'opérateur n'a plus besoin de spécifier celle-ci que par une commande binaire (ouverture / fermeture de l'organe de préhension), ce qui peut être accompli au moyen d'un dispositif très simple.

L'idée de ne pas restituer la pression correspond bien à un cas d'assistance par filtrage de la perception tel que nous l'avons décrit précédemment. En effet :

- Le traitement opéré sur l'information perceptive consiste à n'en conserver qu'une certaine partie, sans modification. L'information de pression est complètement filtrée du retour perceptif destiné à l'opérateur.
- Ce type d'assistance a également la particularité de traiter l'information perceptive *après* acquisition complète de celle-ci par le manipulateur. En effet, l'information de pression, bien que filtrée, est cependant acquise par le manipulateur car celui-ci en a besoin pour le dosage automatique.

4.8 Commande virtuelle

Dans cette section, nous procédons à la symétrisation du cas «Exécution virtuelle» décrit en page 83. La description synthétique de ce cas d'assistance peut être reformulée de la manière suivante :

Reformulation: Commande virtuelle

Dans un cas d'assistance à l'opérateur par ***commande virtuelle****, le système simule les processus de commande et de restitution au lieu de faire contrôler réellement l'opération concernée par l'opérateur.* ▼

4.8.1 Sémantique et localisation

D'un point de vue sémantique, les processus d'assistance exploités dans ces exemples consistent à simuler un opérateur en train de contrôler l'opération. Ces processus sont caractérisés par les points suivants :

- Dans une telle situation, l'environnement est mis temporairement en interaction avec une instance virtuelle du couple opérateur / interface et ce jusqu'à ce que l'opération soit exécutée. Nous sommes donc ici devant un cas d'assistance par *commande virtuelle* d'une opération, dans la mesure ou l'opération s'exécute comme si l'opérateur l'avait réellement dirigée.
- Afin de mener à bien cette commande virtuelle, il est nécessaire de disposer des connaissances suffisantes pour représenter non seulement l'opérateur et ses moyens d'action, mais également ses *réactions*.

En termes de localisation, l'opération se déroule normalement, comme si l'opérateur la contrôlait réellement. L'exécution en est donc inchangée. Réciproquement, le retour perceptif correspond à l'exécution normale. La phase d'acquisition est donc elle aussi identique. Il en résulte que de tels processus d'assistance interviennent au sein du système, en coupant le chemin normal d'information ($P_{ms} \Longrightarrow A_{sm}$). La figure 4.7 localise ces processus en indiquant (en gras) le chemin d'information affecté.

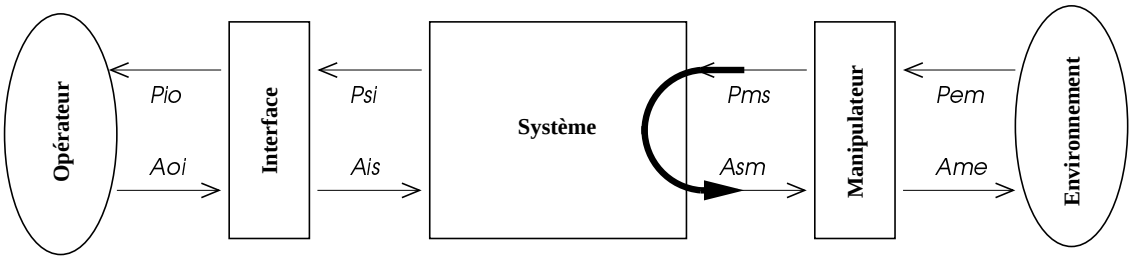

FIG. 4.7: Commande virtuelle

4.8.2 Application

Le cas d'assistance obtenu ici par symétrisation nous permet encore une fois d'aboutir à un champ applicatif déjà largement développé : celui des systèmes de robotique autonome.

Prenons l'exemple bien connu de l'évitement d'obstacle. Un robot autonome doit parvenir à une certaine position, éventuellement déterminée au préalable par un opérateur. Ce robot se déplace de manière autonome pour atteindre la position requise.

Le principe de l'évitement d'obstacle consiste à conférer au robot suffisamment d'intelligence pour pouvoir prendre des décisions relatives au changement de trajectoire face à un obstacle, y compris dans le cas ou cet obstacle n'existait pas au début du déplacement. Ce type de problème est largement abordé par les roboticiens et constitue en fait un cas d'assistance par commande virtuelle, tel que nous l'avons décrit précédemment. En effet :

- Dans une telle situation, le couple manipulateur / environnement est mis temporairement en interaction avec une instance virtuelle du couple opérateur / interface et ce jusqu'à ce que l'opération soit exécutée. En effet, le système a la charge de simuler les processus de détection d'obstacle et de changement consécutif de trajectoire qu'un opérateur réel aurait déclenché afin de contrôler le déplacement.
- D'autre part, afin de mener à bien cette commande virtuelle, le système doit également simuler les capacités de raisonnement et de réaction de l'opérateur, en particulier face aux situations imprévues, ce qui requiert une connaissance importante de l'opérateur et de son interaction naturelle avec l'environnement.

Ce type d'application de la robotique autonome entre plus précisément dans la quatrième catégorie, selon la classification de Giralt et al. (1993). Cette catégorie regroupe les robots autonomes programmables au niveau tâche. Contrairement à ceux de la troisième catégorie (robotique autonome purement réactive), ceux-là sont capables d'exécuter des opérations de manière autonome, après programmation de leur tâche par un opérateur, mais surtout en fonction de leur propre contexte perceptif. Cela signifie que le processus d'exécution d'une opération particulière pourra varier selon les événements et le contexte de travail courant, ce qui implique une certaine intelligence, et une certaine capacité de raisonnement face à l'imprévu, similaire à celle dont un opérateur humain ferait preuve.

☛ Nous constatons à nouveau ici que par une méthode systématique de symétrisation d'un cas d'assistance à l'opérateur utilisant des techniques de réalité virtuelle, nous retrouvons une notion déjà largement utilisée, mais dont le lien avec la réalité virtuelle n'avait pas pour autant été établi de manière explicite.

4.9 Résultats

La figure 4.8 résume l'ensemble des cas d'assistance que nous avons décrits dans ce chapitre et le précédent. Cette figure montre que notre procédé systématique de symétrisation nous permet d'aboutir à une vision plus équilibrée de la notion d'assistance à l'opérateur, affectant aussi bien l'information dans le sens des actions que dans celui des perceptions.

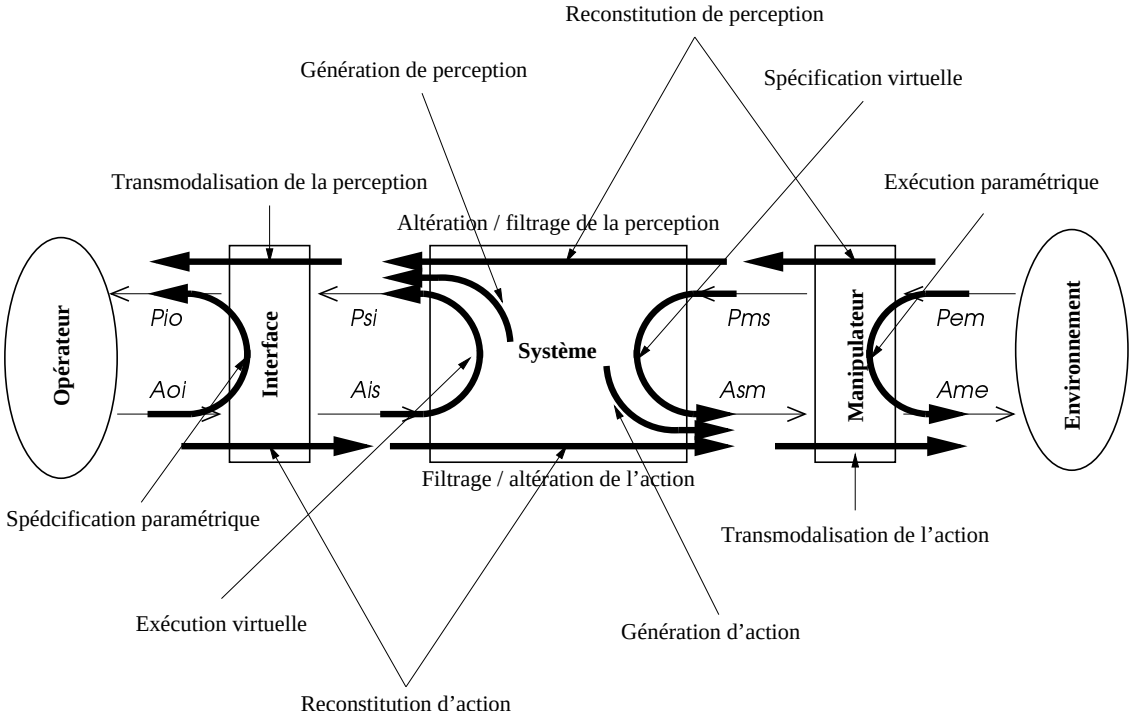

FIG. 4.8: Résumé global des d'assistance

4.9.1 Classes sémantiques d'assistance

Ce chapitre ainsi que le précédent ont fait apparaître sept classes sémantiques d'assistance à l'opérateur.

- ➤ Cinq d'entre elles affectent l'information en sens unique. Il s'agit des cas de génération, altération, reconstitution, filtrage et transmodalisation d'information. Chacune de ces cinq classes contient des cas intéressants d'assistance aussi bien dans le sens des actions que dans celui des perceptions.
- ➤ Deux d'entre elles affectent l'information en boucle. Il s'agit des cas de paramétrisation et de simulation. Ici encore, chacune de ces deux classes contient des cas intéressants d'assistance pour des boucles d'information orientées vers l'opérateur ou l'environnement.

L'intérêt de cette étude réside dans la démonstration du fait qu'une même classe sémantique de traitement de l'information peut s'exprimer dans les deux sens, et déboucher ainsi sur des cas réciproques d'assistance à l'opérateur, qu'*a priori* rien ne rassemble.

4.9.2 La réalité augmentée vue en sens inverse

Tous les cas d'assistance à la perception décrits dans le chapitre précédent ont des applications dans le domaine de la réalité augmentée. Pour la plupart d'entre eux, les exemples choisis en étaient justement des représentants. Force est de constater que l'activité «réalité augmentée » dans les applications d'aujourd'hui se focalise essentiellement sur l'augmentation des *perceptions* de l'utilisateur (Azuma, 1997). Mais peut-on parler de réalité augmentée au niveau des actions ?

Comme nous l'avons vu dans la première partie ainsi que dans le chapitre précédent, la vision actuelle de la réalité augmentée revient à mélanger les perceptions réelles avec des perceptions virtuelles, pour un opérateur en contact avec son environnement direct. Les développements effectués dans cette partie nous montrent que par symétrie, l'idée de mélanger les actions réelles avec des actions virtuelles pour un opérateur interagissant avec son propre environnement devrait également être considérée comme de la réalité augmentée. Si notre procédé de symétrisation a plutôt été illustré par des exemples relevant de la télé-opération, nous n'en pouvons pas moins l'utiliser pour décrire des cas de réalité augmentée par l'action.

L'idée du mélange entre actions réelles et actions virtuelles a déjà donné lieu à des applications intéressantes, notamment dans le domaine de l'assistance chirurgicale ou de l'aide aux personnes handicapées. Le projet Spartacus du CEA (Detriché et al., 1988) puis ses descendants Master 1 et 2 sont représentatifs de cette idée : par «génération d'action», le tétraplégique peut commander un bras manipulateur en mode semi-automatique dans lequel le système est en charge d'une certaine partie des commandes. Par «altération de l'action», on peut assister le télé-chirurgien ou le tétraplégique pilotant ce bras manipulateur dans leur commande de mouvement, par exemple en éliminant les tremblements ou les mouvements saccadés. Par «filtrage de l'action», on peut plus radicalement contraindre le mouvement sur certains degrés de liberté uniquement...

Il n'existe donc aucune différence conceptuelle entre ces exemples et ceux décrits dans le chapitre précédent. La seule distinction légitime est l'orientation (action ou perception) de l'information affectée par le processus d'assistance. De plus, cette distinction n'étant pas d'ordre sémantique, mais plutôt d'ordre «géographique», il en découle que ces nouveaux cas d'assistance doivent également être considérés comme des cas de réalité augmentée, plus précisément, de réalité augmentée par l'action. Si la réalité augmentée traditionnelle consiste à mélanger perception réelle et virtuelle, elle doit aussi consister à mélanger action réelle et virtuelle.

Chapitre 5

Niveau Conceptuel de l'Assistance

Les deux chapitres précédents ont montré qu'il est difficile de définir à proprement parler la notion d'assistance à l'opérateur. C'est une notion déjà largement répandue et dotée d'un sens commun non négligeable : assister l'opérateur, c'est soulager sa charge de travail, augmenter ses capacités, amoindrir ses défaillances. D'un point de vue cognitif, assister l'opérateur consisterait à lui donner l'impression qu'un deuxième humain, doté d'une intelligence et d'une connaissance similaires mais de capacités techniques supplémentaires, est là pour lui venir en aide.

Plutôt que d'en proposer une *définition*, nous préférons donc travailler sur les modes d'*expression* de la notion d'assistance : quels facteurs, quels indices révèlent la présence d'un processus d'assistance à l'opérateur à un instant donné. Dans les deux chapitres qui précèdent, nous avons quelques fois utilisé l'expression «système intelligent» ou «assistance intelligente». Il se trouve que l'étude des niveaux conceptuels de l'information circulant au travers du modèle ***L$_i$SA*** est un bon indicateur de la présence de processus d'assistance. C'est ce que nous nous proposons de décrire dans ce chapitre.

5.1 Analogies entre *M$_R$IC* et *L$_i$SA*

Les modèles ***M$_R$IC*** et ***L$_i$SA***, reproduits à nouveau sur la figure 5.1 en page suivante, traitent du même problème, l'interaction entre un opérateur et un environnement de travail, mais selon des axes différents. Bien que ce problème soit abordé sous un angle cognitif dans la première partie, puis sous un angle fonctionnel dans cette deuxième partie, il existe un rapport étroit entre les différentes composantes de ces deux modèles.

L'analogie la plus évidente est celle existant entre l'agent externe (selon ***M$_R$IC***) et l'environnement (selon ***L$_i$SA***). Il s'agit en fait d'une relation bien plus étroite que la simple analogie, dans la mesure où l'environnement est la représentation matérielle de l'agent externe dans une situation virtualisée.

Cependant, nous pouvons pousser cette relation entre ***M$_R$IC*** et ***L$_i$SA*** jusqu'à englober les trois agents du modèle ***M$_R$IC*** constituant l'opérateur, à savoir les agents mental, de transcription et opératoire :

- Dans une situation de télé-opération décrite par ***L$_i$SA***, c'est l'opérateur qui est doté de la connaissance, de l'intelligence et des processus de décision. Il joue donc le rôle que tient l'agent de contrôle dans le modèle ***M$_R$IC***.

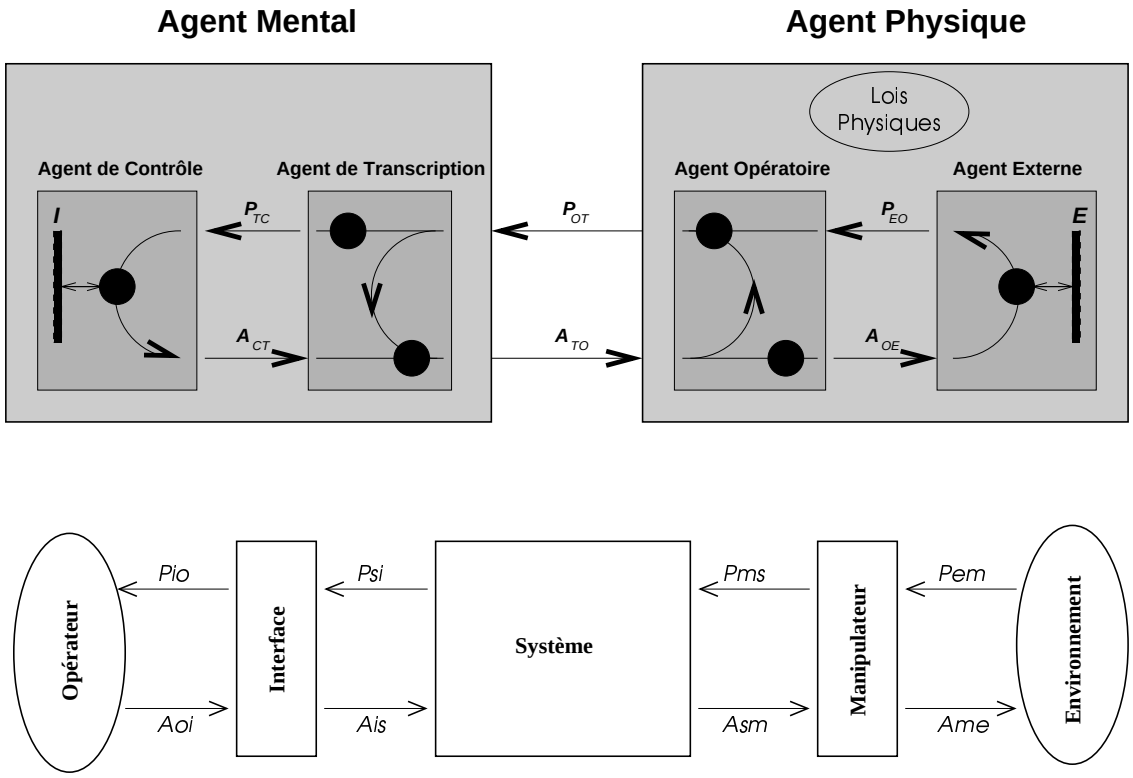

FIG. 5.1: Les modèles M_RIC et L_iSA

➤ Dans cette même situation, l'opérateur exprime ses intentions, génère des actions et acquière des perceptions par l'intermédiaire exclusif de l'interface. Cette interface traduit par conséquent des ordres exprimés intentionellement, en séquences d'information de plus bas niveau. Elle joue donc, pour l'opérateur, un rôle analogue à celui que tient l'agent de transcription dans le modèle M_RIC.

➤ Finalement, dans une situation de télé-opération correctement virtualisée, l'opérateur «oublie» qu'il utilise une interface de contrôle, et s'identifie physiquement au manipulateur qu'il dirige. Le manipulateur joue par conséquent le rôle de l'agent opératoire virtuel.

La figure 4.6 en page 33 nous a montré que le niveau conceptuel de l'information véhiculée au travers du modèle M_RIC décroît quand on se rapproche de l'environnement. Étant donnée la grande similitude existant entre nos deux modèles, il semble donc intéressant de situer le niveau conceptuel de l'information transitant au travers du modèle L_iSA par rapport à celui détaillé dans la première partie. En d'autre termes, la courbe de niveau représentée sur la figure 4.6 en page 33 constitue la courbe de *référence* (celle de la situation réelle) par rapport à laquelle nous allons maintenant positionner celles correspondantes à des situations virtualisées.

5.2 Télé-Opération Immersive

Reprenons le cas idéal d'une situation de télé-opération immersive : un opérateur doté de tous les périphériques d'immersion habituels, et un manipulateur totalement isomorphe au corps humain, en bijection permanente avec l'opérateur.

➤ Le niveau conceptuel de l'information circulant entre le manipulateur et l'environnement est identique à celui de l'information circulant entre les agents opératoire et externe du

modèle ***MRIC*** : il s'agit d'une information d'un niveau purement physique (par exemple, un mouvement de matière).

- ➤ Dans la mesure où l'interface est de type immersive, le système n'agit que comme un convoyeur d'information : il transporte par exemple la configuration spatiale de la combinaison de données telle quelle, de manière à appliquer une configuration identique au manipulateur. En cela, le niveau conceptuel de l'information n'est pas altéré en traversant le système, et il se trouve donc être équivalent à celui qui transite entre l'agent de transcription et l'agent opératoire (par exemple, des commandes motrices).
- ➤ Enfin, et toujours compte tenu du fait que l'interface est de type immersive, l'opérateur n'a aucun travail cognitif particulier à fournir pour exprimer ses intentions. Il agit de manière naturelle, et s'exprime donc à un niveau égal à celui de l'information émanant de l'agent de contrôle.

Il résulte de tout ceci que la courbe de niveau correspondant à une situation de télé-opération immersive est exactement la même que celle de la situation réelle décrite par ***MRIC*** (voir la figure 5.2).

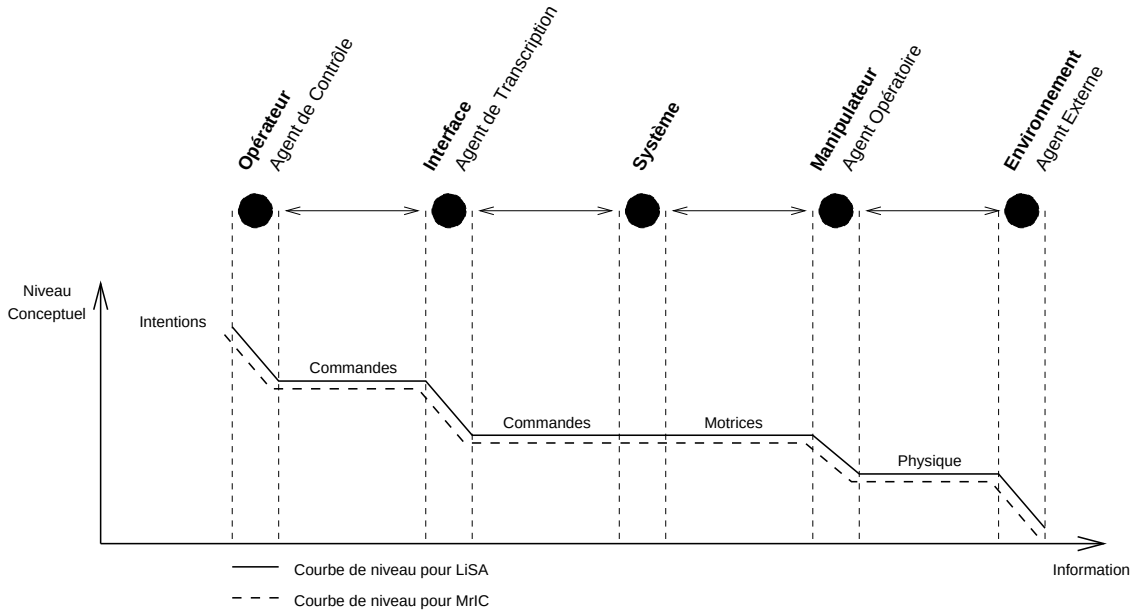

FIG. 5.2: Niveau conceptuel de l'information en télé-opération immersive

Ce résultat est conforme à notre attente, étant donné qu'une situation d'immersion idéale (que ce soit en environnement de synthèse ou en environnement réel) constitue le cas le plus réaliste de réalité virtuelle.

5.3 Télé-Opération «brute»

Considérons maintenant le cas opposé d'une situation de télé-opération immersive idéale : au lieu de disposer d'une interface immersive lui permettant de se comporter de manière naturelle, l'opérateur dispose de l'interface la plus primaire qui soit. Concrètement, au niveau de l'action par exemple, cela signifierait que l'opérateur dispose d'une commande distincte, par exemple un curseur, lui permettant de régler en temps réel la vitesse de chaque moteur du manipulateur. Le déplacement d'un bras articulé nécessiterait par exemple le contrôle simultané des vitesses de l'épaule, du coude et du poignet de ce bras.

Dans une telle situation, le niveau conceptuel de l'information transitant entre l'opérateur et l'interface est affecté : l'opérateur n'a plus la possibilité d'effectuer le geste naturel de mouvement du bras, mais est obligé de décomposer mentalement ce mouvement en une information de plus bas niveau correspondant au mouvement des articulations, afin de pouvoir reproduire ce mouvement sur le manipulateur. Or ce processus de décomposition a été décrit comme le rôle exact de l'agent de transcription dans la première partie : l'information de mouvement décomposé des articulations n'est pas normalement connue explicitement de l'opérateur, et celui-ci doit fournir un travail cognitif non négligeable pour l'expliciter. Il en résulte que le niveau conceptuel de l'information à destination de l'interface est cette fois-ci égal à celui de l'information émanant de l'agent de transcription. La courbe de niveau correspondant à cette situation est représentée sur la figure 5.3.

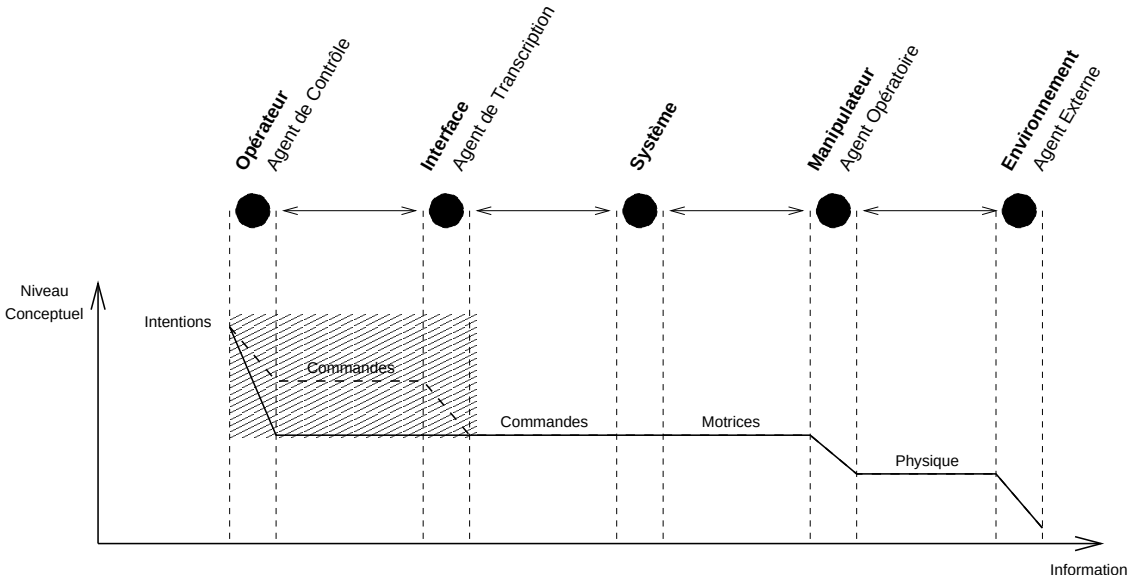

FIG. 5.3: Niveau conceptuel de l'information en télé-opération «brute»

La superposition de cette nouvelle courbe avec la courbe de référence fait apparaître un «puits conceptuel», représenté en grisé sur la figure. En d'autres termes, le niveau conceptuel de l'information concernée est *trop bas* par rapport à la situation de référence qui est l'interaction naturelle. Dans l'exemple de manipulation d'un bras robotique, ce puits conceptuel a pour conséquence d'obliger l'opérateur à expliciter mentalement des processus qui sont normalement à la charge de l'agent de transcription.

Ce phénomène est ce que l'on a coutume d'appeler un manque d'ergonomie de l'interface. Une interface ergonomique est en premier lieu une interface qui requiert le minimum d'apprentissage de la part de l'opérateur. Pour reprendre à nouveau l'exemple du bras robotique, il existe des degrés variables d'ergonomie pour une interface de contrôle : entre l'interface «brutale» et la combinaison de données ou l'exosquelette complet, on trouvera par exemple des joysticks permettant à l'opérateur de se concentrer sur la position de la pince robotique, laissant du même coup au système le soin de calculer les vitesses effectives à appliquer sur chaque articulation du bras. De telles interfaces tendent à se rapprocher de la manipulation naturelle, c'est à dire qu'elles soulagent l'opérateur en se chargeant du rôle de l'agent de transcription.

☛ La mise en évidence de ce puits conceptuel, ou plus exactement de la manière dont une interface ergonomique est censé le combler, est révélateur d'un premier niveau d'assistance à l'opérateur. À ce niveau réside l'assistance par reconstitution d'action, décrite dans le chapitre précédent. Contrairement à ce que l'on est tenté de croire, une interface de contrôle conviviale et simple à utiliser contient *déjà* des processus d'assistance à l'opérateur, dans la mesure où il y a modification du niveau conceptuel de l'information (par rapport à une situation «brute») pour le ramener à un niveau aussi proche que possible du niveau naturel. Même si la notion d'ergonomie des interfaces est abondamment débattue dans la communauté «Interfaces Homme-Machine», il nous semble intéressant de la mettre en évidence sous l'angle du niveau conceptuel de l'information.

5.4 Télé-Opération semi-autonome

Parallèlement à la télé-opération, une autre branche importante de la robotique se consacre à l'autonomie. La robotique autonome prend toute son importance lorsque qu'il est impossible de télé-opérer dans de bonnes conditions. À titre d'exemple, la télé-opération d'un robot d'exploration sur une autre planète (Lumia et Albus, 1988, Schroer, 1998, Schenker, 1991) pose des problèmes majeurs dans la mesure où les distances ne permettent pas une communication en temps réel. Il faut par exemple plusieurs minutes pour le transfert d'une image.

Face à ces contraintes, la télé-opération semi-autonome implémente une solution hybride (Giralt et al., 1993) dans laquelle une partie des compétences est transférée au manipulateur, ce qui permet à l'opérateur de ne transmettre que des ordres de plus haut niveau. Pour reprendre l'exemple d'un robot d'exploration et contrairement au cas d'une télé-opération totale qui nécessiterait un télé-guidage permanent, l'opérateur planifiera uniquement les grandes étapes du déplacement, laissant au système le soin d'exécuter les étapes de déplacement lui-même, grâce à des compétences proches de celles de l'opérateur, telles la capacité à éviter les obstacles.

Nous voyons que dans cette situation, l'information émanant de l'opérateur est de plus haut niveau conceptuel que la commande de déplacement correspondante : au lieu de diriger en temps réel le manipulateur, par exemple grâce à un joystick, l'opérateur n'a ici qu'à proférer des ordres de plus haut niveau du type «déplace-toi ici, puis là, puis là». En conséquence, il appartient au système de développer ces ordres comme l'aurait fait l'opérateur dans une situation de télé-opération directe. La courbe de niveau correspondant à ce nouveau type de télé-opération est illustrée par la figure 5.4 en page suivante.

Cette nouvelle courbe appelle deux remarques :

☛ À la différence du cas précédent, l'effet du processus d'assistance sur le niveau conceptuel de l'information n'est pas de *rétablir* celui-ci à un niveau acceptable, mais bien de le *surélever* par rapport au niveau naturel. L'interaction entre l'opérateur et le système se passe maintenant à un niveau conceptuel *supérieur* à celui d'une situation de télé-opération immersive classique. Nous pensons que dans ces conditions, il est justifié de parler d'«assistance intelligente» à l'opérateur, dans la mesure où pour la première fois, le système est doté de compétences se rapprochant de celle de l'agent de contrôle. Le critère d'«intelligence» est donc à associer avec la surélévation du niveau conceptuel de l'information, par rapport au niveau naturel.

☛ D'autre part, le fait que le niveau conceptuel d'une telle situation soit supérieur à celui d'une situation de télé-opération immersive classique n'est pas sans rappeler le débat concernant l'utilité de la sensation d'immersion dans la télé-opération. Bien au contraire, il tendrait à défendre l'idée qu'un système vraiment performant, et en particulier ne demandant que peu de travail à l'opérateur, n'est pas nécessairement un système immersif. Un système immersif tend à *restaurer* le niveau conceptuel naturel de l'information. Un système *intelligent* tend à le dépasser.

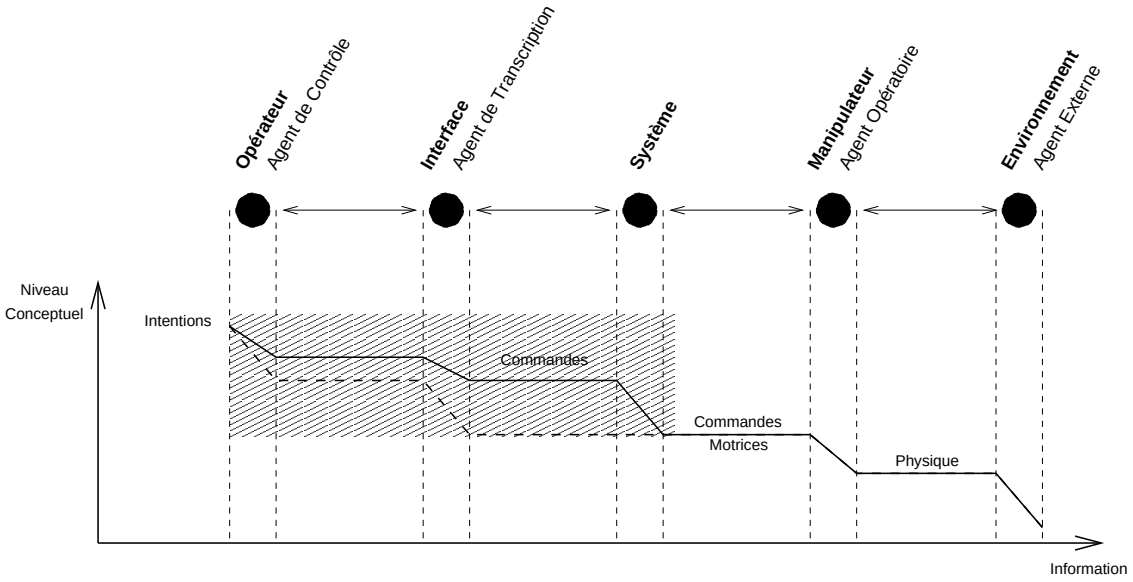

FIG. 5.4: Niveau conceptuel de l'information en télé-opération «semi-autonome»

5.5 Télé-Opération et Détection d'Intentions

Dans la première partie, nous avons constaté que le degré ultime de virtualisation consiste en une situation (fictive) où l'opérateur ne communique plus avec le système de réalité virtuelle qu'au travers de ses intentions. Du point de vue du niveau conceptuel de l'information, une telle situation implique que l'information atteignant le système soit du niveau du celle émanant de l'agent de contrôle. L'interface ne joue alors qu'un rôle de transport, dans la mesure où le niveau conceptuel de l'information est déjà maximal. Cette situation est illustrée par la figure 5.5 en page suivante.

Une telle courbe pourrait être approchée par un système de télé-opération basé sur une interface langagière. Le langage est vraisemblablement le moyen le plus naturel pour un humain d'exprimer ses intentions, quel que soit le manipulateur sous son contrôle. On imagine mal comment le travail de l'opérateur pourrait encore être diminué, dès lors qu'il se situe seulement au niveau de l'expression verbale de souhaits tels «je voudrais que cette fuite de radioactivité soit colmatée». Arriver à ce niveau conceptuel de communication, c'est à dire à un niveau égalitaire avec l'opérateur remet en fait en question la nécessité même de la présence de l'opérateur dans cette situation. De tels systèmes sont de toute manière encore du domaine de la conjecture.

Cependant, dans un objectif consistant à s'approcher du système optimal, tout en ayant conscience de ne jamais l'atteindre, la notion de détection d'intentions resurgit. Dans la première partie, nous avons perçut cette notion comme un moyen de se rapprocher de la virtualisation maximale (tous les agents virtuels, à l'exception de l'agent de contrôle). Ici encore, nous percevons cette idée comme un moyen de se rapprocher de la courbe de niveau optimale : les échanges entre l'opérateur et l'interface se font à des niveau de commande du même ordre que ceux impliqués dans une situation de télé-opération semi-autonome, mais le système, en fonction des commandes reçues est capable d'anticiper l'objectif de l'opérateur puis de le remplacer pour l'exécution. Si l'on reprend l'exemple du bras manipulateur, l'approche d'un objet (commandée par l'opérateur) permettrait au système de détecter une intention de saisie, puis de terminer le mouvement à la place de l'opérateur. Dans une telle situation, on voit bien que même si le niveau conceptuel de l'information transitant entre l'opérateur et l'interface reste d'un niveau inférieur à celui de l'intention, le système, quant à lui, doit *reconstruire* une information d'un tel niveau à partir des

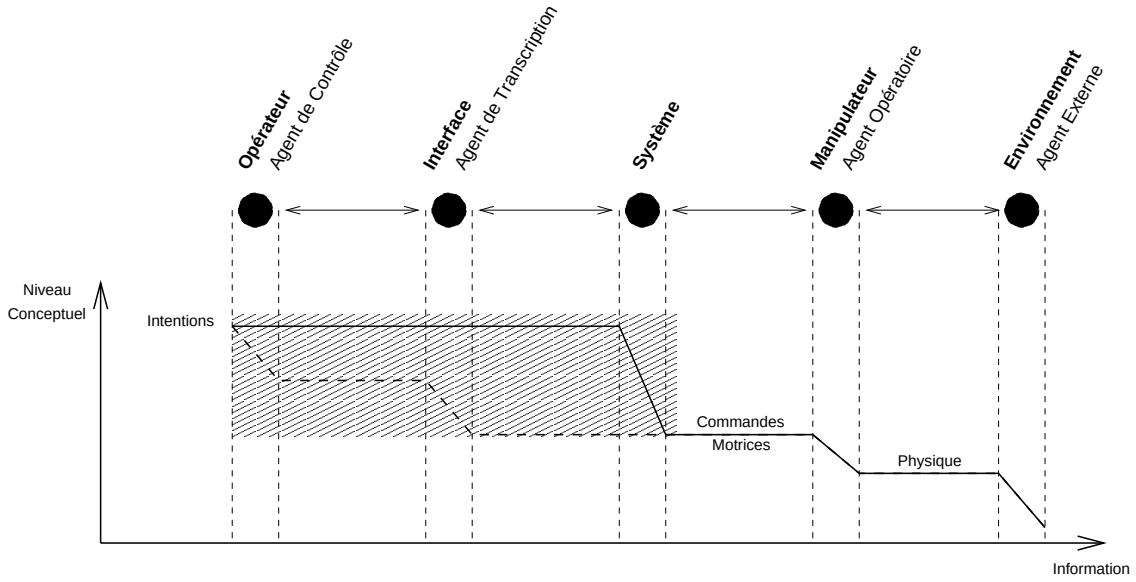

FIG. 5.5: Niveau conceptuel optimal de l'information en télé-opération

données dont il dispose. Cette nouvelle courbe est illustrée par la figure 5.6 en page suivante.

5.6 Résumé

Pour rassembler tous les aspects que venons d'aborder dans cette section, la figure 5.7 en page 109 propose une superposition de toutes les situations que nous venons d'envisager.

- La courbe de «niveau minimal» est celle d'une situation où l'opérateur n'aurait à sa disposition que les commandes de plus bas niveau.
- La courbe de «niveau naturel» est celle d'une situation purement immersive, ou correspondant à une interaction avec son environnement réel.
- La courbe de «niveau optimal» est celle correspondant à une situation dans laquelle l'opérateur n'aurait qu'à exprimer ses intentions.
- Les processus d'assistance à l'opérateur consistent à faire tendre la courbe de niveau conceptuel du système vers la courbe de niveau optimal.

Ces précisions faites, nous pouvons maintenant plus formellement décrire ce que nous entendons par «assistance» :

Vocabulaire: Assistance

Étant donnée une situation de télé-opération décrite par le modèle ***LiSA****, nous dirons qu'il y a* ***assistance*** *à l'opérateur lorsque la courbe de niveau conceptuel de l'information correspondant au système se trouve au dessus de la courbe de niveau minimal. De plus, nous parlerons d'****assistance intelligente*** *lorsque cette courbe se trouve également au dessus de la courbe de niveau naturel.* ●

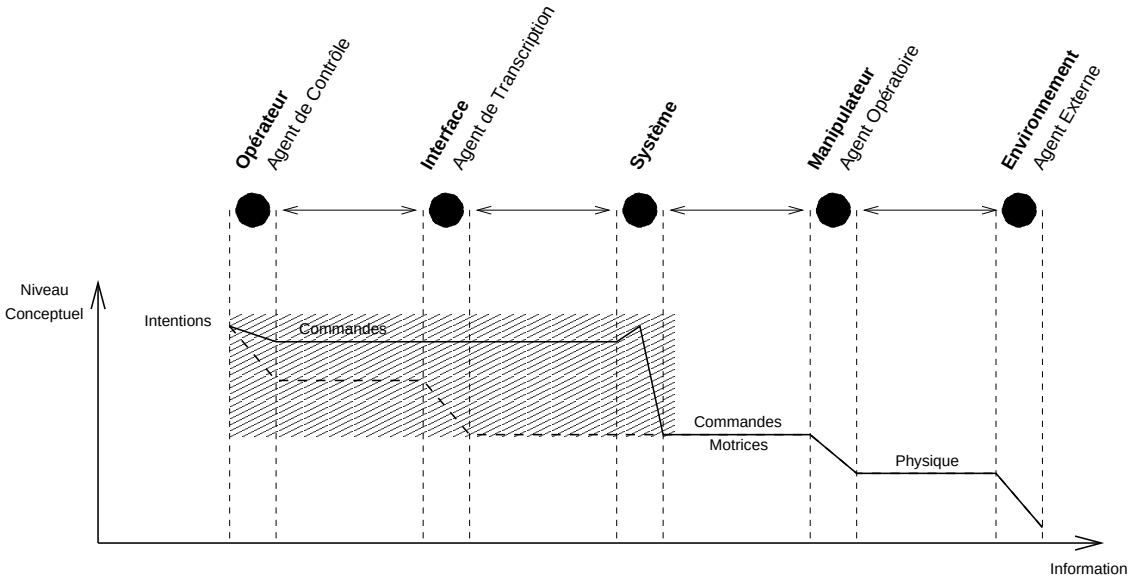

FIG. 5.6: Niveau conceptuel de l'information par détection d'intentions

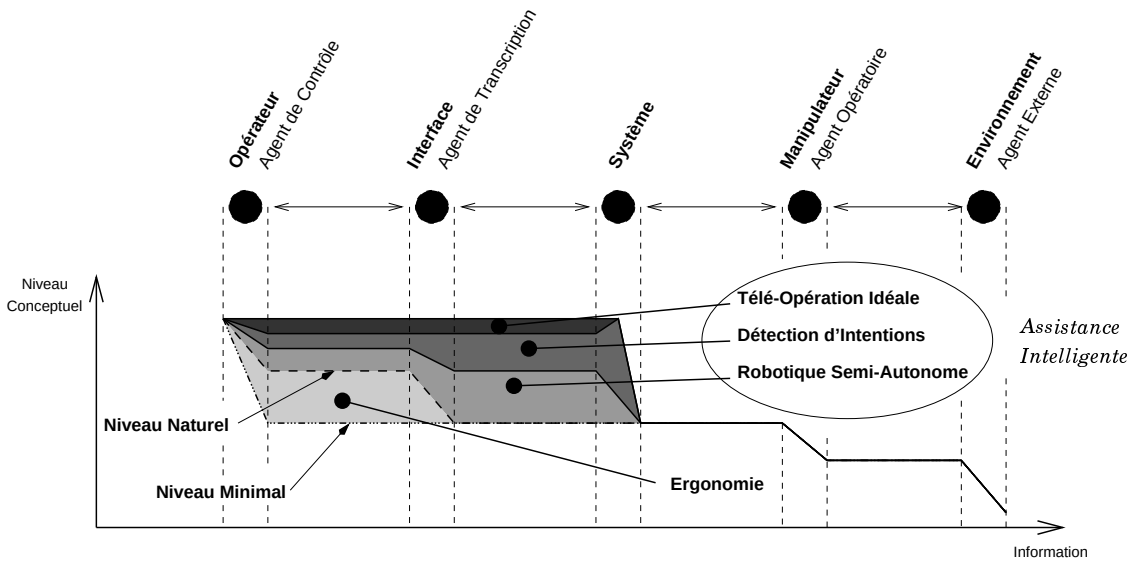

FIG. 5.7: Comparaison des niveaux conceptuels

Chapitre 6

Conclusion

«Didier Verna (rapport de thèse p.111)»
Didier Verna *(rapport de thèse p.111)*

Dans cette deuxième partie, nous avons tenté d'approfondir la notion d'assistance à l'opérateur, et plus particulièrement de déterminer en quoi les concepts tirés de la réalité virtuelle permettent de concevoir et d'implémenter des cas d'assistance particuliers.

Pour ce faire, nous avons tout d'abord élaboré un modèle décrivant les différentes composantes d'un système de télé-opération ou de réalité virtuelle, le modèle ***LiSA***. Nous avons ensuite exposé les cas d'assistance à l'opérateur les plus représentatifs de l'utilisation actuelle des concepts de réalité virtuelle dans l'interaction homme-machine. Ces cas d'assistance ont été analysés selon deux axes : un axe de *localisation* tout d'abord, c'est à dire à quel(s) endroit(s) du modèle ***LiSA*** l'information est affectée par le processus d'assistance, et un axe de *sémantique* ensuite, c'est à dire comment cette information est-elle affectée par le traitement qui lui est infligé.

Cette première analyse a fait ressortir un certain déséquilibre dans l'exploitation des concepts de réalité virtuelle pour la mise en œuvre de processus d'assistance : la grande majorité des cas traite uniquement de l'information perceptive et laisse de côté l'information actionnelle. Pour combler cette lacune, nous avons proposé une méthode systématique d'inversion des cas d'assistance, tenant compte de la grande symétrie de notre modèle. Ce procédé de symétrisation nous a permis d'aboutir méthodiquement à de nouvelles formulations de cas d'assistance, de même sémantique que les cas initiaux, mais orientés en sens inverse. Nous pensons que ce procédé présente trois bénéfices majeurs :

- Tout d'abord, il nous a permis d'unifier conceptuellement les domaines de la télé-opération et de la réalité virtuelle, initialement distincts. En effet, à partir de l'analyse de cas d'assistance à l'opérateur par des techniques de réalité virtuelle, nous avons dans certains cas retrouvés, en les symétrisant, des idées déjà largement exploitées en robotique. Nous avons par la même montré que ces idées, bien que n'ayant jamais jusqu'ici été associées à la réalité virtuelle, relèvent en fait des même concepts.
- Deuxièmement, il nous a permis de mettre en évidence les possibilités d'élargissement de certaines notions comme celle de réalité augmentée. Si la réalité augmentée désigne traditionnellement le fait de mélanger les perceptions réelles et virtuelles, notre analyse des cas d'assistance montre qu'elle devrait également désigner le fait de mélanger les actions réelles et virtuelles, dans la mesure ou la sémantique de ces procédés d'assistance est la même.

Cette étude a d'autre part fait apparaître plusieurs cas d'assistance décrits comme nécessitant une certaine «intelligence» de la part du système, notion devant par conséquent être précisée. Nous avons ainsi montré que l'étude du niveau conceptuel de l'information circulant au travers du modèle ***L$_i$SA*** est un bon révélateur de la présence de processus d'assistance à l'opérateur. Plus précisément, nous avons pu distinguer les cas d'assistance simple des cas d'assistance intelligente : un processus d'assistance simple a pour effet de ramener la courbe de niveau conceptuel de l'information à un niveau naturel, correspondant à une immersion totale, tandis qu'un processus d'assistance intelligente a pour effet de dépasser ce niveau conceptuel, conduisant ainsi le système à manipuler une information relevant normalement de l'agent de contrôle.

Assistance intelligente à la Télé-Opération : Détection d'Intentions

1 Introduction **115**
2 Généralités **117**
2.1 Capacités d'un système de détection d'intentions 117
2.2 Détecter, ou *ne pas* détecter ? 119
2.3 Résumé 121
3 Représentation des Connaissances **123**
3.1 Expérimentation 123
3.2 Antécédents 125
3.3 Structure Générique 127
3.4 Catégories d'action 133
3.5 Application à $TOAS_t$ 140
3.6 Formalisation 142
3.7 Caractéristiques du modèle 146
4 Détection d'Intentions **153**
4.1 Taux de vraisemblance d'une opération 153
4.2 Algorithmes de détection 157
4.3 Cas des actions terminales 161
4.4 Algorithme système 164
5 Exécution et corollaires **169**
5.1 Algorithmes d'exécution 169
5.2 Cas des actions terminales 172
5.3 Corollaires 173
5.4 Limitation 175
6 Un exemple de scénario **179**
7 Conclusion **189**

Chapitre 1

Introduction

«I was doing object oriented assembly at one year old.
For some reason, my mom insists on calling it "playing with blocks".»
Anonyme

Les réflexions menées dans chacune des deux parties précédentes nous ont à chaque fois conduit à envisager, sous des angles différents, le thème de la détection d'intentions. Ce thème est cependant apparu dans les deux cas comme l'aboutissement de la démarche théorique proposée. En effet :

- Dans la première partie, nous avons adopté un point de vue cognitif sur cette idée. La détection d'intentions est alors apparue comme un bon moyen de se rapprocher du degré ultime de virtualisation, c'est à dire celui où l'opérateur ne communiquerait plus que par l'esprit avec le monde virtuel.
- Dans la seconde partie, nous nous sommes focalisés sur l'influence des processus d'assistance à l'opérateur sur le niveau conceptuel de l'information. La détection d'intentions est alors apparue comme un bon moyen de se rapprocher du niveau conceptuel ultime de l'information, c'est à dire celui de l'information manipulée par l'agent de contrôle.

D'un point de vue opérationnel, le principe de la détection d'intentions, débouche sur une catégorie de systèmes de réalité virtuelle faisant montre d'un haut degré d'intelligence. En effet, dans le principe de la détection d'intentions est intrinsèquement contenu le fait qu'il existe un opérateur dans la situation considérée. Un système de détection d'intentions consiste alors à s'approcher au maximum du niveau d'intelligence de l'opérateur, sans pour cela priver celui-ci du contrôle global et permanent des opérations. Si le système disposait réellement d'une intelligence égale à celle l'opérateur, celui-ci n'aurait alors qu'à exprimer, par exemple en langage naturel, son but final, et le système pourrait alors s'exécuter de manière autonome.

Dans cette troisième et dernière partie, nous étudions le principe de la détection d'intentions dans ses aspects théoriques et opérationnels. Chaque aspect théorique de l'étude fait systématiquement l'objet d'une illustration par un exemple concret. Cette partie de notre travail de recherche a en particulier donné lieu à l'implémentation d'un démonstrateur appelé ***TOAST*** * (acronyme de **T**ele-**O**peration **A**ssistance **S**yst**e**m) sur lequel nous nous fonderons pour choisir les exemples. Ce démonstrateur simule la télé-opération d'un bras manipulateur doté de capacités de saisie, transport et pose d'objets.

Dans un premier temps, nous introduisons le thème de la détection d'intentions d'un point de vue général. Cette section s'attache à déterminer dans quel cadre l'application du concept de

*Le lecteur est invité à lire l'annexe 1 pour une description générale du démonstrateur.

détection d'intentions peut s'avérer utile, et quelles sont les fonctionalités importantes d'un tel système.

Dans un deuxième temps, nous présentons un modèle de représentation des connaissances servant à décrire les actions susceptibles d'êtres entreprises par l'opérateur. Les actions ainsi décrites sont connues du système, et peuvent par conséquent faire l'objet d'une détection.

Dans un troisième temps, nous élaborons un mécanisme de détection d'intentions exploitant le modèle de représentation des connaissances précédemment décrit. Le mécanisme proposé s'exprime sous la forme d'un ensemble de critères de détection se rapportant à la connaissance du système, ainsi que d'un jeu d'algorithmes mettant en œuvre ces critères.

Enfin, nous montrons que d'une part, grâce à la généricité du modèle de représentation des connaissances proposé, et d'autre part, grâce à l'élaboration du mécanisme de détection d'intentions précédent, nous pouvons, à un coût très faible, étendre les capacités de notre système à des fonctionalités telles que l'exécution ou la simulation des opérations que celui-ci est initialement capable de détecter.

Chapitre 2

Généralités

L'intérêt théorique du concept de détection d'intentions ne fait aucun doute : nous avons montré que ce concept débouche sur les systèmes de réalité virtuelle les plus proches du degré maximal de virtualisation, tout en restant réalisables techniquement parlant. Reste cependant à démontrer l'intérêt *opérationnel* de la détection d'intentions : est-il bénéfique sur un plan pratique de concevoir de tels systèmes, et si oui, dans quel contexte trouvent-ils leur utilité ? Ce chapitre s'attache à cerner le cadre applicatif dans lequel la détection d'intentions peut s'avérer opportune.

2.1 Capacités d'un système de détection d'intentions

La première question que nous devons nous poser est celle des capacités d'un éventuel système de détection d'intentions : quelles sont les fonctionalités qui rendent un tel système utile ? Quelles compétences est-il légitime d'en attendre ?

2.1.1 Le compromis asservissement / autonomie

Dans la première partie de cette étude, nous avons mis à jour certains problèmes auxquels les systèmes actuels de télé-opération sont confrontés :

- Face à la nécessité d'exécuter les opérations de manière précise et efficace, une première solution consiste à rendre ces systèmes immersifs : en plongeant l'opérateur dans l'environnement cible, on espère que celui-ci fournira un travail de meilleure qualité. Cependant, les systèmes immersifs sont aussi lourds à mettre en œuvre qu'à utiliser, et le manque de confort des périphériques d'immersion contrecarre les avantages de celle-ci.
- Une deuxième solution consiste alors à se tourner vers les systèmes autonomes : il s'agit alors de transférer une certaine quantité de compétences de l'opérateur vers le manipulateur. Celui-ci devient alors capable d'exécuter des opérations complexes (telle le déplacement avec évitement d'obstacles) et l'opérateur contrôle occasionellement les opérations par des directives de plus haut niveau.

Les deux parties précédentes ont eu pour corollaire de positionner le concept de détection d'intentions comme un compromis entre asservissement et autonomie : un tel système devrait pouvoir communiquer avec l'opérateur à un très haut niveau conceptuel (le plus proche possible du niveau des informations manipulées par l'agent opératoire), sans pour autant pouvoir se passer de lui. Un système de détection d'intentions doit donc être une application pratique de cette idée.

Nous devons alors nous demander ce que peut signifier, dans la pratique, l'idée de manipuler des concepts d'un niveau proche de ceux que manipule l'opérateur, sans pour cela tomber dans l'autonomie opératoire, c'est à dire en ayant un besoin simultané des compétences de l'opérateur et de celles du système. Afin de répondre à cette question, il est primordial d'avoir une vision précise de ce que nous entendons par «compétence de l'opérateur» et «compétence du système».

2.1.2 Répartition des compétences

Il existe une répartition naturelle des compétences entre l'homme et la machine. L'homme dispose de la connaissance et des compétences de haut niveau, ce qui lui confère prioritairement un rôle de décision et de contrôle. La machine, quant à elle, dispose beaucoup plus facilement de compétences techniques (rapidité d'exécution, précision…) ce qui lui confère prioritairement un rôle d'exécutant. Il est intéressant de constater que les deux axes de recherche que sont la télé-opération immersive d'une part, et l'autonomie d'autre part, consistent précisément à intervertir ces domaines de compétence :

- Comme nous l'avons vu à plusieurs reprises, l'intégration de la composante immersive aux systèmes de télé-opération a pour but essentiel de rendre l'opérateur plus efficace. Prenons pour exemple le positionnement spatial d'un bras manipulateur. En immergeant l'opérateur dans l'environnement de travail (vision en relief, possibilité de modifier le point de vue…) celui-ci pourra effectuer le mouvement requis d'une manière plus précise. Si la machine connaissait la position exacte souhaitée, elle pourrait de toute manière l'atteindre avec plus de précision que l'opérateur encombré de périphériques d'immersion. Il s'agit donc ici d'améliorer les capacités d'*exécution* de l'opérateur.
- L'autonomie, quant à elle, tend à se dispenser de la présence d'un opérateur. Prenons pour exemple le déplacement d'un robot mobile. Un tel robot, doté de suffisamment de compétences (pour éviter les obstacles imprévus, se déplacer sur un terrain inconnu…) n'aurait alors besoin que de la destination souhaitée pour exécuter l'opération requise. Si le robot était asservi, l'opérateur serait en charge de la trajectoire. Il s'agit donc ici d'améliorer les capacités de *contrôle* et de *décision* du manipulateur.

Si nous nous positionnons en termes de compromis entre asservissement et autonomnie, nous devons donc établir ce compromis sur le plan pratique, au niveau de la répartition des compétences entre l'opérateur et le système. Cela signifie que notre but n'est *ni* de doter le système de capacités de contrôle et de décision démesurées, *ni* de doter l'opérateur de compétences d'exécutant exceptionnelles. Établir un compromis signifie au contraire que l'opérateur et le système doivent conserver leurs compétences propres, tout en ayant suffisamment de compétences mutuelles pour pouvoir interagir.

2.1.3 Niveau conceptuel et complexité exécutoire

Dans une dernière étape de raisonnement, nous devons finalement nous demander comment achever ce compromis de compétences. Dans la deuxième partie de cette étude, nous avons abondamment parlé du niveau conceptuel des informations manipulées par l'opérateur et le système qu'il utilise. Il existe cependant un aspect de cette question qui n'a pas été abordé jusqu'ici, et dont l'éclaircissement permettra finalement de cerner correctement le cadre d'utilité applicative de la détection d'intentions. Ce point est celui des relations entre le niveau conceptuel d'une opération, et sa complexité exécutoire.

L'idée que nous voulons développer ici est que contrairement aux apparences, il est possible de découpler le niveau conceptuel d'une opération de sa difficulté d'exécution. Considérons le cas d'un robot mobile. À un niveau élémentaire, on trouvera des actions telles que le déplacement dans une direction donnée, ou en suivant un rayon de courbure constant. Ces actions sont d'un niveau conceptuel très bas, et ne sont pas difficiles à exécuter. Remontons à présent vers des actions de plus haut niveau, telles le déplacement à un endroit précis de l'environnement. Une telle

action se développe nécessairement comme une succession d'actions de niveau élémentaire. Cependant, sa complexité exécutoire est grande, dans la mesure ou le déplacement peut impliquer une planification de trajectoire complexe, avec évitement d'obstacles, passage par des chemins obligatoires etc.

De prime abord, il semble donc que plus la tâche est abstraite, plus son exécution est difficile. Cependant, quelle que soit l'action considérée, il est toujours possible de trouver des cas où son exécution est triviale. Par exemple, en l'absence d'obstacle, le déplacement d'un point à un autre peut s'effectuer simplement par une trajectoire en ligne droite. D'autre part, une action, même relativement complexe, finit toujours par être simple à exécuter. Par exemple, une fois que l'opérateur à évité tous les obstacles lors d'un déplacement, il existe toujours un moment où le déplacement en ligne droite suffit à terminer celle-ci. Nous voyons donc que la complexité exécutoire d'une opération n'est pas nécessairement proportionnelle à son niveau conceptuel, et mieux, qu'il existe toujours un contexte dans lequel une opération, même de très haut niveau conceptuel, peut se réduire à une exécution simple. C'est ici que si situe l'intérêt du le compromis asservissement / autonomie.

2.1.4 Application à $TOAS_t$

Imaginons donc un système de bras manipulateur capable de saisir, transporter et poser des objets. Supposons que ce système dispose de connaissances lui décrivant *de manière triviale* les opérations en question. Par exemple, l'opération de saisie est décrite comme une séquence d'ouverture de l'organe de préhension, positionnement à l'endroit de la saisie, puis fermeture de l'organe de préhension. Pour le système, le déplacement à l'endroit requis consiste simplement à s'en rapprocher en ligne droite. L'opérateur peut donc par exemple entamer un mouvement de saisie, éviter les éventuels obstacles et se positionner *à peu près* correctement. La proximité relative au but pousse alors le système à détecter l'intention de saisie et à terminer le mouvement. Un tel système est avantageux pour deux raisons :

- Tout ce qui relève de la complexité exécutoire (évitement d'obstacles. . .) reste sous le contrôle de l'opérateur, dont le système ne peut donc pas se passer. Cependant, ce contrôle ne constitue pas une tâche difficile pour lui. Automatiser ce type de contrôle de haut niveau est par contre un travail délicat relevant de l'autonomie.
- Au contraire, la terminaison des actions, au moment où leur exécution peut se dérouler de manière simple, est une tâche qui incombe au manipulateur. La simplicité d'exécution rend la mise en œuvre de ces actions aisée. Laisser cette tâche aux soins de l'opérateur est par contre gênant, car celui-ci aurait par exemple plus de problèmes qu'une machine à effectuer des mouvement nécessitant une grande précision.

☛ Nous voyons que dans une telle situation, chaque acteur (opérateur et système) garde ses compétences propres, mais ne peut se passer des compétences de l'autre. Nous ne sommes ni dans l'autonomie, ni dans l'asservissement. D'autre part, une telle situation est envisageable si la connaissance manipulée par le système est une connaissance de haut niveau, même si elle est exprimée de manière simpliste. Cela montre l'utilité du découplage entre niveau conceptuel et complexité exécutoire.

2.2 Détecter, ou *ne pas* détecter ?

Dans l'élaboration de notre système de détection d'intentions, la deuxième question que nous devons nous poser est la question de savoir ce que l'on doit (veut) détecter, et ce que l'on ne doit (veut) pas détecter. Cette question se décline selon plusieurs axes.

2.2.1 Hiérarchie des actions

Sans trop anticiper sur le prochain chapitre dans lequel nous élaborerons un modèle de représentation des connaissances, on imagine aisément que chaque action de l'opérateur est structurée en une succession d'actions de plus bas niveau pour lesquelles l'achèvement de l'une conditionne la mise en œuvre de la suivante. Par exemple, une action naturelle de saisie d'un objet passe par plusieurs étapes, en particulier le positionnement de la main autour de l'objet, puis la fermeture des doigts, *après* achèvement de l'étape précédente.

Face à cette structure hiérarchisée, nous devons nous demander à quel niveau doit se situer notre système de détection d'intentions : est-il raisonnable de détecter des actions de très bas niveau ? qu'est-ce qui détermine l'utilité d'une telle détection ?

- ➤ Un premier élément de réponse consiste à se remémorer un des aspects de la modélisation de l'opérateur introduite par le modèle ***MIP*R*IC*** dans la première partie. L'introduction de l'agent de transcription nous a permis de faire la distinction entre les actions explicitement requises par l'opérateur, et les développements implicites de ces actions en composantes réflexes. Cette distinction nous permet ici de justifier l'idée qu'il serait inutile, voire même encombrant de détecter des actions d'un niveau conceptuel inférieur à celui que l'agent de contrôle peut gérer. Un tel mécanisme obligerait en effet l'opérateur à prendre explicitement conscience de certaines étapes, normalement réflexes, d'une action, ce qui implique comme nous l'avons vu une charge cognitive importante.
- ➤ Cette limite inférieure à la détection est-elle suffisante, ou peut-elle s'avérer à son tour trop basse ? Considérons à nouveau l'exemple de la saisie d'un objet par un bras manipulateur. Cette saisie passe, comme nous l'avons dit, par une étape de positionnement de l'organe de préhension autour de l'objet en question. Cette étape, dans le cas naturel comme dans le cas télé-opéré est explicite : l'opérateur n'aurait aucun mal à la décrire. Cependant, pour un système de bras manipulateur dont le but serait exclusivement les fonctionalités de saisie, transport et pose, il apparaît vain de vouloir la détecter en elle-même, dans la mesure où cette étape n'a de sens que pour la saisie de l'objet. Dans ce cas, autant détecter immédiatement une intention de saisie. Cet exemple illustre le fait qu'une action peut être du domaine de l'explicite pour l'opérateur, sans pour cela qu'il soit nécessairement utile de la détecter.

Ces deux points montrent que le choix d'un ensemble de détections utiles est un problème ne pouvant faire l'objet d'une théorie générale, mais devant être traité au cas par cas. Étant donné une situation particulière, c'est à dire un type de manipulateur et l'ensemble des opérations qu'il est capable d'effectuer, le concepteur du système devra tout d'abord déterminer empiriquement l'ensemble des opérations relevant du domaine de l'explicite pour l'opérateur, et définir ensuite le sous-ensemble de ces opérations pour lesquelles une détection peut s'avérer utile.

2.2.2 Multiplicité de la détection

Une fois que l'ensemble des actions dont la détection peut s'avérer utile est défini, le problème inverse de cette sélection se pose : peut-il se produire simultanément plusieurs possibilités de détection, et que doit-on faire dans un tel cas ?

A priori, une telle situation est susceptible d'intervenir si la sélection précédente a laissé plusieurs possibilités de détection au sein d'une même hiérarchie. Reprenons l'exemple du bras manipulateur, et supposons que notre système soit capable de détecter des opérations de saisie, transport, et pose d'objets. Les opérations de saisie et de pose constituent deux étapes intermédiaires de l'opération de transport. Que doit-il donc se passer si une intention de saisie d'un objet est détectée ? Le transport de celui-ci doit-il être automatiquement détecté à son tour ?

- ➤ Il existe des cas pour lesquels une telle question est implicitement résolue : certains objets peuvent par exemple être saisissables mais non transportables. Une poignée de porte ou un levier de contrôle en sont des exemples. Dans un tel cas, la détection de l'intention de saisie n'entraînera jamais celle de l'intention de transport. Cependant, de telles situations

ne sont pas désambiguïsées par le système de détection d'intentions lui-même, mais par les connaissances dont il dispose. En effet, le système n'écarte pas l'hypothèse de transport parce qu'elle est peu probable, mais bien parce que ses connaissances lui indiquent que cette opération n'est *jamais* possible.

- Pour les cas où la détection multiple est possible, quel doit être le comportement du système ? Ici encore, il semble n'y avoir aucune réponse fixe à cette question, si ce n'est que la détection d'une action donnée ne doit pas nécessairement entraîner celle des actions dont elle constitue un intermédiaire. Par exemple, supposons une intention de saisie d'un objet détectée. L'intention de transport de celui-ci pourrait devenir plausible à condition que d'autres paramètres jouent en sa faveur, comme la présence d'un lieu de pose pertinent. Si au contraire, aucun autre élément que la saisie elle-même ne vient corroborer l'hypothèse du transport, il serait vain de la détecter.

Ces deux points montrent que dans les cas où la connaissance du système ne permet pas d'exclure la détection simultanée de plusieurs opérations, nous avons besoin de définir une «taux de vraisemblance» des actions, c'est à dire un indice déterminant à quel point chaque action possible est susceptible d'être effectivement entreprise par l'opérateur. Une action devient alors plausible si son taux de vraisemblance dépasse un certain seuil, et dans le cas où plusieurs actions seraient détectées simultanément, chaque hypothèse pourrait alors être examinée par ordre décroissant de vraisemblance.

2.3 Résumé

Nous proposons de résumer les considérations établies dans ce chapitre, par l'ensemble des points suivants :

- Le concept de détection d'intentions a été positionné en termes de compromis asservissement / autonomie dans les deux parties précédentes. D'un point de vue applicatif, ce positionnement s'exprime comme un compromis dans la répartition des compétences de l'opérateur et du système : l'opérateur doit, dans une certaine mesure, s'adapter au système en détaillant les actions qu'il souhaite entreprendre. Réciproquement, le système doit, dans une certaine mesure, s'adapter à l'opérateur, en manipulant des concepts de plus haut niveau que les seuls mouvements simples dont le manipulateur est capable.

- Cette répartition des compétences peut être concrétisée en dotant le système d'un modèle de représentation des connaissances lui permettant de manipuler des concepts de haut niveau, mais représentés d'une manière simpliste. Cette représentation permet au système d'exécuter avec succès les actions qui lui sont connues dans des cas ne présentant aucune difficulté majeure de raisonnement. Ces difficultés restent sous le contrôle de l'opérateur, qui se trouve par contre déchargé des problèmes que le manipulateur saura mieux résoudre, comme les problèmes de précision dans les mouvements.

- Afin d'inculquer au système la connaissance nécessaire à la mise en œuvre d'un mécanisme de détection d'intentions, une étude au cas par cas s'avère nécessaire afin d'éliminer de la détection les actions de trop bas niveau (que l'opérateur effectue normalement sans en avoir conscience) d'une part, puis de déterminer le sous ensemble des actions pour lesquelles la détection peut s'avérer utile d'autre part.

- Enfin, la détection d'intentions ne peut s'exprimer comme une fonction booléenne de la connaissance du système. Nous avons besoin d'une mesure continue de la plausibilité de chaque action connue du système.

Chapitre 3

Représentation des Connaissances

La première étape concrète dans l'élaboration d'un système de détection d'intentions consiste à établir l'ensemble des connaissances nécessaires à son interaction avec l'opérateur. Ces connaissances doivent être représentées au sein du système sous une forme propice à leur utilisation. L'objet de ce chapitre est de proposer un modèle générique de représentation de ces connaissances.

☛ Avant d'entrer en détail dans l'élaboration de ce modèle, précisons que les connaissances que nous nous attachons à modéliser dans ce chapitre sont exclusivement les connaissances relatives aux *actions* que l'opérateur est susceptible de vouloir effectuer. Dans le cas de $TOAS_t$, nous nous intéresserons par exemple à la manière de représenter l'action de saisie d'un objet, mais pas à la manière de représenter l'objet lui-même. Les problèmes de représentation ou d'acquisition de données perceptives (analyse de scène...) ne sont pas du ressort de cette étude et sont supposés résolus. Le démonstrateur $TOAS_t$ a été réalisé sous forme de simulation dans cette optique.

3.1 Expérimentation

Afin de nous aider dans l'élaboration du modèle de représentation des connaissances, une expérience a été réalisée. Le but de cette expérience était de recueillir des données dont on pouvait s'inspirer pour la conception du modèle.

3.1.1 L'expérience

L'expérience a été réalisée sur une populations de douze sujets. Chaque sujet était assis devant un bureau sur lequel étaient disposés un verre vide et une bouteille d'eau pleine, fermée, tous deux à portée de main. Les sujets ne connaissaient pas le but de l'expérience.

Dans une première phase, l'expérimentateur donnait au sujet la consigne : «Buvez un peu d'eau, en n'utilisant qu'un seul bras», et lui laissait toute liberté concernant la manière de s'y prendre.

Dans une deuxième phase, l'expérimentateur donnait au sujet la consigne : «Décrivez verbalement ce que vous venez de faire». Les sujets décrivaient alors une suite d'étapes par lesquelles ils étaient passés. Leur description était enregistrée.

Dans une troisième phase, l'expérimentateur leur faisait prendre conscience de cette structuration en étapes, puis leur donnait une nouvelle consigne : «Détaillez chacune des étapes explicitées». Pour chaque sujet ce processus était reproduit jusqu'à ce que le niveau de détail obtenu paraisse suffisant à l'expérimentateur, ou que le sujet déclare ne pas voir comment il était possible de détailler plus l'étape considérée.

3.1.2 Données recueillies

Nous proposons ici quelques extraits des verbalisations recueillies auprès des sujets de l'expérience. L'utilisation de l'ensemble de ces données sera décrite dans deux paragraphes ultérieurs.

Sujet A

Première étape de verbalisation :

> *«Il a fallu que je prenne la bouteille, pour ça, j'ai utilisé le bras droit. Ensuite, j'ai penché la bouteille jusqu'à ce que le liquide qui se trouve dedans puisse couler dans le verre, j'ai ensuite reposé la bouteille, avant de prendre le verre, j'ai ensuite porté le verre jusqu'à mes lèvres, je l'ai incliné jusqu'à ce que le liquide coule dans ma bouche, j'ai bu plusieurs gorgées d'un trait, ensuite, j'ai reposé le verre.»*

Après avoir demandé des détails sur la saisie de la bouteille :

> *«Sachant que j'ai essayé d'être précis... J'ai soulevé mon bras, ma main était déjà partiellement ouverte, mais je l'ai ouverte d'avantage, avant de saisir la bouteille.*

Après avoir fait remarqué au sujet que le mouvement du bras ne s'arrêtait pas au simple fait de le soulever :

> *«Ah, oui, je déplace le bras jusqu'au contact, avant de refermer la main.»*

Après avoir demandé des détails sur la saisie du verre :

> *«J'ai agi à peu près de la même façon que pour la bouteille : j'ai dirigé le bras vers le verre, les doigts se sont écartés et ils se sont refermés sur le verre. Ensuite, j'ai levé le verre.»*

Sujet B

Première étape de verbalisation :

> *«J'ai attrapé la bouteille, je l'ai inclinée en ayant soin de ne pas renverser le verre, puis après, j'ai reposé la bouteille, puis j'ai pris le verre, puis, je l'ai bu, et j'ai reposé le verre.».*

Après avoir demandé des détails sur la saisie de la bouteille :

> *«Ben, j'ai mis ma main dessus, puis je l'ai levée.»*

Après demande de détails supplémentaires :

> *«J'ai déplié mon bras, puis j'ai attrapé la bouteille en dessous du goulot... la main était ouverte... enfin, je veux dire, dans le même mouvement de déplacement.»*

Sujet C

Première étape de verbalisation :

> *«Avec mon bras droit, j'ai pris la bouteille, j'ai mis de l'eau dans le verre, je l'ai remise à la position initiale (je pense que j'ai dû faire le même mouvement que pour la prendre), j'ai pris le verre, j'ai bu, puis je l'ai reposé.»*

Après avoir demandé des détails sur la saisie de la bouteille :

«J'ai avancé mon bras vers la bouteille, j'ai saisi la bouteille entre mon pouce et les autres doigts, j'ai soulevé la bouteille.»

Après demande de détails supplémentaires :

«Mes doigts avaient déjà la forme de la bouteille, la main était déjà prête à saisir quelque chose, elle était déjà ouverte.»

Sujet D

Première étape de verbalisation :

«J'ai saisi la bouteille avec la main droite, ensuite j'ai levé la bouteille, ensuite j'ai versé un verre d'eau, j'ai reposé la bouteille, ensuite je me suis saisi du verre, dans la main, donc, ensuite j'ai bu, enfin, je l'ai porté aux lèvres, pardon, et j'ai bu, et ensuite j'ai reposé le verre sur la table.»

Après avoir demandé des détails sur la saisie de la bouteille :

«J'ai déplacé mon bras en direction de la bouteille, et là, seulement, j'ai saisi la bouteille.»

Après demande de détails supplémentaires :

«Ce qui se passe, c'est que déjà, la main se prépare à saisir la bouteille, au préalable.»

Puis, en récapitulant, le sujet se ravise :

«Enfin, ça se déclenche au début du mouvement, mais ça s'effectue en même temps. Il y avait un mouvement du bras, et en même temps, la main qui est un peu indépendante.»

Sujet E

Une verbalisation de l'action de saisie de la bouteille un peu différente :

«Bouger le bras, puis saisir. On peut décomposer, c'est à dire qu'on sépare le pouce des autres doigts, et on serre avec le pouce d'un côté et les autres doigts de l'autre.»

Puis, après récapitulation :

«Pivoter le bras, préparer la main, peu être un petit peu avant ?... Si je le refais, je vois plutôt que non. Je pense que c'est au dernier moment.»

3.2 Antécédents

D'un point de vue général, la représentation des connaissance est un problème ancien et abordé par de nombreuses disciplines (sciences cognitives, intelligence artificielle, robotique...). Les problèmes liés à la représentation des connaissances d'action sont cependant apparus plus récemment : on évalue l'aube des travaux portant explicitement sur cette question à ceux de Norman et Rumelhart (1975) en sciences cognitives, et à ceux de Sacerdoti (1977), Van Lhen et Brown (1980) en intelligence artificielle.

3.2.1 Composantes d'une action

Traditionnellement, une action est décrite en trois composantes essentielles. Ces composantes sont les *prérequis* de l'action, son *déroulement* et son *résultat*.

- Le résultat d'une action est une composante déclarative de celle-ci : il exprime un changement d'état consécutif à l'exécution de l'action, ou décrit l'état résultant de cette exécution. À l'inverse, le déroulement d'une action est une composante procédurale de celle-ci : il propose une méthode pour parvenir au but souhaité.
- Il existe deux raisons duales ayant en particulier poussé les chercheurs en intelligence artificielle et planification à distinguer le déroulement d'une action de son but : premièrement, certaines actions sont décrites par leur but plutôt que par leur déroulement, ce qui signifie que plusieurs actions peuvent conduire au même but. Par exemple, «chauffer la pièce» constitue un but qui peut être atteint soit en allumant un radiateur, soit en faisant du feu dans la cheminée. Réciproquement, si deux actions peuvent conduire au même but, une seule action peut avoir des buts différents : par exemple, «faire du feu dans la cheminée» est une action dont le but peut être de chauffer la pièce, mais aussi de brûler des vieux journaux.
- Les prérequis d'une action expriment un ensemble de conditions qui doivent être vérifiées pour qu'une action puisse s'exécuter. Par exemple, pour transporter un objet il est nécessaire que cet objet soit préalablement saisi. Avant d'effectuer le transport, on doit donc tout d'abord s'assurer que ce prérequis est vérifié, et le cas échéant, exécuter une action permettant de le satisfaire.

3.2.2 Formalisme de représentation

Richard (1990) décrit un mode de représentation des action sous la forme de réseau procédural, classiquement utilisé en sciences cognitives. Un réseau procédural est un graphe orienté représentant l'ensemble des sous-buts à réaliser pour atteindre un objectif particulier. Un tel graphe a pour sommet principal l'objectif en question, et se développe en sous-buts devant être atteint séquentiellement de bas en haut et de gauche à droite. Ainsi, une opération de déplacement d'un objet peut être représentée sous la forme d'un réseau procédural, comme illustré par la figure 3.1. L'arborescence indique par exemple qu'avant de pouvoir prendre l'objet X en Y, il est nécessaire,

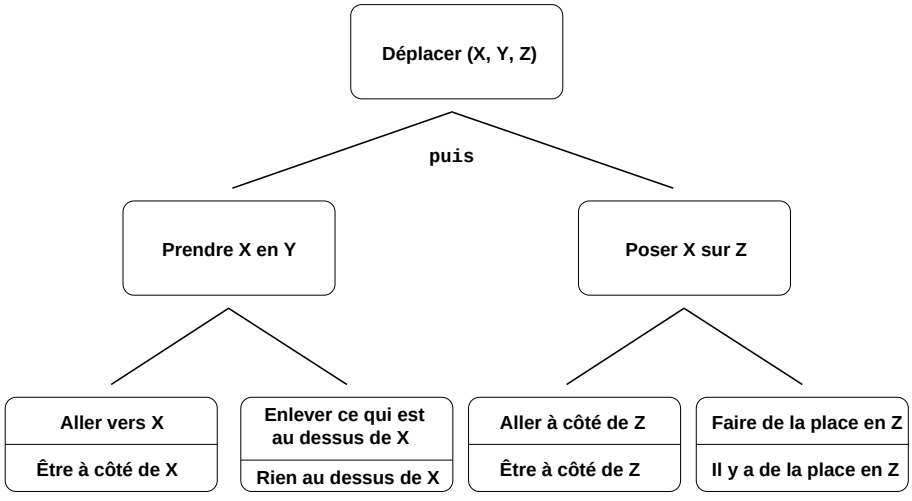

FIG. 3.1: Exemple de réseau procédural, d'après Richard (1990)

premièrement d'être à côté de X, et deuxièmement qu'il n'y ait rien au dessus de X. Les cases

contenant des prérequis contiennent également l'action nécessaire pour satisfaire le prérequis en question.

3.3 Structure Générique

Dans cette section, nous présentons la structure générique de notre modèle de représentation des connaissances, c'est à dire les mécanismes permettant d'articuler les connaissances entre elles. Cette structure est illustrée au travers de l'exemple de la saisie d'un objet.

3.3.1 Représentation arborescente des actions

Données expérimentales

En nous intéressant particulièrement, dans l'expérience décrite précédemment, à la verbalisation des étapes d'action (certains sujets ont également décrit des étapes mentales ou perceptives, comme la recherche visuelle du verre), les points suivants sont à noter :

- Lors de la première phase de verbalisation, les sujets ont explicité une séquence de quatre à six étapes. Deux sujets seulement ont dépassé ce nombre et ont atteint huit étapes dès la première phase (les sujets A et D). Un de ces deux sujets a spontanément déclaré avoir «fait un effort de précision» (le sujet A).
- Les sujets ont tous verbalisé leurs actions en tant que séquence ordonnée de sous-actions. 50% d'entre eux ont également décrit spontanément certaines actions comme une composition de sous-actions exécutées en parallèle (en particulier les sujets B et D).
- Lors de la phase d'explicitation des différentes étapes du mouvement, tous les sujets ont spontanément décrit la première action de saisie, puis les autres comme «le même mouvement» que la première.

Modes de parcours

Ces résultats nous poussent à utiliser, comme dans le cas des réseaux procéduraux, une représentation des actions sous forme d'arbre orienté. Ils laissent d'autre part supposer qu'il serait bénéfique de préciser le mode de parcours, séquentiel ou parallèle, des branches de cet arbre. Notre modèle de représentation des connaissances suit cette idée :

- Un mode de parcours séquentiel indique que la résolution d'une action passe par la résolution *successive* de ses sous-actions : une sous-action doit être résolue avant de pouvoir passer à la suivante.
- Un mode de parcours parallèle indique que la résolution d'une action passe par la résolution de ses sous-actions *dans un ordre quelconque* : chaque résolution de sous-action est indépendante des autres.

Représentation graphique

Afin d'illustrer le développement de certaines actions selon ce modèle, nous adopterons dans tout le reste de cette partie une représentation graphique de l'arbre développé de haut en bas. De plus, afin de faciliter la lecture des schémas, les sous-actions constituant le développement de l'action considérée seront séparées selon les cas par les signes «→» ou «//», comme illustré sur la figure 3.2 en page suivante.

☛ Notons que pour le cas d'un développement séquentiel, l'ordre d'apparition des sous-actions a une importance, et est orienté de gauche à droite. Pour le cas d'un développement parallèle, cet ordre est quelconque.

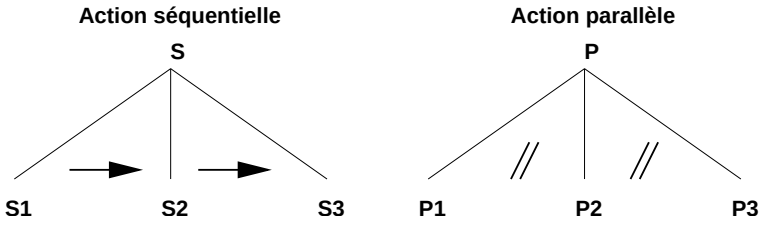

FIG. 3.2: Représentation graphique des modes de parcours

3.3.2 Formalisation syntaxique préliminaire

Unicité du mode de parcours

D'un point de vue théorique, il n'y a aucune raison pour réduire chaque action à un seul mode de parcours : une action pourrait être décomposée en sous-actions dont certaines seraient à effectuer en parallèle, et d'autres en séquence. Certains sujets de notre expérience ont d'ailleurs décrit l'action de «saisir le verre» par «déplier le bras tout en ouvrant la main, puis prendre le verre», ce qui correspond à cette idée : les trois sous-actions sont mises sur le même plan, mais deux d'entre elles sont exécutées en parallèle, avant la troisième.

Par soucis de simplicité et de lisibilité, nous préférons conserver l'unicité du mode de parcours pour chaque action. Ceci est rendu possible par la simple introduction d'une sous-action intermédiaire dans la description de l'action concernée, comme illustré sur la figure 3.3.

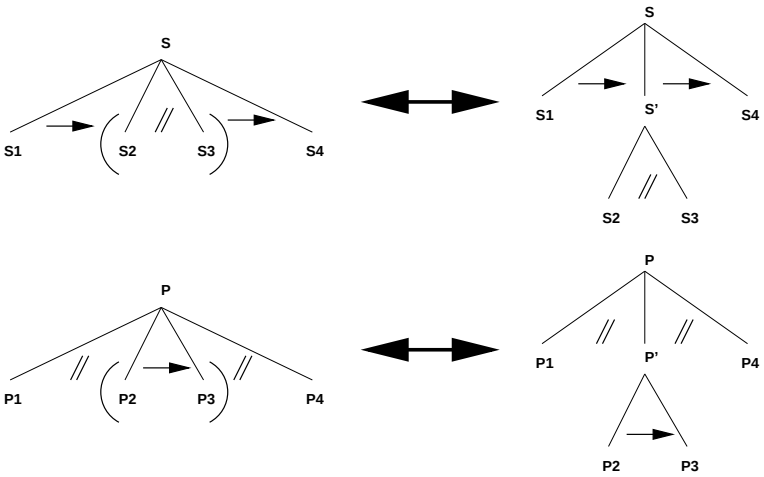

FIG. 3.3: Unicité du mode de parcours des actions

BNF préliminaire de la syntaxe

De cette manière, nous pouvons considérer que les actions ont un mode de parcours unique, valable pour toutes les branches de leur développement. Nous pouvons donc fournir une description syntaxique formelle préliminaire de notre modèle de représentation des connaissances, selon le format BNF suivant (Cf. glossaire en page 215) :

```
A ::= S | P | a
S ::= A [-> A] ... [-> A]
P ::= A [// A] ... [// A]
```

- ➤ A représente une action, qui peut être parallèle ou séquentielle. La notation a dénote une action qui ne possède aucun développement en sous-actions. Ces actions particulières constituent les feuilles de l'arbre, nous reviendrons ultérieurement sur leur définition.
- ➤ Une action séquentielle, S, est une suite ordonnée d'une ou plusieurs sous-actions, tandis qu'une action parallèle, P, est un ensemble non ordonné de sous-actions.

À titre d'exemple, les deux actions S et P représentées sur la figure 3.3 ci-contre sont décrites de la manière suivante :

```
S  ::= S1 -> S' -> S4
S' ::= S2 // S3

P  ::= P1 // P' // P4
P' ::= P2 -> P3
```

3.3.3 Formalisation sémantique préliminaire

Après cette première formalisation syntaxique, nous devons nous attacher à formaliser la sémantique des actions décrites selon les termes de notre modèle de représentation des connaissances. La fonctionalité que nous recherchons à travers les actions est celle qui consiste à obtenir un but précis. Il est donc intéressant de raisonner en termes de *changement d'état* des buts des actions, de l'état «non atteint» vers l'état «atteint». Afin de formaliser la sémantique de ce changement d'état, nous devons identifier tous les paramètres qui influencent l'état du but d'une action à un instant donné, ainsi que la manière dont ce but est susceptible de changer d'état, en fonction de ces paramètres.

- ➤ Dans une première approche, on peut considérer qu'une action donnée passe par trois états successifs : «à exécuter» si elle n'a pas encore été lancée, «en cours d'exécution» quand elle travaille à l'obtention de son but, et «terminée» une fois le but atteint.
- ➤ Représenter une action A comme un développement arborescent en sous-actions implique que l'exécution de A consiste en l'exécution de ses sous-actions. Par conséquent, l'obtention du but de A dépend de manière *exhaustive* de l'état des sous-actions de A, c'est à dire du niveau d'obtention des sous-buts de A.
- ➤ Finalement, à un instant donné, les circonstances dans lesquelles le but de A est *susceptible* de changer d'état, en l'occurrence d'être atteint, sont ces mêmes changements d'état au niveau des sous-actions de A. Notons que si un tel changement d'état intervient, sans pour cela que le but de A change également d'état, il se produit au moins la terminaison de la sous-action considérée.

Nous pouvons donc formaliser la sémantique des actions décrites selon les termes de notre modèle de représentation des connaissances en explicitant tous les cas de figures possibles dans lesquelles un but (ou un sous-but) change d'état. Ceci nous amène à distinguer les actions selon leur mode de parcours.

Sémantique préliminaire des actions parallèles

Soit une action parallèle `P ::= P1 // ... // Pn`. Par simplification, on considère que toutes les sous-actions de P démarrent en même temps, ce qui implique qu'à un instant donné, elles sont toutes en cours d'exécution, ou terminées. Les circonstances suivantes décrivent alors complètement la sémantique des actions parallèles :

Circonstance 1 :

Quand : *Une seule sous-action est en cours d'exécution.*
Toutes les autres sous-actions sont terminées.
Si : *Cette sous-action atteint son but.*

Alors : Cette sous-action se termine.
P atteint son but.

Circonstance 2 :

Quand : *Au moins deux sous-actions sont en cours d'exécution.*
Les autres sont soit en cours d'exécution, soit terminées.
Si : *L'une de ces sous-actions atteint son but.*
Alors : Cette sous-action se termine.

Sémantique préliminaire des actions séquentielles

Soit une action séquentielle S. Contrairement au cas des actions parallèles, les sous-actions de S ne démarrent pas toutes en même temps, mais successivement. Elles sont donc dans trois états possibles : «en cours d'exécution» et «terminée», comme dans le cas des action parallèles, et «à exécuter» quand la séquence d'exécution ne les a pas encore atteintes. Les circonstances suivantes décrivent alors complètement la sémantique des actions séquentielles :

Circonstance 1 :

Quand : *La dernière sous-action est en cours d'exécution.*
Toutes les autres sous-actions sont terminées.
Si : *Cette sous-action atteint son but.*
Alors : Cette sous-action se termine.
S atteint son but.

Circonstance 2 :

Quand : *Quelques premières sous-actions sont terminées.*
La sous-action suivante est en cours d'exécution.
Celles d'après sont à exécuter.
Si : *La sous-action en cours d'exécution atteint son but.*
Alors : Cette sous-action se termine.
La sous-action suivante passe en exécution.

3.3.4 Exemple : saisie d'un objet

Afin de fixer les idées, nous proposons de montrer comment notre modèle de représentation des connaissances, dans son état actuel, peut être utilisé pour décrire l'opération de saisie d'un objet.

Données expérimentales

L'étude des différentes verbalisations fournies par les sujets de l'expérience décrite précédemment fait apparaître les points suivants :

- Lors de la première phase de description («boire un peu d'eau»), 60% des sujets ont immédiatement explicité des actions de saisie sans autre description de mouvement (les termes utilisés étaient «saisir», «prendre» ou «attraper»). Les 40% restant ont d'abord parlé de «déplacement du bras», puis de saisie à proprement parler. Tous les sujets sont restés homogènes dans leurs descriptions des deux procédures de saisie (bouteille et verre) : par exemple, le mouvement du bras était explicité séparément les deux fois, aucune, mais jamais une fois seulement.

- ➤ Chez tous les sujets ayant seulement explicité une action de saisie, la deuxième phase de description les a conduit à distinguer une étape de mouvement, puis une étape à nouveau qualifiée de «saisie», le plus souvent en utilisant le même terme que pour désigner l'action de niveau supérieur (par exemple, les sujets B et D).
- ➤ Selon les cas, dans la deuxième ou troisième phase de description, 50% des sujets ont spontanément décrit la séquence initiale de déplacement du bras comme parallèle à l'ouverture (ou la mise en position de saisie) de la main. Un seul sujet a certifié que l'ouverture de la main intervenait «au dernier moment», c'est à dire juste avant un contact avec le verre (le sujet E). Les autres sujets n'ont cependant pas donné de précision quant à l'articulation des actions de déplacement du bras et d'ouverture de la main. Ils ont décrété que la main était «ouverte assez tôt».

☛ La réutilisation du même vocable («saisir», «attraper» ou «prendre») pour à la fois une action et l'une de ses composantes est une donnée intéressante. Sébillotte (1983) constate ce phénomène dans une étude sur les tâches administratives : par exemple, l'envoi d'un bon de commande sert à désigner aussi bien l'action de remplir le bon que celle de le mettre au courrier. En fait, une action est souvent désignée par son but global plutôt que par son déroulement, même quand il existe un nombre non nul de sous-buts préalables à remplir.

Représentation

- ➤ Les deux premiers points semblent montrer que le concept de «saisie» se décompose en deux étapes majeures, la première (déplacement du bras et ouverture de la main) étant considérée plutôt comme un préliminaire et la deuxième (mouvement final et fermeture de la main) plutôt comme l'action proprement dite.
- ➤ Bien que le taux de sujets ayant décrit un parallélisme dans le mouvement préliminaire ne soit pas significatif, les autres sujets n'ont pas infirmé cette hypothèse, appuyant uniquement le fait que l'ouverture de la main était une condition préalable à la continuation du mouvement.

Nous pouvons donc représenter graphiquement l'action de saisie, selon la structure syntaxique proposée dans la section précédente, comme illustré sur la figure 3.4.

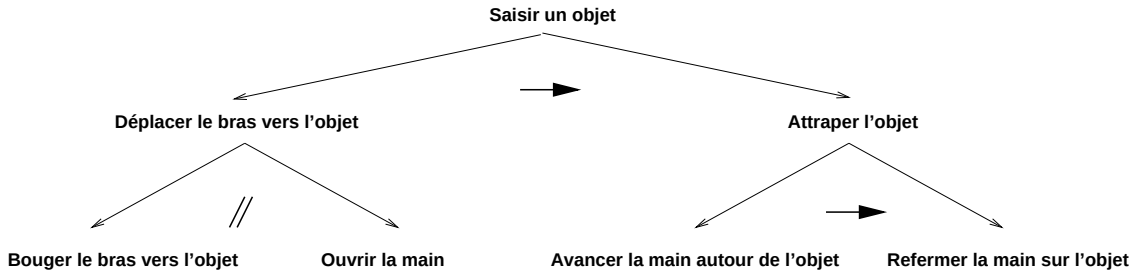

FIG. 3.4: Structure de l'action de «saisie»

☛ Le fait que cette représentation ne fasse apparaître que des développements binaires est un hasard. Nous n'avons jamais formulé de contrainte sur le nombre de sous-actions constituant le développement d'une action donnée.

Déroulement

Du point de vue sémantique, le déroulement de l'action de saisie ainsi représentée peut être explicité en adoptant le formalisme décrit dans la section précédente. Nous ne reproduisons ici

que trois étapes majeures de son déroulement, correspondant à l'obtention du premier sous-but, au lancement de la deuxième branche et à l'obtention du but final.

Circonstance 1 (parallèle) :

Quand : *L'action «Bouger le bras vers l'objet» est en cours d'exécution.*
L'action «Ouvrir la main» est terminée.
Si : *L'action «Bouger le bras vers l'objet» atteint son but.*
Alors : L'action «Bouger le bras vers l'objet» se termine.
L'action «Déplacer le bras vers l'objet» atteint son but.

Circonstance 2 (séquentielle) :

Quand : *L'action «Déplacer le bras vers l'objet» est en cours d'exécution.*
L'action «Attraper l'objet» est à exécuter.
Si : *L'action «Déplacer le bras vers l'objet» atteint son but.*
Alors : L'action «Déplacer le bras vers l'objet» se termine.
L'action «Attraper l'objet» entre en exécution.

Circonstance 1 (séquentielle) :

Quand : *L'action «Bouger le bras vers l'objet» est terminée.*
L'action «Attraper l'objet» est en cours d'exécution.
Si : *L'action «Attraper l'objet» atteint son but.*
Alors : L'action «Attraper l'objet» se termine.
L'action «Saisir l'objet» atteint son but.

3.3.5 Comparaison avec les réseaux procéduraux

Le modèle de représentation des connaissances que nous commençons à élaborer présente des similitudes avec le modèle de réseau procédural classiquement utilisé en sciences cognitives : l'utilisation d'un arbre orienté pour représenter les actions, ainsi que l'aspect séquentiel du parcours des branches lui sont empruntés. Il existe toutefois quelques différences sur lesquelles il convient de revenir :

Parallélisme *vs.* séquentialité

Nous avons dès le début introduit un deuxième mode de parcours de l'arbre de représentation des actions, le mode parallèle, qui n'apparaît pas dans un modèle de réseau procédural. Ce mode de parcours nous est nécessaire dans la mesure où il a été explicitement décrit par un nombre significatif de sujets et qu'il aura donc une influence sur les capacités de détection de notre système. En effet, notre système ne *doit pas* s'attendre à un déroulement séquentiel des événements là où un opérateur exécuterait les sous-actions dans un ordre quelconque ou en même temps.

La représentation des actions par réseau procédural est utilisée pour modéliser les mécanismes de raisonnement et de planification chez l'humain. Ces problèmes sont donc de plus haut niveau que ceux qui nous occupent ici. En particulier, les études cognitives sur le raisonnement et la planification ne descendent jamais jusqu'à la modélisation de l'action de saisie elle-même (Richard, 1990,p.83), mais considèrent celle-ci comme une entité non décomposable.

Organisation de l'arbre

Une différence plus fondamentale entre notre modèle de représentation des connaissance et un réseau procédural s'annonce dès à présent au niveau de l'organisation de l'arbre.

Dans un arbre tel que celui présenté sur la figure 3.4 en page 131, le développement représente bien la décomposition d'une action en actions de plus bas niveau, et ceci quel que soit le degré de profondeur auquel on se place. La sémantique des arcs est donc identique à tous les niveaux. Par exemple, la figure 3.4 en page 131 indique que l'action «saisir un objet» se *décompose* en l'action «déplacer le bras vers l'objet» puis en l'action «attraper l'objet». De même, l'action «attraper l'objet» se *décompose* en l'action «avancer la main autour de l'objet» puis en l'action «refermer la main». Il en découle que les seuls endroits de l'arbre contenant une information de nature véritablement actionnelle sont les feuilles de cet arbre. Tous les autres nœuds sont de nature logique : il décrivent des articulations entre concepts.

La situation est différente sur un réseau procédural tel que celui présenté sur la figure 3.1 en page 126. Une tel réseau n'est en fait pas complètement homogène quant à la sémantique de ses arcs. Au premier niveau, on retrouve une sémantique identique à la notre : l'action «déplacer X de Y en Z» se *décompose* en l'action «prendre X en Y», puis en l'action «poser X sur Z». Par contre, au deuxième niveau, les arcs n'ont plus la même signification : ils indiquent la présence de *prérequis* qui doivent être vérifiés avant l'exécution de l'action concernée, et non plus une décomposition de cette action en actions de plus bas niveau.

Nous voyons que dans une représentation de ce type, l'information actionnelle n'est pas cantonnée aux feuilles de l'arbre ; il nous est donc par exemple impossible, sur ce réseau procédural, de poursuivre la description de l'action «prendre X en Y». Notre représentation permet au contraire de poursuivre la description jusqu'à n'importe quel niveau de détail. Dans notre représentation, les «prérequis» d'une action sont représentés à gauche de celle-ci plutôt qu'en dessous. Ainsi, sur la figure 3.4 en page 131, l'action «déplacer le bras vers l'objet» (en quelque sorte l'équivalent de l'action «aller vers X» sur la figure 3.1 en page 126) peut être considérée comme un prérequis de l'action «attraper l'objet».

3.4 Catégories d'action

La structure arborescente, ainsi que les deux modes de parcours proposés dans la section précédente constituent la base de notre modèle de représentation des connaissances. Ces deux éléments sont pourtant insuffisants à l'élaboration d'un mécanisme de détection d'intentions. L'objet de cette section est de décrire les insuffisances de cette représentation, puis de proposer une amélioration adéquat du modèle.

3.4.1 Problématique

Afin de mettre en évidence les insuffisances de cette représentation, nous décrivons un ensemble de comportements naturels pour un opérateur, mais cependant problématiques dans la mesure où notre modèle, tel qu'il est défini pour l'instant, ne permet pas de les décrire. Ces comportements sont décrits en suivant l'exemple de la saisie d'un objet.

- Quand l'opérateur décide de saisir un objet, il est possible que l'action correspondante soit déjà partiellement accomplie. Par exemple, la main peut déjà se trouver ouverte et à proximité de l'objet, auquel cas la première étape de la saisie, telle que décrite sur la figure 3.4 en page 131 est inutile. L'opérateur passe donc directement à la deuxième étape. Notre modèle, au contraire, ne permet pas de préciser les circonstances dans lesquelles chaque action doit ou ne doit pas être entreprise : il indique seulement si chaque action est en cours d'exécution ou terminée. En conséquence, un système de détection d'intentions basé sur ce modèle s'attendrait à ce que l'opérateur exécute systématiquement chaque sous-action, même si certaines d'entre elles sont inutiles. Dans le cas d'une saisie partiellement effectuée par exemple, le système mémoriserait les actions de l'opérateur, constaterait que celui-ci *n'a pas* exécuté la première moitié de l'action de saisie telle qu'elle lui a été décrite, et en déduirait que l'action en cours est certainement autre chose qu'une saisie.

- Quand l'opérateur exécute une saisie d'objet, il est non seulement capable de décider si telle ou telle étape de l'action doit être exécutée, mais aussi de revenir en arrière dans l'exécution en cas d'imprévu. Par exemple, supposons qu'au moment où l'opérateur termine la première partie de l'action de saisie, l'objet se déplace (ou soit déplacé) soudainement. L'opérateur reviendra alors en arrière et entamera à nouveau cette première étape afin de déplacer lu bras vers la nouvelle position de l'objet. Notre modèle, au contraire, ne permet pas d'indiquer qu'une sous-action, même déjà effectuée, peut nécessiter une réexécution dans certaines circonstances : l'état «terminée» des actions indique seulement que le but a été atteint dans le passé, même si ce but est à nouveau invalide. Dans le cas de la saisie par exemple, le système pourrait tout à fait interpréter un déplacement de la main vers l'objet, mais en travers, comme la deuxième moitié de l'intention de saisie, car il ne saurait pas que la première étape a besoin d'être réexécutée.
- Si l'opérateur a conscience que certains résultats intermédiaires doivent être vérifiés en permanence (au besoin en réexécutant l'action correspondante), il a également conscience que certaines de ces étapes ne sont que des intermédiaires dont le résultat n'importe que pendant l'exécution, et n'influe pas sur le but global. Par exemple, l'action qui consiste à positionner la main autour de l'objet doit nécessairement s'effectuer avec la main ouverte. Cependant, une fois l'objet entouré, le fait que la main soit ouverte n'importe plus. Au contraire, il est même crucial que l'ouverture de la main ne soit pas un résultat à conserver en permanence, dans la mesure où l'action de saisie se termine par une fermeture de celle-ci, ce qui aboutirait à une contradiction. Dans l'état actuel, notre modèle ne permet pas d'indiquer que le résultat de certaines étapes ne doit pas être valide en permanence, mais seulement pour un certain nombre d'étapes consécutives.

3.4.2 Affinements du modèle

Utilisation des sous-buts

Les trois points précédents montrent qu'une action donnée est susceptible de prendre en compte ses sous-buts de *deux* manières différentes, et que notre modèle doit donc être affiné afin d'indiquer, pour chaque sous-action, de quelle manière son but est utilisé :

- Premièrement, il existe des sous-actions dont le but doit être constamment vérifié pour que celui de l'action supérieure le soit. Reprenons l'exemple de la saisie d'un objet. Le but de la sous-action finale («Attraper l'objet») doit être obtenu afin que celui de la saisie le soit. Cependant, cette condition est nécessaire mais non suffisante pour décrire complètement la sémantique de cette action. En effet, si ultérieurement, la sous-action «Attraper l'objet» est invalidée, alors le but de l'action de saisie est lui aussi invalidé aussitôt. Nous voyons donc que pour ce type de sous-action, l'obtention du sous-but est nécessaire à l'obtention du but supérieur, mais également à sa *préservation*. On parlera dans ce cas d'étape *persistante* dans une action.
- Deuxièmement, il existe des sous-actions dont le but ne doit être vérifié qu'un certain temps, le temps de l'obtention d'autres sous-buts. Reprenons l'exemple de la saisie d'un objet. Le but de la sous-action initiale («Déplacer le bras vers l'objet») doit être obtenu afin de pouvoir entamer la sous-action suivante. Cependant, cette condition est nécessaire mais non suffisante pour décrire complètement la sémantique de cette action. En effet, si ultérieurement, après obtention du but global (saisissement de l'objet), le sous-but initial est invalidé (ce qui est le cas puisque la main n'est plus «vers l'objet», mais «sur l'objet»), alors le but global reste malgré tout valide. Nous voyons donc que pour ce type de sous-action, l'obtention du sous-but est nécessaire à l'obtention du but supérieur, mais *pas* à sa préservation (dans ce cas précis, la préservation du sous-but serait même contradictoire avec celle du but global). On parlera dans ce cas d'étape *auxiliaire* dans une action.

☛ Cette nouvelle distinction nous rapproche du formalisme «MAD» proposé par Scapin (1986) décrivant également une tâche par une hiérarchie de sous-buts. Notre notion d'étape «persistante» ou «auxiliaire» doit être mise en correspondance avec la distinction que fait Scapin sur les «prérequis» et les «préalables» des actions : un prérequis est une condition nécessaire pur l'exécution d'une procédure tandis qu'un préalable est une condition qui doit simplement s'exécuter avant.

Représentation graphique

Afin de faire apparaître graphiquement ces deux types d'utilisation des sous-buts par l'action supérieure, on représentera la liaison d'une action avec ses sous-actions dites persistantes par un trait gras, et celle avec ses sous-actions dites auxiliaires par un trait fin, comme illustré sur la figure 3.5.

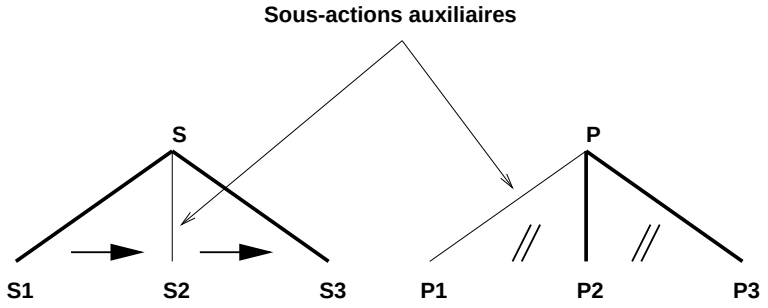

FIG. 3.5: Représentation graphique des catégories de sous-actions

3.4.3 Formalisation syntaxique affinée

Nous pouvons maintenant affiner la grammaire BNF décrivant la syntaxe des actions représentées selon les termes de notre modèle de la manière suivante :

```
A ::= S | P | a
S ::= B [-> B] ... [-> B]
P ::= B [// B] ... [// B]
B ::= Pers (A) | Aux (A)
```

L'introduction de l'entité intermédiaire `B` permet de spécifier la catégorie de la sous-action correspondante, c'est à dire la manière persistante (`Pers (A)`) ou auxiliaire (`Aux (A)`) d'utiliser son but.

À titre d'exemple, les deux actions S et P représentées sur la figure 3.5 sont décrites de la manière suivante :

```
S ::= Pers (S1) -> Aux  (S2) -> Pers (S3)
P ::= Aux  (P1) // Pers (P2) // Pers (P3)
```

3.4.4 Formalisation sémantique affinée

Avec l'apparition des deux catégories de sous-actions définies précédemment, la description de l'état d'une action comme un choix parmi «à exécuter», «en cours d'exécution» et «terminée» devient inappropriée pour les raisons suivantes :

➤ Nous avons montré la nécessité de la notion de «surveillance» d'une action : dire que le but d'une sous-action, doit être vérifié à la fois pour l'obtention et pour la préservation du but de niveau supérieur revient à introduire un nouvel état des actions, un état de surveillance, dans lequel le but de l'action est atteint, mais celle-ci surveille néanmoins qu'il reste valide, et signale un changement d'état si pour quelque raison que ce soit, son but est invalidé.

- La notion d'action «terminée» n'a donc plus de sens dans la mesure ou il est maintenant important de prendre en compte les événements consécutifs à la terminaison d'une action, suivant qu'elle est utilisée de manière persistante ou auxiliaire. Une action peut par conséquent être «terminée» avec ou sans surveillance de son but.

Une meilleure description des états possibles est donc proposée sous la forme des trois cas suivants :

- Une action est «en cours d'exécution» quand son but n'est pas atteint, mais que celle-ci travaille à son obtention. Cet état ne change pas par rapport à la description préliminaire qui en avait été faite.
- Une action est «en veille» quand son but est atteint, donc quand celle-ci ne s'exécute pas, mais surveille néanmoins que son but reste valide. L'invalidation de son but est susceptible de générer l'invalidation du but de niveau supérieur.
- Une action est «inactive» quand elle n'est ni en exécution, ni en veille. Cet état intervient quand l'action n'a pas encore été lancée, ou quand elle a atteint son but, sans nécessité de surveillance ultérieure.

Un deuxième point important change dans cet affinement du modèle : maintenant que l'état d'une action est une donnée ternaire et non plus binaire, les changements d'état ne sont plus implicites : à partir d'un état donné il existe plusieurs possibilités pour l'état suivant. L'état de l'action elle-même fait donc maintenant partie des paramètres qui jouent sur son changement d'état.

En fonction de ces deux nouveautés, nous pouvons maintenant affiner la formalisation sémantique des actions décrites selon notre modèle de représentation des connaissances, toujours en distinguant ces actions selon leur mode de parcours.

Sémantique affinée des actions parallèles

Soit une action parallèle P. On considère toujours par simplification que toutes les sous-actions sont démarrées en même temps. Les circonstances suivantes décrivent alors complètement la sémantique des actions parallèles :

Circonstance 1 :

Quand : *P est en cours d'exécution.*
Une sous-action est en cours d'exécution.
Toutes les autres sous-actions sont en veille.
Si : *Cette sous-action atteint son but.*
Alors : *P* atteint son but.

Circonstance 2 :

Quand : *P est en cours d'exécution.*
Au moins deux sous-actions sont en cours d'exécution.
Toutes les autres sous-actions sont en veille.
si : *L'une de ces sous-action atteint son but.*
Alors : Cette sous-action passe en veille.

Circonstance 3 :

Quand : *P est en cours d'exécution.*
Au moins une sous-action est en veille.
Les autres sont soit en veille, soit en cours d'exécution.

si : *Le but de cette sous-action est invalidé.*
Alors : Cette sous-action repasse en exécution.

Circonstance 4 :

Quand : *P est en veille.*
Toutes les sous-actions persistantes sont en veille.
Toutes les sous-actions auxiliaires sont inactives.
si : *Le but de l'une des sous-actions persistantes est invalidé.*
Alors : Le but de P est invalidé.

Sémantique affinée des actions séquentielles

Soit une action séquentielle S. Contrairement au cas des actions parallèles, toutes les sous-actions de S ne sont pas démarrées en même temps. Les circonstances suivantes décrivent alors complètement la sémantique des actions séquentielles :

Circonstance 1 :

Quand : *S est en cours d'exécution.*
La dernière sous-action est en cours d'exécution.
Toutes les autres sous-actions sont en veille.
si : *Cette sous-action atteint son but.*
Alors : S atteint son but.

Circonstance 2 :

Quand : *S est en cours d'exécution.*
Quelques premières sous-actions sont en veille.
La sous-action suivante est en cours d'exécution.
Les sous-actions suivantes sont inactives.
si : *Cette sous-action atteint son but.*
Alors : Cette sous-action passe en veille.
La sous-action suivante passe en exécution.

Circonstance 3 :

Quand : *S est en cours d'exécution.*
Quelques premières sous-actions sont en veille.
La sous-action suivante est en cours d'exécution.
Les sous-actions suivantes sont inactives.
si : *Le but d'une des sous-action en veille est invalidé.*
Alors : Toutes les sous-actions suivantes redeviennent inactives.
Cette sous-action repasse en exécution.

Circonstance 4 :

Quand : *S est en veille.*
Toutes les sous-actions persistantes sont en veille.
Toutes les sous-actions auxiliaires sont inactives.
si : *Le but de l'une des sous-action persistante est invalidé.*
Alors : Le but de S est invalidé.

☛ Notons que pour les deux types d'action (parallèle ou séquentielle), deux descriptions sur les quatre signalent un changement d'état du but de l'action considérée. Ces descriptions peuvent apparaître moins rigoureuses que les autres dans la mesure où il n'est pas fait mention du changement d'état de la sous-action dont le but a changé d'état, ni du changement d'état de l'action elle-même. En fait, ces changements d'états sont gérés par le niveau supérieur, qui pourrait être une autre action, ou directement le contrôle de l'humain si l'action considérée est celle de plus haut niveau.

Plus précisément, quand l'une de ces deux circonstances s'applique, le changement d'état de l'action considérée est déterminé soit par l'humain lui-même (il décide par exemple d'abandonner la tentative, ou de réessayer), ce qui sort du cadre de cette modélisation, soit par la première action supérieure pour laquelle c'est l'une des deux autres transitions qui s'applique. Cette description formelle est donc malgré tout rigoureuse, car le «flou» laissé à un niveau d'action est corrigé à un niveau supérieur par le même ensemble de descriptions.

3.4.5 Exemple : saisie d'un objet

Catégorisation des sous-actions

Suite à cet affinement du modèle, nous pouvons maintenant reprendre l'exemple de la saisie d'un objet et préciser, pour chaque sous-action, la catégorie à laquelle elle appartient.

- L'action «Saisir un objet» elle-même était décomposée en deux étapes. Sur ces deux étapes, seule la dernière compte dans la préservation du résultat final. En effet, lors de l'exécution, l'étape «Déplacer le bras vers l'objet» peut s'avérer utile, mais une fois l'objet saisi, peu importe qu'elle ait eu lieu. Ce qui compte, c'est que l'étape «Attraper l'objet» reste valide. La première étape est donc *auxiliaire*.
- L'action «Attraper l'objet» était décomposée en deux étapes. Sur ces deux étapes, seule la dernière compte dans la préservation du résultat final. En effet, lors de l'exécution, l'étape «Avancer la main autour de l'objet» peut s'avérer utile, mais une fois l'objet attrapé, peu importe qu'elle ait eu lieu. Ce qui compte, c'est que l'étape «Refermer la main sur l'objet» reste valide. La première étape est donc ici encore *auxiliaire*.
- Dans l'étape initiale, seule la sous-action «Bouger le bras vers l'objet» compte dans la préservation du résultat final. En effet, c'est cette sous-action qui détermine que la main se trouve bien dans l'axe pour terminer le mouvement de saisie. La sous-action «Ouvrir la main» ne doit pas être préservée, sans quoi elle entrerait en conflit avec l'étape finale de fermeture. Cette sous-action est donc également *auxiliaire*.

Approfondissement

En plus de cette catégorisation des sous-actions déterminées expérimentalement, nous devons également pousser le niveau de description de la première étape un cran plus loin. En effet, l'action «bouger le bras vers l'objet» ne peut être conservée telle quelle, dans la mesure où certaines parties de son résultat doivent être préservées, et d'autres non. Le but réel de ce mouvement est de positionner le bras *dans l'axe* de l'objet, de manière à pouvoir ensuite avancer la main jusqu'à l'entourer. Il faut donc distinguer d'une part le positionnement dans l'axe de l'objet, et d'autre part le positionnement de la main dans le plan vertical. Tandis que le positionnement dans l'axe est une condition qui doit être préservée, la position de la main ne doit pas être surveillée ultérieurement dans la mesure où l'étape d'avancement viendra la modifier. Elle appartient donc à la catégorie des étapes auxiliaires.

À ce stade, la modélisation de l'action de saisie pose un problème : l'étape d'ouverture de la main a été expérimentalement située dans la première étape du mouvement, le déplacement global du bras. Cependant, et bien qu'aucun sujet ne l'ai précisé, nous savons que l'étape consistant à avancer la main jusqu'à entourer l'objet est conditionnée par l'ouverture de la main. Le problème

est donc que la description actuelle n'indique pas que la main doit rester ouverte pendant l'étape d'avancement. En effet, nous avons vu que l'étape d'ouverture est auxiliaire, ce qui l'empêche d'avoir une influence sur l'étape d'avancement.

Ce problème peut être résolu en ajoutant, au sein même de l'étape d'avancement, une deuxième instance, auxiliaire également, d'ouverture de la main. Dans ce cas, la première instance est à interpréter comme celle décrite par les sujets de l'expérience : «ouvrir la main», et la deuxième plutôt comme un garde fou : «s'assurer que la main reste ouverte».

Représentation graphique affinée

Une nouvelle version de l'action de saisie est donc proposée sur la figure 3.6.

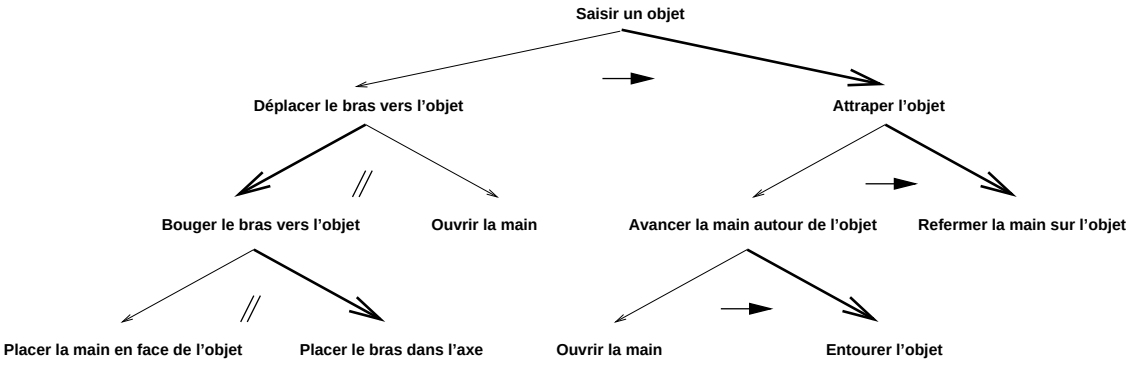

FIG. 3.6: Représentation de l'action de «saisie»

☛ La redondance de l'action d'ouverture de la main est en théorie inutile : seule la deuxième instance est réellement nécessaire pour représenter correctement l'action de saisie dans son ensemble. Cependant, la représentation actuelle est plus proche de la description fournie par les sujets de l'expérience. Il est d'autre part intéressant de constater que si l'on supprime la première instance d'ouverture de la main, nous retombons sur la description fournie par l'unique sujet ayant décrété que l'ouverture de la main se faisait «au dernier moment».

Déroulement

Afin d'illustrer la description sémantique affinée que nous avons établie précédemment, nous reproduisons ici deux cas de figure dont un cas pathologique. Le premier cas illustre le lancement de la dernière étape de la saisie (la fermeture de la main). Le deuxième cas illustre un «accident» durant lequel l'objet change de place (il ne se trouve plus «dans l'axe») alors que la main n'est pas encore refermée dessus.

Circonstance 2 (séquentielle) :

Quand : *L'action «Attraper l'objet» est en cours d'exécution.*
La sous-action «Avancer la main autour de l'objet» est en cours d'exécution.
La sous-action «Refermer la main sur l'objet» est inactive.
si : *La sous-action «Avancer la main autour de l'objet» atteint son but.*
Alors : La sous-action «Avancer la main autour de l'objet» passe en veille.
La sous-action «Refermer la main sur l'objet» passe en exécution.

Circonstance 3 (séquentielle) :

Quand : *L'action «Saisir l'objet» est en cours d'exécution.*
La sous-action «Déplacer le bras vers l'objet» est en veille.
La sous-action «Attraper l'objet» est en cours d'exécution.
si : *Le but de la sous-action «Déplacer le bras vers l'objet» est invalidé.*
Alors : La sous-action «Attraper l'objet» redevient inactive.
La sous-action «Déplacer le bras vers l'objet» repasse en exécution.

3.5 Application à TOASt

Afin de fournir un exemple concret de mise en pratique de ce modèle de représentation des connaissances dans sa version affinée, nous proposons de décrire la manière dont celui-ci est utilisé dans le démonstrateur TOASt, au travers notamment de la description de l'action de saisie.

3.5.1 Notion d'action terminale

Jusqu'ici, nous avons envisagé le concept d'action comme un développement arborescent de sous-actions. Ce développement doit cependant prendre fin, c'est à dire qu'il existe nécessairement des actions d'un type particulier, qui ne se développent pas en sous-actions. Ces actions sont dites «terminales».

Certaines études cognitives ont mis en évidence un concept très voisin : celui d'«action élémentaire». Sébillotte (1988) montre qu'il existe un niveau en dessous duquel les sujets n'arrivent plus à fournir d'explication sur la manière de réaliser une action, chose que nous avons également constaté dans notre expérience. Une action élémentaire n'est donc pas cognitivement décomposable en actions de plus bas niveau. Plus précisément, il n'est pas possible d'assigner un but aux composantes de cette action.

Tandis que la notion d'action élémentaire dépend uniquement d'une étude cognitive des sujets, notre problématique est légèrement différente (c'est la raison qui nous conduit à utiliser plutôt le terme d'«action terminale») : certaines contraintes risquent en effet d'être imposées par le système avec lequel l'opérateur interagit. Par exemple, les cogniticiens considèrent que les actions telles que «prendre un objet» sont des actions élémentaires. Par contre, dans un système de télé-opération non immersif, l'opérateur dispose en général de commandes différentes pour déplacer le bras manipulateur et l'organe de préhension. L'action de saisie doit par conséquent être décomposée mentalement en actions de plus bas niveau. Nos actions terminales sont donc d'un niveau conceptuel inférieur aux actions élémentaires de Sébillotte.

Plus précisément, dans le cas d'une interaction avec un système artificiel, les actions terminales doivent répondre aux caractéristiques suivantes :

- Leur niveau conceptuel doit être suffisamment bas pour que toute opération exécutable par l'opérateur soit représentable selon les termes de notre modèle d'une part, et de manière à pouvoir distinguer si nécessaire les sous-actions auxiliaires et persistantes d'autre part.
- Leur niveau conceptuel doit être suffisamment élevé pour représenter au moins les actions les plus élémentaires permises par l'interface du système. Il serait en effet inutile de fournir des actions terminales auxquelles l'opérateur n'aurait de toute manière jamais accès.

3.5.2 Les actions terminales de TOASt

Afin de pouvoir représenter convenablement les actions que l'opérateur est susceptible d'entreprendre, et notre système est susceptible de détecter, il importe en tout premier lieu de définir l'ensemble des actions terminales dont nous aurons besoin.

TOASt a été conçu pour simuler la télé-opération d'un bras manipulateur. L'opérateur dispose pour cela d'un simple joystick, lui aussi simulé, et actionné par les trois boutons de la souris.

Grâce à ce joystick, les commandes élémentaires suivantes sont disponibles, et sont représentées graphiquement sur la figure 3.7 :

- Opérer une rotation du bras autour de son axe vertical. Une seule dimension du joystick est alors utilisée.
- Opérer un déplacement de l'organe de préhension dans le plan du bras. Le plan du joystick est alors associé au plan du bras, et sa position indique un vecteur vitesse appliqué à l'organe de préhension.
- Opérer une rotation de l'organe de préhension dans le plan vertical. Une seule dimension du joystick est alors utilisée.
- Opérer l'ouverture ou la fermeture de l'organe de préhension. Une seule dimension du joystick est alors utilisée.

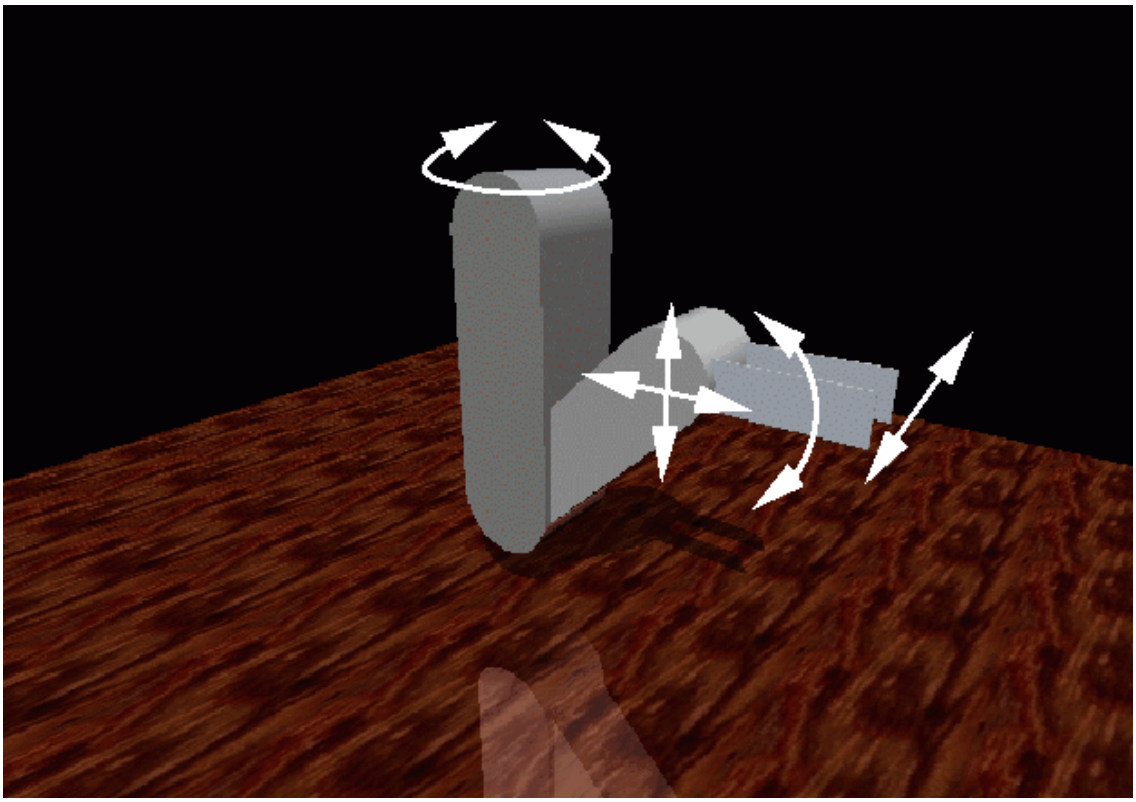

FIG. 3.7: Commandes élémentaires de TAS$_t$

L'opérateur travaille au moyen exclusif de ces commandes. C'est donc par leur entremise qu'il exprime ses intentions. Les actions terminales du système, comme toutes autres actions, doivent par conséquent s'exprimer également en fonction des commandes disponibles. En particulier, à chacune de ces commandes correspond une action terminale simple, comme suit :

- Opérer une rotation du bras autour de son axe vertical implique un changement de position angulaire de celui-ci. L'action terminale simple correspondante consiste donc à donner au bras une position angulaire θ autour de l'axe vertical.
- Opérer un déplacement de l'organe de préhension dans le plan du bras implique un changement de position de celui-ci. L'action terminale simple correspondante consiste donc à donner à l'organe de préhension une position (x, y) dans le plan vertical.

- Opérer une rotation de l'organe de préhension dans le plan vertical implique un changement de position angulaire de celui-ci. L'action terminale simple correspondante consiste donc à donner à l'organe de préhension une position angulaire α dans le plan vertical.
- Finalement, l'opération d'ouverture ou de fermeture de l'organe de préhension a été simplifiée pour aboutir soit à une ouverture totale, soit à une fermeture totale, cette fermeture étant sur un objet, ou sur l'organe de préhension lui-même.

3.5.3 Action de saisie

L'action de saisie, décrite sur la figure 3.6 en page 139 peut maintenant s'exprimer plus formellement en fonction des actions terminales que nous venons d'expliciter.

- L'action «Placer la main en face de l'objet» correspond à l'action terminale de positionnement de l'organe de préhension en (x, y) dans le plan vertical.
- L'action «Placer le bras dans l'axe de l'objet» correspond à l'action terminale de positionnement angulaire du bras en θ selon l'axe vertical.
- L'action «Entourer l'objet» correspond à nouveau à un positionnement de l'organe de préhension en (x, y) dans le plan vertical.
- Les actions d'ouverture et de fermeture de la main sont évidentes.

La figure 3.8 illustre de manière plus formelle l'opération de saisie, appliquée au cas du bras manipulateur de $\mathcal{TOAS}_t$.

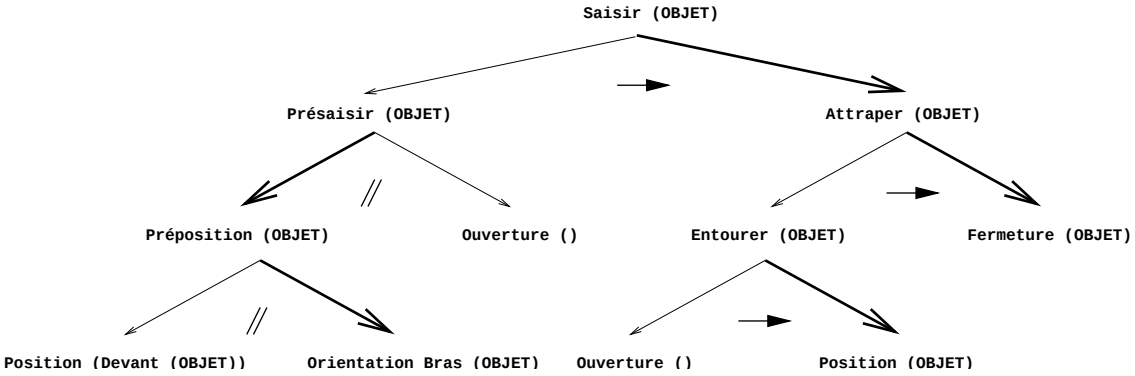

FIG. 3.8: Action de saisie pour $\mathcal{TOAS}_t$

☛ Remarquons que la représentation de l'opération de saisie que nous avons élaborée précédemment est significative de la manière dont l'humain exécute cette opération *manuellement*, et non pas aux commandes d'un système artificiel. Il n'y a aucune raison pour que le contrôle de cette opération au moyen d'un bras manipulateur se déroule exactement de la même manière. Cependant, cette représentation reste bien adaptée au déroulement de la même opération dans $\mathcal{TOAS}_t$, et nous continuerons par conséquent à l'utiliser.

D'un point de vue théorique, la modélisation d'une opération donnée est susceptible de varier suivant le type de manipulateur et le type d'interface utilisés. Une nouvelle expérimentation serait donc nécessaire à chaque fois que l'un de ces paramètres est modifié.

3.6 Formalisation

Avec l'introduction de la notion d'action terminale, nous disposons finalement de tous les éléments nécessaires à la description formelle de notre modèle de représentation des connais-

sances dans sa version finale. Dans cette section, nous résumons les différentes étapes d'élaboration du modèle que nous venons de suivre par un ensemble de définitions et de points de vocabulaire.

3.6.1 Les actions et leur mode

Définition 15: Action

*Une **action** est soit une action terminale, soit un* arbre *composé d'un ensemble de* branches *et doté d'un* mode d'exécution *de ces branches. Ce mode d'exécution est soit* parallèle, *soit* séquentiel, *auquel cas l'arbre est orienté. Le nombre de branches d'un tel arbre est supérieur ou égal à un.* ■

Vocabulaire: Mode d'une action

*Dans le cas d'une action non terminale, on parlera simplement de **mode d'une action** pour désigner le mode d'exécution des branches qui la composent. Une action est donc de mode séquentiel ou parallèle. De manière encore plus concise, on pourra dire qu'une action* est *séquentielle ou parallèle.* ●

Les deux modes possibles d'une action sont définis comme suit :

- ➤ Si une action (non terminale) est séquentielle, les sous-actions menant à sa résolution doivent être exécutées l'une après l'autre : pour entamer une certaine sous-action, il est nécessaire d'attendre la résolution de la sous-action précédente.
- ➤ Si une action (non terminale) est parallèle, les sous-actions menant à sa résolution peuvent être exécutées en même temps, ces sous-actions étant indépendantes les unes des autres.

3.6.2 Les branches et leur qualité

Définition 16: Branche

*Une **branche** est l'association d'une action et de la manière dont l'action supérieure l'utilise. Une action peut utiliser une sous-action de manière* persistante *ou* auxiliaire. ■

Vocabulaire: Qualité d'une branche

*On parlera simplement de **qualité d'une branche** pour désigner la manière dont l'action associée est utilisé. Une branche est donc de qualité persistante ou auxiliaire. De manière encore plus concise, on pourra dire qu'une branche* est *persistante ou auxiliaire.* ●

Les deux qualités possibles d'une branche sont définies comme suit :

- ➤ Si une branche est persistante, sa résolution conditionne la résolution de l'action supérieure. D'autre part, la préservation de son but est nécessaire à la préservation du but supérieur.
- ➤ Si une branche est auxiliaire, sa résolution conditionne la résolution de l'action supérieure. Cependant, la préservation du but du but supérieur n'implique pas la préservation du but de l'action associée à la branche.

3.6.3 Syntaxe formelle du modèle

En fonction des définitions précédentes, la structure syntaxique d'une action représentée selon les termes de notre modèle est décrite formellement par la grammaire BNF suivante :

```
A ::= S | P | t
S ::= B [-> B] ... [-> B]
P ::= B [// B] ... [// B]
B ::= Pers (A) | Aux (A)
```

- A représente la notion d'action, qui peut être séquentielle (S), parallèle (P) ou terminale (t).
- B représente la notion de branche, qui correspond à une utilisation persistante (`Pers (A)`) ou auxiliaire (`Aux (A)`) des actions.

3.6.4 Dynamique du modèle

Définition 17: État d'une action

*Soit une action A représentée selon les termes de notre modèle. L'**état** de l'action A défini son comportement vis-à-vis du but qui lui est associé. Cet état peut être* inactif, en cours d'exécution *ou* en veille. ■

Les différents états d'une action sont définis comme suit :

- Lorsqu'une action est inactive, le but correspondant est dans un état quelconque. L'action ne s'intéresse pas à son but.
- Lorsqu'une action est en cours d'exécution, le but correspondant n'est pas atteint. L'action travaille à son obtention.
- Lorsqu'une action est en veille, le but correspondant est atteint. L'action ne s'exécute pas, mais surveille la préservation de validité de son but.

☛ Par soucis de concision, on pourra parler directement de l'«état d'une branche», ce qui se réfère à l'état de l'action associée à la branche.

Définition 18: Contexte d'une action

*Soit une action A. Le **contexte** de A, noté C, est défini comme l'ensemble composé de l'état de A ainsi que des états de toutes les branches de A. Dans le cas d'une action terminale, le contexte est réduit à l'état de l'action elle-même.* ■

Définition 19: Événement

*Soit une action A. On appelle **événement**, noté ε, un changement d'état du but de A, si A est une action terminale, ou un changement d'état de l'un des sous-buts de A sinon.* ■

Définition 20: Transition

*Soit une action A. Un événement entraîne soit une modification du contexte C de A, soit une modification de l'état du but de A. La règle décrivant cette modification est appelée **transition** ; elle est notée $\Theta(A, C, \varepsilon)$.* ■

Grâce à ces définitions, nous pouvons maintenant décrire formellement la sémantique des actions représentées selon les termes de notre modèle, en explicitant de manière exhaustive l'ensemble de leurs transitions.

Processus d'évolution des actions parallèles

Transition 1 :

C : *P est en cours d'exécution.*
Une branche est en cours d'exécution.
Toutes les autres branches sont en veille.
ε : *Cette branche atteint son but.*
$\Theta(P, C, \varepsilon) \rightarrow P$ atteint son but.

Transition 2 :

C : *P est en cours d'exécution.*
Au moins deux branches sont en cours d'exécution.
Toutes les autres branches sont en veille.
ε : *L'une de ces branche atteint son but.*
$\Theta(P, C, \varepsilon) \rightarrow$ Cette branche passe en veille.

Transition 3 :

C : *P est en cours d'exécution.*
Au moins une branche est en veille.
Les autres branches sont soit en veille, soit en cours d'exécution.
ε : *Le but de cette branche est invalidé.*
$\Theta(P, C, \varepsilon) \rightarrow$ Cette branche repasse en exécution.

Transition 4 :

C : *P est en veille.*
Toutes les branches persistantes sont en veille.
Toutes les branches auxiliaires sont inactives.
ε : *Le but de l'une des branches persistantes est invalidé.*
$\Theta(P, C, \varepsilon) \rightarrow$ Le but de P est invalidé.

Processus d'évolution des actions séquentielles

Transition 1 :

C : *S est en cours d'exécution.*
La dernière branche est en cours d'exécution.
Toutes les autres branches sont en veille.
ε : *Cette branche atteint son but.*
$\Theta(S, C, \varepsilon) \rightarrow S$ atteint son but.

Transition 2 :

C : *S est en cours d'exécution.*
Quelques premières branches sont en veille.
La branche suivante est en cours d'exécution.
Les branches suivantes sont inactives.
ε : *Cette branche atteint son but.*
$\Theta(S, C, \varepsilon) \rightarrow$ Cette branche passe en veille.
La branche suivante passe en exécution.

Transition 3 :

C : *S est en cours d'exécution.*
Quelques premières branches sont en veille.
La branche suivante est en cours d'exécution.
Les branches suivantes sont inactives.
ε : *Le but de l'une des branches en veille est invalidé.*
$\Theta(S, C, \varepsilon) \rightarrow$ Toutes les branches suivantes redeviennent inactives.
Cette branche repasse en exécution.

Transition 4 :

C : *S est en veille.*
Toutes les branches persistantes sont en veille.
Toutes les branches auxiliaires sont inactives.
ε : *Le but de l'une des branches persistantes est invalidé.*
$\Theta(S, C, \varepsilon) \rightarrow$ Le but de S est invalidé.

Processus d'évolution des actions terminales

Transition 1 :

C : *T est en cours d'exécution.*
ε : *T atteint son but.*
$\Theta(T, C, \varepsilon) \rightarrow$.

Transition 2 :

C : *T est en veille.*
ε : *Le but de T est invalidé.*
$\Theta(T, C, \varepsilon) \rightarrow$.

3.7 Caractéristiques du modèle

La section précédente décrit complètement et formellement notre modèle de représentation des connaissances tel qu'il sera utilisé par la suite. La principale caractéristique de ce modèle est son approche syntaxique de la représentation des actions. Il existe des approches non syntaxiques de ce type de problème, par exemple l'utilisation de grammaires ensemblistes, l'approche orientée objet (Poitrenaud, 1998), ou encore la représentation par contraintes (Richard, 1999).

Ces formalismes, quand ils ne rélèvent pas directement de la psychologie cognitive, sont en général destinés à des application du type planification ou analyse de tâche (OTAN, 1991), et débordent par conséquent du champ d'application de notre étude, plus focalisée sur la représentation d'actions précises que sur la manière dont l'humain se construit ces mêmes représentations. Dans ce contexte, l'approche syntaxique reste particulièrement bien adaptée, et débouche sur un certain nombre de spécificités qui caractérisent notre modèle, et qu'il nous semble important de souligner dès à présent.

3.7.1 Mutuelle référence

Les deux principales définitions décrivant ce modèle, à savoir les définitions des «actions» et des «branches» sont mutuellement référentes : le concept d'action est en réalité défini par lui-même, comme un ensemble d'actions de plus bas niveau, les sous-actions. Du point de vue de la mise en œuvre, cela confère à ce modèle une très grande généricité : seules les actions terminales doivent être «câblées», pour chaque système particulier. La syntaxe et la sémantique

des actions représentées selon les termes de notre modèle sont totalement indépendantes des actions terminales. Par conséquent, toute la connaissance non terminale d'un système utilisant ce modèle peut lui être inculquée dynamiquement, en particulier au démarrage de celui-ci. Dans *TAO*$_{ASt}$, cette connaissance est exprimée sous forme de fichiers LISP qui sont interprétés et pour lesquels aucune compilation n'est donc nécessaire.

3.7.2 Réutilisabilité des actions

Bien que ce ne soit pas apparu explicitement jusqu'ici, les actions représentées par notre modèle sont *réutilisables*, c'est à dire que rien n'empêche de définir une action, puis de l'utiliser simultanément en tant que sous-action dans d'autres actions. Un tel exemple de réutilisation existe déjà dans la représentation de l'action de saisie : l'action «ouverture de la main» apparaît deux fois. Cet exemple constitue un cas particulier dans la mesure où l'ouverture de la main est une action terminale, mais l'on peut imaginer des cas où des actions non terminales, comme la saisie, sont utilisées en tant que sous-actions dans plusieurs opérations distinctes, ou plusieurs fois dans une même opération.

Cette caractéristique de notre modèle est conforme au résultat expérimental montrant que les sujets décrivaient spontanément la première occurrence d'une action, puis parlaient du «même mouvement» pour décrire les occurrences suivantes, même celles se rapportant à un autre objet.

Cette caractéristique est également à l'origine de la notion de branche : étant donné que les actions sont *a priori* réutilisables, la notion de qualité définie précédemment ne peut leur être associée. Une action donnée est indépendante de la manière dont son but sera utilisé au niveau supérieur. La notion de qualité est donc à associer à l'action supérieure, et non à l'action elle-même.

3.7.3 Représentation implicite des buts

Une autre caractéristique intéressante de notre modèle est la représentation implicite des buts. Cette caractéristique provient d'une part de la décomposition de chaque action en actions terminales, et d'autre part de l'introduction de la notion de qualité d'une branche. Normalement, toute action devrait être associée *explicitement* à un but. Par exemple, l'action de saisie devrait être associée au but «que l'objet soit saisi». Cela signifie en particulier que nous devrions également nous doter d'un modèle de représentation des buts, en plus du modèle de représentation des actions.

En fait, notre modèle de représentation des connaissances rend inutile cette dualité, car il a la particularité de représenter aussi bien les actions que leurs buts : chaque action terminale est associée à un but terminal, qui lui aussi est par conséquent câblé. Dans *TAO*$_{ASt}$ par exemple, l'action terminale `Fermeture (OBJET)` contient une routine câblée pour évaluer le contact entre l'objet et l'organe de préhension. Dans la mesure où toute action se décline en actions terminales, tout but non terminal se décline en buts terminaux selon la même logique. Plus précisément, un but non terminal est implicitement décrit par notre modèle comme la conjonction de tous les buts terminaux joignables par des branches persistantes.

En reprenant l'exemple de la saisie, décrit sur la figure 3.8 en page 142, nous voyons que le but global «que l'objet soit saisi» se décline simplement en le but terminal «que l'organe de préhension soit refermé sur l'objet».

☛ Cette caractéristique de notre modèle ne suit pas l'idée de séparation but / action développée en psychologie cognitive, mais ceci ne constitue pas pour autant un inconvénient. En effet, rappelons que contrairement aux études concernées, nous ne cherchons pas à modéliser les capacités de raisonnement et de planification de l'humain : notre but n'est donc pas de construire dynamiquement une méthode de résolution d'un problème donné, mais bien de modéliser des mécanismes de résolution prédéterminés pour des actions précises.

3.7.4 Opérationalisation

Anderson (1983) introduit le concept d'«opérationalisation» désignant les processus consistant à transformer un ensemble de connaissances déclaratives sous une forme exécutable. Pour Richard (1990), un réseau procédural décrivant une action est une connaissance déclarative : cette connaissance ne produit pas elle-même l'action considérée. Il est donc nécessaire d'opérationaliser cette connaissance, en fournissant par exemple des algorithmes permettant de l'utiliser à des fins *effectives* d'exécution.

Le modèle de représentation des connaissances que nous avons proposé est présenté d'une manière *déclarative*, mais nous pensons que contrairement à un réseau procédural, il contient intrinsèquement une part de connaissance déjà opérationalisée, sans quoi nous n'aurions pu établir la dynamique du modèle dans la section précédente). En effet, l'introduction du concept de «qualité» d'une branche relève plus des aspects opérationnels que déclaratifs : il donne des indications sur le comportement *dynamique* de l'action représentée, en fonction de circonstances comme des changements d'état. En cela, notre modèle comporte bien des composantes qui ne peuvent être décrites autrement que sous forme opérationnelle. Dans les prochains chapitres de cette partie, nous poursuivrons l'opérationalisation de notre modèle en fournissant un ensemble d'algorithmes liés à la détection d'intentions.

3.7.5 Actions (terminales) et schèmes

Il est intéressant de confronter notre modèle de représentation des connaissances à la notion piagétienne de «schème» (Cellerier, 1979), de la même manière que Richard (1990,p.243) effectue cette comparaison avec ses «schémas d'action».

Il existe des similitudes entre les actions représentées selon les termes de notre modèle et les schèmes. Ces similitudes sont les suivantes :

- Les conditions précises dans lesquelles nos actions sont applicables sont parfaitement définies : en tant que sous-actions, elles dépendent du mode (parallèle ou séquentiel) de l'action supérieure, de la qualité (auxiliaire ou persistante) de leur branche, et de l'état des sous-actions précédentes. La section décrivant la dynamique du modèle s'est justement attaché à décrire les conditions d'applicabilité des sous-actions d'une action particulière.
- Comme nous l'avons déjà noté, nos actions sont réutilisables : elles peuvent servir à plusieurs reprises en tant que sous-actions d'une action supérieure, ou même dans des actions supérieures différentes. De plus, elles présentent un caractère de généricité : l'action de saisie, par exemple, n'est pas définie pour un objet en particulier et pourra s'appliquer telle-quelle pour un nouvel objet.
- Enfin, nos actions contiennent implicitement de l'information sur leur but : ce but est exprimé en fonction des sous-buts de l'action concernée. La finalité d'une action est donc comprise dans la description de l'action elle-même. De plus, le concept de qualité d'une branche donne à l'action des capacités de contrôle sur son propre déroulement, par exemple, en lui indiquant que si un sous-but persistant est invalidé, il s'est produit une erreur ou un accident dans l'exécution.

La principale différence entre nos actions et les schèmes est que le schème est par lui-même opérationnel : il n'est pas déclaratif, et n'a nul besoin d'être opérationalisé : il s'applique de lui-même. Au contraire, les actions décrites selon les termes de notre modèle ont encore besoin d'une étape d'opérationalisation avant de pouvoir être appliquées.

Cependant, il existe dans notre modèle un cas particulier : celui des actions terminales. Toutes les similitudes avec les schèmes précédemment décrites restent valables dans le cas des actions terminales. Mais ces actions ont la particularité de n'être pas décrites de manière déclarative dans notre modèle : nous avons vu que ces actions doivent être codées en dur dans le système. Cela signifie qu'il n'existe aucune version déclarative de nos actions terminales, mais que celles-ci n'existent dans le système que par le programme informatique qui les met en œuvre, autrement

dit qui les opérationalise. En cela, notre concept d'action terminale se rapproche considérablement de la notion de schème piagétien.

3.7.6 Correction *vs.* anticipation

Nous avons précédemment défini une action parallèle comme une action dont les sous-actions «peuvent être exécutées en même temps, chacune étant indépendante de la résolution des autres». Ainsi, dans la première étape de l'opération de saisie, l'ouverture de la main est indépendante du déplacement du bras. Il est à noter que dans cette définition du mode parallèle, le parallélisme *n'est pas obligatoire* : il est tout à fait possible d'exécuter les sous-actions l'une après l'autre, dans un ordre quelconque, ou même par petits morceaux. Face à cette caractéristique de notre modèle, nous devons nous interroger sur l'existence et le statut d'éventuelles opérations dont le parallélisme serait *nécessaire*.

De telles opérations peuvent s'envisager quand les actions en lesquelles elles se développent ne sont pas indépendantes les unes des autres. Considérons par exemple les actions terminales «déplacer le bras à tel endroit» et «orienter la main selon tel angle». Imaginons de plus une opération de transport nécessitant que l'objet soit maintenu dans une position verticale. Il est naturel d'envisager que cette opération se décompose en un parallélisme *obligatoire* entre deux branches persistantes composées des actions «déplacer le bras à l'endroit souhaité» et «orienter la main horizontalement». Cette dernière serait d'ailleurs mieux qualifiée de «conserver l'horizontalité de la main» dans ce contexte.

Étant donnée notre définition du mode parallèle, une telle décomposition n'est pas représentable par notre modèle. Cependant, il ne s'agit pas d'une faiblesse de ce modèle, mais bien d'une incorrection dans la représentation de l'opération. En effet, les deux actions considérées ne sont pas totalement indépendantes l'une de l'autre, puisque le déplacement du bras (sans mouvement du poignet) entraîne une modification de l'orientation de la main. La représentation proposée pour l'opération de transport suggère que cette opération est effectuée par *correction* des effets de bords introduits par le déplacement du bras : l'action «déplacer le bras» entraîne une modification de l'orientation de la main, et l'action «conserver l'horizontalité de la main» vient corriger *après coup* l'inclinaison de celle-ci. Cette représentation est incorrecte sur le plan cognitif car l'humain «connaît» inconsciemment ces effets de bords, sait anticiper la modification de l'inclinaison de sa main et génère simultanément des commandes musculaires pour le bras et le poignet. Il opère donc par *anticipation* et non par *correction*, et considère l'orientation de la main comme un attribut du déplacement plutôt que comme une action distincte. L'opération de transport reste par conséquent un tout non dissociable.

Ce point suggère donc que notre modèle ne souffre d'aucune insuffisance, mais que si le besoin d'un parallélisme obligatoire se fait sentir, l'action en question doit être considérée comme terminale de manière à pouvoir *anticiper* tous ses paramètres plutôt que d'en *corriger* certains. Ceci implique également que par soucis d'ergonomie, l'opérateur devrait ne disposer que d'*une seule* commande afin de l'exécuter.

3.7.7 Validation syntaxique et sémantique des opérations

Le modèle de représentation des connaissances que nous venons de décrire permet d'articuler entre eux des concepts d'action. Cela signifie que toute opération représentée selon les termes de ce modèle est formulée selon des règles syntaxiques parfaitement définies. Un système utilisant ce modèle de représentation des connaissances aura par conséquent la faculté de vérifier la *validité syntaxique* des connaissances exprimées par son moyen.

Ce modèle de représentation des connaissances ne traite cependant en aucune manière de la *validité sémantique* des actions, et ce n'est d'ailleurs pas son rôle. En particulier, il est tout à fait possible de représenter des actions syntaxiquement correctes, mais sémantiquement erronées. Par exemple, on pourrait définir une action parallèle, composée de deux branches persistantes, l'une ouvrant la main, l'autre la fermant. Une telle action ne veut évidemment rien dire. De même, la réutilisabilité des actions nous laisse la possibilité syntaxique d'introduire des boucles :

il est syntaxiquement correct d'écrire que la saisie d'un objet passe par la saisie de cet objet. Ces actions, quoi que sémantiquement erronées, n'en sont pas moins représentable selon les termes de notre modèle. Lors de l'élaboration d'un système de détection d'intentions utilisant ce modèle, il incombe au concepteur du système de s'assurer que l'ensemble des connaissances inculquées au système sont sémantiquement correctes.

3.7.8 Configurations particulières

Il existe un ensemble conséquent de combinaisons possibles entre mode d'action et qualité de branche. Parmi toutes ces combinaisons, deux d'entre elles appellent des remarques spécifiques.

Terminaison par une branche auxiliaire

Le premier cas particulier est celui d'une opération séquentielle se terminant par une branche auxiliaire. Concrètement, une telle configuration signifie que le but final de l'opération est atteint *avant* que la dernière action soit accomplie. Cette dernière action est donc en quelque sorte «optionnelle». À titre d'exemple, nous pourrions décider que l'opération de saisie d'un objet se termine par une branche consistant à soulever partiellement l'objet en question de manière à supprimer le contact avec l'endroit de pose. Une telle branche doit donc être auxiliaire dans la mesure où même si l'objet n'est pas légèrement soulevé, il n'en est pas moins saisi. Un tel cas de figure reste cependant exceptionnel.

Absence de branche persistante

Le deuxième cas particulier est celui d'une action, séquentielle ou parallèle, qui ne contiendrait que des branches auxiliaires. Concrètement, une telle configuration signifie que cette action, une fois résolue, ne peut jamais être invalidée. Ceci n'a visiblement aucun intérêt pour des opérations de haut niveau, mais peut éventuellement servir pour des actions n'apparaissant qu'en tant qu'élément d'une action supérieure. Dans ce cas, la sémantique d'une telle configuration indique qu'il est nécessaire et suffisant de résoudre l'action concernée une et une seule fois pour toutes les instances de l'opération supérieure. Par exemple, pour se connecter à de multiples reprises à une machine Unix, il est nécessaire d'obtenir un compte utilisateur une et une seule fois. Ici encore, un tel cas de figure reste exceptionnel dans la mesure où la supposition est faite qu'aucun événement extérieur ne peut réellement invalider l'action concernée (par exemple, le compte n'est jamais supprimé).

3.7.9 Équivalence sémantique d'actions séquentielles

Dans le cas d'une action séquentielle, où l'ordre d'exécution des sous-actions est parfaitement réglé, il est intéressant de constater que certaines configurations syntaxiques différentes sont équivalentes sur le plan sémantique. La figure 3.9 en page suivante illustre ce phénomène dans une configuration particulière qui nous servira par la suite.

Le théorème donné ci-dessous introduit une équivalence sémantique particulière qui nous servira par la suite (la démonstration est fournie en annexe). Ce théorème indique en fait que pour toute action séquentielle, n'importe quelle partie *finale* de la séquence peut être regroupée en une nouvelle branche persistante, à condition que chaque branche initiale conserve sa qualité son ordre d'apparition. Ce résultat est relativement intuitif, dans la mesure où la résolution d'une sous-action conditionne en fait la résolution de tout le reste de la séquence, mais pas ce qui la précède.

Théorème: Théorème d'équivalence sémantique

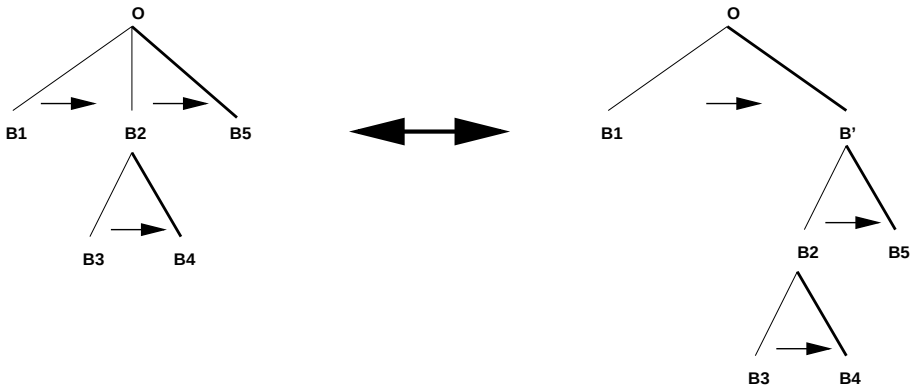

FIG. 3.9: Exemple d'équivalence sémantique

Soit $S = \{B_1 \rightarrow B_2 \rightarrow \ldots \rightarrow B_n\}$ *une action séquentielle.*
Soit $A = \{B_i \rightarrow B_{i+1} \rightarrow \ldots \rightarrow B_n\}$ *une action séquentielle, avec* $i > 1$.
Soit $S' = \{B_1 \rightarrow \ldots \rightarrow B_{i-1} \rightarrow B\}$ *une action séquentielle,* B *étant une branche de qualité persistante dont l'action associée est* A.
Alors, S *et* S' *sont sémantiquement équivalentes.* ∎

Chapitre 4

Détection d'Intentions

Grâce au modèle de représentation des connaissances développé dans le chapitre précédent, un système artificiel peut connaître l'ensemble des actions qu'un opérateur est susceptible d'entreprendre, ainsi que la manière dont ces actions doivent être exécutées. Dans ce chapitre, nous nous intéressons à la mise en œuvre de ce modèle, c'est à dire à la manière dont les connaissances représentées peuvent être exploitées à des fins de détection d'intentions. Pour ce faire, nous introduisons un certain nombre de concepts supplémentaires s'appliquant aux connaissances décrites par notre modèle, et une collection d'algorithmes permettant d'implémenter un mécanisme de détection d'intentions.

4.1 Taux de vraisemblance d'une opération

Dans le chapitre *Généralités* de cette troisième partie, nous avons montré le besoin de définir un «taux de vraisemblance» d'une action : le fait qu'une action soit *peut-être* en cours d'exécution, et qu'il faille par conséquent la détecter ne peut être un facteur booléen. Nous avons au contraire besoin d'une mesure continue de la vraisemblance, pour laquelle une détection sera déclenchée au delà d'un certain seuil de confiance. Cette section a pour objet de définir une telle mesure.

4.1.1 Composition des taux de vraisemblance

Dans notre modèle de représentation des connaissances, la notion d'action est autoréférente, c'est à dire qu'une action est par définition décomposée en actions de plus bas niveau, ses sous-actions. Cette caractéristique de notre modèle nous impose de suivre la même philosophie pour définir un taux de vraisemblance : le taux de vraisemblance d'une action doit être une composition, selon certaines règles, des taux de vraisemblance de ses sous-actions. Étant donné que notre modèle définit deux catégories d'actions (les actions parallèles et les actions séquentielles), il est naturel d'examiner la sémantique du taux de vraisemblance dans ces deux cas.

Actions parallèles

Une action parallèle consiste à résoudre plusieurs buts de plus bas niveau simultanément, ces buts étant indépendants les uns des autres. De plus, le parallélisme est optionnel : il signifie simplement que l'ordre de résolution des sous-buts n'a pas d'importance. Cette caractéristique

suggère une composition additive des taux de vraisemblance : le taux de vraisemblance d'une action parallèle doit être une fonction de la somme des taux de vraisemblance de ses sous-actions. Intuitivement, cela signifie que pour qu'une action parallèle soit vraisemblable, il faut que chacune de ses sous-actions le soit.

En première approximation, nous obtenons donc le résultat suivant : soit une action parallèle P décomposée en sous-actions $\{A_1, A_2 \dots A_N\}$. On note τ_i le taux de vraisemblance, dans l'intervalle $[0, 1]$, de A_i. Le taux de vraisemblance $\tau(P)$ de P est alors défini par :

$$\tau(P) = \frac{\sum_{i=1}^{N} \tau_i}{N}$$

Le rapport à N permet d'assurer que le taux de vraisemblance est une valeur comprise dans l'intervalle $[0, 1]$.

Actions séquentielles

La notion de vraisemblance pour une action séquentielle est un peu moins intuitive que dans le cas précédent : une action séquentielle consiste à résoudre plusieurs buts de plus bas niveau les uns après les autres, chaque résolution conditionnant de démarrage de la résolution suivante. Cette caractéristique suggère l'intervention d'une composition multiplicative dans le taux de vraisemblance. En fait, il n'est pas tout à fait exact de dire que la résolution d'une sous-action conditionne celle de la suivante. Elle conditionne en effet la résolution de *tout le reste* de la séquence : si une sous-action n'est pas résolue, ce n'est pas seulement la suivante qui ne peut pas démarrer, mais toutes les autres. Par conséquent, le taux de vraisemblance d'une sous-action ne doit pas seulement influer sur celui de la sous-action suivante, mais sur celui de tout le reste de la séquence. Ceci suggère l'introduction d'un taux de vraisemblance par *séquence* et non plus seulement par *action*, taux qui serait pondéré par le taux de vraisemblance de la sous-action précédente. Intuitivement, cela signifie qu'une séquence est d'autant plus vraisemblable que la sous-action qui la précède l'est.

En première approximation, nous obtenons donc le résultat suivant : soit une action S séquentielle décomposée en sous-actions $\{A_1, A_2 \dots A_N\}$. On note τ_i le taux de vraisemblance, dans l'intervalle $[0, 1]$, de A_i et λ_i celui de la séquence $\{A_i, A_{i+1} \dots A_N\}$, dans l'intervalle $[0, N - i + 1]$. Le taux de vraisemblance $\tau(S)$ de S est alors défini par :

$$\lambda_i = \tau_i.(1 + \lambda_{i+1})$$

$$\tau(S) = \frac{\lambda_1}{N}$$

ou bien encore, une fois développé :

$$\tau(S) = \frac{\tau_1 + \tau_1.[\tau_2 + \tau_2.(\dots + \tau_N)]}{N}$$

De même que dans le cas des actions parallèles, le rapport à N permet d'assurer que le taux de vraisemblance reste dans l'intervalle $[0, 1]$. Ainsi, pour une action séquentielle S décomposée en seulement deux sous-actions $\{A_1, A_2\}$, on obtient $\tau(S) = \lambda_1/2 = \tau_1(1 + \tau_2)/2$.

4.1.2 Pondération des actions

La composition additive ou multiplicative des taux de vraisemblance selon le mode des actions est assez intuitive. Elle ne suffit cependant pas à définir ce taux d'une manière satisfaisante, car elle ne prend pas en compte la complexité des actions considérées. Dans notre modèle de représentation des connaissances, il est possible de trouver à un même niveau de profondeur des actions de complexité très différente. À titre d'exemple et en vertu du théorème d'équivalence sémantique donné en page 150, l'action de saisie peut se représenter de manière sémantiquement

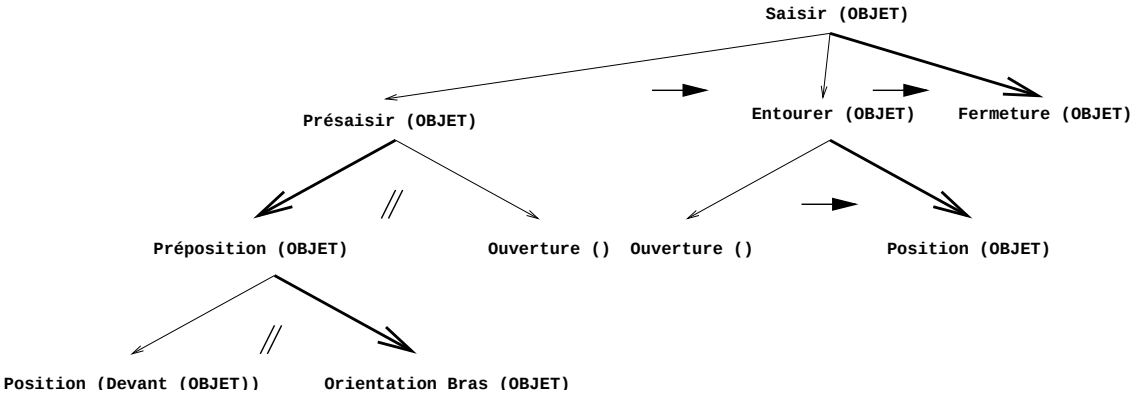

FIG. 4.1: Représentation équivalente de la saisie

équivalente en supprimant l'action «attraper l'objet» et en la décomposant directement au niveau supérieur, comme illustré sur la figure 4.1.

Même si cette représentation est plus éloignée du résultat expérimental, aucune donnée théorique ne nous permet d'écarter l'hypothèse d'une configuration analogue dans d'autres situations. Dans ce cas précis, nous voyons que l'arbre est déséquilibré dans la mesure où il positionne sur un même plan une action terminale et une action à deux degrés de profondeur.

Les taux de vraisemblance, tels qu'ils sont pour l'instant définis, aboutissent à deux conséquences fâcheuses, l'une pragmatique et l'autre théorique.

- Considérons notre nouvelle représentation de l'opération de saisie, et imaginons que les deux premières étapes sont résolues. Leurs taux de vraisemblance respectifs seraient donc à 1. Le taux de vraisemblance de l'opération de saisie vaut dans ce cas 2/3, ce qui signifie que la saisie de l'objet est résolue à 66%. Ce résultat paraît très peu conforme à notre attente. D'un point de vue pragmatique, on aurait tendance à trouver l'opération de saisie bien plus achevée qu'à 66% quand il ne reste plus qu'à fermer la main.

- Mais bien plus important que cela, ce résultat n'est pas satisfaisant sur un plan théorique, dans la mesure où le calcul du taux de vraisemblance donne des résultats différents pour des représentations équivalentes d'une même action. En effet, toujours selon les même hypothèses, le taux de vraisemblance de l'opération de saisie, représentée dans sa version d'origine sur la figure 3.8 en page 142 est évalué à 75%, taux supérieur à celui obtenu dans la représentation équivalente décrite sur la figure 4.1.

Ces deux arguments montrent qu'il est essentiel de prendre en compte la complexité des actions dans le calcul du taux de vraisemblance. Plus une action est complexe, plus son influence doit être grande. Nous sommes par conséquent conduit à pondérer les actions, c'est à dire qu'à chaque action est associé un poids, et que les taux de vraisemblance des actions sont pondérées par ce facteur. D'une manière naturelle, le poids d'une action sera défini comme la somme des poids de ses sous-actions.

☛ Il est intéressant de constater que dans ces conditions, les actions de poids minimum sont les actions terminales, et que les poids augmentent en même temps que la complexité des opérations. Le poids d'une action est donc d'une certaine manière un indicateur du niveau conceptuel de celle-ci.

4.1.3 Formalisation

Nous disposons maintenant de tous les éléments nécessaires à la formalisation du concept de taux de vraisemblance. Cette section présente formellement ce concept grâce à un ensemble de définitions.

Définition 21: Poids d'une action

*Le **poids** d'une action est défini comme la somme des poids des sous-actions qui la constituent. Ainsi, soit une action A développée en un ensemble de sous-actions $\{A_1, A_2 \dots A_N\}$. On note ω_i le poids de l'action A_i. Le poids $\omega(A)$ de A est alors défini par :*

$$\omega(A) = \sum_{i=1}^{N} \omega_i$$

■

Définition 22: Taux de vraisemblance (parallèle)

Soit une action parallèle P développée en un ensemble de sous-actions $\{A_1, A_2 \dots A_N\}$. On note τ_i le taux de vraisemblance de A_i, dans l'intervalle $[0, 1]$. Le taux de vraisemblance $\tau(P)$ de P est alors défini par :

$$\tau(P) = \frac{\sum_{i=1}^{N} \omega_i \tau_i}{\sum_{i=1}^{N} \omega_i} = \frac{\sum_{i=1}^{N} \omega_i \tau_i}{\omega_P}$$

■

Définition 23: Taux de vraisemblance (séquentiel)

Soit une action séquentielle S développée en un ensemble de sous-actions $\{A_1, A_2 \dots A_N\}$. On note τ_i le taux de vraisemblance de A_i, dans l'intervalle $[0, 1]$, et λ_i celui de la séquence $\{A_i, A_{i+1} \dots A_N\}$, dans l'intervalle $[0, N - i + 1]$. Le taux de vraisemblance λ_i de la séquence $\{A_i, A_{i+1} \dots A_N\}$ est alors défini par :

$$\lambda_i = \tau_i.(\omega_i + \lambda_{i+1})$$

et le taux de vraisemblance de S est alors défini par :

$$\tau(S) = \frac{\lambda_1}{\sum_{i=1}^{N} \omega_i} = \frac{\lambda_1}{\omega_S}$$

ou encore, une fois développé, par :

$$\tau(S) = \frac{\omega_1\tau_1 + \tau_1.[\omega_2\tau_2 + \tau_2.(\dots + \omega_N\tau_N)]}{\omega_S}$$

■

Enfin, comme nous l'avons déjà suggéré dans le chapitre *Généralité*, une action devrait être considérée comme détectable si son taux de vraisemblance dépasse un certain seuil :

Définition 24: Seuil de confiance

*On défini le **seuil de confiance** du système de détection d'intentions comme la valeur minimum du taux de vraisemblance pour laquelle une action est considérée comme détectée.* ■

Vocabulaire: Plausibilité d'une action

*On dira qu'une action est **plausible** si son taux de vraisemblance est au moins égal au seuil de confiance du système de détection d'intentions.* ●

☛ Notons que la généricité de notre modèle de représentation des connaissances nous a à nouveau permis de fournir un ensemble de définition autoréférentes. Pour une application donnée, seuls les poids et les taux de vraisemblance des actions terminales doivent être codés en dur. Tous les autres paramètres en découlent.

4.2 Algorithmes de détection

Dans cette section, nous décrivons l'ensemble des algorithmes qui permettent d'implémenter un mécanisme de détection d'intentions. Ces algorithmes utilisent les connaissances inculquées au système selon le modèle décrit précédemment. Les algorithmes ayant pour but de calculer le taux de vraisemblance de chaque action, ainsi que ceux concernant la gestion des détections par le système sont décrits.

4.2.1 Algorithme préliminaire

Pour détecter une action, nous devons théoriquement connaître en permanence son taux de vraisemblance, puis signaler au système quand ce taux a dépassé le seuil de confiance de celui-ci. Dans une première approche, ce comportement simple est décrit par l'algorithme 1.

Algorithme 1: Algorithme préliminaire

```
DÉTECTER(Action)
(1)   // Initialisation
(2)   Lancer la détection de toutes les branches.
(3)   // Boucle infinie
(4)   Pour toujours:
(5)     Recalculer le taux de vraisemblance.
(6)     si l'action est plausible et détectable:
(7)       Signaler la détection au système.
```

➤ Le lancement de la détection de toutes les sous-actions, à la ligne 2, fait appel à ce même algorithme, et assure en particulier que le taux de vraisemblance de chaque sous-action est disponible en permanence.

➤ La deuxième partie du test figurant à la ligne 6 fait référence au problème du choix dans les actions à détecter décrit dans le chapitre *Généralités*. Les actions détectables sont supposées déclarées par le concepteur du système.

Il existe deux raisons qui rendent cette version simpliste de la détection d'une action non satisfaisante :

➤ Premièrement, le fait de lancer systématiquement toutes les détections de plus bas niveau est un comportement grossier et susceptible de gaspiller des ressources. En effet, il existe des situations dans lesquelles il est inutile de lancer certaines détections. Pour les actions séquentielles, par exemple, on peut toujours déterminer un ensemble $\{A_1, A_2 \dots A_k\}$ de sous-actions et un ensemble $\{\tau_1, \tau_2 \dots \tau_k\}$ de taux de vraisemblance associés pour lesquels la configuration (A_i, τ_i) rend impossible la plausibilité de l'action elle-même. L'exemple le plus trivial est celui de la configuration $(A_1, 0)$. Lorsqu'une telle situation apparaît, il devient inutile de lancer la détection de tout les reste de la séquence.

- Deuxièmement, cet algorithme ne permet pas de gérer correctement le problème de la multiplicité des détections, problème également évoqué dans le chapitre «Généralités». Ce problème apparaît, comme nous l'avons dit, quand plusieurs actions détectables se trouvent imbriquées au sein d'une même hiérarchie. Dans un tel cas, plusieurs configurations devraient être gérées : premièrement, si deux actions imbriquées deviennent plausibles au même moment, elles devraient être proposées à l'opérateur par ordre de niveau conceptuel décroissant. On ne peut donc pas simplement signaler une détection au système comme cela est fait à la ligne 7 de l'algorithme 1 en page précédente. Deuxièmement, en admettant que l'opérateur confirme la détection de l'une de ces actions, son taux de vraisemblance prend instantanément une valeur de 1 puisqu'elle est effectivement intentée par l'opérateur. Cette nouvelle valeur du taux de vraisemblance est alors susceptible de modifier l'évaluation de toutes les autres détections en cours.

4.2.2 Généralisation du seuil de confiance

Les problèmes soulevés par l'algorithme préliminaire suggèrent en fait de généraliser l'usage du seuil de confiance à la gestion des branches d'une action aussi bien qu'à l'action elle-même.

L'idée est la suivante : l'usage actuel du seuil de confiance indique que le système de détection d'intentions n'est «intéressé» par une action que lorsque celle-ci devient plausible. Le reste du temps, il ne se soucie pas de la valeur de son taux de vraisemblance. Nous avons exploité cette idée dans l'algorithme préliminaire à la ligne 7 par l'utilisation d'un *signal* destiné au système. Algorithmiquement parlant, cela signifie que le calcul du taux de vraisemblance d'une action est à sa propre charge, et que le système a pour fonction principale d'*attendre* que ce taux de vraisemblance dépasse le seuil de confiance pour agir.

Cette idée peut être généralisée à la gestion des branches d'une action : l'algorithme de détection d'une action ne serait alors «intéressé» par une branche que lorsque celle-ci serait plausible. Le reste du temps, il considérerait son taux de vraisemblance comme nul. La fonction principale d'un tel algorithme serait en conséquence d'attendre que les branches deviennent plausibles, et réagir de manière appropriée. Cette idée est développée séparément dans les cas des actions parallèles et séquentielles comme suit.

4.2.3 Détection d'une action parallèle

Une action parallèle est résolue quand chacune de ses sous-action l'est. De la même manière, nous considérons qu'une telle action est plausible quand chacune de ses sous-actions l'est. Cela signifie en particulier que tant qu'une branche est non plausible, il est inutile de mettre à jour le taux de vraisemblance de l'action. Un nouveau schéma pour la détection d'une action parallèle est donc proposé par l'algorithme 2 ci-contre.

- La procédure d'initialisation démarrant à la ligne 2 tient compte des deux types de qualité d'une branche. Quand une action est résolue, seules ses branches persistantes peuvent invalider son résultat. Par conséquent, la détection des branches auxiliaires ne doit pas être poursuivie.
- Le test figurant à la ligne 8 permet de bloquer (à la ligne 9) l'exécution du processus de détection tant que toutes les branches ne sont pas devenues plausibles. Le taux de vraisemblance de l'action est pendant ce temps considéré comme nul. Si ce test échoue, le processus de détection doit tourner en permanence de manière à mettre constamment à jour le taux de vraisemblance de l'action. Dans ces conditions, le processus de détection doit cependant rester réceptif aux signaux émis par les sous-actions, d'où le test non bloquant figurant à la ligne 11.
- Le taux de vraisemblance de l'action est calculé à la ligne 12. Cette ligne est malgré tout susceptible d'être exécutée (une seule fois) alors qu'une sous-action est non plausible (auquel cas le taux de vraisemblance sera considéré comme nul) : ce cas de figure intervient si une

Algorithme 2: Détection d'une action parallèle

```
DETECTER(Action parallèle)
(1)   // Initialisation
(2)   si l'action est résolue:
(3)     Lancer la détection des branches persistantes seulement.
(4)   sinon:
(5)     Lancer la détection de toutes les branches.
(6)   // Boucle infinie
(7)   Pour toujours:
(8)     si une branche est non plausible:
(9)       Attendre un signal de changement d'état.
(10)    sinon:
(11)      Tester (sans attendre) l'arrivée d'un signal de changement d'état.
(12)    Recalculer le taux de vraisemblance.
(13)    si l'action change d'état:
(14)      Interrompre toutes les branches.
(15)      si l'action est détectable et plausible:
(16)        L'ajouter à la liste des détections.
(17)      Signaler le changement d'état.
(18)    // Pas de changement d'état
(19)    sinon:
(20)      si un signal a été reçu et la liste des détections est non vide:
(21)        Interrompre toutes les branches.
(22)        Signaler la détection.
(23)      // Pas de signal reçu
(24)      sinon:
(25)        si l'action est résolue:
(26)          Reprendre la détection des branches persistantes seulement.
(27)        sinon:
(28)          Reprendre la détection de toutes les branches.
```

sous-action change d'état pour devenir non plausible, ce qui génère un signal récupéré à la ligne 9.

- Une action en cours de détection peut avoir trois états : non plausible, plausible et résolue. Cependant, toutes les transitions ne sont pas considérées comme des changements d'état dans le test de la ligne 13. En effet, la transition de l'état plausible à l'état résolu n'a pas à être signalée. L'action supérieure (ou le système) sera par contre intéressé(e) par toutes les autres transitions.
- Les lignes 17 et 22 interrompent le déroulement de l'algorithme. Celui-ci sera éventuellement repris par le passage à l'équivalent des lignes 26 ou 28 de l'algorithme supérieur, ou sur commande du système.
- Les lignes 16, 17et 22 permettent de gérer le problème des détections multiples : quand une action détectable devient plausible, le signal de son changement d'état remonte jusqu'au système, même si les actions supérieures qu'il traverse ne sont toujours pas plausibles.

4.2.4 Détection d'une action séquentielle

Dans le cas d'une action séquentielle, pour qu'une sous-action soit résolue, il faut en particulier que toutes les sous-actions précédentes le soient. De la même manière, nous considérons que pour qu'une sous-action soit plausible, il faut également que les sous-actions précédentes le soient. Cela signifie en particulier que tant qu'une branche est non plausible, il est inutile de

lancer la détection des branches suivantes. Un nouveau schéma pour la détection d'une action séquentielle est donc proposé par l'algorithme 3.

Algorithme 3: Détection d'une action séquentielle

```
DETECTER(Action séquentielle)
(1)   // Initialisation
(2)   si l'action est résolue:
(3)     Lancer la détection des branches persistantes seulement.
(4)   sinon:
(5)     Lancer la détection de toutes les branches, jusqu'à la première branche
        non plausible.
(6)   // Boucle infinie
(7)   Pour toujours:
(8)     si la première branche est non plausible:
(9)       Attendre un signal de changement d'état.
(10)    sinon:
(11)      Tester (sans attendre) un signal de changement d'état.
(12)    Recalculer le taux de vraisemblance.
(13)    si l'action change d'état:
(14)      Interrompre toutes les branches.
(15)      si l'action est détectable et plausible:
(16)        L'ajouter à la liste des détections.
(17)      Signaler le changement d'état.
(18)    // Pas de changement d'état
(19)    sinon:
(20)      si un signal a été reçu et la liste des détections est non vide:
(21)        Interrompre toutes les branches.
(22)        Signaler la détection.
(23)      // Pas de signal reçu
(24)      sinon:
(25)        si l'action est résolue:
(26)          Reprendre la détection des branches persistantes seulement.
(27)        sinon:
(28)          Reprendre la détection de toutes les branches, jusqu'à la première
              branche non plausible.
```

Cet algorithme est très similaire à celui de la détection d'une action parallèle. Les différences sont les suivantes :

- Les lignes 5 et 28 tiennent compte de la séquentialité de l'action : lorsqu'une branche est non plausible, il est inutile de lancer la détection de toute la séquence qui suit.
- La ligne 8 bloque le processus de détection seulement pendant la période où la première branche est non plausible. Ceci pourrait faire l'objet d'une optimisation plus poussée, en déterminant, comme nous l'avons dit précédemment jusqu'à quelle sous-action et pour quel taux de vraisemblance la plausibilité de l'action concernée est impossible.

4.2.5 Actions englobantes

Lors de l'élaboration de notre modèle, nous avons noté que les actions sont réutilisables, c'est à dire qu'une même action peut faire partie de plusieurs opérations distinctes. Par exemple, dans TAS_t, l'action `Position ()` intervient plusieurs fois dans l'opération de saisie, et intervient également dans celle de pose. Réciproquement, pour un ensemble de connaissances donné, il existe

un certain nombre d'actions qui ne sont *jamais* utilisées en tant que sous-action d'une autre action. Ces actions, que l'on pourrait qualifier d'actions «englobantes», sont en fait les opérations de plus haut niveau que le système connaisse, et constituent en quelque sorte les «racines de l'arborescence des connaissances, l'opposé du concept d'action terminale.

A priori, il semble que pour instancier toute la connaissance E du système, c'est à dire toutes les actions pour lesquelles il possède une représentation, il suffise d'instancier toutes ses actions englobantes. En effet, une action non englobante est par définition utilisée comme sous-action, et sera donc instanciée au moment de l'instanciation de l'action supérieure. Supposons par exemple que le système connaisse les opérations de saisie, transport et pose d'un objet. Seule l'opération de transport est *a priori* une action englobante. En effet, les actions de saisie et pose sont utilisées en tant que sous-action par cette opération.

Cependant, supposons maintenant que certains objets de la situation considérée soient saisissables, mais *non transportables* (une poignée de porte par exemple). Dans ce cas, l'action de saisie devient également englobante *pour ces objets*. En effet, comme il est impossible d'instancier une opération de transport de ces objets, l'action de saisie devient l'action de plus haut niveau que le système puisse détecter pour ces objets.

Ce point montre que la notion d'action englobante n'est pas définie pour un ensemble de connaissances E données, mais pour toutes les instanciations possibles de cette connaissance à un instant donné, en fonction de l'environnement. Ceci nous permet d'introduire la définition suivante :

Définition 25: Action englobante

Étant donné un ensemble $E = \{A_1, A_2 \dots A_n\}$ d'actions décrites selon notre modèle, et une instance de E, c'est à dire l'ensemble de toutes les actions possibles à un moment donné, on appelle ***action englobante*** *une action A_i qui ne figure jamais en tant que sous-action d'une autre action.* ■

Les algorithmes que nous venons de décrire ont une conséquence intéressante vis à vis des actions englobantes : comme tout signal de plausibilité d'une action détectable remonte vers le système, celui n'a en fait besoin de lancer initialement que la détection des actions englobantes, les autres étant implicitement incluses dans le développement de celles-ci. Par exemple, si notre système tente de détecter une intention de transport, il n'est pas nécessaire de lancer parallèlement la détection de l'intention de saisie, puisque celle-ci figure implicitement dans le développement de l'action englobante «transport». Cette caractéristique vient encore diminuer la quantité de ressources nécessaires à la mise en œuvre du mécanisme de détection d'intentions.

4.3 Cas des actions terminales

L'application des algorithmes décrits précédemment à un cas pratique ne nécessite plus que l'élaboration de deux points relatifs aux actions terminales : le calcul du taux de vraisemblance et les algorithmes de détection. Ces points constituent l'objet de cette section.

4.3.1 Taux de vraisemblance

Nous proposons de définir la notion de taux de vraisemblance d'une action terminale au travers d'un exemple implémenté dans ***TOAST*** : celui-ci de la mise en position de l'organe de préhension dans le plan vertical. Sur la figure 3.8 en page 142 cette action terminale était notée `Position ()`. Rappelons que l'opérateur de ***TOAST*** dispose d'un joystick afin d'opérer un déplacement du bras manipulateur dans ce même plan.

Problématique

Considérons donc que l'opérateur est en train de déplacer le bras manipulateur et que la détection d'une intention de positionnement à un endroit précis `(X,Y)` est en cours. Dans une première approche, on imagine que plus l'organe de préhension est proche de la position en question, plus l'action de positionnement est vraisemblable. Le taux de vraisemblance de cette action serait donc inversement proportionnel à la distance entre la position actuelle et la position souhaitée.

Cette définition possible n'est cependant pas entièrement satisfaisante : il peut en effet arriver que l'opérateur passe à proximité de la position en question, sans pour cela y diriger véritablement le bras manipulateur (par exemple, en un mouvement tangent). Dans ce cas, son but est autre, et la détection ne devrait pas avoir lieu.

Niveau d'activation

Ce problème montre que le seul examen du *résultat* d'une action est insuffisant : il faut également examiner la *manière* dont l'action est effectuée. Cela signifie en particulier que pour une action terminale donnée, le système de détection d'intentions doit avoir une idée de la méthode par laquelle celle-ci se résous, et comparer cette méthode avec celle qu'emploie l'opérateur. Si ces deux méthodes sont proches, *alors* une détection est envisageable selon le critère initial, c'est à dire la distance au but. La comparaison des méthodes d'exécution peut s'exprimer sous la forme d'une *mesure*, que nous appelons «niveau d'activation» d'une action. Plus la méthode d'exécution employée par l'opérateur est proche de celle qu'*aurait* employé le système, plus le niveau d'activation est important.

Exemple

Reprenons l'exemple de l'action `Position ()`. Quelle méthode de résolution doit-on inculquer au système ? En vertu des considérations développées dans le chapitre *Généralités*, nous avons établi que la connaissance du système devait décrire les opérations d'une manière simpliste. La méthode d'exécution la plus simple pour déplacer l'organe de préhension à une position donnée est bien entendu la ligne droite ; c'est également la méthode qui sera utilisée par l'opérateur lorsque le mouvement sera pratiquement achevé. Le niveau d'activation de l'action de positionnement s'obtient donc par comparaison des directions des vecteurs vitesse de déplacement ; celui utilisé couramment par l'opérateur d'une part, et celui donnant une ligne directe vers le point de destination d'autre part. Des directions identiques donnent un niveau d'activation unitaire, des directions opposées donnent un niveau nul.

Par la suite, le taux de vraisemblance de cette même action, si le seuil d'activation est franchi (ou si cette action n'est pas en cours d'exécution), est donné par une distance planaire entre les positions courantes et souhaitée. Des positions égales donnent un taux de vraisemblance unitaire, des positions suffisamment éloignées donnent un taux nul.

Prise en compte du niveau d'activation

La prise en compte du niveau d'activation d'une action terminale n'intervient pas dans toutes les situations : telle que nous l'avons décrite, elle trouve son utilité quand l'action terminale en cours de détection est également en cours d'exécution, et qu'elle n'est pas encore plausible. De même que pour les opérations de plus haut niveau, il existe des cas où le processus de détection sera en cours alors que l'action est déjà plausible : une telle situation se produit par exemple lorsque l'opération supérieure n'est pas encore plausible, ou bien, comme nous le verrons ultérieurement, lorsque l'opérateur n'a pas confirmé l'hypothèse de détection proposée. Un tel état est en fait proche de l'état de «veille» décrit dans le chapitre précédent. Dans de telles situations, l'action terminale redeviendra non plausible lorsque l'on se sera suffisamment écarté de son but, et ceci quelle que soit la méthode d'exécution employée par l'opérateur.

4.3.2 Pondération

Afin de prendre en compte les actions terminales dans le calcul du taux de vraisemblance des opérations, il est non seulement nécessaire de connaître leur propre taux de vraisemblance, décrit dans la section précédente, mais également leur poids. Le poids d'une action terminale est nécessairement câblé. Nous devons donc nous demander comment le déterminer.

La première idée qui vient est d'assigner un poids unitaire aux actions terminales. Cependant, comme nous l'avons noté précédemment, le poids d'une action quelle qu'elle soit est représentatif de sa complexité opératoire. Ceci devrait donc être également le cas pour les actions terminales. Dans TAS_t, il existe des actions terminales de complexité variable. Certaines d'entre elles comme l'orientation du bras autour de son axe vertical agissent en dimension 1, tandis que d'autres, comme le positionnement de l'organe de préhension dans le plan vertical agissent en dimension 2. Nous avons choisi d'assigner les poids des actions terminales en fonction de ce critère. Une démarche similaire doit être entreprise pour chaque type de manipulateur.

La figure 4.2 représente à nouveau l'opération de saisie, en faisant figurer l'ensemble des poids de chaque action. Seuls les poids terminaux sont connus à l'avance.

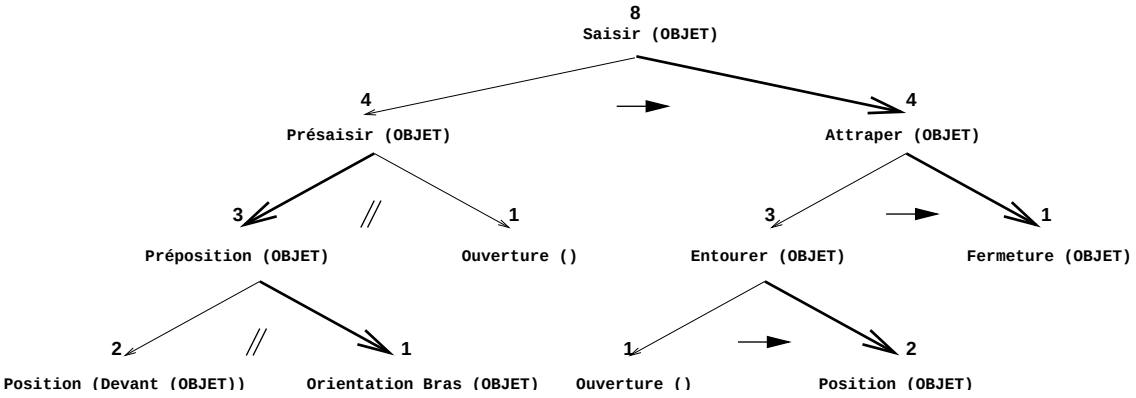

FIG. 4.2: Pondération de l'opération de saisie.

Pour reprendre le cas de figure cité précédemment, en supposant que l'objet à saisir soit déjà entouré, le taux de vraisemblance de l'opération de saisie s'évalue à $[4 + (3 + 0)]/8 = 7/8$, soit 87,5%.

4.3.3 Algorithme de détection

Le schéma de détection pour une action terminale est donné par l'algorithme 4 en page suivante.

- Les considérations sur le changement d'état et la multiplicité des détections évoquées pour les actions parallèles et séquentielles restent valables dans le cas d'une action terminale.
- Les tests figurant de la ligne 2 à la ligne 15 permettent d'assurer que le niveau d'activation de l'action est utilisé à bon escient.

4.3.4 Formalisation

À l'issue de ces considérations, nous proposons la formalisation suivante, pour les actions terminales, constituée de trois définitions :

Définition 26: **Niveau d'activation**

Algorithme 4: Détection d'une action terminale

```
DETECTER(Action terminale)
(1)   Pour toujours:
(2)     si l'action n'est pas en cours d'exécution:
(3)       Recalculer le taux de vraisemblance (distance au but).
(4)     // L'action est en cours d'exécution
(5)     sinon:
(6)       si l'action est plausible ou résolue:
(7)         Recalculer le taux de vraisemblance (distance au but).
(8)       // L'action est non plausible
(9)       sinon:
(10)        Calculer le niveau d'activation.
(11)        si le niveau d'activation est supérieur au seuil d'activation:
(12)          Recalculer le taux de vraisemblance (distance au but).
(13)        // Le taux d'activation est trop faible
(14)        sinon:
(15)          Le taux de vraisemblance est nul.
(16)    si l'action change d'état:
(17)      si l'action est détectable et plausible:
(18)        L'ajouter à la liste des détections.
(19)      Signaler le nouvel état.
```

Soit une action terminale T, pour laquelle le système de détection d'intentions dispose d'une méthode de résolution M_s. Soit M_o la méthode de résolution employée par l'opérateur. On définit le ***niveau d'activation*** *de l'action T comme une distance dans l'intervalle $[0, 1]$ entre M_s et M_o.* ■

Définition 27: Seuil d'activation

On définit le ***seuil d'activation*** *comme la valeur du niveau d'activation à partir de laquelle une action terminale pourra être détectée.* ■

Définition 28: Taux de vraisemblance d'une action terminale

Quand il y a lieu de calculer le taux de vraisemblance d'une action terminale, ce taux est défini comme une distance au but comprise dans l'intervalle $[0, 1]$. Dans le cas contraire, ce taux est nul. Les conditions dans lesquelles le taux de vraisemblance doit être calculé sont définies par l'algorithme 4. ■

4.4 Algorithme système

Nous disposons maintenant de l'ensemble des algorithmes et des critères de détection requis pour utiliser convenablement les connaissances décrites au moyen de notre modèle. Dans cette section nous décrivons le cœur du système de détection d'intentions, c'est à dire l'algorithme destiné à gérer l'interaction avec l'opérateur, contrôler le manipulateur, et procéder à la détection d'intentions. L'algorithme 5 ci-contre décrit le fonctionnement du système de détection d'intentions. Cet algorithme est présenté de manière simplifiée, afin de mettre en évidence les principaux aspects théoriques développés dans cette partie. Pour une description plus complète des fonctionnalités qui ne sont pas représentées ici, le lecteur pourra se référer au chapitre suivant ainsi qu'à

l'annexe 1.

Algorithme 5: Algorithme système

```
SYSTÈME()
(1)   Pour toujours:
(2)     Attendre un événement.
(3)     cas d'un événement extérieur:
(4)       Redéfinir l'ensemble des opérations englobantes.
(5)       Initialiser et lancer les nouvelles détections.
(6)     cas d'une détection:
(7)       tant qu'il reste des questions à poser à l'opérateur:
(8)         Demander confirmation à l'opérateur pour la détection de plus
            haut niveau.
(9)         si l'opérateur confirme:
(10)          Considérer que le taux de vraisemblance vaut 1, et recalculer la
              plausibilité des détections supérieures en cours.
(11)        sinon:
(12)          Cette opération entre en phase d'antidétection.
(13)      si une action est confirmée:
(14)        Interrompre et réinitialiser toutes les détections.
(15)        Lancer l'exécution de cette action.
(16)    cas d'une antidétection:
(17)      Il s'agit d'une action englobante : relancer sa détection.
(18)    cas d'une action autonome:
(19)      Informer l'opérateur du résultat.
(20)      si l'opération a échoué:
(21)        Éventuellement, laisser l'opérateur agir, puis reprendre l'exécu-
            tion.
(22)      sinon:
(23)        Relancer les détections.
(24)    cas d'une commande opérateur:
(25)      si une action est en cours d'exécution:
(26)        Éventuellement, donner temporairement le contrôle à l'opérateur,
            ou annuler l'exécution.
```

Le système de détection d'intentions est essentiellement une boucle infinie dont la fonction principale est de traiter des événements. Ces événements sont d'origines diverses (opérateur, environnement, système lui-même) et déclenchent des comportements appropriés.

Chaque action détectable peut être dans deux phases différentes. La première est la phase de détection normale : l'opération n'est pas résolue et le système surveille son taux de vraisemblance. La deuxième est la phase d'*antidétection* : comme nous l'avons indiqué précédemment, si l'opérateur ne confirme pas l'hypothèse d'une détection, nous ne devons pas la lui proposer à nouveau tant que l'action correspondante n'a pas à nouveau franchi le seuil de confiance. Les actions non confirmées doivent donc être malgré tout surveillées jusqu'à ce qu'elles redeviennent non plausibles, puis plausibles à nouveau. On dira qu'elles sont en phase d'antidétection.

➤ À la ligne 3, on traite le cas des «événements extérieurs», c'est à dire ceux qui proviennent de l'environnement. Plus précisément, on appelle «événement extérieur» un événement perceptif susceptible de modifier l'ensemble des actions englobantes. Cela signifie en particulier qu'une modification de l'environnement ne constitue pas nécessairement un événement extérieur. Prenons le cas de $\mathcal{TOAS}_t$, qui connaît les actions de saisie et de pose. L'apparition (la disparition) d'un objet dans la scène constitue un événement extérieur dans la mesure où ces actions deviennent possibles (impossibles). Par contre, la modification de la

position de cet objet ne doit pas générer un tel événement.

- À la ligne 5, on lance les processus de détection des actions englobantes. Ces processus s'effectuent en parallèle avec le processus système et communiquent avec lui au moyen d'événements qui sont considérés comme *internes* au système.
- À la ligne 6, un tel événement est reçu : le processus de détection d'une action englobante a détecté la plausibilité d'une (ou plusieurs) sous-action(s) de son développement. Le système doit donc examiner la liste des actions plausibles, et demander confirmation à l'opérateur. Si l'opérateur confirme, on peut alors considérer que le taux de vraisemblance de l'action en question vaut 1, ce qui est susceptible de rendre d'autres actions de plus haut niveau plausibles. Cette évaluation doit être faite récursivement afin de proposer éventuellement une nouvelle détection d'un niveau supérieur à l'opérateur. La figure 4.3 montre $TOAS_t$ en phase de confirmation, après qu'une intention de saisie ait été détectée.

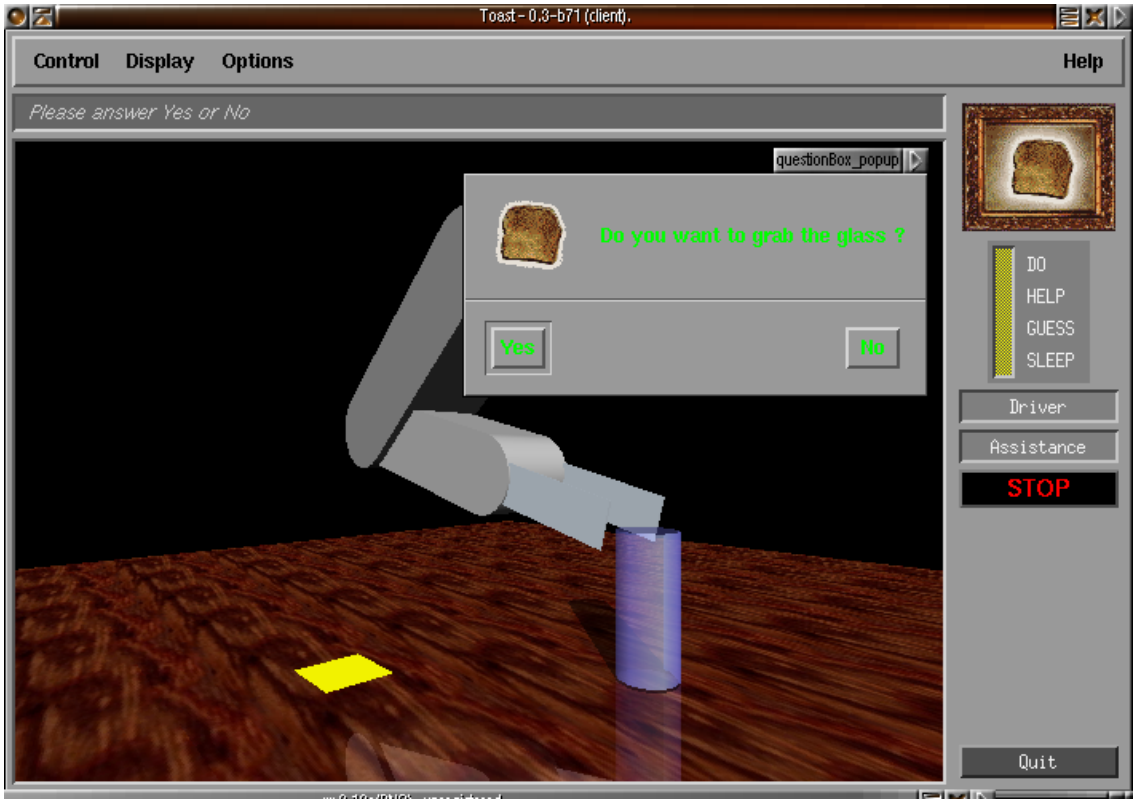

FIG. 4.3: $TOAS_t$ en phase de confirmation de saisie

Si à l'issue de ce processus, aucune action n'est confirmée, l'action englobante rentre en phase d'antidétection. Si une action est confirmée, le système interrompt tous les processus de détection et passe en phase d'exécution (Cf. chapitre suivant).

☛ Notons que si une action détectée n'est pas confirmée par l'opérateur mais que celui-ci l'exécute néanmoins, celle action reste à l'état plausible pendant toute la durée de l'exécution, donc ne change pas d'état, et donc ne génère aucun nouvel événement à l'attention du système. C'est une phase analogue à celle de l'antidétection, dans laquelle l'opérateur mène l'action à son terme manuellement.

- À la ligne 16, une antidétection (autre événement interne) est signalée. Cela signifie qu'une action englobante précédemment infirmée par l'opérateur vient de repasser en dessous du seuil de confiance. Dans ce cas, l'action en question est considérée comme à nouveau détectable, et son processus de détection est relancé.
- À la ligne 18, une action précédemment déclenchée se termine. Si l'exécution se termine sur un échec (Cf. chapitre suivant), le système de $\mathcal{TAS}_t$ peut éventuellement laisser l'opérateur corriger la situation, puis reprendre l'exécution. Dans le cas contraire, toutes les détections sont relancées.
- À la ligne 24, une commande en provenance de l'opérateur est reçue. Si le système est en phase d'exécution, $\mathcal{TAS}_t$ permet à l'opérateur de prendre temporairement ou définitivement le contrôle du manipulateur. Dans ce dernier cas, l'exécution en cours est annulée.

Chapitre 5

Exécution et corollaires

Au début de cette partie, nous avons déterminé que le cadre d'utilité applicative de la détection d'intentions se situe dans le compromis asservissement / autonomie : on ne demande pas à l'opérateur de compétences techniques très poussées, et on ne requiert pas du système une intelligence démesurée. Cette idée a conduit notre pensée vers des systèmes qui, hormis leur capacité de détection d'intentions, seraient dotés de capacités d'exécution dans des cas ne présentant pas de difficulté majeure. L'objet de ce chapitre est de montrer que compte tenu du modèle de représentation des connaissances proposé d'une part, et des algorithmes et des critères de détection élaborés d'autre part, la conception de mécanismes d'exécution voire de simulation et de détection d'erreur est pratiquement immédiate et peut être obtenue à un coût très faible.

5.1 Algorithmes d'exécution

Le modèle de représentation des connaissances que nous avons élaboré est destiné en priorité à l'utilisation dans un mécanisme de détection d'intentions. Cependant, ce modèle a pour vocation de décrire la démarche d'*exécution* qu'un opérateur suivrait s'il n'était pas assisté. Il peut donc naturellement être utilisé à des fins véritables d'exécution et pas seulement de détection. Dans cette section, nous décrivons les algorithmes requis pour la mise en œuvre d'un mécanisme d'exécution. De même que dans le chapitre précédent, deux algorithmes sont fournis, un par mode d'action.

5.1.1 Relation entre détection et exécution

Les algorithmes proposés dans le chapitre précédent prennent déjà en compte toutes les caractéristiques de notre de modèle de représentation des connaissances :

- Le mode des actions, parallèle ou séquentiel, est utilisé afin de déterminer à un instant donné, l'ensemble des branches dont la détection (ne) devrait (pas) avoir lieu.
- Le lancement effectif de la détection de ces branches est conditionné par leur qualité, persistante ou auxiliaire, en conjonction avec l'état actuel de l'action, résolue ou non.

Il en découle que ces algorithmes suivent en fait exactement la logique d'une exécution, mis à par le fait que les changements d'états sont basés sur les transitions plausible / non plausible plutôt que sur les transitions résolue / non résolue. En d'autres termes, ces algorithmes se comportent comme des algorithmes d'exécution qui considèrent que la résolution d'une action est obtenue

quand son taux de vraisemblance atteint le seuil de confiance du système au lieu de la valeur maximale.

Une conséquence particulièrement intéressante de cette remarque est que les algorithmes d'exécution peuvent s'obtenir par *simplification* des algorithmes de détection, la simplification consistant à considérer que le seuil de confiance du système est unitaire. Cette idée est développée dans les deux sections suivantes, selon les modes d'opération.

5.1.2 Exécution d'une action parallèle

Le schéma d'exécution d'une action parallèle est présenté par l'algorithme 6.

Algorithme 6: Exécution d'une action parallèle

EXÉCUTER(Action parallèle)
(1) *// Initialisation*
(2) **si** l'action est résolue («en veille»)**:**
(3) Lancer l'exécution des branches persistantes seulement («en veille»).
(4) **sinon:**
(5) Lancer l'exécution de toutes les branches.
(6) *// Boucle infinie*
(7) **Pour toujours:**
(8) Attendre un signal de changement d'état.
(9) *// Un changement d'état est reçu*
(10) **si** l'action est non résolue («en cours d'exécution»)**:**
(11) **si** la branche signalée est invalidée**:**
(12) L'état ne change pas.
(13) **sinon:**
(14) L'action est résolue (le but est atteint) si toutes les sous-actions le sont (les sous-buts le sont).
(15) *// L'action est résolue («en veille»)*
(16) **sinon:**
(17) *// Le signal vient nécessairement d'une branche persistante*
(18) *// Cette branche est nécessairement invalidée*
(19) L'action est invalidée.
(20) **si** l'action change d'état**:**
(21) Interrompre toutes les branches.
(22) Signaler le changement d'état.
(23) *// Pas de changement d'état*
(24) **sinon:**
(25) *// L'action est nécessairement non résolue (le but non atteint)*
(26) Reprendre l'exécution de la branche signalée («en veille» ou «en cours d'exécution»).

- L'initialisation débutant à la ligne 2 est similaire à celle de l'algorithme de détection correspondant.
- Le processus d'exécution d'une action parallèle ne tourne *jamais* en permanence, mais passe véritablement son temps à attendre un changement d'état (ligne 8) puisque l'état d'exécution est effectivement binaire : résolu ou non. Il n'y a donc pas de test à faire ici afin de déterminer si l'attente d'un signal doit être bloquante ou non.
- Le calcul du nouvel état débutant à la ligne 10 est moins coûteux que celui de l'algorithme de détection, puisque celui-ci ne s'effectue que ponctuellement, après un changement d'état d'une action, et qu'il ne travaille que sur des états binaires.

- La ligne 22 interrompt le déroulement de l'algorithme. Celui-ci sera éventuellement repris par le passage à l'équivalent de la ligne 26 de l'algorithme supérieur, ou sur commande du système.
- Le travail de reprise en cas de non changement d'état est également simplifié, puisqu'il s'agit simplement de relancer la branche signalée.

5.1.3 Exécution d'une action séquentielle

Le schéma d'exécution d'une action séquentielle est présenté par l'algorithme 7.

Algorithme 7: Exécution d'une action séquentielle

EXÉCUTER(Action séquentielle)
(1) *// Initialisation*
(2) **si** l'action est résolue («en veille»)**:**
(3) Lancer l'exécution des branches persistantes seulement («en veille»).
(4) **sinon:**
(5) Lancer l'exécution de toutes les branches («en veille»), jusqu'à la première branche non résolue («en cours d'exécution»).
(6) *// Boucle infinie*
(7) **Pour toujours:**
(8) Attendre un signal de changement d'état.
(9) *// Un changement d'état est reçu*
(10) **si** l'action est non résolue («en cours d'exécution»)**:**
(11) **si** la branche signalée est invalidée**:**
(12) L'état ne change pas.
(13) **sinon:**
(14) **si** la branche signalée est la dernière branche**:**
(15) L'action est résolue (le but atteint).
(16) **sinon:**
(17) L'état ne change pas.
(18) *// L'action est résolue («en veille»)*
(19) **sinon:**
(20) *// Le signal vient nécessairement d'une branche persistante*
(21) *// Cette branche est nécessairement invalidée*
(22) L'action est invalidée.
(23) **si** l'action change d'état**:**
(24) Interrompre toutes les branches.
(25) Signaler le changement d'état.
(26) *// Pas de changement d'état*
(27) **sinon:**
(28) *// L'action est nécessairement non résolue (le but non atteint)*
(29) **si** la branche signalée est résolue (son but atteint)**:**
(30) Reprendre son exécution (la mettre «en veille»).
(31) *// Il y a nécessairement une branche suivante*
(32) Lancer l'exécution de la branche suivante.
(33) *// La branche signalée est invalidée*
(34) **sinon:**
(35) Interrompre l'exécution de toutes les branches suivantes en cours.
(36) Reprendre l'exécution de cette branche.

- L'initialisation débutant à la ligne 2 est similaire à celle de l'algorithme de détection correspondant, à ceci près que le lancement des exécutions des branches s'effectue jusqu'à la

première branche non *résolue* et non pas non *plausible*.

- Les considérations relatives à l'attente bloquante d'un signal (ligne 8) ou à l'interruption / reprise de l'algorithme (lignes 25, 30 et 36) développées pour l'algorithme d'exécution d'une action parallèle restent de mise dans le cas d'une action séquentielle.
- Il en va de même pour le calcul du nouvel état de l'action qui débute à la ligne 10 et raisonne sur des changements d'état binaires.

5.2 Cas des actions terminales

De même que pour le cas de la détection, l'application des algorithmes décrits précédemment à un cas pratique ne nécessite plus que l'élaboration de deux points relatifs aux actions terminales : leur algorithme d'exécution et leur implémentation.. Ces points constituent l'objet de cette section.

5.2.1 Algorithme d'exécution

Le schéma d'exécution d'une action terminale est donné par l'algorithme 8.

Algorithme 8: Exécution d'une action terminale

EXÉCUTER(Action terminale)
(1) **Pour toujours:**
(2) **si** l'action n'est pas résolue (le but non atteint)**:**
(3) Calculer la méthode d'exécution requise.
(4) Appliquer cette méthode.
(5) Revérifier l'état de l'action.
(6) **si** l'action change d'état**:**
(7) Signaler le nouvel état.

Nous voyons que cet algorithme est extrêmement simple.

- Les actions terminales, contrairement aux autres, sont les seules dont le processus d'exécution tourne en permanence.
- Tant que l'action n'est pas résolue, le processus d'exécution détermine la manière d'atteindre le but de l'action, et l'exécute. Si l'action est résolue, le processus d'exécution consiste alors simplement à vérifier qu'elle le reste.
- Un changement d'état, c'est à dire le passage de l'état non résolu à l'état résolu, ou inversement, interrompt le processus d'exécution et signale le changement d'état à l'opération du niveau supérieur, ou au système lui-même.

5.2.2 Application à $TOAS_t$

Le problème restant à résoudre est celui du calcul et de l'application de la méthode de résolution. Nous proposons de résoudre ce problème au travers de l'exemple de l'action terminale `Position ()` implémentée dans $TOAS_t$. Rappelons que cette action a pour but de positionner l'organe de préhension du bras manipulateur à une position (X, Y) dans le plan vertical. La philosophie générale de cette partie étant de représenter les actions, y compris les actions terminales d'une manière simple, la méthode d'exécution que devrait employer le système est, ici encore, le déplacement en ligne droite vers le point de destination.

☛ Il est particulièrement intéressant ici de constater que l'implémentation des actions terminales en termes d'exécution est très peu coûteuse. En effet, le calcul de la méthode d'exécution d'une action terminale est déjà fait puisqu'il est utilisé par le processus de détection. Par exemple, pour ce qui concerne l'action `Position ()` de $\mathcal{TOAS}_t$, l'implémentation de son processus de détection nous a conduit à écrire une routine de calcul du vecteur vitesse de déplacement du point courant au point de destination. La comparaison de ce vecteur vitesse avec la vitesse courante de déplacement nous permettait de calculer le taux d'activation de l'action. Afin d'implémenter le processus d'exécution de cette même action, le calcul de la méthode de résolution est déjà fourni par la routine précédente. Il suffit donc d'appliquer directement son résultat au manipulateur.

5.3 Corollaires

Nous venons de voir que la mise en œuvre des mécanismes d'exécution est une tâche peu coûteuse : les seules actions devant véritablement agir sur le manipulateur sont les actions terminales, et les méthodes d'exécution dont elles ont besoin sont déjà fournies car nécessaires au système de détection d'intentions. La mise en place de ces mécanismes d'exécution a deux conséquences très intéressantes, que cette section s'attache à décrire.

5.3.1 Simulation

Reprenons l'exemple de l'action terminale `Position ()` de $\mathcal{TOAS}_t$, et supposons cette action en cours d'exécution. À un instant donné et dans le plan vertical du bras manipulateur, l'organe de préhension se trouve à une position (P_x, P_y). Le point destination est noté (D_x, D_y).

L'algorithme d'exécution de cette action terminale calcule en permanence le vecteur direction $\overrightarrow{PD}$ et en déduit un vecteur vitesse $\overrightarrow{V_{PD}}$ à appliquer au bras manipulateur. Connaissant ce vecteur vitesse, le processus d'exécution peut en déduire à peu de frais qu'à l'instant $t + \delta t$, la position de l'organe de préhension devrait valoir, si tout s'est bien passé, $P + \delta \overrightarrow{V_{PD}}$.

Supposons maintenant qu'au lieu d'appliquer effectivement les commandes de déplacement au manipulateur, le processus d'exécution ne fasse que les calculer et mettre à jour l'état d'une instance virtuelle du manipulateur en fonction de ses prévisions. Nous obtenons ici, et encore une fois à peu de frais, un mécanisme de *simulation* des opérations. Ces mécanismes sont implémentés dans $\mathcal{TOAS}_t$. La figure 5.1 en page suivante montre $\mathcal{TOAS}_t$ en phase de simulation de l'opération de pose d'un objet. L'instance simulée de la scène est tracée en fil de fer.

☛ Notons que pour passer de l'exécution à la simulation, le coût de développement est encore plus faible que pour passer de la détection à la simulation. En effet, les algorithmes requis pour la simulation sont *exactement* les mêmes, y compris ceux des actions terminales, que ceux requis pour l'exécution. Le seul et unique point qui change se trouve dans l'implémentation des actions terminales, où les commandes motrices sont calculées mais non transmises au manipulateur.

5.3.2 Détection d'erreur

Dans la section précédente, nous avons évoqué le terme de «prévision» : l'implémentation des mécanismes de simulation nous a conduit à prévoir les modifications infinitésimales de la scène. Or une capacité de prévision implique une capacité de vérification : connaissant le résultat attendu dans le prochain intervalle de temps, il devient alors possible de comparer ce résultat avec celui véritablement obtenu.

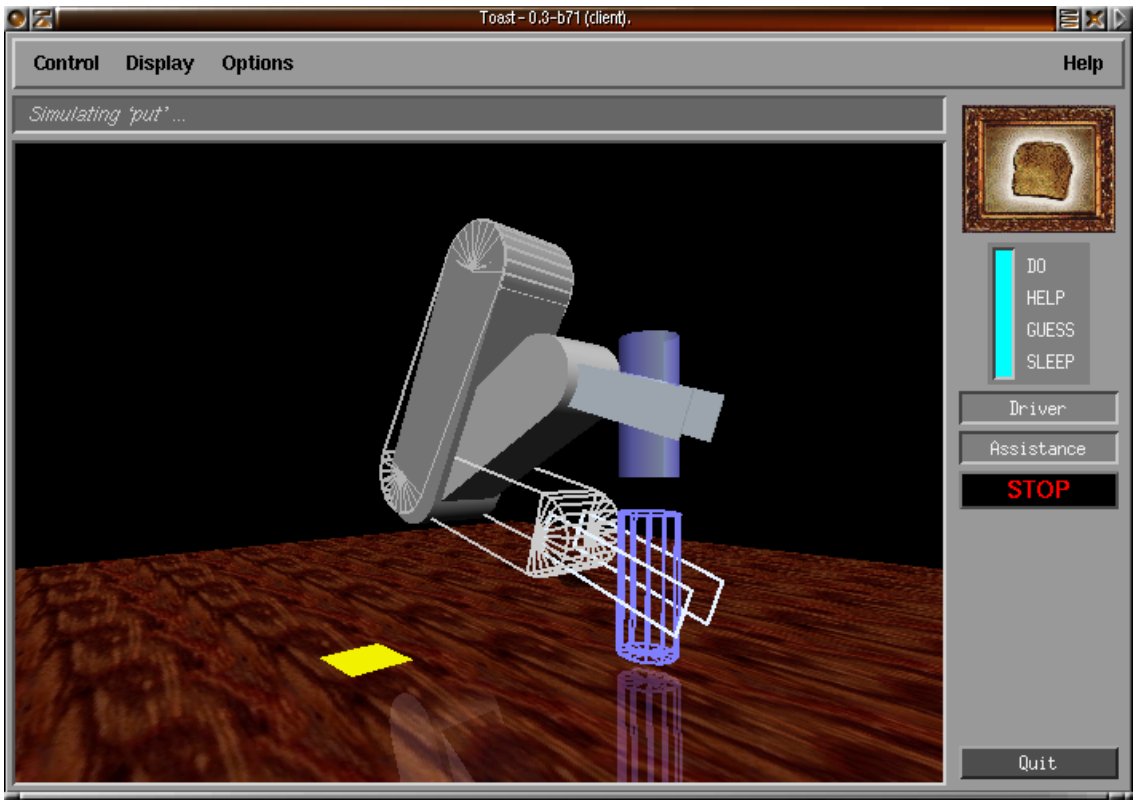

FIG. 5.1: $TOAST_t$ en phase de simulation de pose.

Dans le cas de l'action terminale `Position ()` de $TOAST_t$ par exemple, à chaque passage dans la boucle, le processus d'exécution peut comparer la position courante de l'organe de préhension à celle qui avait été prévue ($P + \delta \vec{V_{PD}}$) et en extraire d'éventuelles discordances.

Ce mécanisme a été implémenté dans $TOAST_t$: en phase d'exécution, chaque action terminale compare en permanence le résultat qu'elle obtient avec celui qu'elle avait prévu. Une discordance entre les deux résultats persistant au delà d'un certain nombre d'itérations génère un signal d'erreur qui remonte jusqu'au système. La figure 5.2 en page suivante montre $TOAST_t$ après détection d'une erreur. Dans cet exemple, $TOAST_t$ a été préalablement mis dans des conditions telles que l'opération de saisie échoue : le bras manipulateur ne peut descendre à la position requise car il est bloqué par l'objet. L'action terminale `Position ()` détermine que malgré les commandes motrices qu'elle produit, la distance entre la position actuelle et la position requise ne diminue pas, et détecte par conséquent une erreur.

Ce mécanisme nécessite une légère modification des procédures d'exécution. En ce qui concerne les actions parallèles et séquentielles, cette variante consiste à comprendre un nouvel état, l'état «erreur», en plus des états «résolu» et «non résolu». Lorsqu'une action reçoit un signal de changement d'état vers l'état erreur, elle doit interrompre son déroulement et propager le signal vers le haut jusqu'au système. Cette modification des algorithmes est minime et ne justifie pas de fournir ici une nouvelle description complète. Une variante pour les actions terminales est par contre donnée par l'algorithme 9 en page 176.

La partie modifiée dans cet algorithme concerne le calcul du taux d'erreur et son influence sur l'état de l'action de la ligne 3 à la ligne 10. Il n'est pas concevable de détecter une erreur dès la première incohérence entre la prévision et le résultat obtenu. En effet, le système peut réagir

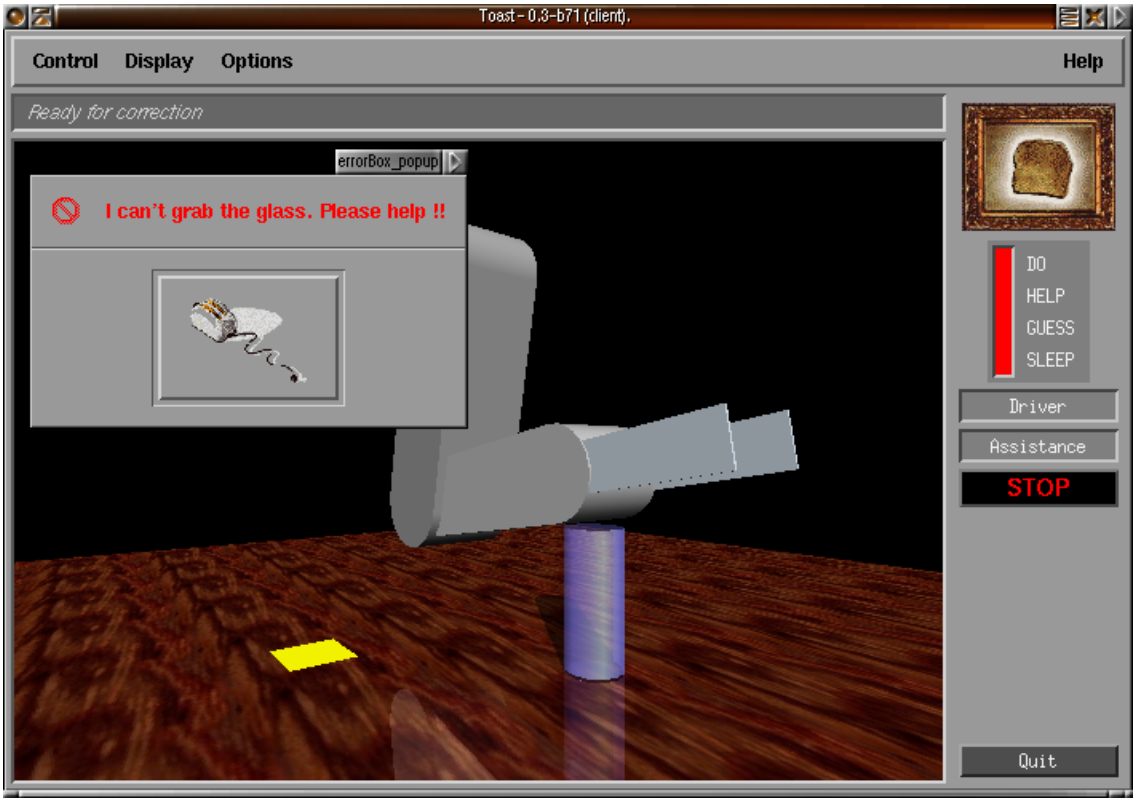

FIG. 5.2: $TOAST_t$ en phase de détection d'erreur

avec un temps de retard sur certains événements, comme le changement intempestif de position pour un objet dont la procédure de saisie est en cours. Ce qui détermine la détection d'une erreur dans notre algorithme, c'est une série d'incohérences *consécutives* pendant suffisamment longtemps (ligne 7). Dès que la prévision redevient conforme au résultat obtenu, le taux d'erreur est réinitialisé (ligne 6).

5.4 Limitation

Nous venons de montrer que notre modèle de représentation des connaissances, ainsi que sa philosophie autoréférente qui conduit à déporter tout le véritable travail sur les actions terminales offre, à un très faible coût, des fonctionalités puissantes comme l'exécution, la simulation ou la détection d'erreur, en plus de sa fonctionalité de détection pour laquelle il est initialement conçu. Ce paradigme a cependant une limitation qu'il convient de préciser. Cette limitation peut être décrite au travers du concept de «proprioception étendue».

5.4.1 Proprioception étendue

Le terme de proprioception se réferre à la faculté de percevoir son propre corps de manière interne, c'est à dire indépendamment de l'environnement. C'est le sens proprioceptif qui renseigne l'humain sur son orientation dans l'espace, la position de ses membres...

Dans le cas de $TOAST_t$, une certaine quantité de données proprioceptives est disponible : état des articulations du bras manipulateur, position spatiale de certains points stratégiques, état d'ouverture ou de fermeture de l'organe de préhension... Ces données sont indépendantes de l'environ-

Algorithme 9: Exécution d'une action terminale avec détection d'erreur

```
EXÉCUTER(Action terminale)
(1)   Pour toujours:
(2)     si l'action n'est pas résolue (le but non atteint):
(3)       si le résultat prévisionnel diffère du résultat obtenu:
(4)         Incrémenter le taux d'erreur.
(5)       sinon:
(6)         Annuler le taux d'erreur.
(7)       si le taux d'erreur est trop important:
(8)         L'action est considérée infaisable (son état passe à «erreur»).
(9)       sinon:
(10)        Revérifier l'état de l'action.
(11)      si l'action change d'état:
(12)        Signaler le nouvel état.
(13)      sinon:
(14)        Calculer la méthode d'exécution requise.
(15)        Calculer le résultat prévisionnel.
(16)        Appliquer cette méthode.
```

nement. Supposons maintenant qu'un objet soit saisi et que le bras manipulateur se déplace. Cet objet est fixe par rapport à l'organe de préhension. Par conséquent, pour connaître le déplacement de l'objet entre deux instants donnés, il suffit de connaître celui du bras manipulateur, ce qui relève de la proprioception. Il est inutile de passer par un mécanisme d'acquisition externe (analyse de scène). Dans une telle circonstance, nous voyons qu'une donnée normalement extéroceptive est acquise par des moyens exclusivement proprioceptifs.

☛ Ce phénomène est connu en télé-opération, mais il n'est pas nommé. Nous proposons de le désigner par l'expression «proprioception étendue». Un exemple d'utilisation caractéristique de cette idée se retrouve dans les systèmes de suivi dynamique d'objet, ou suivi de cible : prenons un système de plusieurs caméras, et une cible visuelle (par exemple un objet) à conserver au centre des images. Certains algorithmes se basent sur les variations des images elle-mêmes afin de calculer automatiquement les facteurs de rotation à appliquer aux caméras pour le suivi de l'objet. Dans un tel système, on peut également obtenir la position spatiale de l'objet considéré en fonction de celles des caméras elle-mêmes. Nous retrouvons ici la notion de proprioception étendue, dans la mesure où la position de l'objet, donnée extéroceptive, est obtenue à partir de la position des caméras (donnée proprioceptive).

5.4.2 Conséquence

La limite de notre modèle se situe précisément au niveau de la proprioception étendue. Nos actions terminales sont parfaitement capables de gérer (c'est à dire détecter, exécuter, simuler) les commandes relatives au manipulateur lui-même. La situation se complique pour les commandes relatives aux objets.

Dans le cas du transport d'un objet par exemple, le déplacement de celui-ci s'effectue par un déplacement du bras manipulateur lui-même. L'action `Position (OBJET)` indique donc au système que pour déplacer l'*objet* à la position P, c'est en fait le *bras manipulateur* lui-même que l'on doit déplacer à cette position. Nous sommes ici dans le cas de la proprioception étendue et les choses se déroulent bien.

Par contre, notre modèle ne permet pas d'inculquer au système une action telle que «pousser l'objet jusqu'à ce qu'il tombe» car dans ce cas, qui ne relève pas du domaine de la proprioception étendue, il y n'y a pas de connexion simple entre le mouvement du bras et celui de l'objet dans sa chute. Nos actions terminales sont donc incapables de se représenter ce phénomène.

Chapitre 6

Un exemple de scénario

Dans ce chapitre, nous proposons d'expliciter le déroulement d'un scénario particulier (saisie – transport – pose d'un objet), tel qu'il peut se produire dans T_OAS_t. Ce scénario nous permet de montrer le mécanisme de détection d'intentions de T_OAS_t à l'œuvre, mais également d'illustrer certains autres types d'assistance implémentés dans le système et correspondant à des cas analysés dans la deuxième partie.

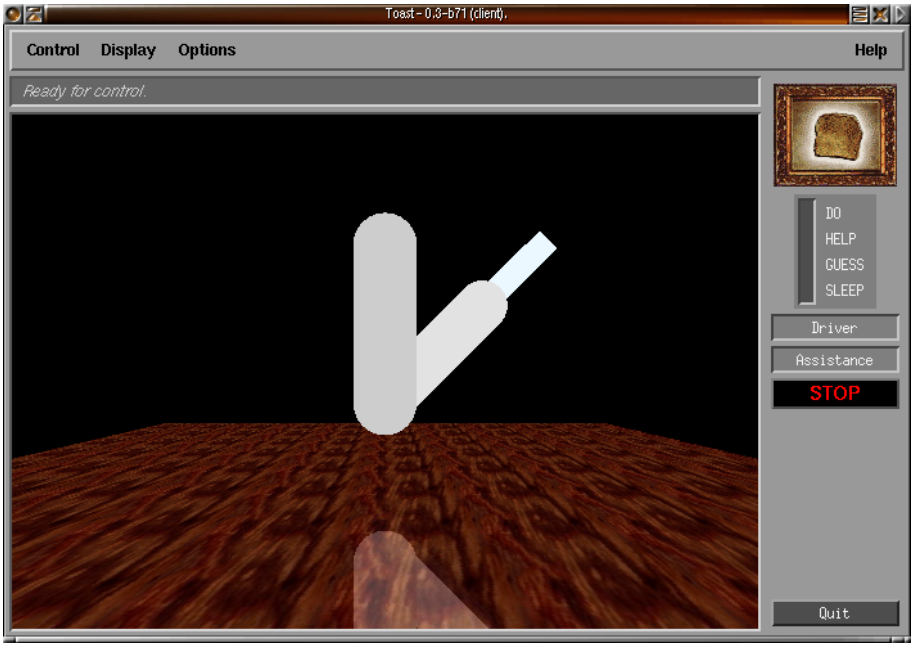

FIG. 6.1: Scénario – première étape

Première étape Au départ, aucun objet n'est présent dans la scène. Le système a connaissance des actions de saisie, telle que décrite sur la figure 3.8 en page 142, et de pose des objets. Pour déplacer le bras manipulateur, l'opérateur dispose d'un simple joystick, simulé à l'écran dans le cas de T_OAS_t (Cf. annexe 1), et pour lequel chacun des trois boutons de la souris lui confère une des actions terminales précédemment décrites (Cf. figure 3.7 en page 141).

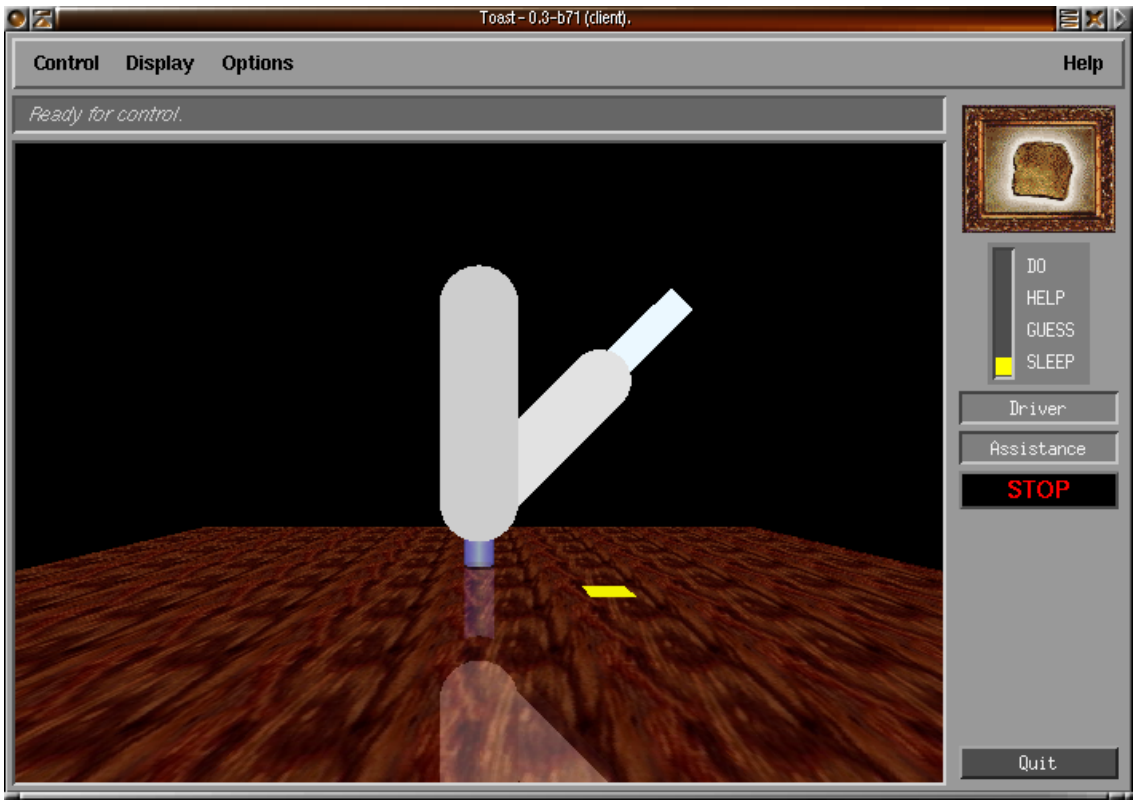

FIG. 6.2: Scénario – deuxième étape

Deuxième étape Deux objets sont apparus dans la scène (un verre et une zone colorée sur le plan de travail, symbolisant une aire de remplissage). L'apparition de ces objets constitue un «événement extérieur» pour le système (Cf. algorithme système en page 165) qui redéfinit alors l'ensemble de ses actions englobantes. Les hypothèses de saisie et pose pour ces deux objets sont envisageables, et le système lance alors leur détection.

☛ L'aire de remplissage n'est pas un objet réellement saisissable. Cependant, cette connaissance n'a pas été inculquée au système volontairement, pour permettre la démonstration de certaines fonctionalités comme la détection et la correction d'erreurs. Par contre, le système sait que le plan de travail ne constitue pas un objet saisissable. Cette hypothèse ne fera donc jamais l'objet d'une détection.

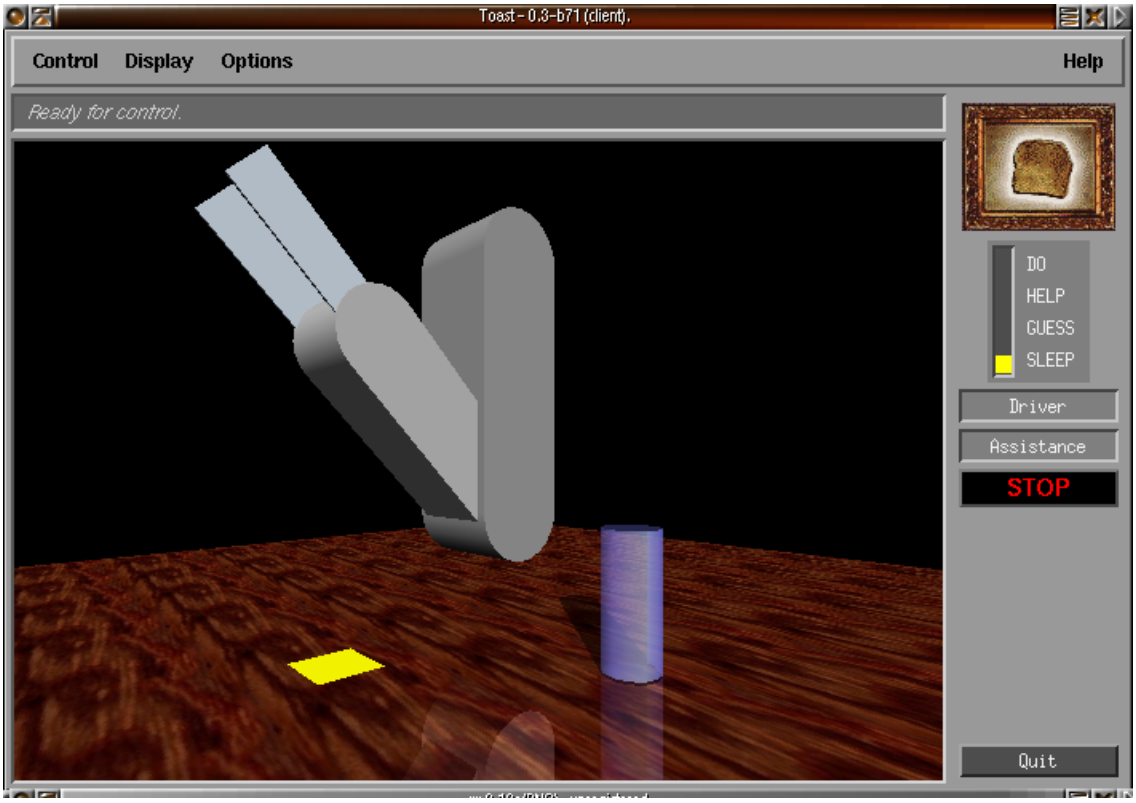

FIG. 6.3: Scénario – troisième étape

Troisième étape Dans un troisième temps, l'opérateur déplace son point de vue afin de mieux percevoir la position spatiale de l'objet et de déplacer le bras correctement. Ceci donne lieu à l'explicitation d'un premier exemple d'assistance de bas niveau :

En se plaçant de l'autre côté du bras, l'opérateur inverse son point de vue de scène. Si la commande de mouvement du bras était câblée, elle serait donc elle-aussi inversée : en déplaçant le joystick vers la droite, l'opérateur obtiendrait cette fois un déplacement du bras vers la gauche. En fait, il est plus naturel pour l'opérateur d'opérer des mouvements du bras par un déplacement *analogue* du joystick, par rapport à son champ de vision. Cette fonctionalité est implémentée dans $\mathcal{TAS}_t$. Elle implique d'interpréter dynamiquement les commandes appliquées au joystick en fonction du point de vue actuel de l'opérateur. Ce type d'assistance relève simplement de l'ergonomie : il consiste à fournir à l'opérateur les commandes de base les plus naturelles possibles.

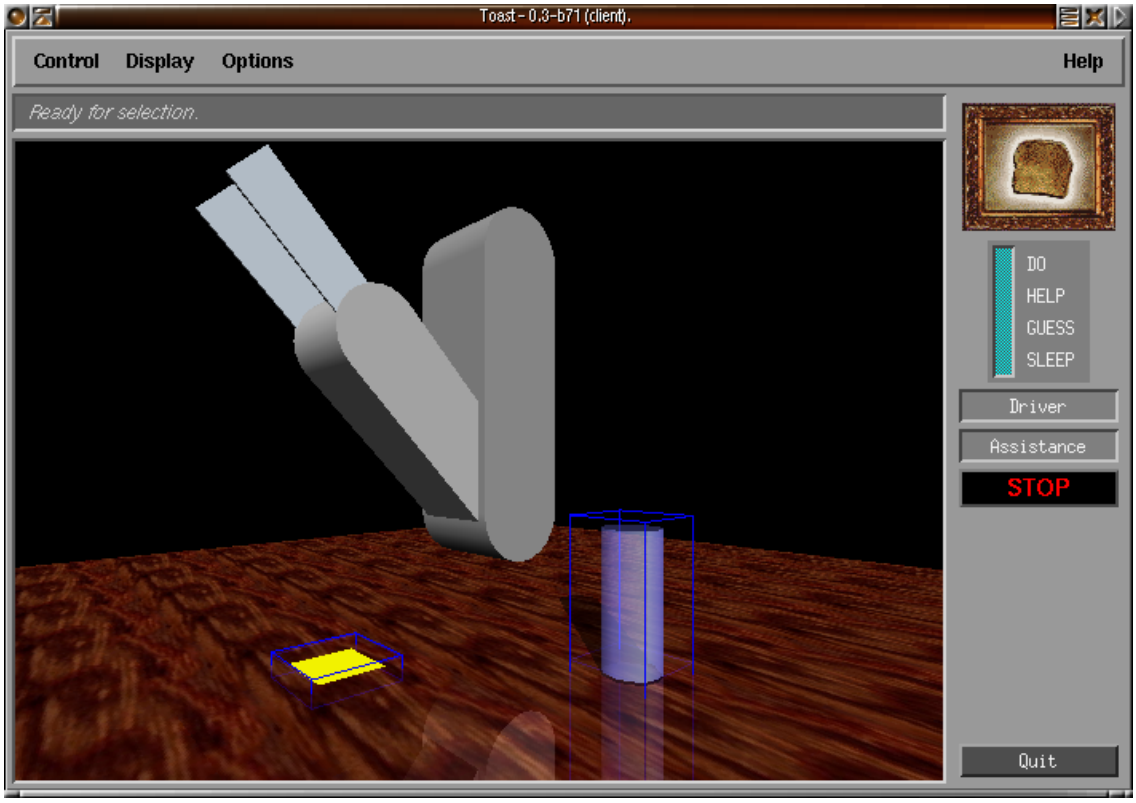

FIG. 6.4: Scénario – quatrième étape

Quatrième étape L'opérateur se demande maintenant si la situation est suffisamment simple pour que l'exécution directe de la saisie se déroule sans accroc. Afin d'obtenir une réponse à cette question, il demande une simulation au système. Ceci donne lieu à l'explicitation d'un deuxième exemple d'assistance :

L'opérateur demande une simulation de la saisie à partir d'un menu de l'interface graphique (Cf. annexe 1) qui ne permet pas de spécifier l'objet en même temps que l'action. Le système examine alors l'ensemble des objets pour lesquels cette action est possible, et opère une pré-sélection des objets concernés. Ces objets sont visuellement distinguables des autres par un cadre virtuel tracé en fil de fer bleu. L'opérateur doit maintenant sélectionner l'objet de son choix, mais seulement parmi la présélection : les objets de la présélection sont cliquables, et toute action de la souris à un autre endroit de l'image est sans effet. Ce type d'assistance correspond à ce que nous avons appelé une assistance par *commande paramétrique* (Cf. section 3.6 en page 81)

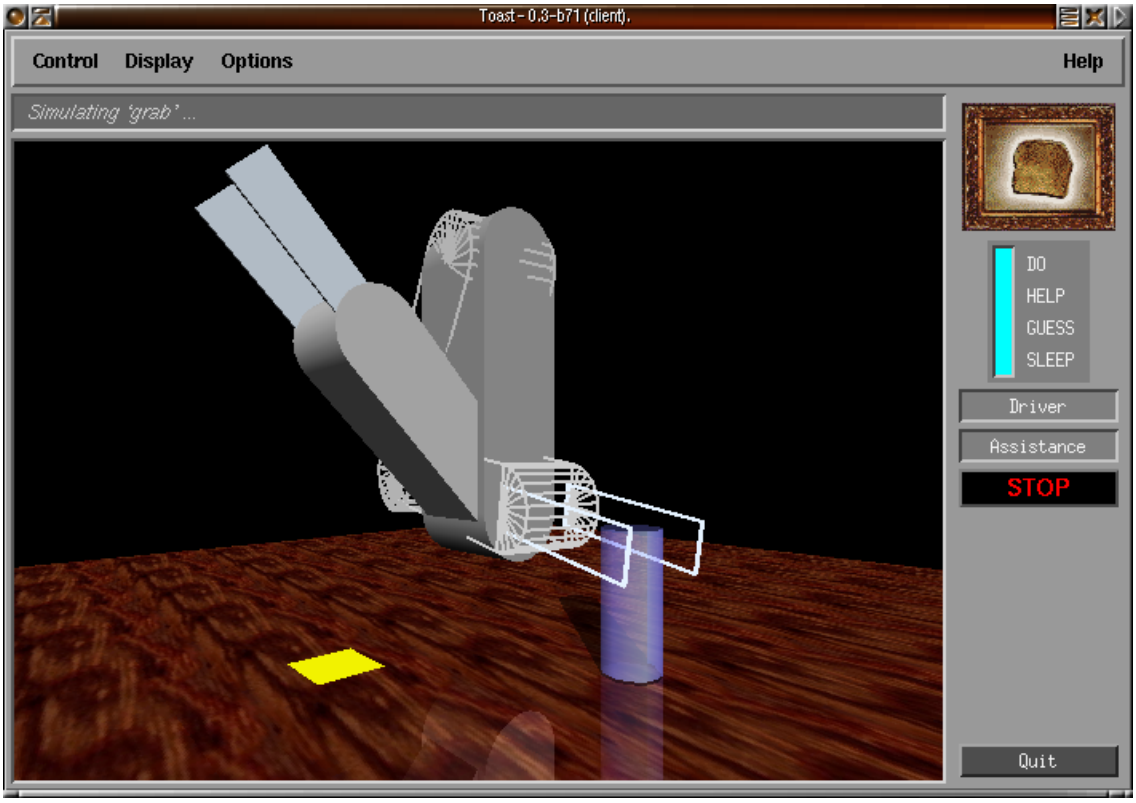

FIG. 6.5: Scénario – cinquième étape

Cinquième étape Après que l'opérateur ait sélectionné le verre sur l'image, le système lance la simulation de la saisie. Cette simulation apparaît tracée en fil de fer, et montre que la situation courante ne permet pas au système d'exécuter proprement l'opération avec les connaissances dont il dispose. En effet, cette simulation montre que le bras manipulateur «traverse» l'objet. Dans la réalité l'objet serait donc poussé sur le côté par le bras. L'opérateur doit donc commencer lui-même le mouvement.

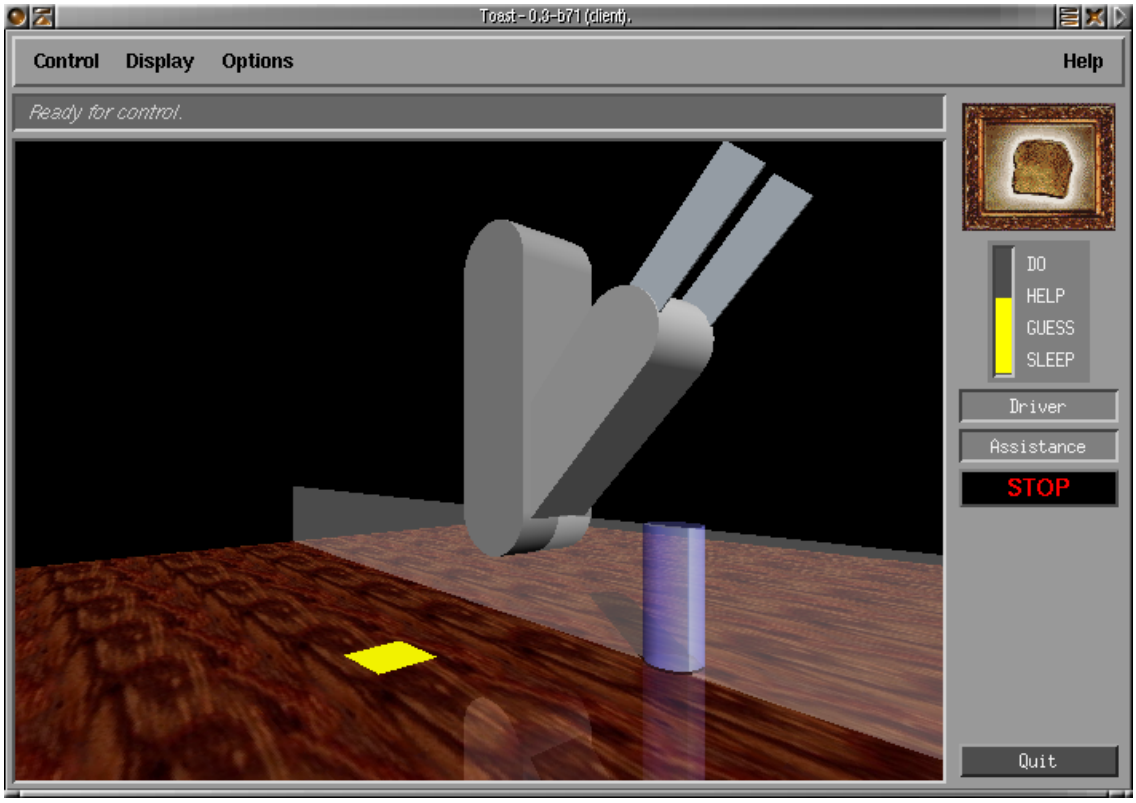

FIG. 6.6: Scénario – sixième étape

Sixième étape L'opérateur entame donc l'opération de saisie par une rotation du bras, de manière à venir positionner celui-ci dans l'axe de l'objet. Ceci donne lieu à l'explicitation d'un troisième exemple d'assistance :

Face à un simple écran d'ordinateur, il est difficile de percevoir à quel moment le bras manipulateur sera bien dans l'axe de l'objet. Pour palier cette difficulté, $\mathcal{TOAS}_t$ implémente un cas d'assistance décrit dans la deuxième partie comme un cas de *génération de perception* (Cf. section 3.1 en page 73). Lorsqu'un objet est présent (non saisi) dans la scène, toute commande de rotation du bras émise par l'opérateur déclenche l'apparition d'un plan vertical virtuel, translucide et solidaire du bras manipulateur, donnant une indication au sol sur l'orientation de celui-ci. Il devient donc plus facile à l'opérateur de se positionner dans l'axe de l'objet.

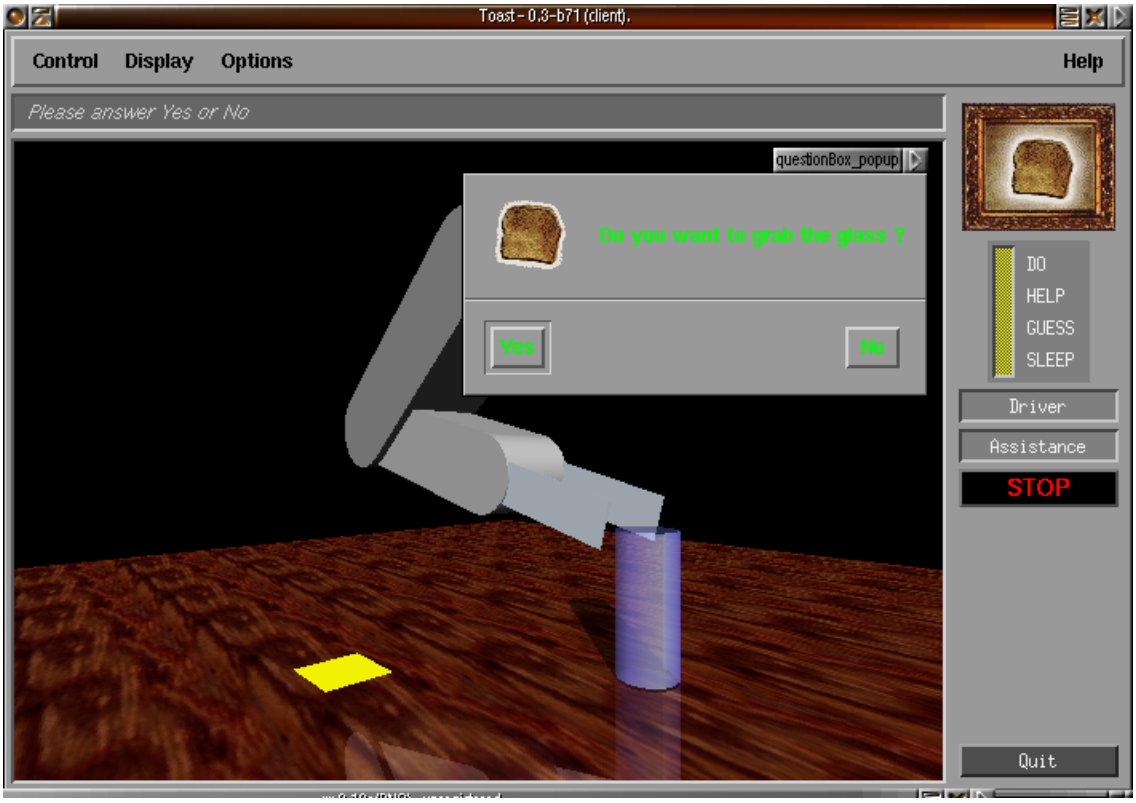

FIG. 6.7: Scénario – septième étape

Septième étape L'opérateur s'est maintenant positionné *à peu près* dans l'axe de l'objet, et entamme à nouveau un déplacement du bras manipulateur en vue de la saisie. À une certaine distance du but, le taux de vraisemblance de l'action de saisie en cours de détection dépasse le seuil de confiance du système, ce qui entraîne l'interruption des opérations. Confirmation est alors demandée à l'opérateur.

☛ Dans $TOAST_t$, il existe également une option de fonctionnement selon laquelle aucune confirmation n'est demandée à l'opérateur : quand le système est suffisamment sur de lui, il prend directement le relais des opérations.

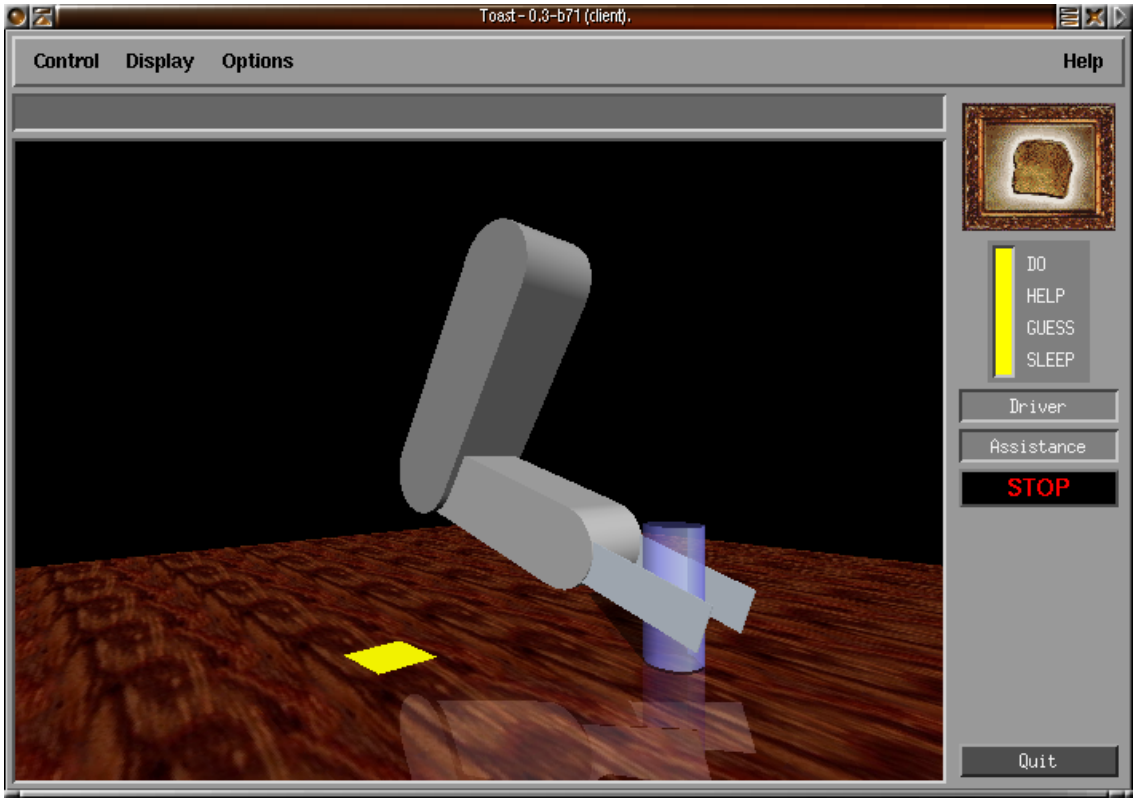

FIG. 6.8: Scénario – huitième étape

Huitième étape L'opérateur confirme que son intention était bien la saisie de l'objet, et le système termine alors lui-même le mouvement. Durant cette période, l'opérateur peut s'il le souhaite interrompre complètement l'exécution (bouton STOP) ou reprendre momentanément le contrôle en appliquant une commande au bras manipulateur.

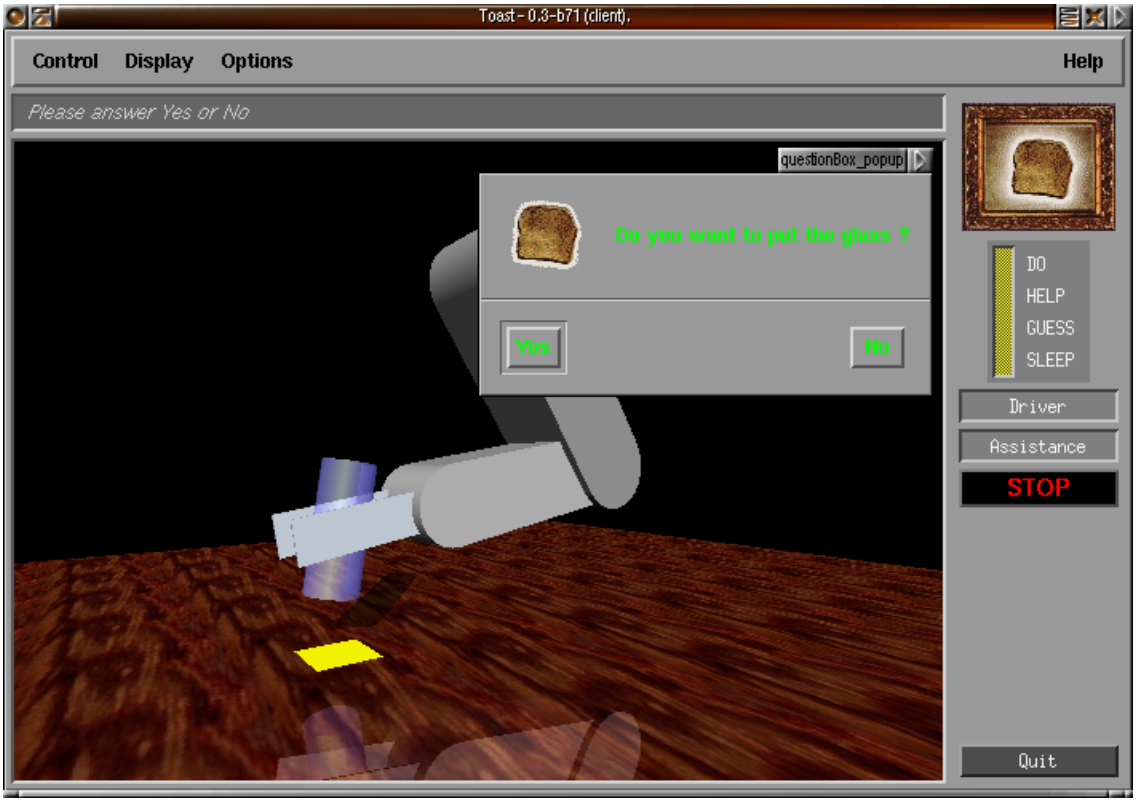

FIG. 6.9: Scénario – neuvième étape

Neuvième étape L'opérateur transporte lui-même l'objet à proximité de l'aire de remplissage. Durant ce mouvement, l'assistance par génération de perception intervient comme précédemment, et l'objet, solidaire de l'organe de préhension, adopte l'inclinaison de celui-ci. À une certaine distance du sol, l'hypothèse de pose dépasse le seuil de confiance du système, et celui-ci demande confirmation à l'opérateur, de même que dans le cas de la saisie.

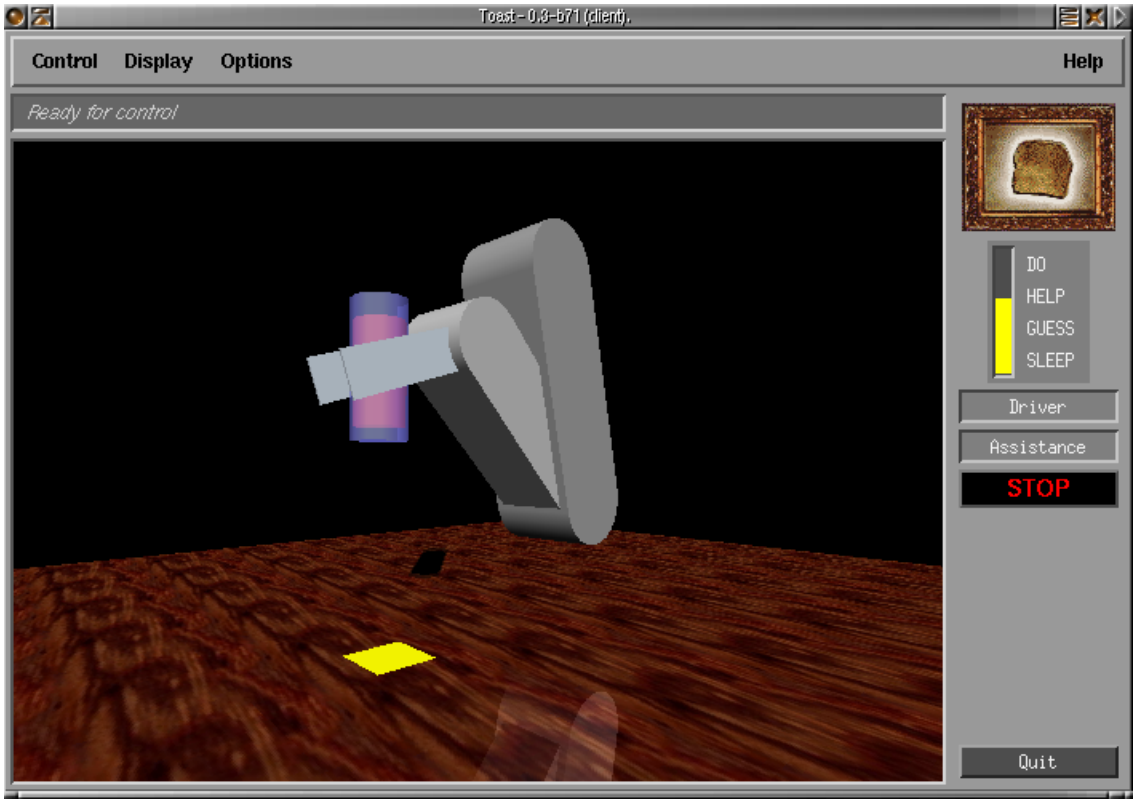

FIG. 6.10: Scénario – dixième étape

Dixième étape Une fois posé, le verre est rempli. L'opérateur entame à ce moment un nouveau transport de l'objet. Ceci donne lieu à l'explicitation d'un quatrième exemple d'assistance :

Contrairement à la première phase de transport, il devient ici important de conserver la verticalité du verre, dans la mesure où celui-ci est rempli. $\mathcal{TAS}_t$ implémente dans ce cas une assistance décrite dans la deuxième partie comme une *génération d'action* (Cf section 4.2 en page 88) : l'opérateur dispose des *mêmes* commandes que dans la phase de transport précédente. La différence est que cette fois-ci, le système connaît la nécessité de préservation de la verticalité, et génère lui-même les commandes d'orientation de l'organe de préhension permettant de conserver la verticalité de l'objet. L'opérateur n'a donc pas à se soucier de ce détail.

Chapitre 7

Conclusion

«I'd like to build a computer that would be proud of me.»
Danny Hillis

Dans cette troisième et dernière partie, nous avons approfondi la notion de détection d'intentions sous un angle applicatif. D'un point de vue conceptuel, la détection d'intentions a été vu, dans les deux parties précédentes, comme le concept débouchant sur les systèmes de réalité virtuelle manipulant des connaissances du plus haut niveau sans pour cela pouvoir se passer de l'opérateur. Cette idée trouve une concrétisation dans le compromis asservissement / autonomie : chaque composante du système (opérateur et manipulateur) garde ses compétences propres, mais la communication s'effectue à un niveau conceptuel proche de celui que l'opérateur utilise naturellement.

Le cadre d'utilité applicative de la détection d'intentions se situe par conséquent au niveau de systèmes capables de se représenter des opérations d'assez haut niveau, mais de ne les exécuter que quand leur résolution ne pose plus de problème majeur. Notre mécanisme de détection d'intentions a été élaboré dans cette optique.

- Dans un premier temps, nous avons élaboré un modèle de représentation des connaissances destiné à fournir au système une description des actions que l'opérateur est susceptible d'entreprendre. Ce modèle permet de décrire chaque action sous la forme d'un développement arborescent en actions de plus bas niveau. Ce développement est également doté de deux caractéristiques supplémentaires : le *mode* d'une action défini l'agencement temporel (parallèle ou séquentiel) de son développement. La *qualité* d'une branche défini le degré de persistance de son but. Ce modèle débouche sur la notion d'«action terminale» constituant les feuilles du développement arborescent. Ces actions terminales sont les seules à être câblées pour un système donné et représentent les commandes élémentaires dont l'opérateur dispose pour contrôler le système.
- Dans un deuxième temps, nous avons proposé un jeu d'algorithmes et un ensemble de critères destinés à implémenter un mécanisme de détection d'intentions. Ce mécanisme passe par l'évaluation du «taux de vraisemblance» d'une opération, qui donne à un instant donné une mesure de sa plausibilité. De même qu'une opération de développe en actions de plus bas niveau, son taux de vraisemblance de défini en fonction des taux de vraisemblance des actions qui constituent son développement. Ici encore, seul le calcul du taux de vraisemblance des actions terminales doit être câblé, et dépend du système considéré.

La principale caractéristique du mécanisme de détection d'intentions que nous proposons est sa grande généricité : l'autoréférence omniprésente dans les définitions et les algorithmes fait que le véritable travail de détection d'intentions est fait par les actions terminales, qui sont les seules

dépendantes du système considéré. Le moteur de détection d'intentions peut en conséquence être utilisé tel quel, sans aucune modification dans n'importe quel système. Compte tenu d'un ensemble d'actions terminales, la connaissance du système peut être décrite formellement selon les termes de notre modèle et lui être inculquée dynamiquement, par exemple à son lancement.

La généricité totale de ce modèle amène un certain nombre de conséquences particulièrement intéressantes : en plus du mécanisme de détection d'intentions, il est possible d'implémenter à un coût extrêmement faible, des fonctionalités d'exécution avec détection d'erreur et de simulation des opérations que le système est à l'origine conçu pour détecter. Les algorithmes requis pour mettre en œuvre ces fonctionalités sont peu différents voire plus simples que les algorithmes de détection d'intentions, et ils sont eux aussi complètement génériques. L'implémentation de ces algorithmes dans le cas des actions terminales est elle aussi pratiquement immédiate.

Nous pensons avoir poussé à son paroxysme l'idée selon laquelle toute la connaissance relative aux opérations peut se résumer aux actions de bas niveau. Cette idée, formulée autrement, signifie que la connaissance que le système a du monde s'arrête à celle de ses actions élémentaires. Les limites du cadre applicatif de cette philosophie se trouvent d'une part dans la simplicité de la représentation de ces opérations et donc dans les conditions restreintes selon lesquelles elles peuvent être détectées, simulées ou exécutées, et d'autre part dans le concept de proprioception étendue.

Conclusion Générale

«question = (to)? be :! be;»
William Shakespeare

Le travail présenté ici s'est décomposé en trois axes majeurs, correspondant chacun à l'une des trois parties de ce document. Ces trois parties s'articulent de la même manière, que ce soit sur le plan logique ou chronologique, et débouchent sur un certain nombre de résultats que nous nous proposons de rappeler. À l'issue de ce travail, quelques réflexions sur les améliorations, les ouvertures et les perspectives envisageables sont également proposées.

*

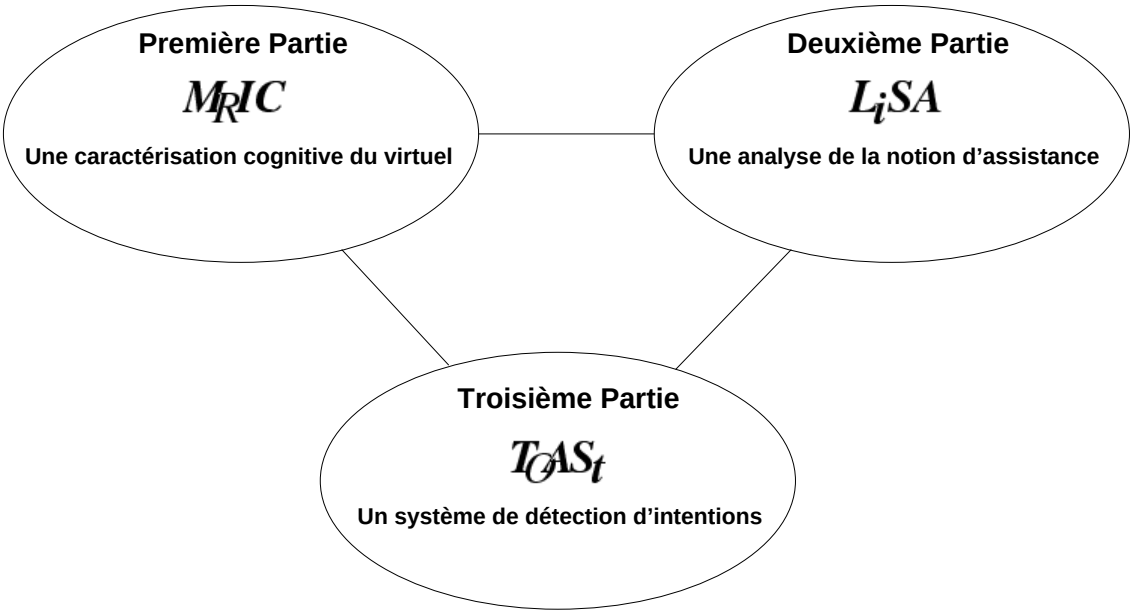

FIG. 1: *MRIC*, *LiSA*, *TOASt* : trois apports scientifiques

Résultats

Les apports de ce travail se situent respectivement sur un plan théorique, avec la proposition d'une définition cognitive du virtuel puis avec l'étude des relation entre les notions de réalité virtuelle et d'assistance à la télé-opération, et sur un plan applicatif avec la mise en œuvre du concept de détection d'intentions.

Définir cognitivement le virtuel

Dans la première partie, nous avons éclairci le sens que l'on peut attribuer à la notion de réalité virtuelle. Un panorama des propositions existantes a montré la nécessité de mettre en avant la dimension cognitive dans l'élaboration d'une définition. Dans cette optique, nous avons proposé un modèle cognitif de description des interactions entre un humain et son environnement : le modèle ***M*$_R$*IC***. Nous avons ensuite défini non pas le «virtuel», mais les «processus de virtualisation» qui font que l'humain accède au virtuel à partir du réel.

L'apport de ce modèle réside principalement dans la capacité que nous avons eue de rassembler les propositions existantes et leurs caractéristiques en une seule et unique vision suivant une approche cognitive. Grâce à cette approche, les différents aspects caractéristiques du virtuel ont pu être mis en évidence directement sur le modèle, et la terminologie actuelle a pu être précisée.

Par delà le bénéfice unificateur de notre démarche, nous avons également pu, par l'analyse théorique, retrouver deux conclusions jusqu'ici mises en évidence par des moyens plus heuristiques :

- Premièrement, la distinction majeure entre les types de réalités virtuelles se situe véritablement entre les applications «de bureau» où l'interaction se fait par des périphériques de contrôle nécessitant un apprentissage minimum, et les applications «immersives» dans lesquelles l'interaction se fait d'une manière naturelle à l'opérateur.
- Deuxièmement, il n'existe pas de différence conceptuelle entre les environnements réels et les environnements de synthèse. En effet, les processus de virtualisation sont les mêmes dans les deux cas.

Finalement, il est à noter que dans le cadre du modèle ***M*$_R$*IC***, le concept de détection d'intentions permet d'obtenir le plus haut degré de virtualisation tout en conservant un opérateur dans la situation.

Tandis que l'approche de cette première partie, fondée sur un modèle cognitif d'interaction, nous a permis de mettre en évidence les liens étroits entre télé-opération et réalité virtuelle, nous poursuivons logiquement cette analyse dans la deuxième partie, en examinant les mêmes phénomènes d'un point de vue fonctionnel.

Assistance en réalité virtuelle

Dans la deuxième partie, nous avons approfondi la notion d'assistance à l'opérateur, et déterminé en quoi le concept de réalité virtuelle permet de concevoir des cas d'assistance particuliers. Un modèle décrivant les situations de télé-opération en réalité virtuelle a été élaboré : le modèle ***L*$_i$*SA***. Ce modèle, version duale du modèle ***M*$_R$*IC***, a permis d'analyser les principaux cas d'assistance connus, et a mis en évidence un certain déséquilibre dans l'exploitation des concepts de réalité virtuelle pour l'assistance. Une méthode systématique de symétrisation des cas analysés a permis de corriger ce déséquilibre.

Nous pensons que les apports de cette théorie de l'assistance sont doubles :

- Tout d'abord, elle permet d'unifier les domaines conceptuels de l'assistance à la télé-opération et de la réalité virtuelle. En effet, nous avons montré, par symétrisation, que certains cas d'assistance à la télé-opération bien connus relèvent des mêmes concepts que ceux exploités en réalité virtuelle.
- Deuxièmement, elle permet de mettre en évidence les possibilités d'élargissement de certaines notions comme celle de réalité augmentée. Si la réalité augmentée désigne traditionnellement le fait de mélanger les perceptions réelles et virtuelles, elle devrait également désigner le fait de mélanger les actions réelles et virtuelles, dans la mesure ou la sémantique de ces procédés d'assistance est la même.

Nous avons par ailleurs montré que l'étude du niveau conceptuel de l'information circulant au travers du modèle ***LiSA*** est un bon révélateur de la présence de processus d'assistance à l'opérateur. Le concept de détection d'intentions est à nouveau apparu, dans le cadre de ce modèle, comme le degré d'assistance à l'opérateur où le niveau conceptuel de l'information manipulée est maximal.

Tandis que par l'analyse théorique des deux premières parties, nous avons fait apparaître l'intérêt conceptuel de la notion de détection d'intentions, nous poursuivons logiquement cette idée dans la troisième partie, en lui donnant un axe applicatif.

Détection d'intentions

Dans la troisième partie, nous avons approfondi la notion de détection d'intentions sous un angle applicatif, pour des systèmes capables de se représenter des opérations d'assez haut niveau, sans pour cela être capables de les exécuter dans leur généralité. Nous avons donc élaboré un modèle de représentation arborescente des opérations, défini la notion de «taux de «vraisemblance» et proposé un jeu d'algorithmes destinés à implémenter le mécanisme de détection d'intentions. L'ensemble de ces éléments à fait l'objet d'une implémentation dans le démonstrateur ***TOASt***.

Les principaux bénéfices de ce mécanisme se situent selon nous à plusieurs niveaux :

- La généricité du mécanisme élaboré fait que le véritable travail de détection d'intentions est fait par les actions terminales, qui sont les seules dépendantes du système considéré. Le moteur de détection d'intentions peut en conséquence être utilisé tel quel, sans modification.
- Cette généricité a également d'autres conséquences particulièrement intéressantes : en plus du mécanisme de détection d'intentions, il est possible d'implémenter à un coût extrêmement faible, des fonctionalités d'exécution avec détection d'erreur et de simulation des opérations que le système est à l'origine conçu pour détecter.

Enfin, nous avons poussé à son paroxysme l'idée selon laquelle toute la connaissance relative aux opérations peut se résumer aux actions élémentaires. Les limites du cadre applicatif de cette approche ont été localisées au niveau du concept pour lequel nous avons proposé l'expression «proprioception étendue».

Perspectives

Les perspectives de ce travail se situent essentiellement au niveau de mise en œuvre du mécanisme de détection d'intentions proposé, puis d'un point de vue plus général sur le plan applicatif. Nous proposons ici quelques pistes intéressantes.

Mise en œuvre

Au niveau du mécanisme de détection d'intentions lui-même, un certain nombre d'évolutions liées à sa mise en œuvre nous paraissent intéressantes.

Représentation des connaissances non opérationelles

Nous avons rappelé plus haut que le problème de l'acquisition et de la représentation des connaissances perceptives n'était pas du ressort de cette étude. Ce point a néanmoins été partiellement abordé, par nécessité, pour la représentation des opérations se rapportant aux objets (par exemple, la saisie). L'intervention de cette connaissance perceptive est pour l'instant statique. Cela signifie que les opérations que le système est capable d'effectuer ou de détecter font appel à certains paramètres stockés dans des *variables* des objets. Ainsi, l'opération de saisie passe par une étape de déplacement autour de l'objet. Dans $\mathcal{TAS}_t$, la position correspondante est, par simplification, considérée comme le centre de l'objet lui-même.

Cette simplification, due à des contraintes techniques (Cf. annexe 1), doit être évitée dans le cas général. Une bonne modélisation objet conduit dans une première étape à ajouter un champ «position de saisie» à la représentation interne de l'objet considéré, puis dans une deuxième étape, à introduire directement une *méthode* de saisie spécifique à l'objet, méthode qui serait alors décrite dans les termes de notre modèle de représentation des connaissances. Ce point confirme qu'une plus grande intégration entre la représentation des connaissances opérationelles et non opérationelles est nécessaire.

Découplage action / but

À l'issue de la dernière partie, nous avons montré que notre mécanisme de détection d'intentions permettait de gérer tout ce qui est du domaine de la proprioception étendue. Celui-ci n'est donc pas adapté à la représentation de connaissances opérationelles pour lesquelles une action doit être effectuée non pas pour elle même, mais dans un but perceptif découplé. Les opérations de ce type sont en général d'assez haut niveau. À titre d'exemple citons des opérations comme «allumer la lumière» ou «remplir le verre». L'opération «allumer la lumière» passerait par une action du type «appuyer sur le bouton jusqu'à ce que la lumière soit allumée». L'opération «remplir le verre» passerait par une action du type «ouvrir le robinet jusqu'à ce que le verre soit rempli». Le problème de ces actions est que la relation entre leur exécution et leur but véritable est une relation non terminale, une connaissance de haut niveau. Par exemple, le but terminal de l'action «appuyer sur le bouton» est «que le bouton soit enfoncé». Le fait que cette action ait pour conséquence que la lumière s'allume relève d'une connaissance de plus haut niveau que les actions elle-mêmes.

À vrai dire, nous pensons que notre modèle de représentation des connaissances n'est pas si loin de pouvoir résoudre ce genre de problèmes. La raison en est que les buts problématiques dont nous venons de donner des exemples ne sont en réalité pas associés à des actions. Ils en sont seulement des conséquences *indirectes*, déterminées par une connaissance de plus haut niveau conceptuel. Ces buts n'étant pas associés à de véritables actions, nous pourrions alors ajouter à notre système une action terminale particulière : l'action vide, «ne rien faire». L'opération «allumer la lumière» s'exprimerait alors par la séquence «Appuyer sur le bouton» puis «ne rien faire (jusqu'à ce que la lumière soit allumée)». L'opération «remplir le verre» s'exprimerait quant à elle par la séquence «ouvrir le robinet» puis «ne rien faire (jusqu'à ce que le verre soit plein)».

Apprentissage

Une dernière perspective, naturelle quand on adopte le point de vue d'un cogniticien, est celle de l'apprentissage. Cette idée se place en fait plus dans le domaine de la conjecture que dans l'ordre de la perspective à court ou moyen terme. Étant donnée la structure de notre modèle de représentation des connaissances, et un jeu d'actions terminales pour un manipulateur

particulier, serait-il possible d'envisager un système d'apprentissage des opérations que nous avons jusqu'ici décrites explicitement ? Un tel système, s'il était concevable, devrait déterminer par lui-même la décomposition des opérations en question, et serait vraissemblablement conduit à effectuer un apprentissage sur les poids de celle-ci…

Discussion

L'apport principal de ce travail se situe dans la proposition de théories permettant d'unifier des concepts issus de la réalité virtuelle avec des champs d'applications tels que celui de la télé-opération. En cela, nous avons retrouvé, par l'analyse théorique, des résultats qui jusqu'ici avaient été constatés par des moyens plus empiriques. L'état de l'art actuel en matière d'assistance peut se positionner selon deux points par rapport au travail présenté dans ce document :

- Dans les deuxième et troisième parties principalement, notre propos a été illustré par une série d'exemples tirés de cas réels, mais qui n'en sont pour la plupart qu'au stade de l'expérimentation dans les laboratoires de recherche.
- D'autre part, un certain nombre de cas d'assistance que nous avons décrits (par exemple l'assistance à la manipulation d'objets) peuvent très bien exister indépendamment du concept de détection d'intentions.

Il est donc intéressant, d'un point de vue général, de suivre le développement des idées à l'heure actuelle encore sujettes à expérimentation, et en particulier de porter attention à la place de la détection d'intentions dans les prochaines applications industrielles.

Annexes

A Description de $TGAS_t$ 199
B Preuve du théorème 207
C Table des figures 209
D Liste des définitions 213
E Glossaire 215
F Index 219
G Bibliographie 223

Annexe A

Description de TOASt

«TOASt : it's 'butter' to get some help.»

Cette annexe a pour objet de décrire brièvement certains aspects techniques du démonstrateur informatique TOASt auquel nous avons souvent fait référence dans la troisième partie de ce document. Nous proposons tout d'abord une description générale de l'architecture du système, puis nous revenons sur l'implémentation du mécanisme de détection d'intentions.

Architecture du système

L'application se présente sous la forme de deux programmes distincts écrits en `C`, `C++` et `Lisp`. Le premier programme constitue l'interface utilisateur (le «client»). Il contient les mécanismes d'interaction avec l'environnement virtuel (prise de contrôle, visualisation, commandes du bras manipulateur…) et de contrôle du niveau ou des types d'assistance. Le deuxième programme constitue le «serveur» d'environnement virtuel. Il contient les mécanisme de gestion de cet environnement (contrôle de la présence des objets disponibles, gestion temporelle et physique de l'environnement, manipulation du bras, détection des collisions…).

Communication

Bien que l'architecture de base du système soit plutôt orientée selon une philosophie «client / serveur» (étant donné que la gestion de l'environnement virtuel est centralisée), les deux applications utilisent un support multicast (Diot et al., 1997) pour échanger l'information, ce qui les rend symétriques en termes de communication réseau. L'utilisation du principe de la communication de groupe permet d'autoriser la présence de plusieurs clients au sein d'une même instance du système. Pour un groupe multicast donné, les contraintes de fonctionnement du système sont les suivantes :

- Un unique serveur d'environnement est autorisé par groupe. Si une tentative de connexion d'un deuxième serveur est détectée, celui-ci n'est pas autorisé à partager le groupe, en est averti, et meurt aussitôt. Un groupe multicast utilisé par le système peut exister sans serveur. Aucun contrôle n'est cependant possible jusqu'à l'arrivée d'un serveur.
- Tout client qui le souhaite peut joindre un groupe multicast utilisé par l'application. Cependant, un unique client peut avoir le contrôle du manipulateur à la fois. Une fois que le contrôle est attribué (par le serveur) à un client, les autres clients sont restreints au simple

rôle de «spectateur» : ils peuvent assister à la scène et modifier leur point de vue, sans pour cela intervenir de manière active sur le déroulement de la situation.

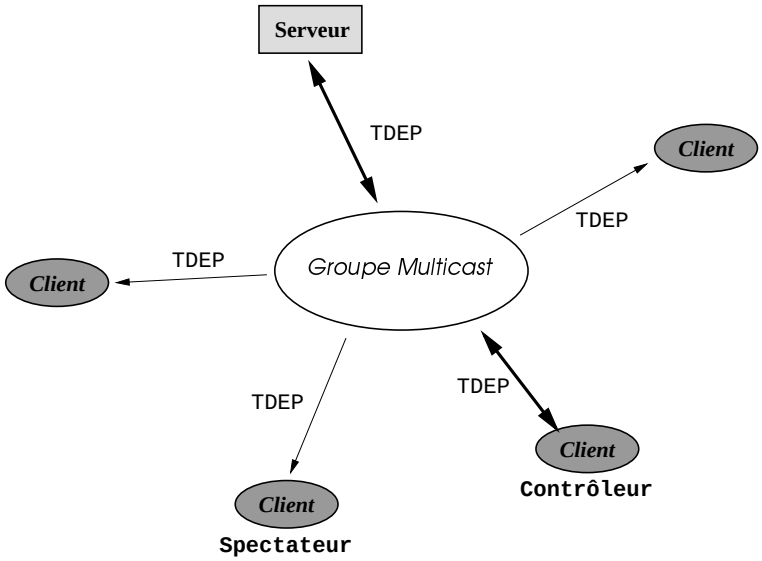

FIG. A.1: $TOAS_t$: architecture réseau

La figure A.1 illustre ce mécanisme. Le système a été conçu initialement selon cette optique dans une perspective d'extension au travail coopératif où plusieurs client se partageraient le contrôle du manipulateur. Les communications de groupe, au dessus des couches UDP / IP-multicast, suivent un protocole applicatif nommé «TDEP» ($TOAS_t$ Data Exchange Protocol). Ce protocole permet de gérer les transactions de départ / arrivée des clients ou des serveurs sur un groupe, demande / obtention / relâche du contrôle du manipulateur, et d'action / perception sur et de l'environnement.

Multithreading

Les clients $TOAS_t$ ont à leur charges différentes choses, parmis lesquelles on trouve la gestion de l'interface graphique, du rendu visuel et des mécanismes d'assistance par détection d'intentions. La coexistence de ces différentes fonctionalités ne va pas sans poser quelques difficultés de conception. À titre d'exemple, notons que tandis que le rendu visuel de l'environnement à un instant donné peut être assez long et risque par conséquent de bloquer les paramètres décrivant l'état de l'environnement, il est au contraire crucial pour le mécanisme de détection d'intentions de disposer des informations les plus récentes, et donc éventuellement d'abandonner des informations devenues obsolètes.

Ces considérations nous ont conduit à utiliser des techniques de multithreading (Winacott, 1993, Powell et al., 1992, Sun, 1996) afin de paralléliser les différentes fonctionalités de l'application. Ainsi, pour obtenir une image toujours cohérente, le moteur d'affichage se construit périodiquement une copie de l'état de l'environnement afin d'en effectuer le rendu. Parallèlement, le gestionnaire réseau peut recevoir des nouvelles données et les mettre à jour en permanence, sans pour cela devoir attendre que le moteur d'affichage ait fini le rendu d'une image.

La figure A.2 ci-contre illustre l'architecture parallèle des clients $TOAS_t$. Chaque fil d'exécution est intrinsèquement une boucle infinie tournant en parallèle avec les autres.

➤ Le fil d'exécution principal de l'application n'a pour objet que de créer les différents threads attribués à chaque fonctionnalité. Cependant, il est également en charge de la réception de tous les signaux (erreurs, interruptions extérieures ou utilisateur). Une fois les threads crées, celui-ci sera bloqué en attente d'un événement.

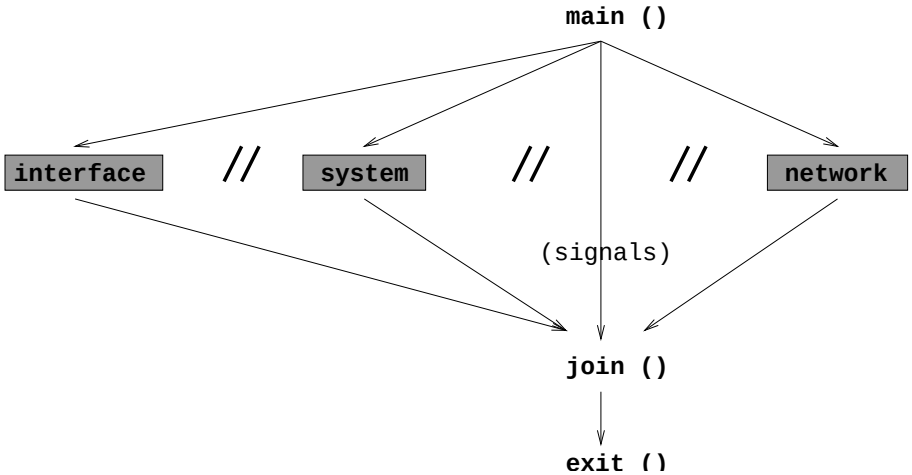

FIG. A.2: $TOAS_t$: architecture parallèle

- Un premier thread est en charge de toutes les fonctionalités de l'interface graphique, de la boucle d'événements et de l'affichage. L'affichage de la scène est implémenté directement au dessus de la bibliothèque OpenGL.
- Un deuxième thread est en charge du système d'assistance et des fonctionalités de détection d'intentions. Les algorithmes correspondants sont eux-même complètement parallélisés (voir section suivante).
- Enfin, un dernier thread est en charge de l'écoute réseau. Il a la responsabilité de mettre à jour les données reçues le plus rapidement possible. Ce thread n'est cependant pas en charge de l'émission des messages. Ceux-ci sont envoyés de manière asynchrone depuis n'importe quelle partie de l'application.

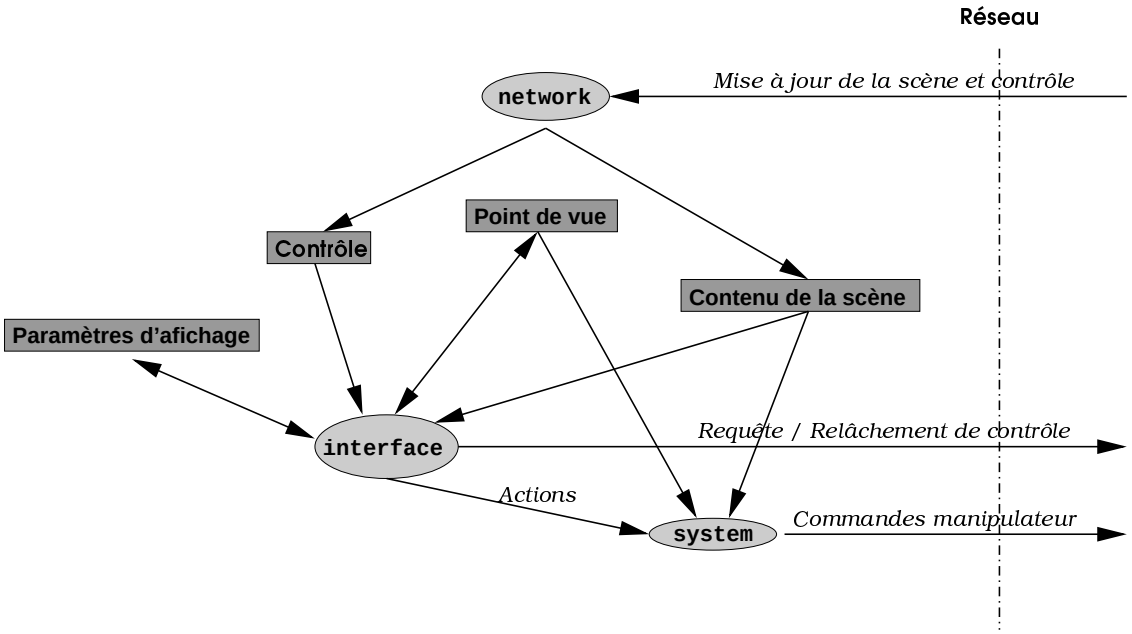

FIG. A.3: $TOAS_t$: synchronisation

Les trois threads principaux de l'application sont en accès concurrent aux données manipulées par le système. Il est donc nécessaire de synchroniser les accès aux données pour éviter toute incohérence ou corruption dans le traitement. La figure A.3 en page précédente donne une vision d'ensemble des types de données manipulées par le systèmes, ainsi que des différents fils d'exécution y accédant :

- Les données d'affichage contiennent tous les paramètres spécifiés par l'utilisateur (mode de tracé, lumière, texture, réflexions...) et ne sont manipulées que par le thread gérant l'interface graphique.
- L'information sur le contrôle du bras manipulateur est reçue par le réseau. Les requêtes de prise ou de relâchement du contrôle sont transmises sur le réseau de manière asynchrone, directement par l'interface graphique.
- Les données concernant le point de vue sur la scène sont modifiées seulement par l'interface graphique, mais sont cependant nécessaires au système de détection d'intentions pour certains cas d'assistance.
- L'information relative au contenu de la scène est réceptionnée depuis le réseau, et utilisée à la fois pour l'affichage et pour la détection d'intentions.
- Enfin, il est à noter que les actions requises par l'utilisateur ne sont jamais transmises directement sur le réseau, mais communiquées au système, dans la mesure où elles sont sujettes à interprétation.

Chaque fois que plusieurs threads sont en accès concurrent aux même données, une synchronisation est nécessaire. Les outils traditionnels (mutex, variables de conditions, sémaphores) ont été utilisés dans ce but.

Détection d'Intentions

Les algorithmes proposés pour la détection d'intentions ont été présentés dans la troisième partie de ce document. Dans un soucis de rester aussi fidèle que possible à leur description formelle, une grande attention a été portée à leur implémentation. Nous proposons de décrire ici quelques aspects techniques de leur fonctionnement, en commençant par l'interface permettant d'inculquer les connaissances au système de détection d'intentions.

Représentation des Connaissances

Comme nous l'avons indiqué précédemment, toutes les connaissances actionnelles, à l'exception des actions élémentaires qui sont par définition codées en dur, peuvent être inculquées dynamiquement au système. Les clients TAS$_t$ n'ont aucune autre connaissance initiale que ces mêmes actions élémentaires, et chargent les connaissances actionnelles requises au lancement de l'application, à partir d'un fichier descriptif (traditionnellement d'extension `.kno` pour «knowledge»).

Ce fichier contient une représentation déclarative des actions que l'utilisateur souhaite définir, et utilise pour ce faire un langage d'extension `Scheme`, variante de la famille `Lisp` (un interpréteur `Lisp` est intégré au système). Pour fixer les idées, reprenons l'exemple de la saisie décrit sur la figure 3.8 en page 142. La figure A.4 en page suivante présente la déclaration `Lisp` équivalente utilisée dans TAS$_t$.

Il existe quatre fonctions `Lisp` de base servant à construire les actions : les deux fonctions définissant la qualité des branches (persistantes ou auxiliaires) et les deux fonctions définissant le mode de parcours de celles-ci (parallèle ou séquentiel). La combinaison d'appels récursifs de ces deux fonctions (qui doit nécessairement se terminer par des appels aux actions terminales) produit un objet `C++` représentant l'action définie par l'utilisateur.

Dans notre théorie, nous avons montré que les actions sont réutilisables à n'importe quel niveau. Plutôt que de définir une action jusqu'à son développement terminal, nous offrons donc,

```
;; Grab an object:
(define grab
  (sequential "Grab"
    ;; Move towards the object and start opening the plier.
    (auxiliary (parallel "_prepare grab"
                  (persistent (parallel "_prepare grab: arm position"
                                (auxiliary plier-position 'self '(- object 10))
                                (persistent arm-orient)
                                ))
                  (auxiliary plier-open)
                  ))
    ;; Finally, really grab it (surround it and close the plier).
    (persistent (sequential "_really grab"
                  (auxiliary (sequential "_really grab: surround"
                                (auxiliary plier-open)
                                (persistent plier-position 'self 'object
                                            'self 'object 1)
                                ))
                  (persistent plier-close)))
    ))

;;; Local Variables:
;;; mode: scheme
;;; eval: (put 'sequential 'scheme-indent-function 1)
;;; eval: (put 'parallel   'scheme-indent-function 1)
;;; End:
```

FIG. A.4: Déclaration `Lisp` de l'action de saisie

par l'intermédiaire de la fonction standard `define` de `Scheme`, la possibilité de définir des actions et de les référencer directement dans d'autres actions. Cette fonctionalité était de toute manière nécessaire à cause des actions terminales. La figure A.5 en page suivante présente une version modularisée de l'action de saisie, qui en devient du même coup plus lisible.

Le lecteur attentif aura d'autre part noté un passage d'argument au niveau des actions terminales, en particulier, les arguments `'self` et `'object`. Chaque action terminale possède une *source* et une *destination*. La source représente ce qui doit être modifié par l'action terminale considérée, et la destination représente l'état final devant être atteint par la source. Les paramètres `'self` et `'object` sont des paramètres spéciaux désignant directement la composante appropriée du manipulateur lui-même, ou de l'objet considéré. Des valeurs numériques sont également possibles. Pour fixer les idées, prenons l'exemple de l'orientation du bras (selon l'axe vertical). Pour demander que le bras manipulateur s'oriente dans la direction d'un objet, nous écrirons `(arm-orient 'self 'object)`. L'action terminale prendra alors automatiquement comme source l'orientation courante du bras manipulateur, et la positionnera à la même valeur que celle de l'objet. Supposons maintenant qu'un objet soit saisi et que nous voulions le positionner au degré zéro d'orientation. Nous écrirons alors `(arm-orient 'object 0.0)`. La source, c'est à dire l'ensemble des paramètres à modifier, est maintenant l'orientation de l'objet et non plus celle du bras. L'orientation du bras ne constitue que le moyen d'y parvenir. C'est là le bénéfice du concept de proprioception étendue.

Initialement, il était prévu de généraliser le mécanisme de passage d'arguments. On aurait voulu par exemple pouvoir décrire une action de pose d'un objet à un endroit précis, ou même sur un autre objet. Malheureusement, cela n'a pas été possible, pour des raisons purement techniques et non pas théoriques : après intégration de l'interpréteur `Lisp` (GNU Guile) dans le système, il s'est avéré que celui-ci ne pouvait fonctionner correctement en environnement multithreadé. Il était par conséquent impossible d'interpréter du code `Lisp` en temps réel à partir du fil d'exécution du système de détection d'intentions. L'interprétation `Lisp` a par conséquent dû être restreinte au chargement initial des connaissances, avant que les différents threads soient créés. Pour la même raison, il est impossible de changer de base de connaissance quand le système est en cours de fonctionnement, bien que toutes les structures nécessaires à cette fonctionalité soient en place.

```
;; Surround the object:
(define really-grab-surround
  (sequential "_really grab: surround"
    (auxiliary plier-open)
    (persistent plier-position 'self 'object 'self 'object 1)
    ))

;; Do the actual grabbing:
(define really-grab
  (sequential "_really grab"
    (auxiliary really-grab-surround)
    (persistent plier-close)
    ))

;; Move towards the object:
(define prepare-grab-arm-position
  (parallel "_prepare grab: arm position"
    (auxiliary plier-position 'self '(- object 10))
    (persistent arm-orient)
    ))

;; Prepare for grabbing:
(define prepare-grab
  (parallel "_prepare grab"
    (persistent prepare-grab-arm-position)
    (auxiliary plier-open)
    ))

;; Toplevel GRAB action:
(define grab
  (sequential "Grab"
    (auxiliary prepare-grab)
    (persistent really-grab)
    ))

;;; Local Variables:
;;; mode: scheme
;;; eval: (put 'sequential 'scheme-indent-function 1)
;;; eval: (put 'parallel   'scheme-indent-function 1)
;;; End:
```

FIG. A.5: Version modulaire de l'action de saisie

Implémentation des algorithmes

Dans la troisième partie de ce document, nous avons décrit l'ensemble des algorithmes utilisés pour l'exécution, la simulation et la détection d'intentions. Du point de vue des fonctionalités requises, le détail de ces algorithmes a fait apparaître un certain nombre de points pour lesquels nous avons eu le soucis de fournir une implémentation aussi proche que possible de leur description formelle. En particulier :

- Nous avons besoin d'un mécanisme permettant un véritable parallélisme des actions, que ce soit en exécution, simulation, ou détection d'intentions. Le mode de parcours parallèle des branches le requiert.
- Nous avons besoin d'un mécanisme de communication événementielle entre les actions et leurs sous-actions. En effet, les algorithmes ont été décrits essentiellement en termes de «signaux» de changement d'état.
- Enfin, les algorithmes nous imposent également la capacité d'interrompre ou de reprendre le déroulement des sous-actions.

L'utilisation massive des techniques de multithreading (Winacott, 1993, Powell et al., 1992, Sun, 1996) précédemment décrites, et en particulier des spécifications Posix permet de résoudre élégamment l'ensemble de ces problèmes :

- Chaque action est, comme nous l'avons dit, un objet C++ héritant selon le cas d'une classe d'actions parallèles ou séquentielles. Les actions terminales sont les seules exceptions à cette règle.
- À chacun de ces objets est associé un fil d'exécution particulier. Plus précisément, le lancement d'une exécution, d'une simulation ou d'une détection d'intentions s'effectue au moyen d'une méthode spécifique qui créé un thread propre à l'objet d'action considéré.
- La communication événementielle entre les actions et leurs sous-actions, c'est à dire concrètement entre plusieurs threads de l'application est implémentée grâce à des sémaphores. Le comportement typique d'un thread d'action non terminale est de créer les threads nécéssaires de ses sous-actions, puis d'attendre que son sémaphore soit lisible, c'est à dire qu'un changement d'état d'une des sous-actions ait été signalé.
- Enfin, les spécifications Posix imposent l'existence d'une fonction d'annulation des threads, `pthread_cancel ()`, qui reproduit exactement notre fonctionnalité d'interruption d'une action. Notons que les threads de Sun n'offrent pas cette fonctionalité, mais proposent en revanche un mécanisme de suspension et de reprise au moyen des fonctions `thr_suspend ()` et `thr_continue ()`.

Cette optique d'implémentation est résumée sur la figure A.6.

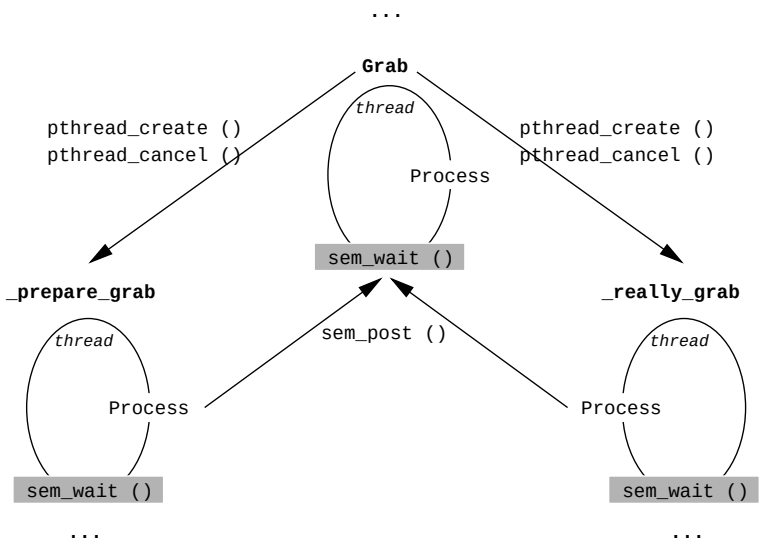

FIG. A.6: Implémentation des algorithmes

Chaque action possède son propre thread, et contrôle la création, la destruction, et l'interruption des sous-threads de l'arborescence. Nous voyons que les threads en question sont essentiellement des boucles infinies qui passent leur temps à attendre un signal de changement d'état, le traiter selon la logique de l'action correspondante, puis se remettre en attente.

L'intérêt de cette approche est que la majeure partie des threads de l'application sont véritablement bloqués, et pas en attente active. À un instant donné, les seuls threads consommant véritablement des ressources sont par conséquent les actions terminales en cours d'exécution, ainsi que les branches «en veille».

Annexe B

Preuve du théorème

Nous proposons ici la démonstration du théorème d'équivalence sémantique énoncé en page 150. Compte tenu du fait que la sémantique d'une opération est entièrement définie par ses transitions, pour montrer que deux opérations sont sémantiquement équivalentes, il suffit de montrer qu'il y a équivalence entre leurs transitions. La démarche de démonstration est donc la suivante : pour chaque transition $\Theta(S, C, \varepsilon)$ de S, nous montrons premièrement qu'il y a équivalence des contextes C et C', c'est à dire que toutes les branches B_j sont dans le même état. Deuxièmement, en fonction de cette équivalence contextuelle, nous montrons que l'événement ε génère la même transition dans les cas de S et de S'.

Rappel des notations :

$S = \{B_1 \rightarrow \ldots \rightarrow B_n\}$
$B = \{B_i \rightarrow \ldots \rightarrow B_n\}$
$S' = \{B_1 \rightarrow \ldots \rightarrow B_{i-1} \rightarrow B\}$

Transition 1 :

C : *S est en cours d'exécution.*
B_n est en cours d'exécution.
$B_1 \ldots B_{n-1}$ sont en veille.
ε : *B_n atteint son but.*
$\Theta(S, C, \varepsilon) \rightarrow O$ atteint son but.

Dans le cas de S', B_n est la dernière branche de B.
B_n en cours d'exécution $\Rightarrow B_i \ldots B_{n-1}$ en veille, et B cours d'exécution.
B en cours d'exécution $\Rightarrow B_1 \ldots B_{i-1}$ en veille et S' est en cours d'exécution.
$\Longrightarrow C \equiv C'$
B_n atteint son but $\Rightarrow B$ atteint son but $\Rightarrow S'$ atteint son but.
$\Longrightarrow \Theta(S, C, \varepsilon) \equiv \Theta(S', C, \varepsilon)$

Transition 2 :

C : *S est en cours d'exécution.*
$B_1 \ldots B_{k-1}$ sont en veille.
$B_k (k < n)$ est en cours d'exécution.

$B_{k+1} \dots B_n$ *sont inactives.*
ε : B_k *atteint son but.*
$\Theta(S, C, \varepsilon) \rightarrow B_k$ passe en veille.
B_{k+1} passe en exécution.

Dans le cas de S', il y a deux possibilités, selon que B_k fait partie ou non de B.

- Supposons que $B_k \notin B$ ($k < i$).
  B_k en exécution $\Rightarrow B_1 \dots B_{k-1}$ en veille et $B_{k+1} \dots B$ inactives.
  B inactive $\Rightarrow B_i \dots B_n$ inactives.
  $\Longrightarrow C \equiv C'$
  B_k atteint son but $\Rightarrow B_k$ passe en veille.
  Si $k < i - 1$, B_{k+1} passe en exécution.
  Si $k = i - 1$, B passe en exécution $\Rightarrow B_i$ passe exécution.
  $\Longrightarrow \Theta(S, C, \varepsilon) \equiv \Theta(S', C, \varepsilon)$
- Supposons que $B_k \in B$ ($k \geq i$).
  B_k en exécution $\Rightarrow B_i \dots B_{k-1}$ en veille, $B_{k+1} \dots B_n$ inactives et B en exécution.
  B en exécution $\Rightarrow B_1 \dots B_{i-1}$ en veille, et S' en exécution.
  $\Longrightarrow C \equiv C'$
  B_k atteint son but $\Rightarrow B_k$ passe en veille et B_{k+1} passe en exécution.
  $\Longrightarrow \Theta(S, C, \varepsilon) \equiv \Theta(S', C, \varepsilon)$

Transition 3 :

C : *S est en cours d'exécution.*
$B_1 \dots B_{k-1}$ *sont en veille.*
$B_k (k < n)$ *est en cours d'exécution.*
$B_{k+1} \dots B_n$ *sont inactives.*
ε : *Le but de* $B_l (l < k)$ *est invalidé.*
$\Theta(S, C, \varepsilon) \rightarrow B_{l+1} \dots B_k$ redeviennent inactives.
B_l repasse en exécution.

Dans le cas de S', il y a deux possibilités, selon que B_l fait partie ou non de B. Le contexte C de O étant identique à celui de la transition précédente, sa démonstration de l'équivalence des contextes s'applique. Il nous reste à démontrer l'équivalence des transitions.

- Supposons que $B_l \notin B$ ($l < i$).
  B_l invalidé $\Rightarrow B_{l+1} \dots B$ redeviennent inactives et B_l repasse en exécution.
  B redevient inactive $\Rightarrow B_i \dots B_n$ redeviennent inactives.
  $\Longrightarrow \Theta(S, C, \varepsilon) \equiv \Theta(S', C, \varepsilon)$
- Supposons que $B_l \in B$ ($l \geq i$).
  B_l invalidé $\Rightarrow B_{l+1} \dots B_n$ redeviennent inactives et B_l repasse en exécution.
  $\Longrightarrow \Theta(S, C, \varepsilon) \equiv \Theta(S', C, \varepsilon)$

Transition 4 :

C : *S est en veille.*
Toutes les branches persistantes sont en veille.
Toutes les branches auxiliaires sont inactives.
ε : *Le but de l'une des branches persistantes* B_k *est invalidé.*
$\Theta(S, C, \varepsilon) \rightarrow$ Le but de S est invalidé.

Dans le cas de S' :
B persistante $\Rightarrow B$ en veille.
$\Longrightarrow C \equiv C'$ Si $B_k \notin B$, alors B_k invalidée $\Rightarrow S$ invalidée.
Si $B_k \in B$, alors B_k invalidée $\Rightarrow B$ invalidée $\Rightarrow S'$ invalidée.
$\Longrightarrow \Theta(S, C, \varepsilon) \equiv \Theta(S', C, \varepsilon)$

Annexe C

Table des figures

RÉALITÉ VIRTUELLE

3.1 Temps-Lieu-Interaction 20
3.2 Immersion-Interaction-Imagination 21
3.3 Les quatre modes d'être 22

4.1 Opérateur *vs.* Environnement 25
4.2 Mental *vs.* Physique 26
4.3 Illusion d'optique 28
4.4 Explicite *vs.* Réflexe 28
4.5 Modèle M_RIC complet 30
4.6 Niveau conceptuel de l'information 33
4.7 Les quatre «I» du réel 34

5.1 M_RIC-*e* : virtualisation de l'agent externe. 40
5.2 M_RIC-*m* : virtualisation du monde. 44
5.3 M_RIC-*me* : Réalité virtuelle au deuxième degré 48
5.4 Réalité Virtuelle et Triangle Sémiotique 49
5.5 Immersion et Triangle Sémiotique 50
5.6 M_RIC-*o* : pseudo-immersion. 51
5.7 M_RIC-*ae* : Réalité Augmentée par altération de l'agent externe 52
5.8 M_RIC-*ao* : Réalité Augmentée par altération de l'agent opératoire 52
5.9 Transmutation virtuelle 56
5.10 M_RIC-*i* : détection d'intentions 57
5.11 M_RIC-*l* : réalité virtuelle et langage 60

ASSISTANCE

2.1 Modèle L_iSA complet 70

3.1 Génération de perception 75

3.2 Altération de la perception ... 76
3.3 Transmodalisation de la perception ... 78
3.4 Reconstitution de perception ... 79
3.5 Filtrage de l'action ... 81
3.6 Commande paramétrique ... 82
3.7 Exécution virtuelle ... 84
3.8 Résumé des cas d'assistance ... 84

4.1 Génération d'action ... 89
4.2 Altération de l'action ... 90
4.3 Transmodalisation de l'action ... 91
4.4 Reconstitution de l'action ... 93
4.5 Exécution paramétrique ... 94
4.6 Filtrage de la perception ... 96
4.7 Commande virtuelle ... 97
4.8 Résumé global des d'assistance ... 98

5.1 Les modèles M_RIC et L_iSA ... 102
5.2 Niveau conceptuel de l'information en télé-opération immersive ... 103
5.3 Niveau conceptuel de l'information en télé-opération «brute» ... 104
5.4 Niveau conceptuel de l'information en télé-opération «semi-autonome» ... 106
5.5 Niveau conceptuel optimal de l'information en télé-opération ... 107
5.6 Niveau conceptuel de l'information par détection d'intentions ... 108
5.7 Comparaison des niveaux conceptuels ... 109

DÉTECTION D'INTENTIONS

3.1 Exemple de réseau procédural ... 126
3.2 Représentation graphique des modes de parcours ... 128
3.3 Unicité du mode de parcours des actions ... 128
3.4 Structure de l'action de «saisie» ... 131
3.5 Représentation graphique des catégories de sous-actions ... 135
3.6 Représentation de l'action de «saisie» ... 139
3.7 Commandes élémentaires de T_CAS_t ... 141
3.8 Action de saisie pour T_CAS_t ... 142
3.9 Exemple d'équivalence sémantique ... 151

4.1 Représentation équivalente de la saisie ... 155
4.2 Pondération de l'opération de saisie. ... 163
4.3 T_CAS_t en phase de confirmation de saisie ... 166

5.1 T_CAS_t en phase de simulation de pose. ... 174
5.2 T_CAS_t en phase de détection d'erreur ... 175

6.1 Scénario – première étape ... 179
6.2 Scénario – deuxième étape ... 180
6.3 Scénario – troisième étape ... 181

6.4 Scénario – quatrième étape 182
6.5 Scénario – cinquième étape 183
6.6 Scénario – sixième étape 184
6.7 Scénario – septième étape 185
6.8 Scénario – huitième étape 186
6.9 Scénario – neuvième étape 187
6.10 Scénario – dixième étape 188

1 M_RIC, L_iSA, T_OAS_t : trois apports scientifiques 191

A.1 T_OAS_t : architecture réseau 200
A.2 T_OAS_t : architecture parallèle 201
A.3 T_OAS_t : synchronisation 201
A.4 Déclaration `Lisp` de l'action de saisie 203
A.5 Version modulaire de l'action de saisie 204
A.6 Implémentation des algorithmes 205

Annexe D

Liste des définitions

Définitions

1 Opérateur 26
2 Environnement 26
3 Monde 27
4 Agent de Contrôle 29
5 Agent de transcription 29
6 Agent Opératoire 29
7 Agent Externe 29
8 Agent Mental 29
9 Agent Physique 29
10 Interaction 34
11 Virtualisation 38
12 Interface 68
13 Manipulateur 69
14 Système 70
15 Action 143
16 Branche 143
17 État d'une action 144
18 Contexte d'une action 144
19 Événement 144
20 Transition 144
21 Poids d'une action 156
22 Taux de vraisemblance (parallèle) 156
23 Taux de vraisemblance (séquentiel) 156
24 Seuil de confiance 156
25 Action englobante 161
26 Niveau d'activation 163
27 Seuil d'activation 164

28 Taux de vraisemblance d'une action terminale 164

● Vocabulaire

1 Réalité Virtuelle 38
2 Réalité Virtuelle Artificielle 39
3 Réalité Virtuelle Altérée 39
4 Environnement Virtuel 41
5 Monde Virtuel 44
6 Restitution 68
7 Commande 68
8 Acquisition 69
9 Exécution 69
10 Assistance 107
11 Mode d'une action 143
12 Qualité d'une branche 143
13 Plausibilité d'une action 157

▲ Formulations

1 Génération de Perception 74
2 Altération de la Perception 76
3 Transmodalisation de la perception 78
4 Reconstitution de Perception 79
5 Filtrage de l'action 81
6 Commande paramétrique 83
7 Exécution virtuelle 84

▼ Reformulations

1 Génération d'action 88
2 Altération de l'action 90
3 Transmodalisation de l'action 91
4 Reconstitution d'action 92
5 Exécution paramétrique 94
6 Filtrage de la perception 95
7 Commande virtuelle 96

Annexe E

Glossaire

Avatar

. 45

Un avatar est une représentation virtuelle de l'opérateur dans un monde de synthèse. Les environnements virtuels distribués, notamment, utilisent des avatars pour représenter chaque participant et leur permettre de s'identifier. Un avatar peut aller du simple cube à un modèle très complexe de l'humain avec par exemple des textures d'origine photographique pour le visage.

BNF

. 128

Acronyme de «Backus-Naur Form». Une BNF est une manière très populaire d'exprimer une grammaire. C'est la méthode la plus utilisée pour décrire les langages de programmation informatique.

CAO

. 17

Acronyme de «Conception Assistée par Ordinateur».

Exosquelette

. 46

Ce terme désigne une certaine catégorie de périphériques de réalité virtuelle utilisés en général pour la saisie de commandes. Ces périphériques sont composés de segments articulés et recouvrent la plupart du temps le bras de l'opérateur (ou sa main dans le cas de gants), donnant l'impression d'un squelette externe, d'où leur nom. Les plus complexes d'entre eux permettent d'acquérir la position du bras et de la main de l'opérateur très fidèlement, articulation par articulation. Par rapport à de simples capteurs de position, les exosquelettes ont l'avantage de pouvoir produire du retour d'effort. Cependant, ce sont des périphériques bien plus lourds à concevoir et utiliser que de simples capteurs, ou même des combinaisons souples.

Fil de Fer

. 16

En Anglais «Wireframe». Technique infographique ayant pour but de représenter des surfaces en ne traçant que les bords et les angles. Cette technique de tracé est beaucoup

moins coûteuse en temps de calcul que les techniques de remplissage des surfaces.

Gant de Données
. 16
En Anglais «Dataglove». Périphériques recouvrant tout ou une partie de la main, et transmettant des informations sur la position de celle-ci. Ces périphériques sont proposés sous diverses formes. Les plus simples contiennent uniquement un capteur de position pour toute la main. Les plus complexes fournissent une information allant jusqu'à la position de chaque articulation de chaque doigt.

LCD
. 16
Écrans à cristaux liquides très utilisés pour les ordinateurs portables et les jeux vidéos miniatures.

Proprioception
. 21
La proprioception est le fait de percevoir son propre corps de manière interne, c'est à dire indépendamment de l'environnement. La perception visuelle de soi n'est donc pas une perception proprioceptive. Au contraire, c'est le sens proprioceptif qui renseigne l'humain sur lui-même, son orientation, la position de ses membres. . . Ce sens est à mettre en relation avec le sens *kinesthésique* qui informe l'humain sur les mouvements de son propre corps.

Sémiotique
. 49
La «sémiotique» (ou «sémiologie») est la science des signes. Elle étudie la manière dont les systèmes de communication symbolique sont produits et utilisés par des individus ou des collectivités.

Track Ball
. 41
Version étendue de la «souris» des ordinateurs, consistant le plus souvent en une boule et quelques boutons. La boule tient dans la main de l'opérateur et dispose d'au minimum trois degrés de liberté (allant jusqu'à six), ce qui permet de l'associer au déplacement d'objets dans un environnement virtuel par exemple.

Visiocasque
. 16
En Anglais «Head Mounted Display» ou «HMD» dans sa version abrégée. Casque de visualisation en trois dimensions, fournissant à l'utilisateur un mini-écran pour chaque œuil, chacun avec une image légèrement différente de l'autre ce qui permet de restituer le relief. Ces casques vont généralement de pair avec des capteurs de position. Ces capteurs transmettent en temps réel la position de la tête du sujet, ce qui permet au système informatique de recalculer immédiatement les images devant être affichées sur chacun des écrans.

VRML
. 41
VRML, acronyme de «Virtual Reality Modeling Language», est un langage de description d'environnements virtuels, principalement inspiré de la spécification d'OpenGL, le premier véritable standard en matière de bibliothèque graphique. Sa première version, apparue vers le milieu des années 90, permettait de décrire des environnements statiques. La version 2 (aussi connue sous le nom de VRML97, Moving Worlds), ajoute des

fonctionnalités d'animation en temps réel et quelques méthodes d'interaction primaires avec l'utilisateur. Son succès actuel, dû à la politique marketing menée à son endroit bien plus qu'à la qualité de conception du langage, en a fait le standard *de facto* de représentation et d'échange d'environnements virtuels au travers de l'Internet.
Voir le «VRML repository» à : http ://www.web3d.org/vrml/vrml.htm

Annexe F

Index

A

acquisition, **69**
action, **143**
    état d'une, **144**
    altération de l', 90
    catégorie d'une, 133–140
    contexte d'une, **144**
    englobante, **161**
    filtrage de, 81
    génération d', 88
    mode d'une, 127–132, **143**
    poids d'une, 154–155, **156**
    reconstitution d', 92
    terminale, **140**
    transmodalisation de l', 91
agent de contrôle, **29**
agent de transcription, **29**
agent externe, **29**
agent mental, **29**
agent opératoire, **29**
agent physique, **29**
Anderson (1983), 148, 223
assistance, **107**
    intelligente, 105
    niveau conceptuel de, 101–107
avatar
    glossaire, *215*
Azuma (1997), 73, 99, 223

B

Backes et al. (1998), 80, 223
Bajura et al. (1992), 75, 223
Balaguer et Gobetti (1995), 43, 223
Barfield et al. (1995), 58, 223
Berger et al. (1996), 75, 224
biocapteur, 59
BNF
    glossaire, *215*
branche, **143**
    qualité d'une, **143**
Browning et al. (1993), 45, 224
Burdéa
    réalité virtuelle, 20
Burdea et Coiffet (1993), 15, 20, 23, 24, 34, 44, 55, 224
Burdea (1993), 20, 21, 23, 24, 34, 44, 55, 224

C

Côté et Lavallée (1995), 78, 225
Calrsson et Jää-Aro (1994), 17, 224
CAO, 17, 41
    réalité virtuelle et, 41
    glossaire, *215*
Cellerier (1979), 148, 224
Chirac et Tiberi (2000), 224
Coiffet
    immersion, 20
    interaction, 20
    réalité virtuelle, 20
commande, **68**
    paramétrique, 83
    virtuelle, 96
contexte

d'une action, voir action

D

détection d'intentions, 55, 106
Detriché et al. (1988), 99, 225
Diot et al. (1997), 199, 225
Dubois et al. (1997), 17, 81, 225
Duchon et Balet (1996), 41, 42, 225
Dugelay et al. (1998), 74, 225

E

Ellis (1996), 58, 225
environnement, **26**
environnement virtuel, **41**
    altéré, 42
    artificiel, 42
équivalence sémantique, **150**
    démonstration, **207**
Ernadotte et al. (1997), 74, 226
état
    d'une action, voir action
Evans
    visiocasque, 16
événement, **144**
exécution, **69**
    paramétrique, 94
    virtuelle, 84
exosquelette
    glossaire, *215*

F

Fagot (1998), 42, 226
Fernando et al. (1995), 90, 226
Fertey et al. (1995), 46, 73, 226
fil de fer, 16
    glossaire, *215*
Fisher
    gant de données, 16
Fuchs
    interaction, 19
    réalité virtuelle, 19
Fuchs et al. (1999), 49, 50, 58, 226
Fuchs (1996), 19, 20, 23, 24, 45, 226

G

gant de données, 16
    Fisher, voir Fisher
    glossaire, *216*
Giralt et al. (1993), 97, 105, 226
Grumbach et Verna (1996), 9, 49, 227
Grumbach (1994), 49, 227
Grumbach (1995), 49, 227

H

Heilig
    Sensorama, 15
    Théâtre Expérimental, 16
    visiocasque, 16
Heilig (1960), 16, 227
Heilig (1992), 15, 227
Hine et al. (1994), 78, 227

I

immersion, 44, 55
    Coiffet, voir Coiffet
    immersion réelle, 49
    pseudo-, 50
immersive
    réalité virtuelle, 44
interaction, **34**
    Coiffet, voir Coiffet
    Fuchs, voir Fuchs
interface, **67**, **68**

K

Kanade et al. (1999), 39, 227
Kenedy et al. (1997), 35, 57, 228
Kheddar et al. (1997), 55, 228
Kheddar et al. (1998), 55, 228
Kim (1996), 83, 228
Kissileva (1998), 42, 228
Kitamura et al. (1998), 76, 80, 228
Krueger, 42
    réalité artificielle, 22, 38
    réalité virtuelle, 22
Krueger (1991), 22, 24, 38, 39, 42, 228

L

Lévy
    réalité virtuelle, 22
Lévy (1995), 22–25, 229
LCD
    glossaire, *216*

Loomis et al. (1998), 77, 228
Lumia et Albus (1988), 105, 229

M

manipulateur, **69**
Mazeau et Bryche (1995), 56, 229
Mellor (1995), 52, 75, 229
Mendenhall, 42
    réalité altérée, 22, 39
    réalité virtuelle, 22
Miller et al. (1992), 43, 229
mode
    d'une action, voir action
monde, **27**
monde virtuel, **44**

N

Nagao et Rekimoto (1995), 73, 229
niveau conceptuel
    de l'assistance, voir assistance
    et complexité exécutoire, 118
niveau d'activation, **163**
Norman et Rumelhart (1975), 125, 229
Norman (1980), 28, 229

O

Ogden et Richards (1923), 49, 230
opérateur, **26**
opération
    mode d'une, 127–132
OTAN (1991), 146, 230

P

Papin (1999), 17, 230
perception
    altération de la, 76
    filtrage de la, 95
    génération de, 74
    reconstitution de, 79
    transmodalisation de la, 78
Pimentel et Teixeira (1994), 15, 21, 23, 45, 55, 230
plausibilité, **157**
poids
    d'une action, voir action
Poitrenaud (1998), 146, 230
Powell et al. (1992), 200, 204, 230
prédiction, 35, 57
proprioception, 21, 56
    étendue, 175
    Quéau, voir Quéau
    glossaire, *216*

Q

Quéau
    proprioception, 21
    réalité virtuelle, 21
Quéau (1993), 21, 23, 24, 27, 57, 230
qualité
    d'une branche, voir branche

R

réalité, **35**
réalité altérée, **39**
    Mendenhall, voir Mendenhall, voir Mendenhall
    Vanderheiden, voir Vanderheiden, voir Vanderheiden
réalité artificielle, **39**
    Krueger, voir Krueger, voir Krueger
réalité augmentée, **50**
réalité virtuelle, 19–24, **38**, 37–40
    Burdéa, voir Burdéa
    CAO et, 41
    Coiffet, voir Coiffet
    Fuchs, voir Fuchs
    immersive, 44
    Krueger, voir Krueger
    Lévy, voir Lévy
    Mendenhall, voir Mendenhall
    Quéau, voir Quéau
    triangle sémiotique et, 48
    Vanderheiden, voir Vanderheiden
récursion, voir récursion
réseau procédural, 126
Rekimoto (1995), 73, 230
Rekimoto (1997), 51, 73, 231
restitution, **68**
Richard et al. (1995), 77, 231
Richard (1990), 126, 132, 148, 231
Richard (1999), 146, 231

S

sémiotique
    triangle, 48

glossaire, *216*
Sacerdoti (1977), 125, 231
Scapin (1986), 135, 231
Schenker (1991), 105, 231
Schloerb (1995), 58, 231
Schroer (1998), 105, 232
Sensorama
Heilig, voir Heilig
seuil d'activation, **164**
seuil de confiance, **156**
Simsarian et Akesson (1997), 50, 232
State et al. (1996), 75, 232
Sturman (1992), 93, 232
Sun (1996), 200, 204, 232
Sutherland
visiocasque, 16
système, **70**
Sébillotte (1983), 131, 232
Sébillotte (1988), 140, 232

T

télé-opération, **53**
catégories de , 53
Tachi et Yasuda (1994), 54, 58, 232
taux de vraisemblance, 121
taux de vraisemblance, 153–157
parallèle, **156**
séquentiel, **156**
terminal, **164**
Théâtre Expérimental
Heilig, voir Heilig
Thalmann (1998), 45, 233
trackball, 41
glossaire, *216*
triangle sémiotique, voir sémiotique

V

Vanderheiden, 42
réalité altérée, 22, 39
réalité virtuelle, 22
Vanderheiden et Mendenhall (1994), 17, 22, 24, 39, 42, 93, 233
Van Lhen et Brown (1980), 125, 233
Verna et Grumbach (1998a), 9, 233
Verna et Grumbach (1998b), 9, 234
Verna et Grumbach (1999), 9, 234
Verna (1997a), 9, 233
Verna (1997b), 9, 233
Verna (1999), 9, 233
virtualisation, **38**
visiocasque
Evans, voir Evans
Heilig, voir Heilig
Sutherland, voir Sutherland
glossaire, *216*
VRML
glossaire, *216*

W

Wenzel, 17
Winacott (1993), 200, 204, 234

Annexe G

Bibliographie

Anderson, J., (1983).
The Architecture of Cognition. Harvard.
. 148

Azuma, R., (1997).
A Survey of Augmented Reality. Dans *Presence*, volume 6/4, pages 335–385. MIT Press.
. . . . . 73, 99

➤ Un panorama complet des principaux domaines d'application de la réalité augmentée. Ce panorama s'agrémente de considérations techniques sur les problèmes majeurs que toute application de réalité augmentée peut s'attendre à devoir résoudre, faisant par la même de cet article une bonne introduction au domaine pour quiconque souhaite se lancer dans l'élaboration d'une telle application.

Backes, P. G., Peters, S. F., Phan, L., et Tso, K. S., (1998).
Task Lines and Motion Guides. Dans *Presence*, volume 7/5, pages 494–502. MIT Press.
. . . . . 80, 80

➤ Cet article décrit le concept de «lignes de tâche et guides de déplacement». L'idée est de téléopérer un robot en deux phases, la première consistant à prédéfinir un trajet à suivre et la deuxième consistant à ne contrôler en temps réel que la vitesse de déplacement du robot le long de cette trajectoire prédéfinie.

Bajura, M., Fuchs, H., et Ohbuchi, R., (1992).
Merging Virtual Reality with the Real World : Seing Ultrasound Imagery within the Patient. Dans *SIGGRAPH'92*, volume 26, pages 203–210.
. 75

➤ Les auteurs présentent un système de réalité augmentée permettant d'incruster l'image virtuelle d'un fœtus (reconstituée à partir de données ultrasoniques) sur l'image réelle du ventre de la mère. Une illusion de transparence est ainsi donnée au gynécologue.

Balaguer, J.-F. et Gobetti, E., (1995).
i3D : a High Speed Web Browser. Dans *VRML'95*.
. 43

➤ Description de i3D, un navigateur 3D développé au CERN et permettant la visualisation d'environnements virtuels décrits en VRML

Barfield, W., Hendrix, C., Bjorneseth, O., Kaczmarek, K., et Lotens, W., (1995).
Comparison of Human Sensory Capabilities with Technical Specifications of Virtual Environments Equipments. Dans *Presence*, volume 4/4, pages 329–356. MIT Press.
. 58

➤ Un tableau comparatif très complet sur les techniques de réalité virtuelle permettant de restituer de l'information selon les modalités perceptives de l'humain, et qui met en évidence un certain nombre de carences dans ce domaine.

Berger, M.-O., Simon, G., Petitjean, S., et Wrobel-Daucourt, B., (1996).
Mixing Synthesis and Video Images of Outdoor Environments : Application to the Bridges of Paris. Dans *GTRV'96*, pages 8–12, Toulouse.
. 75

➤ Dans cet article, les auteurs présentent un système de réalité augmentée ayant pour but de simuler différentes conditions de lumière sur un ponts de Paris. Ce système permet, par superposition de l'image réelle avec un éclairage virtuel, de choisir le meilleur emplacement des projecteurs sans pour cela faire aucun essai réel.

Browning, Cruz-Neira, Sandin, et deFanti, (1993).
Projection based Virtual Environment and Disability. Electronic visualisation lab, Design visualisation lab, Illinois University.
. 45

➤ CAVE est un système de réalité virtuelle immersive basé sur des techniques de projection. L'utilisateur évolue dans une grande pièce cubique sans casque de visualisation, mais les mouvements de la tête sont pistés par des capteurs disposés sur les parois, et une image calculée en temps réel est projetée sur chaque mur de la pièce en fonction de la position de celui-ci.

Burdea, G., (1993).
Virtual Reality Systems and Applications. Dans *Electro'93 International Conference*, page 164. Edison.
. . . . . 20, 21, 23, 23, 23, 24, 34, 44, 55

Burdea, G. et Coiffet, P., (1993).
La Réalité Virtuelle. Hermès.
. . . . . 15, 20, 20, 20, 23, 23, 23, 23, 24, 34, 44, 55

➤ Ce livre dresse un panorama technologique des interfaces et systèmes de réalité virtuelle, décrit les problèmes liés aux performances de ces systèmes et les champs d'applications, avec les contraintes qu'ils imposent.

Calrsson, C. et Jää-Aro, K.-M., (1994).
The Dive Laboratory. Dans *Presence*, volume 4/1, pages 431–440. MIT Press. http ://www.sics.se/dive/.
. 17

➤ Dive (Distributed Interactive Virtual Environment) est à l'heure actuelle le plus célèbre système de réalité virtuelle distribuée à travers le réseau Internet. Basé sur la technologie du multicast, il permet à un nombre potentiellement indéterminé d'utilisateurs de se rencontrer dans un environnement virtuel partagé en temps réel.

Cellerier, G., (1979).
Structures Cognitives et Schèmes d'Action. *Archives de Psychologie*, (47) :87–122.
. 148

Chirac, J. et Tiberi, J., (2000).
A Statistical Approach to Bogus Computation of the Number of Right Wing Electors in Paris.
. . . . . .

➤ Dans cet article, les auteurs présentent une méthode statistique permettant de surévaluer considérablement le nombre d'électeurs votant à droite dans la capitale Française. Des résultats expérimentaux sur certains arrondissements sont présentés.

Côté, J. et Lavallée, J., (1995).
Augmented Reality Graphic Interface for Upstream Dam Inspection. Dans *SPIE, Photonics East*.

. 78

➤ Un exemple d'application de la réalité augmentée à l'inspection sous marine des structures d'un barrage. Cet article montre notamment comment, grâce à un modèle géométrique tridimensionnel du barrage ainsi qu'à l'utilisation de capteurs de position, on peut générer une image virtuelle de celui-ci, alors que la visibilité réelle est complètement nulle.

Detriché, J.-M., Rouyer, J.-L., et Lesigne, B., (1988).
La Robotique Nucléaire au Service des Handicapés Moteurs. *Clefs CEA*, (10) :37.

. 99

➤ Cet article décrit l'application de technologies de robotique nucléaire à la réhabilitation de personnes handicapées, au travers des projets Spartacus puis Master du CEA.

Diot, C., Dabbous, W., et Crowcroft, J., (1997).
Group Communication. *IEEE Journal on Selected Area in Communication*. Special Issue on Group Communication.

. 199

➤ Ce article présente dans sa généralité le concept de communication de groupe, par opposition à la communication point à point. Les problèmes d'adressage et de routage, les protocoles de routage multicast, le contrôle de traffic, la fiabilité, ainsi que les applications sont discutés.

Dubois, E., Nigay, L., et Troccaz, J., (1997).
Interaction Chirurgien-Système : Gestes Médico-Chirurgicaux Assistés par Ordinateur. Dans *IHM'97*, pages 207–208, Toulouse.

. . . . . 17, 81

➤ Un exemple d'application de la réalité augmentée dans le domaine chirurgical, pour la ponction péricardique. Le système décrit assiste notamment l'opérateur en appliquant un retour d'effort sur le dispositif de guidage de l'aiguille, ce qui empêche les mouvements accidentels de celui-ci.

Duchon, J. et Balet, O., (1996).
La Réalité Virtuelle, un Outil de Prototypage et de Communication en Avant-Projet. Dans *GTRV'96*, pages 57–66, Toulouse.

. . . . . 41, 42

➤ Cet article décrit un système de prototypage illustré par une application de CAO basée sur des techniques de réalité virtuelle. Cette application permet de prototyper des modèles de satellites avant la phase de réalisation proprement dite.

Dugelay, J.-L., Fintzel, K., valente, S., Dubois, P., et Delingette, H., (1998).
Vers la Téléconférence Virtuelle Réaliste. Dans *GTRV'98*, Issy-les-Moulineaux.

. 74

➤ Dans cet article, les auteurs décrivent un certain nombre de techniques et d'outils d'imagerie virtuelle permettant d'élaborer des systèmes de vidéoconférence plus confortables que les systèmes actuels.

Ellis, S., (1996).
Presence of Mind : a Reaction to Thomas Sheridan's "Further Musings on the Psychophysics of Presence". Dans *Presence*, volume 5/2, pages 247–259. MIT Press.

. 58

➤ Ellis revient dans cet article sur la notion de téléprésence dans le sens psychologique du terme, c'est à dire la sensation d'être physiquement présent dans un environnement distant, et montre, grâce à des exemples d'applications, que l'on a tort en général de croire que plus la sensation de présence est élevée, plus la téléopération est efficace.

Ernadotte, D., Fuchs, P., et Laurgeau, C., (1997).
Incrustation d'Objets Virtuels sur des Scènes Réelles en Vue de l'Aide à la Télé-Opération. Centre Robotique de l'école des Mines de Paris.
. 74

➤ Ce tutoriel présente les problèmes techniques majeurs liées à l'apport d'objets virtuels sur des scènes réelles filmées : calibration des caméras, suppression des surfaces cachées. . .

Fagot, A., (1998).
Application d'une Solution Réalité Virtuelle à un Cas Industriel. Dans *GTRV'98*.
. 42

➤ Présentation d'un système de réalité virtuelle permettant de travailler sur l'aménagement d'intérieur dans un immeuble de bureaux. L'aménagement virtuel permet de proposer différentes solutions aux clients afin d'évaluer leurs souhaits et de répondre plus conformément à leurs attentes.

Fernando, T., Fa, M., Drew, P. M., et Mulin, M., (1995).
Constraint Based 3D Manipulation Techniques within Virtual Environments. Dans Earnshaw, R. A., Vince, J. A., et Jones, H., editors, *Virtual Reality Applications*, pages 71–89. Academic Press.
. 90

➤ Les auteurs présentent un ensemble de techniques et d'applications de manipulation tridimensionnelle d'objets dans le cadre de mouvements partiellement contraints.

Fertey, G., Delpy, T., Lapierre, M., et Thibault, G., (1995).
An Industrial Application of Virtual Reality : and Aid for Designing maintenance Operations in Nuclear Plants. Dans *L'interface des Mondes Réels et Virtuels*, pages 151–162.
. . . . . 46, 73

➤ Une application de Réalité Virtuelle destinée à l'entraînement en vue d'opérations réelles dans une centrale nucléaire est présentée.

Fuchs, P., (1996).
Les Interfaces de Réalité Virtuelle. Association des Journées Internationales de l'Informatique de Montpellier-District.
. . . . . 19, 19, 20, 20, 23, 23, 24, 45

➤ Ce livre présente un panorama complet des interfaces et périphériques matériels de réalité virtuelle. Il présente leurs caractéristiques techniques ainsi que les facteurs humains devant être pris en compte pour leur conception et leur utilisation.

Fuchs, P., Nashashibi, F., et Lourdeaux, D., (1999).
A Theoretical Approach of the Design and Evaluation of a Virtual Reality Device. Dans *GTRV'99*, pages 11–20, Laval.
. . . . . 49, 50, 58

➤ Cet article présente un point de vue théorique sur la notion de réalité virtuelle et en particulier de l'immersion, dans le but de fournir une méthode pour la conception et l'évaluation de nouveaux périphériques de réalité virtuelle.

Giralt, G., Chatila, R., et Alami, R., (1993).
Remote Intervention, Robot Autonomy and Teleprogramming : Generic Concepts and Real World Application Cases. Dans *International Conference on Intelligent Robots and Systems*, pages 314–320. IEEE/RSJ, IEEE Computer Society Press.
. . . . . 97, 105

➤ Les auteurs proposent un panorama des applications de robotiques selon quatre grandes classes d'application : la télé-opération où le robot est totalement asservi, la télé-opération «avancée» contenant les cas de téléprésence et la robotique semi-autonome, les systèmes autonomes purement réactifs et les robots autonomes à programmation au niveau tâche.

Grumbach, A., (1994).
Cognition Artificielle. Addison-Wesley.
. 49

➤ Étude et présentation des principaux concepts liés à la cognition. L'auteur présente le principe du triangle sémiotique, cadre dans lequel il présente les modèles classiques de représentation et de traitement de l'information : réseaux neuro-mimétiques, systèmes experts, traitement du langage naturel...

Grumbach, A., (1995).
Modèle Cognitif d'Interaction avec un Monde Virtuel. Dans *L'Interface des Mondes Réels et Virtuels*, pages 275–285, Montpellier.
. 49

Grumbach, A. et Verna, D., (1996).
Assistance Cognitive à la Télé-Opération en Monde Virtuel. Dans *GTRV'96, cinquièmes journés nationales réalité virtuelle*, pages 38–46, Toulouse, France. IRIT.
. . . . . 9, 49

➤ Cet article décrit dans les grandes lignes le principe de l'assistance à la télé-opération en monde virtuel. La relation avec le triangle sémiotique et le concept de détection d'intentions sont introduits.

Heilig, M., (1960).
Stereoscopic Television Apparatus for Individual Use. US patent n°2,955,156.
. 16

➤ Brevet américain déposé par Heilig pour son casque stéréoscopique de visualisation. Ce casque, ancêtre des visiocasques modernes, utilisait des diapositives et comportait également des capacités de stéréophonie et d'inclusion d'odeurs.

Heilig, M., (1992).
Enter the Experimental Revolution. Dans *The Cyberarts Conference*, pages 292–305, Pasadena CA.
. 15

➤ Article autobiographique décrivant les débuts de l'auteur dans le domaine de la Réalité Virtuelle, l'histoire de son Théâtre Expérimental (le Sensorama en est une version simplifiée) et l'incrédulité des scientifiques de l'époque à laquelle il dû faire face.

Hine, B. P., Stoker, C., Sims, M., Rasmussen, D., et Hontalas, P., (1994).
The Application of Telepresence and Virtual Reality to Subsea Exploration. Dans *ROV'94*.
. 78

➤ Cet article décrit l'utilisation de techniques de réalité virtuelle pour le téléguidage de sous-marins : un téléopérateur est mis en immersion dans l'environnement distant, grâce en particulier à des images de synthèse reconstituant l'environnement du sous-marin à partir de données issues de ses capteurs.

Kanade, T., Rander, P., Vedula, S., et Saito, H., (1999).
Virtualized Reality : Digitizing a 3D Time-Varying Event As Is and in Real Time. Dans Yuichi Ohta, H. T., editor, *Mixed Reality, Merging Real and Virtual Worlds*, pages 41–57. Springer-Verlag.
. 39

➤ Cet article décrit un projet consistant à reconstituer artificiellement des image d'une scène réelle prises à des points de vue arbitraire, par extrapollation d'images réelles de cette même scène mais issues de multiples caméras situées à des endroits fixes.

Kenedy, R., Lanham, S., Drexier, J., Massey, C., et Liliental, M., (1997).
A Comparison of Cybersicknesses, Incidences, Symptom Profiles, Measurement Techniques and Suggestions for Future Research. Dans *Presence*, volume 6/6, pages 639–644. MIT Press.

. . . . . 35, 57

➤ Cet article comprend un tableau très intéressant et très instructif sur les grands types de malaises des mondes virtuels («cybersicknesses») et leur sources.

Kheddar, A., Tzafestas, C., Blazevic, P., et Coiffet, P., (1998).
Fitting Tele-Operation and Virtual Reality Technologies towards Teleworking. Dans *FIR'98, 4th French-Israeli Symposium on Robotics*, pages 147–152, Besaçon, France.

. 55

➤ Cet article montre comment les techniques de réalité virtuelle permettent de rendre la télé-opération plus abordable pour des opérateurs non spécialistes, en leur donnant l'illusion de travailler manuellement dans l'environnement distant.

Kheddar, A., Tzafestas, C., et Coiffet, P., (1997).
The Hidden Robot Concept : High Level Abstraction Tele-Operation. Dans *IROS'97, International Conference on Intelligent Robot and Systems*, pages 1818–1824, Grenoble, France.

. 55

➤ Cet article présente le concept de «robot caché» ou de «représentation fonctionnelle intermédiaire» : grâce à des techniques de réalité virtuelle, un opérateur est en interaction avec une version virtuelle d'un environnement de télé-opération dans laquelle le robot a disparu mais où l'opérateur a l'impression d'agir naturellement sur l'environnement distant.

Kim, W. S., (1996).
Virtual Reality Calibration and Preview / Predictive Displays for Telerobotics. Dans *Presence*, volume 5/2, pages 173–190. MIT Press.

. 83

➤ L'auteur présente l'utilisation de systèmes de réalité virtuelle pour assister la télé-opération. En proposant à l'opérateur de téléopérer une version locale et virtuelle du robot, sans délai de communication, l'opérateur peut ainsi préparer son plan en toute sécurité puis le transmettre au robot réel pour exécution.

Kissileva, O., (1998).
Art and Virtual Worlds. Dans *Virtual Worlds 98*, Lecture Notes in Artificial Intelligence, pages 357–359. Spinger Verlag.

. 42

➤ Cet article décrit l'influence des nouvelles technologies, en particulier de la réalité virtuelle sur les disciplines artistiques. L'influence de la réalité virtuelle sur l'art ne se réduit pas à la manière de «fabriquer» des œuvres nouvelles mais offre également de nouvelles percspectives quand aux concepts artistiques sous-jacents.

Kitamura, Y., Yee, A., et Kishino, F., (1998).
A Sophisticated Manipulation Aid in a Virtual Environment Using Dynamic Constraints among Object Faces. Dans *Presence*, volume 7/5, pages 460–477. MIT Press.

. . . . . 76, 80, 80

➤ Cet article présente une technique d'aide au placement relatif d'objets dans un environnement virtuel. La méthode proposée consiste, en fonction des éventuelles collisions entre objets, à contraindre le mouvement de ceux-ci sur plusieurs faces.

Krueger, M., (1991).
Artificial Reality II. Addison-Wesley.

. . . . . 22, 24, 38, 39, 42

Loomis, J. M., Golledge, R. G., et Klatzky, R. L., (1998).
Navigation System for the Blind : Auditory Display Modes and Guidance. Dans *Presence*, volume 7/2, pages 193–203. MIT Press.
. 77

➤ Cet article présente des méthodes d'assistance à la navigation pour des personnes aveugles. L'assistance est obtenue en restituant sous une modalité sonore, une information qui normalement provient de la modalité visuelle (présence et distance aux obstacles, orientation. . .).

Lumia, R. et Albus, J., (1988).
Télé-Opération and Autonomy for Space Robotics. Dans *Robotics*, volume 4, pages 27–33.
. 105

Lévy, P., (1995).
Qu'est-ce que le Virtuel ? Éditions La Découverte.
. . . . . 22, 22, 22, 23, 23, 24, 25

➤ Une réflexion philosophique autour de la notion de Virtuel. Le champ d'investigation recouvre non seulement les traditionnels aspects de réalité simulée, mais également des domaines aussi vastes que la virtualisation du texte, de l'économie, ainsi que les aspects de virtualisation des communautés humaines.

Mazeau, J. et Bryche, X., (1995).
Un Espace Interactif Dispersé Destiné à Supporter des Rencontres Sportives. Dans *L'Interface des Mondes Réels et Virtuels*, pages 189–198, Montpellier.
. 56

Mellor, J., (1995).
Enhanced Reality Visualisation in a Surgical Environment. Rapport technique 1544, M.I.T.
. . . . . 52, 75

➤ Ce rapport technique décrit un système d'assistance à la chirurgie (en particulier la chirurgie du cerveau) dans lequel le chirurgien dispose d'une vision de la tête du patient sur laquelle est surimposé un schéma synthétique indiquant la zone de tissu endommagé.

Miller, G., Hoffert, E., Chen, S., Patterson, E., Blackketter, D., Rubin, S., Applin, S., Yim, D., et Hanan, J., (1992).
The Virtual Museum : Interactive 3D Navigation of a Multimedia Database. Dans John Wiley and Sons, editor, *Journal of Visualization and Computer Animation*, volume 3, pages 183–197.
. 43

➤ Les premiers pas dans l'élaboration d'un musée virtuel avec navigation interactive en temps réel sont présentés. L'utilisateur déclenchait des séquences vidéo préalablement filmées pour la navigation où la visualisation des salles.

Nagao, K. et Rekimoto, J., (1995).
The World Through the Computer : Computer Augmented Interaction with Real World Objects. Dans *the 1995 User Interface Software Technology*, pages 29–36, Pittsburgh.
. 73

➤ Un système d'écran permettant de superposer de l'information textuelle au champ de vision normal de l'utilisateur est présenté, ainsi qu'un exemple d'application à l'acquisition d'information dans une bibliothèque (information sur le contenu des rayons et sur les livres qui y sont disposés).

Norman, D. et Rumelhart, D., (1975).
Explorations in Cognition. Freeman and Co, San Francisco.
. 125

Norman, L., (1980).
Traitement de l'Information et Comportement Humain. Vigot.
. 28

Ogden, C. et Richards, I., (1923).
The Meaning of meaning. Routledge and Keagan.
. 49

➤ Ce livre présente un panorama complet des problèmes abordés par la psychologie expérimentale. Ces aspects vont de la perception et du traitement physiologique de l'information aux modèles cognitifs de représentation des connaissance et de raisonnement.

OTAN, (1991).
Human Engineering, Regulatory Documents in Use in RSG.14 Member Nations. NATO Unclassified, Annex III to AC/243(Panel 8)TR/7, Volume 2, Issue 2.
. 146

➤ Ce rapport de l'OTAN rassemble tous les documents traitant des facteurs humains dans les applications militaires développées au sein des pays membres.

Papin, J.-P., (1999).
Exemples d'Utilisation des Techniques de Réalité Virtuelle dans les Systèmes de Combat. Dans *L'Armement*, volume 67, pages 40–43.
. 17

➤ Cet article dresse un panorama des grands types d'application de la réalité virtuelle dans le domaine militaire : l'aide à la conception, l'aide à l'évaluation, l'aide à la formation et à l'entraînement, et l'aide à préparation et à la réalisation de mission.

Pimentel, K. et Teixeira, K., (1994).
La Réalité Virtuelle . . . de l'autre côté du miroir. Addison-Wesley.
. . . . . 15, 21, 23, 45, 55

➤ Un livre guide recouvrant de nombreux aspects de la Réalité Virtuelle et de son évolution. Il se décompose en trois parties : la première relate l'histoire du développement de la Réalité Virtuelle dans le monde. La deuxième se focalise sur les outils de réalité virtuelle (matériels d'immersion, périphériques. . .). La troisième enfin propose un panorama des applications existantes.

Poitrenaud, (1998).
La Représentation des Procédures chez l'Opérateur : Description et Mise en Œuvre des Savoir-Faire. Thèse de doctorat, Université Paris 8.
. 146

Powell, M., Kleiman, S., Barton, S., Shah, D., Stein, D., et Weeks, M., (1992).
SunOS 5.0 Multithread Architecture. SunSoft.
. . . . . 200, 204

➤ Ce document présente en détail l'implémentation des threads fournie par Sun avec son système d'exploitation Solaris 2 (SunOS 5). L'architecture à deux niveaux (niveau utilisateur et niveau noyan) est présentée, ainsi que les fonctionalités de synchronisation et de gestion des signaux.

Quéau, P., (1993).
Le Virtuel. Champ Vallon, Seyssel.
. . . . . 21, 21, 23, 23, 24, 27, 57

➤ Ce livre présente un panorama de l'état de l'art en matière de réalité virtuelle, en insistant en particulier sur les aspects éthiques et philosophiques du problème, par rapport à notre manière d'interpréter et de nous représenter le monde.

Rekimoto, J., (1995).
The Magnifying Glass Aproach to Augmented Reality Systems. Dans *ICAT'95*.
. 73

➤ L'auteur décrit un système de superposition d'information textuelle au champ de vision de l'utilisateur, grâce à un écran de type PalmTop, à travers lequel l'utilisateur perçoit son environnement.

Rekimoto, J., (1997).
NaviCam : a Magnifying Glass Approach to Augmented Reality. Dans *Presence*, volume 6/4, pages 399–412. MIT Press.
.....51, 73

➤ Cet article présente différentes méthodes pour superposer de l'information synthétique sur le champ visuel d'un opérateur, tout en lui laissant la faculté de percevoir son environnement direct. NaviCam, un système basé sur un écran PalmTop est notamment présenté, ainsi que plusieurs exemples d'applications en réalité augmentée.

Richard, Jean-Fran c., (1999).
Comportements, Buts et Représentations. *Psychologie Française*, 44(1) :75–90.
..... 146

Richard, J.-F., (1990).
Les activités mentales. Armand-Collin.
..... 126, 126, 132, 148, 148

➤ Cet ouvrage traite des processus mentaux d'apprentissage, de raisonnement et de planification chez l'humain. Ces problèmes sont abordés sous l'angle de la psychologie cognitive et se fondent une une étude de la représentation des connaissances.

Richard, P., Burdea, G., et Coiffet, P., (1995).
Substitution Sensorielle et Redondance Informative lors d'Interactions Haptiques avec des Objets Virtuels. Dans *L'Interface des Mondes Réels et Virtuels*, pages 317–325, Montpellier, France.
..... 77

➤ Cet article présente une étude comparative des techniques (éventuellement redondantes) tactiles, auditives ou visuelles pour le retour d'effort à des fins d'interaction avec des environnements virtuels ou en télé-opération.

Sacerdoti, E., (1977).
A Structure for Plans and Behavior. Elsevier Computer Science Library.
..... 125

Scapin, D. L., (1986).
Vers des Outils Formels de Description des Tâches Orientés Conception d'Interfaces. Rapport technique 893, INRIA, Domaine de Voluceau Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France.
..... 135, 135

➤ Ce rapport de recherche présente une analyse de certains modèles de description de tâches puis un nouveau formalisme destiné orienté vers la conception d'interfaces utilisateur.

Schenker, P., (1991).
Intelligent Robotics for Space Applications. Dans *Intelligent Robotics Systems*, pages 545–589. S.G. Tzafestas.
..... 105

Schloerb, D., (1995).
A Quantitative Measure of Telepresence. Dans *presence*, volume 4/1, pages 64–80. MIT Press.
.....58, 58

➤ Dans cet article, l'auteur propose une vision de la téléprésence découplée en «présence subjective», correspondant à la probabilité que l'opérateur se sente présent dans l'environnement de travail, et «présence objective», correspondant à la probabilité que la tâche donnée soit exécutée avec succès. Des méthode d'évaluation quantitative de ces deux paramètres sont également fournies.

Schroer, B., (1998).
Telerobotics Issues in Space Applications. Dans *Robotics*, volume 4, pages 233–244.
. 105

Sébillotte, S., (1983).
Représentation des actions chez l'opérateur : étude de tâches administratives. Rapport technique 256, INRIA, Domaine de Voluceau Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France.
. 131

➤ Ce rapport de recherche présente une étude expérimentale montrant qu'il est possible de représenter les tâches administratives par un ensemble fini de sous-buts dont l'utilisation varie en fonction de la tâche, mais qui restent globalement les mêmes dans toutes les circonstances.

Sébillotte, S., (1988).
Hierarchical Planning as Method for Task Analysis : the Example of Office Task Analysis. *Behaviour and Information Technology*.
. . . . . 140, 140

Simsarian, K. T. et Akesson, K.-P., (1997).
Windows on the World : an Example of Augmented Virtuality. Dans *LLIA*, numéro 123, pages 68–71.
. 50

➤ Les auteurs présentent le concept de «virtualité augmentée» comme l'inverse de celui de «réalité augmentée» : il s'agit d'incruster des textures ou des objets réels sur des scènes virtuelles. Il est alors possible de conserver les aspects souhaités du réel tout en bénéficiant de la souplesse des environnements virtuels.

State, A., Livingstone, M. A., Hirota, G., Garrett, W. F., Whiton, M. C., Fuchs, H., et Pisano, E. D., (1996).
Techniques for Augmented Reality Systems : Realizing Ultrasound-Guided Needle Biopsies. Dans *SIGGRAPH'96*, pages 439–446, New Orleans.
. 75

➤ Les auteurs présentent un système de réalité augmentée permettant d'incruster l'image virtuelle d'un outil chirurgical et d'une tumeur (à partir de données ultrasoniques) sur l'image réelle d'un patient en opération. Une illusion de transparence est ainsi donnée au chirurgien, lui permettant se repérer en trois dimensions dans le corps du patient.

Sturman, D., (1992).
Whole Hand Input. Thèse de doctorat, MIT.
. 93

➤ Dans cette thèse, l'auteur décrit ses recherches en matière d'utilisation de tous les degrés de liberté de la main comme périphérique de spécification d'actions. Plusieurs exemples d'application sont proposés, comme le pilotage d'une créature virtuelle.

Sun, (1996).
Multithreaded Implementations and Comparisons : a White Paper. Sun Microsystems.
. . . . . 200, 204

➤ Cet article propose une étude comparative de différentes implémentations des threads noyau et utilisateurs, notamment pour les architectures Sun, HP, IBM, OSF, DCE et Silicon Graphics. Les interfaces Solaris et Posix sont notamment détaillées.

Tachi, S. et Yasuda, K., (1994).
Evaluation Experiments of a Teleexistence Manipulation System. Dans *Presence*, volume 3/1, pages 35–44. MIT Press.
. . . . . 54, 58

➤ Cet article présente différentes configurations pour un système immersif de télé-opération destiné à commander un bras manipulateur, et montre que l'efficacité maximale est obtenue pour une configuration avec retour visuel stéréoscopique, et mouvements naturels du bras et de la tête de l'opérateur.

Thalmann, D., (1998).
Interagir avec des Êtres Virtuels. Dans *GTRV'98*, pages 7–15.
. 45

➤ Présentation générale des problèmes de représentation de l'humain dans les environnements virtuels. Plusieurs classes de représentation sont proposées, selon qu'il s'agit d'un avatar, d'un agent guidé ou d'un agent autonome. Des exemples d'appication sont également fournis dont celui d'un jeu de tennis virtuel mélangeant avatars et agents autonomes.

Van Lhen, K. et Brown, J., (1980).
Planning Nets : a Representation for Formalizing Analogies and Semantic Models of Procedural Skills. *Aptitude Learning and Instruction*.
. 125

Vanderheiden, G. et Mendenhall, J., (1994).
Use of a Two-Class Model to Analyse Applications and Barriers to the Use of Virtual Reality by People with Disabilities. Dans *Presence*, volume 3/3, pages 193–200. MIT Press.
. . . . . 17, 22, 24, 39, 39, 42, 93

➤ Cet article propose d'étudier l'utilité potentielle de la Réalité Virtuelle pour l'assistance aux personnes handicapées, en catégorisant les applications selon deux axes : selon que l'information peut être métaphoriquement présentée de plusieurs manières, ou que le mode de présentation est intrinsèquement contenu dans l'information elle-même. Cet article introduit également la notion de «Réalité Virtuelle Altérée».

Verna, D., (1997a).
Comment définir le Virtuel ? Le modèle M_RIC. Rapport technique 97 D 008, E.N.S.T.
. 9

➤ Ce rapport interne constitue le premier document écrit relatif au modèle M_RIC élaboré par l'auteur dans sa thèse, et sur lequel se base sa réflexion théorique sur la possibilité de définir le virtuel d'un point de vue cognitif.

Verna, D., (1997b).
Télé-Opération et Réalité Virtuelle : assistance à l'opérateur par modélisation cognitive de ses intentions. Dans *IHM'97, neuvièmes journées sur l'ingénieurie de l'interaction homme-machine*, pages 205–206, Poitiers, France. AFIHM, Cépaduès-Éditions.
. 9

➤ Cet article présente un résumé anticipatif du travail effectué dans la thèse de l'auteur : définition de la notion de réalité virtuelle, étude du concept d'assistance à l'opérateur, modèle de représentation des connaissances et application à la détection d'intentions.

Verna, D., (1999).
Définir le Virtuel : une Vision Cognitive. Dans *ReViCo'99*, Paris, France. ENST.
. 9

➤ Cet article présente une vision cognitive de la réalité virtuelle. Il montre comment l'approche cognitive permet d'unifier les différentes visions du concept, classifier les grands types de situations (réalité virtuelle immersive, réalité virtuelle de type CAO. . .) et traiter le cas des environnements réels tels que ceux que l'on trouve en télé-opération.

Verna, D. et Grumbach, A., (1998a).
Can we define Virtual Reality ? the M_RIC model. Dans Heudin, J.-C., editor, *VW'98, first international conference on virtual worlds*, Lecture Notes in Artificial Intelligence, pages 29–41. Springer-Verlag.
. 9

➤ Une réflexion théorique est présentée sur la possiblité d'une définition de la réalité virtuelle d'un point de vue cognitif et non pas technologique. Cette réflexion s'appuie sur un modèle descriptif représentant l'humain en interaction avec son environnement.

Verna, D. et Grumbach, A., (1998b).
Sémantique et Localisation de l'Assistance en Réalité Virtuelle. Dans *GTRV'98, sixièmes journés nationales réalité virtuelle*, pages 105–112, Issy-Les-Moulineaux, France. CNET.
. 9

➤ Dans cet article, l'auteur présente une analyse du concept d'assistance à l'opérateur dans un contexte de réalité virtuelle au travers de quelques exemples représentatifs. Cette analyse basée sur un modèle de situation le conduit en particulier, par un procédé systématique de symétrisation des exemples, à extrapoler le raisonnement vers de nouvelles perspectives dans l'application du concept d'assistance.

Verna, D. et Grumbach, A., (1999).
Augmented Reality, the Other Way Around. Dans Gervautz, M., Hildebrand, A., et Schmalstieg, D., editors, *EGVE'99, fifth Eurographics Workshop on Virtual Environments*, pages 147–156, Vienna, Austria. ICG, Springer-Verlag.
. 9

➤ Dans cet article, l'auteur montre par une analyse systématique que le concept de réalité augmentée a été perçu d'une manière quelque peu déséquilibrée : principalement à destination de l'utilisateur et au niveau de ses perceptions. L'auteur montre par la suite comment cette vision du concept peut être rééquilibrée afin de se diriger vers l'environnement et d'inclure également les actions de l'utilisateur.

Winacott, R., (1993).
What Is a Thread of Control ? *Sun Post*, pages 40–69.
. . . . . 200, 204

➤ Cet article propose un tour d'horizon du concept de «thread», de l'interface `C` proposée par Sun pour le système Solaris (SunOS 5), les outils de synchronisation et de débugage, le tout illustré par des petits exemples de code.

Télé-opération et réalité virtuelle : assistance à l'opérateur par modélisation cognitive de ses intentions

Compte tenu de l'état de l'art en matière de technologie, les systèmes actuels de réalité virtuelle doivent établir un compromis entre efficacité et utilisabilité: un écran d'ordinateur est d'emploi facile mais ne permet pas une bonne restitution du relief. Réciproquement, un casque de visualisation permet de fournir une vision stéréoscopique, mais constitue un périphérique encombrant. Pour résoudre ce type de dilemme, deux approches sont envisageables: une première approche, plus appliquée, consiste à améliorer la qualité de l'interaction Homme-Machine en perfectionnant les interfaces. Une deuxième approche, plus théorique, consiste à assister l'humain dans sa tâche en diminuant sa charge de travail et donc la quantité d'interaction nécessaire.

Notre travail porte sur la notion d'assistance et se situe par conséquent dans la deuxième approche: nous analysons sous un angle cognitif la notion d'assistance à l'humain dans un contexte de réalité virtuelle, et nous montrons en particulier l'intérêt du concept de détection d'intentions comme source d'assistance intelligente. Le travail est organisé en trois grandes parties: l'élaboration d'un point de vue cognitif sur la notion de réalité virtuelle, une étude théorique sur la notion d'assistance dans les systèmes de réalité virtuelle, et une approche fondée sur la détection d'intentions.

La première partie est consacrée à l'élaboration théorique d'une caractérisation cognitive du virtuel. Plutôt que de définir le virtuel en tant que tel, nous montrons qu'il est plus intéressant de définir les processus de virtualisation, c'est à dire les mécanismes cognitifs qui font qu'à partir d'une situation réelle, l'humain entre progressivement dans le virtuel. Par exemple, à partir de la situation réelle où un humain est devant un écran d'ordinateur, face à un monde de synthèse, on parlera de "virtualisation" au moment où celui-ci oublie qu'il s'agit d'un écran et pense qu'en se déplaçant, son point de vue sera modifié.

Dans un premier temps, un modèle cognitif décrivant l'interaction entre un humain et son environnement est proposé, le modèle MrIC. Les processus de virtualisation sont ensuite définis et nous permettent d'obtenir un certain nombre de variantes de ce modèle, décrivant les grandes catégories de situations virtualisées: situations de type CAO (avec environnement virtuel), situations dites "immersives" (avec représentation virtuelle de l'humain).

La deuxième partie est consacrée à une étude théorique de la notion d'assistance. Assister l'humain, c'est soulager son travail de toutes les manières possibles: lui fournir plus d'information, ou l'information pertinente, ou encore effectuer des opérations à sa place... Notre objectif est donc d'étudier de quelle manière l'assistance à l'humain peut être mise en oeuvre dans les systèmes de réalité virtuelle. Nous commençons par élaborer un modèle décrivant les situations d'interaction entre un humain et de tels systèmes, le modèle LiSA. Ce modèle fait intervenir un certain nombre de composantes (opérateur, interface, système de réalité virtuelle...) qui interagissent en échangeant de l'information. L'apparition d'un processus d'assistance à l'humain dans une situation donnée a pour effet de modifier l'information véhiculée entre les différentes composantes du modèle.

Cette particularité nous permet d'analyser des exemples d'assistance caractéristiques des applications actuelles selon deux axes: quelle(s) composante(s) du modèle contiennent ces processus d'assistance, et comment ceux-ci modifient-ils l'information (enrichissement, filtrage, altération...).

La troisième partie décrit une approche fondée sur la détection d'intentions. L'idée générale est de concevoir un système qui, en fonction des actions de l'opérateur et de l'état de l'environnement, tente de déterminer l'opération que celui-ci est en train d'exécuter, et le cas échéant, prend le contrôle de l'exécution. Dans cette optique, nous commençons par présenter un modèle arborescent de représentation des connaissances servant à décrire les actions susceptibles d'êtres entreprises par l'opérateur. Puis, nous établissons les critères permettant au système d'évaluer la probabilité pour qu'une action dont il a connaissance soit effectivement celle entreprise par l'opérateur, puis nous proposons un jeu de processus permettant d'évaluer ces critères à tout instant.

Le mécanisme de détection d'intentions proposé, ainsi que son extension à des fonctionalités d'exécution et de simulation des opérations qu'il est initialement capable de détecter font l'objet d'une illustration à travers un démonstrateur informatique, le système ToAST.