HAL
open science

ADELE : une architecture à base de tableau noir pour la simulation ergonomique

Aline Alauzet

To cite this version:

Aline Alauzet. ADELE : une architecture à base de tableau noir pour la simulation ergonomique. Informatique [cs]. Université Henri Poincaré - Nancy 1, 1998. Français. NNT : 1998NAN10026 . tel-01754329

HAL Id: tel-01754329

https://hal.univ-lorraine.fr/tel-01754329v1

Submitted on 30 Mar 2018

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

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

UNIVERSITÉ DE LORRAINE

AVERTISSEMENT

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

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

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

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

LIENS

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

Université de Nancy I | INRETS-LESCO | INRIA-Lorraine | LORIA

U.F.R. S.T.M.I.A

Ecole Doctorale IAE+M
D.F.D. Informatique

ADELE : Une architecture à base de tableau noir pour la simulation ergonomique

THÈSE

présentée et soutenue publiquement le **5 mai 1998**

pour l'obtention du

Doctorat de l'Université Henri Poincaré Nancy I

(Spécialité Informatique)

par

Aline ALAUZET

S.C.D. - U.H.P. NANCY 1 SCIENCES BIBLIOTHÈQUE

Composition du jury :

Président :	**Mme Noëlle CARBONELL**
Rapporteurs :	**Mme Suzanne PINSON**
	M. Jacques KOULOUMDJIAN
Directeurs de thèse :	**M. Jean-Paul HATON**
	M. Samuel SELLAM
Examinateur :	**M. Jean-Pierre VERRIEST**
Invité :	**M. Serge RIBOUCHON**

Remerciements

Je tiens à remercier :

André Chapon, directeur du LESCO (Laboratoire Ergonomie Santé COnfort de l'INRETS), qui m'a encouragée et autorisé à entreprendre ce travail de thèse,

Les personnes qui ont accepté d'être les rapporteurs de ce travail : Mme Suzanne Pinson, M. Jacques Kouloumdjian, Mme Noëlle Carbonell (que je remercie aussi pour avoir accepté de présider le jury),

Jean-Paul Haton, qui bien voulu accepter la direction de la thèse et qui en a assumé la charge de manière exemplaire ; je le remercie d'avoir été si présent et si encourageant tout au long de ce travail.

Samuel Sellam qui a co-encadré la thèse de manière tout aussi exemplaire ; je le remercie de tout cœur pour sa présence et son investissement dans ce travail.

Hélène Tattegrain-Veste pour son aide précieuse et son soutien permanent,

Jean-Pierre Verriest, pour son aide et notre collaboration fructueuse sur le sujet,

Serge Ribouchon, pour sa participation dans le cadre de nos réunions informelles,

Bruno Astier, pour le développement de la passerelle ADELE-MAN3D,

Christelle Carrez, pour les modifications de Atome pour la liaison avec GemStone,

mes collègues de l'INRETS et plus particulièrement Blandine, Mireille, Claude G., Valérie, Valérie, Bernard, Claude M.-L., Didier, François, Delphine, Catherine, Jojo, Raquel, Nicolas, Wang, pour leur soutien,

et surtout Jean-Luc, Nina, Léa, Thierry, qui ont eu à supporter mon investissement dans ce travail de thèse.

Table des matières

Introduction générale 7

PARTIE I : Problématique - État de l'art 8

Introduction 9

1 Simuler le comportement humain pour l'analyse ergonomique 10

1.1 La simulation ergonomique 10

1.1.1 La démarche de simulation en ergonomie 10

1.1.2 Les modèles pour la simulation en ergonomie 12

1.2 L'analyse ergonomique dans le processus de conception 14

1.2.1 La place de l'intervention ergonomique 14

1.2.2 La nature et les outils de l'intervention ergonomique 17

1.3 Les modèles géométriques de l'homme utilisés en CAO 19

1.3.1 MAN3D : un modèle de l'homme 19

1.3.1.1 La représentation du corps humain 19

1.3.1.2 Les capacités fonctionnelles 23

1.3.1.3 La modification de la posture 24

1.3.1.4 Comment MAN3D est utilisé 25

1.3.1.5 Aspects mathématique et informatique 26

1.3.1.6 Évolution de MAN3D 26

1.3.2 Les concepts clefs d'un outil d'aide à la conception centré sur un modèle d'opérateur humain 27

1.3.2.1 Le réalisme 27

1.3.2.2 La prédiction de la posture 29

1.3.2.3 La manipulation du mannequin 31

1.3.2.4 Niveau d'automatisation et détermination de la posture 31

1.3.2.5 Le diagnostic ergonomique 33

1.3.3 Les modèles graphiques d'opérateur humain 34

1.4 Conclusion 35

1.4.1 La théorie, l'offre et la demande, pour la simulation ergonomique 35

1.4.2 Les limites des modèles d'opérateur humain pour l'analyse ergonomique 36

2 Paradigmes IA pour la simulation ergonomique **38**

2.1 Approches à bases de connaissances pour l'aide ergonomique à la conception **38**

2.1.1 Revue des systèmes existants 38

2.1.2 La connaissance experte dans les systèmes d'aide à la conception ergonomique 39

2.2 Les systèmes multi-agents **41**

2.2.1 Quelques précisions terminologiques 41

2.2.1.1 IA et systèmes à bases de connaissances 41

2.2.1.2 Intelligence artificielle distribuée et systèmes multi-agents 42

2.2.1.3 Systèmes multi-agents et modèle de tableau noir 43

2.2.2 Fondements théoriques 44

2.2.2.1 Systèmes multi-agents basés sur le paradigme multi-experts 44

2.2.2.2 Modèle de tableau noir 45

2.2.3 Le système ATOME 47

2.2.3.1 Principe général 47

2.2.3.2 Format des sources de connaissance 49

2.2.3.3 Principes de fonctionnement 51

2.2.3.4 L'utilisation de ATOME et ses limites 53

3 Conclusion : problématique et objectifs **55**

3.1 Les objectifs par rapport au domaine **55**

3.1.1 Délimitation du champ 55

3.1.2 Les objectifs 56

3.2 La problématique IA **57**

3.2.1 Formaliser le qualitatif 57

3.2.2 Choix d'une approche 58

PARTIE II : Le système ADELE **59**

Introduction **60**

4 Modélisation des données et des connaissances **61**

4.1 Analyse globale du problème **61**

4.1.1 Quelles limites pour la simulation ? 61

4.1.2 Les connaissances à modéliser 62

4.2 La modélisation de l'activité **63**

4.2.1 Modélisation de l'environnement 63

4.2.2 Modélisation de l'opérateur 65
4.2.2.1 Modèle symbolique 65
4.2.2.2 Modèle graphique 66
4.2.2.3 Liens entre l'opérateur générique et l'opérateur graphique 67
4.2.2.4 Opérateur actif 67
4.2.3 Modélisation des interactions de l'opérateur avec l'environnement 68
4.2.4 Description symbolique des postures 72

4.3 Les stratégies de mouvement **76**
4.3.1 Le cas de la position assise 76
4.3.2 Le cas de la station debout 80

4.4 Le diagnostic ergonomique **81**
4.4.1 Le respect du cahier des charges ergonomique 81
4.4.2 Le confort de la posture 82
4.4.3 Le bilan ergonomique 85

4.5 Conclusion **86**

5 Aspect conceptuels du système ADELE **87**

5.1 Conception d'une architecture **87**
5.1.1 Le raisonnement dans ADELE 87
5.1.2 Nature des informations manipulées 88
5.1.3 Structuration du système 90

5.2 Le système multi-agents **92**
5.2.1 Le rôle de la partie multi-agents dans ADELE 92
5.2.2 Les tableaux noirs 93
5.2.3 Les sources de connaissances spécialistes 96
5.2.4 Le contrôle de la résolution 98

5.3 L'intégration de données du tableau noir dans une BD objet **99**

PARTIE III : Le prototype - son implémentation et son évaluation **102**

Introduction **103**

6 Architecture logicielle du système **104**

6.1 Principes du système **104**
6.1.1 Architecture générale 104

6.1.2 Principes d'utilisation 105
6.1.2.1 Cohérence entre Euclid/MAN3D et GemStone/BaseAdele 105
6.1.2.2 Déroulement d'une session de simulation 106
6.1.3 Limites du prototype 107

6.2 La base de données objets **107**
6.2.1 Quelques mots sur GemStone et son environnement 107
6.2.2 Interface interactive avec la base de données : InterfaceAdele 108
6.2.3 La base de données : BaseAdele 109

6.3 Communication entre les différents modules du système **112**
6.3.1 Communication entre SMA-Adele et MAN3D 112
6.3.2 Communication entre Interface-Adele et SMA-Adele 114

6.4 Le partage de données entre Atome et GemStone **115**
6.4.1 Le problème à résoudre 115
6.4.2 Interface C++ avec GemStone 116
6.4.3 Une nouvelle version de Atome 117
6.4.4 L'accès aux données de BaseAdele par SMA-Adele 118

7 Manipulation et évaluation du système **120**

7.1 Utilisation de ADELE **120**

7.2 Objectifs de l'évaluation **124**

7.3 Mise en œuvre de la phase de test **125**

7.4 Bilan de l'évaluation **126**

Conclusion générale **128**

Bibliographie **129**

ANNEXES **134**

Annexe 1 : Modèles graphiques d'opérateur humain **135**

Annexe 2 : Notions sur la méthode OMT **137**

Annexe 3 : Commandes MAN3D **139**

Annexe 4 : Extrait de la base de données BaseAdele, développée sous GemStone **144**

Introduction générale

Le contexte dans lequel s'insère ce travail est celui de la conception d'environnements tels que les véhicules ou les espaces de travail, dans lesquels l'homme exerce une activité. La question de fond à laquelle nous nous intéressons est celle du test de l'adéquation de ces environnements, au cours de leur conception, à l'activité future de leurs utilisateurs.

La prise en compte tardive du facteur humain dans les processus de conception de produits peut conduire à des surcoûts, liés aux corrections à apporter au produit dans les phases finales du processus, pour assurer cette adéquation. D'où le besoin de faire remonter le plus tôt possible dans le processus de conception la prise en compte du facteur humain, c'est-à-dire de répondre le plus tôt possible à la question de la pertinence du produit vis-à-vis de son futur utilisateur.

Pour tester le futur produit dans cette perspective, on tente de simuler un opérateur humain en activité, sur une ébauche de la solution en cours de conception. A l'ère des maquettes sur plan papier, les concepteurs de bureaux d'études utilisaient pour cette simulation des mannequins articulés plats, qu'ils positionnaient sur le dessin. Avec le passage à la conception assistée par ordinateur (CAO), des mannequins informatisés ont été conçus, d'abord en 2D, à l'image des mannequins plans utilisés, puis en 3D.

Notre travail a pour point de départ l'un de ces mannequins informatisés, MAN3D[1], et s'insère dans une problématique de simulation de l'activité, de prédiction du comportement moteur et de diagnostic ergonomique, pour la conception d'environnements utilisés par l'homme. La thèse que nous défendons est que l'usage des techniques d'intelligence artificielle dans ce contexte peut contribuer à résoudre des problèmes actuellement non résolus par une modélisation basée sur l'usage seul de techniques algorithmiques classiques. Dans cette optique, nous proposons une analyse de la question de la simulation du comportement moteur pour le diagnostic ergonomique, aboutissant à la conception d'un système d'assistance pour l'évaluation ergonomique sur maquette CAO. Ce système est basé sur une modélisation objet et une architecture de tableau noir. Nous avons développé un prototype avec pour objectif de démontrer la faisabilité de cette approche.

La démarche adoptée, dont nous rendons compte ici, a comporté trois phases. Dans un premier temps, nous avons tenté de circonscrire les objectifs, de définir le domaine, d'explorer les liens existant avec les domaines connexes du sujet étudié. Cette phase, accompagnée d'une étape d'analyse de l'état de l'art, a conduit à l'élaboration de la problématique, que nous détaillons dans la partie I. La seconde phase consacrée à la modélisation du problème et à la conception du système est exposée dans la partie II. La partie III est consacrée à la description du prototype de système qui a été réalisé puis testé dans une étape de validation expérimentale.

[1] MAN3D a été développé à l'INRETS en collaboration avec Renault

PARTIE I : Problématique - État de l'art

Introduction

Nous expliquons dans cette partie de quelle manière nous avons positionné notre sujet d'étude, d'une part du point de vue de la question de la simulation du comportement humain (chapitre 1), d'autre part du point de vue de la prise en charge de cette question par une approche informatique (chapitres 2 et 3).

Nous expliquons dans le chapitre 1 quels sont les problèmes posés et les solutions existantes, pour la simulation du comportement humain destinée à l'analyse ergonomique. Nous avons choisi de commencer par décrire le contexte de la simulation ergonomique, qui constitue un cadre « théorique » pour situer ce sujet (section 1.1). Nous décrivons ensuite ce que l'on peut considérer comme la « demande », c'est à dire le contexte réel d'application d'une démarche d'analyse ergonomique dans un processus de conception, ainsi que les besoins des concepteurs de produits (section 1.2). La section 1.3 est consacrée à « l'offre », c'est-à-dire aux outils existants (les modèles géométriques de l'homme utilisés en CAO), dont fait partie le modèle MAN3D. Nous évoquons aussi dans cette section les concepts importants liés à la modélisation d'un opérateur humain dans ce contexte. Enfin, nous concluons ce chapitre (section 1.3) en comparant les trois points précédents (la théorie, l'offre et la demande) et en soulignant les limites des outils existants.

Le chapitre 2 est consacré à la revue des paradigmes de l'Intelligence Artificielle (IA) utiles pour la simulation ergonomique. Nous évoquons tout d'abord les approches à bases de connaissances développées dans des contextes d'aide à la conception (section 2.1), en faisant tout d'abord une revue des systèmes existants, puis en analysant de plus près la nature de la connaissance experte modélisée. La section 2.2 est consacrée aux systèmes multi-agents. Nous commençons par fournir quelques précisions à propos de la place du thème des systèmes multi-agents dans l'IA, des liens entre les thèmes de l'Intelligence Artificielle Distribuée (IAD) et des systèmes multi-agents, et de la place des modèles à base de tableau noir dans les systèmes multi-agents. Nous exposons ensuite les fondements théoriques de ces systèmes, en décrivant plus particulièrement le modèle de tableau noir, puis nous décrivons le système ATOME, basé sur ce modèle, que nous avons utilisé dans ce travail.

Nous concluons cette partie dans le chapitre 3, qui est consacré à l'exposé de la problématique et des objectifs. Nous distinguons deux parties : les objectifs définis par rapport au domaine (section 3.1), et ce qui relève plus d'une problématique informatique et plus spécifiquement de l'IA (section 3.2).

1 Simuler le comportement humain pour l'analyse ergonomique

Le cadre dans lequel nous nous situons est celui de l'aide à la conception de produits industriels, par la modélisation, la prédiction et la simulation du comportement humain. Les démarches de modélisation du comportement humain dans ce cadre sont multiples, et la vision du thème peut être différente, selon que l'on s'intéresse aux démarches de modélisation ou aux outils utilisés réellement dans les processus de conception industriels.

Le modèle de simulation qui constitue le point de départ de notre travail (MAN3D) peut clairement être classé dans la catégorie des outils utilisés dans l'industrie, issus d'approches à caractères anthropométrique et biomécanique. L'objectif de ce chapitre est non seulement d'expliciter le contexte d'utilisation de ce modèle et donc les besoins auxquels il répond (ou doit répondre) afin d'introduire nos objectifs d'enrichissement, mais aussi d'élargir le cadre de réflexion. En effet, notre but est de montrer que l'approche de simulation du comportement moteur de l'opérateur pour l'aide à la conception peut être enrichie si l'on repart d'une approche plus globale, intégrant la simulation de l'activité pour la conception d'une situation de travail. C'est pourquoi la première section (1.1) est consacrée au thème de la simulation ergonomique, qui permet d'introduire les différentes catégories de modèles existants dans ce cadre. A l'opposé de cette description du contexte en terme de modélisation, la deuxième section (1.2) analyse la place de l'intervention ergonomique dans un processus de conception industriel de véhicules, dans le but de décrire les besoins des concepteurs, le contexte d'utilisation de tels systèmes d'aide et les contraintes induites. La troisième section (1.3) est consacrée aux modèles de l'homme utilisés en CAO, c'est-à-dire aux modèles tels que MAN3D. Nous commençons par décrire ce dernier (§1.3.1) afin de favoriser la compréhension des concepts liés à la modélisation de l'opérateur humain que nous évoquons dans le paragraphe 1.3.2. Le dernier paragraphe (1.3.3) évoque brièvement les modèles graphiques d'opérateur humain similaires à MAN3D et décrits dans la littérature du domaine.

Dans la section 1.4 nous synthétisons les différents éléments exposés dans ce chapitre, afin d'introduire les éléments de notre problématique développés dans le chapitre 3.

1.1 La simulation ergonomique

1.1.1 La démarche de simulation en ergonomie

La pratique de l'ergonomie est fondée sur l'analyse de l'activité de travail, tel qu'il est réellement effectué. Cette analyse permet d'appréhender simultanément la tâche originellement prescrite, les contraintes de nature humaine, matérielle ou organisationnelle pesant sur sa réalisation, et surtout la manière dont l'activité se déroule effectivement, et donc les phénomènes d'adaptation, de réappropriation ainsi que les autres stratégies mises en œuvre par l'opérateur. C'est cette analyse qui est à la base des connaissances et compétences que

l'ergonome met en œuvre pour transformer les situations de travail afin d'assurer une meilleure prise en compte des caractéristiques du fonctionnement de l'homme au travail.

Avec cette même optique, l'ergonome peut agir sur des situations de travail existantes ou bien sur des projets de conception. Dans le cadre des processus de conception des systèmes de travail, l'ergonome met en œuvre une approche de nature prédictive. Il doit être en mesure d'utiliser des méthodes lui permettant d'émettre un avis pronostic sur une situation de travail projetée ou en phase de conception. C'est dans ce contexte que s'inscrit la démarche de "simulation ergonomique".

D'après Maline (1),le champ d'application de la démarche de simulation en ergonomie porte sur les interactions qui existent, au cours d'une activité de travail, entre un individu et une machine, entre un collectif d'individus et une ligne de production, ou encore entre un collectif d'individus et une organisation du travail. L'objectif de ce type de démarche est d'appréhender de façon anticipée la situation dans laquelle un ou plusieurs salariés vont se trouver, les contraintes qu'ils risquent de subir, les marges de manœuvre dont ils vont bénéficier dans la mise en œuvre des tâches qui leur sont assignées. La simulation des conditions de réalisation du travail futur se met en place à partir de scénarios regroupant des éléments relatifs aux tâches à accomplir, au déroulement temporel, aux caractéristiques techniques et à l'organisation envisagés.

Le modèle informatique est un des supports utilisables pour cette simulation (les autres supports étant le plan, la maquette et le prototype). Il permet de représenter l'espace de travail, de symboliser un ou des opérateurs, de mettre en œuvre la dynamique du système et d'animer le modèle de la situation de travail. L'animation du modèle de la situation de travail consiste à suivre l'évolution de l'environnement de travail, définir les actions d'un opérateur et introduire dans la simulation les résultats de ces actions, agir sur la dynamique du process pour simuler des situations de travail variables, comparer différents scénarios.

Grâce à la réalisation et à l'animation du modèle de la situation de travail futur, l'ergonome sera en mesure d'émettre des pronostics, d'aider aux choix de conception en mettant en lumière les conséquences possibles et/ou probables que certains choix peuvent avoir sur les conditions futures de réalisation des tâches et les contraintes qui risquent d'en découler pour les salariés et pour la production.

On retiendra, à la suite de cette brève description de la démarche de simulation en ergonomie, qu'elle repose sur la définition de plusieurs modèles : un modèle de l'environnement de travail, un modèle de l'opérateur, un modèle de l'activité de l'opérateur (sous la forme de scénarios regroupant des situations d'action caractéristiques), un modèle de la dynamique du système opérateur/environnement (qui se traduit dans une phase de mise en scène des scénarios).

Sur le plan technique, peu d'outils existent pour servir de support à cette démarche. Il n'existe pas de logiciel intégré, comportant de réelles capacités de simulation pour chacun de ces différents modèles. Nous détaillons dans le paragraphe suivant (1.1.2) les catégories de modèles et outils existants.

1.1.2 Les modèles pour la simulation en ergonomie

En pratique il existe :

- des moyens pour modéliser l'environnement en cours de conception,
- différents modèles concernant l'opérateur,
- des modèles de simulation de tâches, génériques au départ mais applicables au contexte de l'étude ergonomique.

Les logiciels de CAO permettent de construire une maquette graphique du produit en cours de conception et donc de travailler sur le modèle de l'environnement, mais de façon déconnectée de l'aspect opérateur.

Les logiciels de simulation utilisés dans des cadres de conception sont beaucoup plus conçus pour envisager les caractéristiques d'une production, son dimensionnement, son phasage et son ordonnancement que pour intégrer des données relatives à l'homme au travail. Les outils de simulation basés sur un modèle de réseau de tâches peuvent cependant être utilisés pour faire de la simulation de nature ergonomique. C'est le cas par exemple du logiciel SAINT (Systems Analysis of Integrated Networks of Tasks), existant aussi dans une version appelée MicroSaint (2). C'est un outil qui peut être appliqué à tout problème d'analyse de performance humaine pour lequel on s'intéresse à une prédiction quantitative des performances humaines et/ou à l'interaction entre l'homme et son environnement. Il permet de décomposer les performances d'un système en séries de sous-activités ou tâches, dont le séquencement est modélisé par un réseau. À ce réseau sont associées des variables, dont l'évolution dépend de l'exécution des tâches. L'activation du modèle permet d'effectuer différentes simulations, suivant les variables d'entrée et les modèles d'évolution associés aux tâches.

Il existe par ailleurs de très nombreux modèles d'opérateur, de différentes natures et origines et appliqués à différents contextes. Nous nous limiterons ici à l'évocation de modèles implémentés et utilisés dans l'industrie pour l'aide à la conception.

Dans un ouvrage consacré aux modèles de performance humaine appliqués à la conception de systèmes et édité à la suite de la réunion d'un groupe de travail international sur le sujet (3), on trouve une classification exhaustive des différents modèles existants. L'idée était de répertorier les différentes catégories d'outils disponibles, en se focalisant sur les techniques qui permettent des simulations informatisées du fonctionnement de l'homme au sein de systèmes, par opposition aux modèles simplement analytiques ou conceptuels. La classification obtenue comporte six domaines de modélisation :

[1] - <u>les modèles d'allocation de tâche et d'analyse de la charge de travail</u>
Cette catégorie regroupe les techniques permettant d'estimer la charge de travail humaine, lorsque différentes tâches sont réparties entre un homme et une machine.

[2] - <u>les modèles de tâches individuelles</u>
Ils concernent les cas où l'on s'intéresse à la performance d'un opérateur unique réalisant une tâche unique. C'est le domaine le plus ancien de la modélisation de la performance humaine et le plus mathématique car on y retrouve principalement des descriptions mathématiques des mécanismes du fonctionnement humain. Dans cette catégorie on trouve par exemple des modèles concernant les tâches de

contrôle/supervision ou bien des modèles de prise de décision pour le contrôle de véhicules.

[3] - les modèles de situations multi-tâches
Ces modèles concernent le cas d'un seul opérateur réalisant des tâches multiples. Cette catégorie ne recouvre pas la représentation des mécanismes du fonctionnement humain, mais la simulation des résultats en terme de précision et de charge de travail. Cette simulation est effectuée à l'aide d'outils de modélisation de réseaux de tâches (tels que SAINT, dont il est question plus haut).

[4] - les modèles du comportement d'équipes
Ils concernent la modélisation de plusieurs opérateurs réalisant des tâches multiples.

[5] - les modèles basés sur des approches anthropométriques et biomécaniques, appliqués à la conception des espaces de travail
Il s'agit de la modélisation du comportement humain, à un niveau d'analyse plus basique que celui des modèles évoqués dans les catégories précédentes. Ces modèles permettent de prédire la capacité d'un opérateur à voir et atteindre les commandes et les afficheurs, réaliser des tâches de manutention sans risque, pouvoir s'installer aisément dans un espace de travail et en sortir de même. Notons que c'est dans cette catégorie que l'on peut classer le modèle MAN3D.

[6] - les modèles de formation et de maintien des compétences
Ils fournissent des techniques pour aider à la conception et à l'utilisation des systèmes d'apprentissage. Ils sont basés principalement sur la modélisation des courbes d'apprentissage et permettent notamment la prédiction de l'acquisition et du maintien de compétences spécifiques.

On trouve dans ce même ouvrage (3) un ensemble d'articles décrivant des modèles appartenant à ces différentes catégories. Dans un ouvrage plus spécifiquement consacré aux outils d'aide à la conception utilisés dans l'industrie automobile (4), Kroemer (5) rappelle un classement des modèles constituant ici la cinquième des catégories listées ci-dessus, qui distingue :

- les modèles anthropométriques, fournissant une géométrie statique du corps, prenant en charge les dimensions corporelles, des positions d'atteinte du corps et/ou de ses parties, des postures,

- les modèles biomécaniques, fournissant des représentations des activités physiques du corps (à partir des données anthropométriques utilisées en entrée),

- les modèles d'interface homme-machine, combinaisons spécifiques de modèles anthropométriques et biomécaniques pour des représentations des interfaces homme-machine. Ils décrivent les interactions entre la personne modélisée et l'équipement, dans un système comportant une interface homme-machine.

Dans ce dernier classement, MAN3D est du type des modèles d'interface homme/machine. Notons que pour ce type de modèle, la modélisation des interactions entre la personne modélisée et l'équipement se limite le plus souvent aux aspects atteinte et vision.

1.2 L'analyse ergonomique dans le processus de conception

De fait, les modèles pour la simulation ergonomique sont encore peu utilisés dans l'industrie. Les raisons sont multiples et nous ne cherchons pas à les approfondir ici. Elles relèvent certainement des performances limitées des outils existants, du manque de modèles de l'Homme et aussi de la méconnaissance de l'ergonomie dans certains secteurs industriels (avec une exception pour les secteurs de l'aviation et de l'automobile, qui ont depuis longtemps tenté d'intégrer le facteur humain dans la conception).

Dans un article consacré précisément à l'analyse des raisons de l'utilisation peu fréquente des modèles de comportement humain (6) Rouse et Cody introduisent leur analyse en catégorisant les différentes sources utilisées par les concepteurs pour acquérir des informations et régler des problèmes de nature ergonomique. Ils décrivent trois catégories de sources d'information. La première et de loin la plus fréquemment utilisée est le jugement humain. Dans cette catégorie on peut classer l'expérience du concepteur lui-même, ainsi que les souvenirs ou opinions de ses collègues et parfois d'experts du domaine et d'utilisateurs des systèmes. La deuxième source est constituée par les archives : elle regroupe les documentations portant sur des projets de conception terminés, les normes, les pratiques, les règlements, ainsi que la littérature scientifique, technique et commerciale. La troisième et dernière catégorie, de loin la moins fréquemment utilisée, est constituée par les modèles. Dans cette catégorie, les auteurs classent à la fois les études empiriques et les modèles analytiques. Les études empiriques correspondent à des situations où le système est simulé plus ou moins fidèlement, mais où l'opérateur est un sujet humain, inséré dans la boucle de fonctionnement du système. Les modèles analytiques correspondent à des simulations informatisées, dans lesquelles les deux composants (l'humain et le système) sont représentés, de manière symbolique simplement ou bien sous la forme d'un modèle informatisé plus complexe.

La situation décrite par Rouse et Cody concernait une population de concepteurs dans le domaine de l'aérospatial, aux Etats-Unis, en 1989. Elle est néanmoins totalement similaire à celle que nous pouvons observer en France, dans le cadre d'un processus de conception de véhicules que nous avons plus particulièrement étudié, avec l'aide de l'ergonome en charge des aspects ergonomiques dans le cadre des projets de conception associés. C'est sur la base des informations recueillies dans ce cadre que nous décrivons plus en détail dans les paragraphes suivants (1.2.1 et 1.2.2) la place et la nature de l'intervention ergonomique dans un processus de conception industriel.

1.2.1 La place de l'intervention ergonomique

Nous nous attachons ici à situer la place et la nature de l'analyse ergonomique, dans le cadre d'un processus précis de conception de produit, celui des habitacles de poids-lourds, et dans une entreprise : Renault Véhicule Industriel (RVI).

C'est pourquoi nous commençons par décrire ce processus de conception. La Figure 1 déroule un processus de conception type pour un produit de RVI. Sur ce schéma, les interventions ergonomiques sont repérées par des flèches horizontales et un codage "Ei", i étant le numéro d'ordre de l'intervention.

On entend par "intervention ergonomique", toute intervention d'un ergonome, qu'il s'agisse d'une intervention se déroulant dans un cadre formel et prévu (par exemple participation à une réunion de projet) ou bien d'une intervention sous la forme de réponses à des questions des concepteurs par exemple, survenant au cours du processus. Notons que ce processus de questions/réponses a récemment fait l'objet d'une formalisation, le résultat étant un ensemble de fiches, chaque fiche décrivant la question posée et la réponse apportée.

Le maquettage CAO est démarré au niveau de la phase d'implantation des composants. On remarque que les interventions ergonomiques ne s'effectuent pas seulement sur la maquette CAO (seules E3 et E4 se font sur la maquette CAO).

La phase E1 est celle où l'on tente de définir le véhicule idéal en fonction des contraintes du projet. L'intervention ergonomique se traduit par :

- la définition des éléments du poste de conduite : siège, pédalier, volant, levier de vitesses, (qui vont devenir par la suite des éléments CAO),
- la spécification de zones géométriques définissant des limites : visibilité à travers le pare-brise, rétrovision, zone genoux (définie dans le but d'éviter les interférences avec la planche de bord), encombrement gauche (pas d'interférence avec la porte et les éléments associés),
- la définition de cotes d'habitabilité : largeur aux pieds, hauteur pavillon (intérieur cabine), hauteur capot moteur,
- la définition de cotes d'emmarchement : hauteur, position longitudinale, répartition des marches.

L'intervention E2 n'a lieu que lorsque des composants nouveaux (par rapport aux projets précédents) sont introduits. L'ergonome participe dans ce cas à la définition du cahier des charges concernant le nouveau composant.

Au moment de l'implantation des composants (intervention E3), chaque concepteur a en charge le positionnement d'une fonction, c'est-à-dire d'un élément de l'habitacle (siège, volant, pédale, tableau de bord, ...). Dans cette phase, l'ergonome contribue à :

- la répartition géométrique des fonctions (première ébauche),
- la définition de critères d'accessibilité et de visibilité pour les commandes,
- la définition de critères de visibilité des afficheurs.

Bien que l'ergonome fournisse des préconisations pour l'implantation des composants, les solutions retenues sont généralement plus guidées par les aspects techniques. La première ébauche est donc loin d'être optimale pour les critères ergonomiques. La phase suivante a pour but de décider si ceci est acceptable ou non. L'intervention E4 correspond donc à l'évaluation de la dégradation des critères ergonomiques.

Les dernières interventions de l'ergonome (qui nous concernent moins ici), s'effectuent sur la maquette de style.

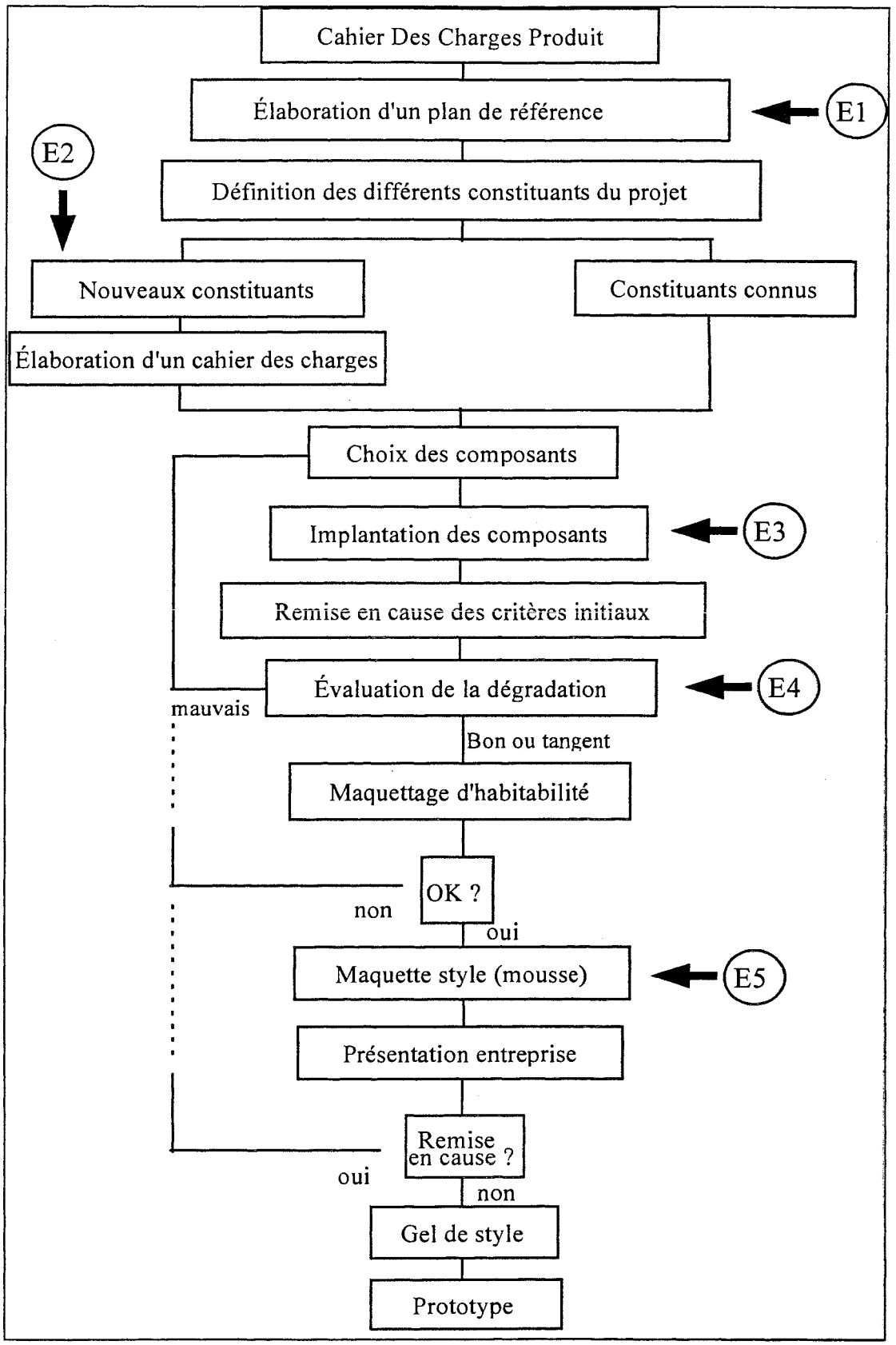

Figure 1 : Les interventions ergonomiques (E1 à E5) dans le processus de conception d'un habitacle de véhicule industriel

1.2.2 La nature et les outils de l'intervention ergonomique

Nous nous intéressons plus particulièrement aux interventions ergonomiques lors de la phase de constitution de la maquette du produit sous logiciel de CAO (c'est-à-dire aux interventions E3 et E4 du schéma de la Figure 1).

Dans le processus décrit dans le paragraphe précédent (1.2.1), l'ergonome intervient donc dans ce cadre soit pour fournir des arguments pour décider du bon positionnement d'une fonction et de la bonne répartition des différents composants, soit pour évaluer la dégradation des critères ergonomiques, une fois qu'une première décision d'implantation a été arrêtée.

On peut classer les "prestations ergonomiques" concernant la cabine en 6 types différents :

- la géométrie du poste de conduite (les aspects dimensionnels, l'implantation des commandes),
- l'accessibilité de la cabine,
- la visibilité (à travers le pare-brise et les ouvrants),
- l'habitabilité (déplacements à l'intérieur de la cabine),
- les rangements (sous forme de préconisations),
- divers autres aspects (accès à la plate-forme arrière, basculement de la cabine, accès au pare-brise par l'extérieur).

Pour répondre aux questions posées par le bureau d'études sur ces différents aspects, l'ergonome utilise trois sources principales d'informations : les modèles, les résultats d'essais expérimentaux, l'analyse de la concurrence.

Les modèles font ici référence aux abaques ou aux algorithmes qui permettent d'obtenir la valeur de critères. Ces critères sont en fait l'association d'indices de qualité ou de satisfaction à des paramètres d'entrée, soit sous la forme de tableaux, soit sous la forme de courbes. Par exemple pour le siège, l'ergonome peut utiliser des modèles de confort fournissant des notes prédictives de confort en relation avec certains paramètres liés aux dimensions du siège, ou à la hauteur de l'assise, ... Il faut noter cependant qu'il est très difficile de trouver des critères permettant de quantifier le confort. Il n'existe pas de modèle générique permettant de prédire le confort. Par contre, certaines approches heuristiques, liées à des domaines spécifiques, permettent de fournir des tendances.

Les essais expérimentaux correspondent à des tests réalisés sur maquette à taille réelle, ou bien sur un dispositif spécifique (tel qu'un siège ergonomique expérimental, équipé de capteurs), ou bien en situation réelle, dans tous ces cas avec de vrais sujets. Ils fournissent aussi des valeurs permettant de juger de l'acceptabilité d'une disposition de fonctions, mais les cas d'entrée (les valeurs des paramètres) sont limités aux valeurs testées dans l'expérimentation. Notons cependant que certaines expérimentations peuvent mener à la production de données permettant de concevoir des modèles plus complets.

L'analyse de la concurrence permet aussi de constituer des intervalles de valeurs acceptables pour certains paramètres, dans les cas où l'on ne dispose ni de modèles, ni de résultats expérimentaux.

Lorsque aucune de ces trois sources d'informations n'est disponible, les réponses fournies par l'ergonome sont de nature subjective. C'est le cas notamment pour le levier de vitesse, les composantes d'effort (au poste de conduite), l'habitabilité. Les réponses de nature subjective concernent aussi les compromis entre critères. Il n'existe en effet pas de données objectives

sur les choix à effectuer lorsque les critères utilisés pour des paramètres différents fournissent des résultats antagonistes.

Par ailleurs, il existe un ensemble de règles et de préconisations, plus ou moins formalisées. Elles s'appliquent en général à un composant donné. Ainsi, certains afficheurs doivent être positionnés dans le champ de vision principal (défini lorsque l'opérateur est en position de conduite) ou bien dans le champ périphérique, ou bien au-delà (en tournant la tête). De même, des règles de visibilité sont associées à certaines commandes. Des règles d'accessibilité sont aussi associées à certaines commandes : par exemple, certaines doivent pouvoir être accessibles le dos collé au siège. Un autre exemple de règle concerne simultanément le mode d'accès et l'utilisation d'un composant donné; ceci est codé selon trois valeurs possibles : bout de doigt, trois doigts (par exemple pour le cas où il faut effectuer une rotation, ou un pincement, ou bien encore pour utiliser le cendrier), ou pleine main (cas du levier de vitesses, cas de la pomme à attraper dans la boite à gants, ...).

Pour ce qui concerne les tests d'atteinte, le concepteur utilise des zones d'atteintes prédéfinies. Il s'agit de zones géométriques calculées a priori à partir d'une longueur du membre considéré (le bras par exemple) variant selon le type d'atteinte préconisé : bout de doigt, trois doigts (50 mm de plus) ou pleine main (50 mm de plus que trois doigts), pour le cas du bras. En pratique le concepteur superpose sur sa maquette graphique le dessin de la zone d'atteinte considérée (celle du bras s'il s'agit d'un composant utilisé avec la main). Si le composant est dans la zone, l'implantation est considérée comme bonne, sinon il y a problème et question à l'ergonome.

Un aspect important de l'intervention ergonomique est la prise en compte des caractéristiques de la population cible des futurs utilisateurs. En effet, les différents tests concernant l'acceptabilité de la solution d'implantation choisie et/ou la dégradation des critères ergonomiques se déclinent aussi en fonction des caractéristiques des futurs utilisateurs. Ces caractéristiques sont liées essentiellement aux aspects dimensionnels. Les résultats fournis par les modèles sont souvent associés à un individu moyen, le "50ème centile"[2], lorsqu'ils ne prennent pas en compte les variables morphologiques. Avec les tests expérimentaux, on peut utiliser des individus de sexe, de taille et de corpulence variables, mais les résultats sont souvent limités du fait du petit nombre de sujets (pour des expérimentations souvent assez lourdes). C'est là que les modèles graphiques d'opérateur humain trouvent leur principale utilité lorsqu'ils permettent de faire varier les morphologies.

Dans le contexte décrit ici, MAN3D est principalement utilisé pour effectuer des tests d'encombrement. Ces tests consistent à vérifier que l'opérateur peut être installé dans l'habitacle, sans interférences avec les éléments tels que le montant latéral pour l'épaule gauche ou le volume central pour la jambe droite, ou bien le volant ou la planche de bord pour les genoux. En pratique, comme pour l'atteinte, il existe des zones géométriques définissant les limites à ne pas dépasser (qui sont spécifiées par l'ergonome au début du processus de conception). Lorsqu'il n'existe pas de zone prédéfinie, le mannequin est positionné "en position de conduite" sur la maquette graphique et le test des interférences se fait de manière visuelle, soit en statique c'est-à-dire sans modifier la posture du mannequin, soit en modifiant la posture pour simuler des attitudes ou des gestes.

---

[2] Il s'agit du 50ème centile pour la stature. Cet individu moyen a donc une stature dont la valeur est celle en-dessous de laquelle se trouve 50% de la population étudiée (la distribution de stature étant gaussienne).

1.3 Les modèles géométriques de l'homme utilisés en CAO

L'apparition des modèles de l'homme en CAO s'est faite dans le contexte du passage des maquettes papier aux maquettes informatisées, sous la forme d'une numérisation des silhouettes planes utilisées auparavant. Dans un premier temps, ce passage ne s'est donc pas accompagné d'un déplacement de la problématique ou d'une modification des objectifs qui auraient pu survenir au vu des nouvelles possibilités offertes par l'informatisation. L'idée était avant tout de répondre à l'objectif de représentation du corps humain dans ses aspects dimensionnels, afin d'effectuer des tests d'encombrement. C'est ce qui permet d'expliquer la nature et les caractéristiques des modèles encore utilisés actuellement dans ce contexte.

Nous décrivons ici dans un premier temps le modèle MAN3D, qui constitue le point de départ de ce travail (§1.3.1), puis nous analysons les concepts clefs liés à la modélisation de l'opérateur humain, pour aboutir à une brève analyse des modèles existants (§1.3.2).

1.3.1 MAN3D : un modèle de l'homme

Le modèle MAN3D de l'INRETS (7) permet de créer et de mettre en situation (c'est-à-dire d'intégrer sous logiciel de CAO dans une maquette de produit en cours de conception), des "sujets" , qui sont des représentations géométriques et fonctionnelles d'hommes, de femmes ou d'enfants, de dimensions variables.

1.3.1.1 La représentation du corps humain

Un sujet est une représentation géométrique résultant de l'association de 4 entités :

- une structure de base décrivant la topologie et les axes de mobilité du squelette, formée de segments, appelés aussi chaînons (Figure 2), reliés entre eux par des articulations (Figure 3 et Figure 4), auxquelles sont rajoutés des points remarquables (Figure 5). Les points remarquables sont soit des points correspondant à des extrémités de chaînons (extrémités des doigts par exemple), soit des points particuliers non liés à des chaînons, rajoutés à la structure (tels que l'origine du bassin, ou bien certains points matérialisant la tête).
- des données de contours définissant l'aspect extérieur du sujet (Figure 6 et Figure 7),
- des données de dimensions associées aux segments,
- un ensemble d'angles intersegmentaires, fournissant l'arrangement géométrique des pièces du squelette, c'est-à-dire l'attitude posturale du sujet.

La structure de base ainsi que les données de contours constituent un modèle figé, qui a été défini à partir des données biomécaniques disponibles. Les dimensions peuvent être modulées, de façon à représenter des sujets de morphologie et de taille différentes. Des procédures sont proposées à l'utilisateur pour lui permettre de représenter les sujets de son choix. La modification des angles intersegmentaires permet de placer un sujet dans des postures différentes.

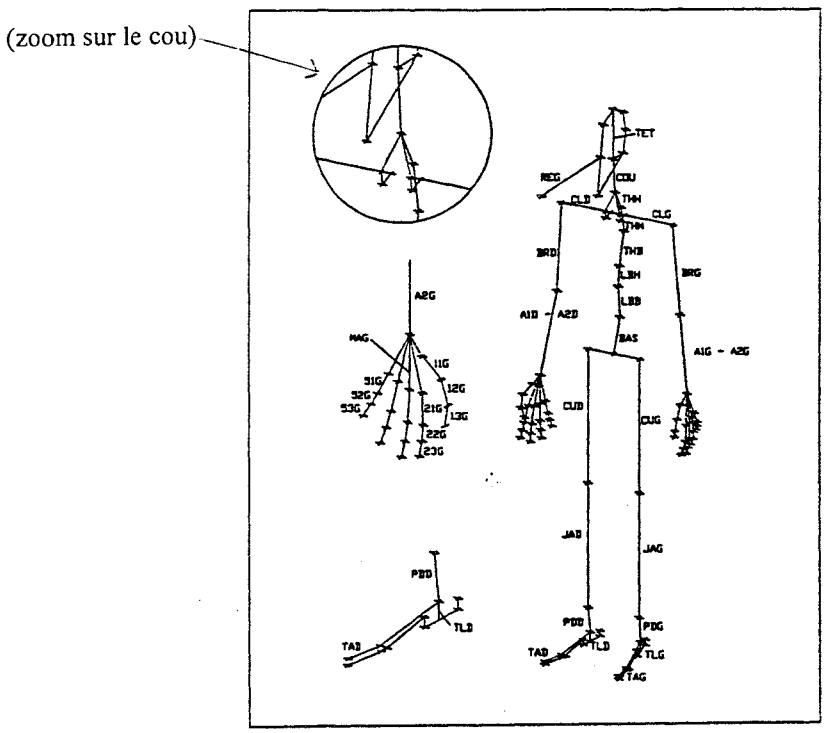

Figure 2 : Les chaînons du squelette de MAN3D [schéma de J.P. Verriest (7)]

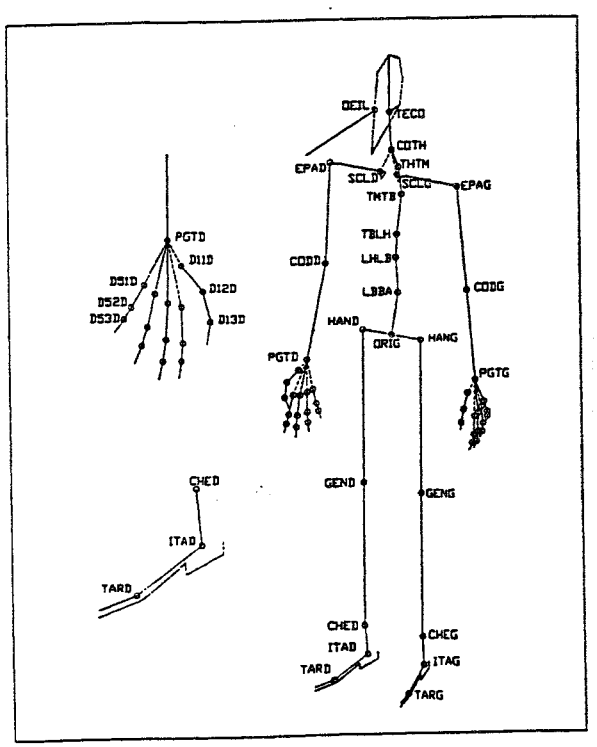

Figure 3 : Articulations connectant les chaînons de MAN3D [schéma de J.P. Verriest (7)]

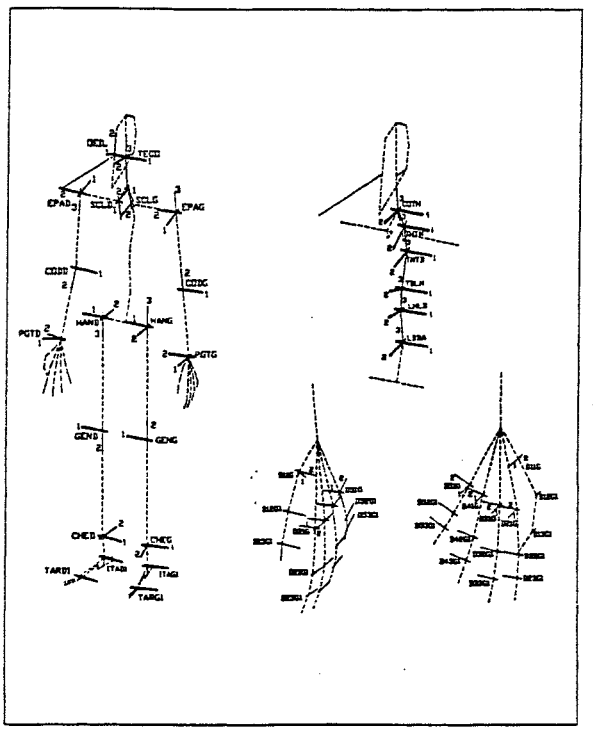

Figure 4 : Axes de mobilité articulaire de MAN3D [schéma de J.P. Verriest (7)]

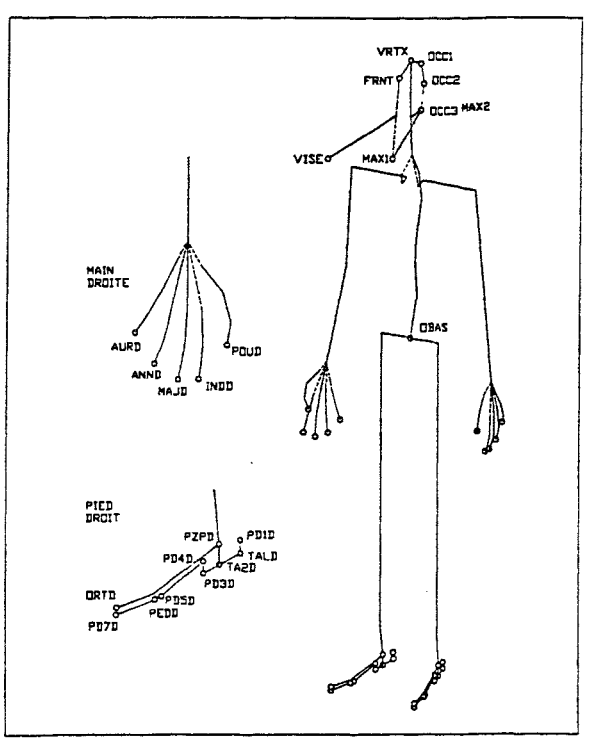

Figure 5 : Localisation des points remarquables autres que les articulations [schéma de J.P. Verriest (7)]

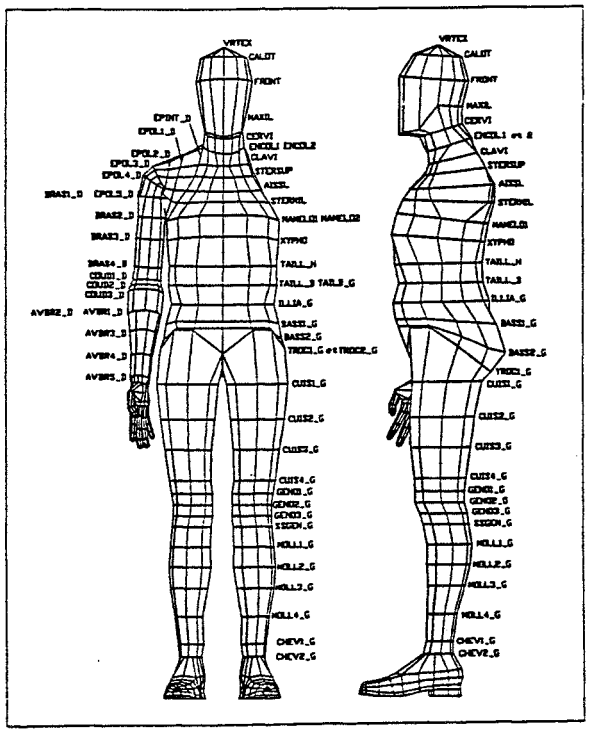

Figure 6 : Localisation des couronnes de contours de MAN3D [schéma de J.P. Verriest (7)]

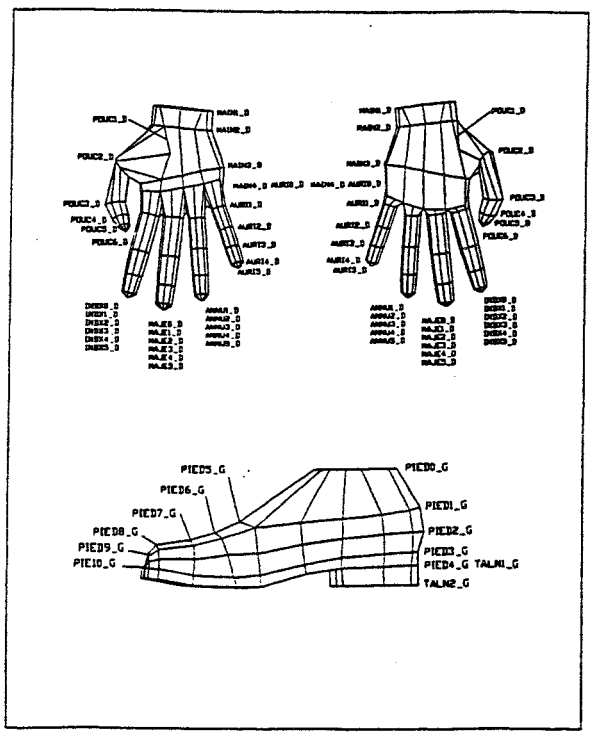

Figure 7 : Localisation des couronnes de contours de MAN3D. Détail des mains et des pieds. [schéma de J.P. Verriest (7)]

1.3.1.2 Les capacités fonctionnelles

La représentation des capacités fonctionnelles (limitée aux capacités fonctionnelles motrices) est réalisée à deux niveaux :

Au niveau de la structure elle-même, la définition de butées articulaires, associées aux articulations connectant entre eux les segments, permet de représenter les limites articulaires réelles de l'homme. Il existe des butées pour chacun des axes de mobilité associés à toute articulation.

Le modèle comporte par ailleurs la définition de " mouvements" permettant de représenter les capacités fonctionnelles motrices. Un mouvement est une opération permettant de passer d'une posture à une autre.

Le modèle s'appuie sur la notion de mouvements élémentaires, définis conformément aux usages de la kinésiologie. Un mouvement élémentaire correspond à un déplacement angulaire ne mettant en jeu qu'un seul axe de mobilité. Les différents types de mouvement sont décrits pour un sujet debout, bras et jambes le long du corps; le plan sagittal est le plan de symétrie du tronc. On distingue :

- l'abduction : mouvement tendant à éloigner un segment du plan sagittal,
- l'adduction : mouvement tendant à rapprocher un segment du plan sagittal,
- la flexion : mouvement tendant à replier le corps sur lui-même,
- l'extension : mouvement tendant à déplier le corps,
- la latéroflexion : mouvement dans le plan frontal perpendiculaire au plan sagittal,
- la rotation axiale : rotation autour de l'axe longitudinal du segment.

Ces mouvements sont représentés sur la Figure 8.

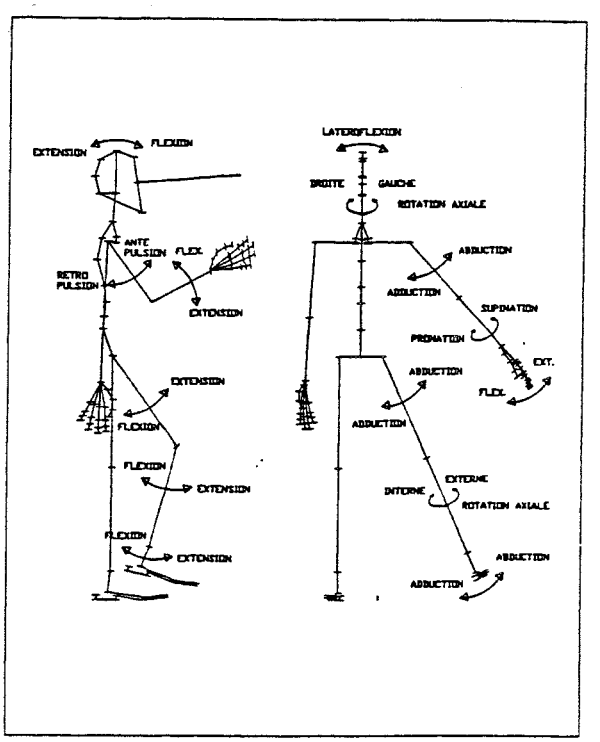

Figure 8 : Principaux types de mouvements élémentaires [schéma de J.P. Verriest (7)]

Ces définitions de mouvements élémentaires supposent que les mouvements sont considérés séparément. Lorsqu'un mouvement est décrit pour un axe de mobilité donné, tous les autres axes de mobilité sont en position neutre. En réalité, dans les activités humaines, les mouvements sont rarement de niveau élémentaire : ils mettent en jeu simultanément plusieurs axes de mobilité. C'est pourquoi des mouvements combinés ont été prédéfinis, associés à certaines articulations : l'épaule, le coude, le poignet, pour le membre supérieur, et la hanche, le genou, et la cheville, pour le membre inférieur. Ces mouvements sont : la flexion-extension, l'orientation, le pivotement, l'atteinte simple et l'atteinte orientée. Ils permettent en principe de simuler n'importe quel mouvement réel.

D'autres mouvements ont été prédéfinis : la combinaison de mouvements élémentaires pour simuler des mouvements d'atteinte d'un point avec les doigts, des mouvements du tronc : flexion sagittale, inclinaison latérale, torsion, ainsi que des combinaisons de ces mouvements (flexion + inclinaison, flexion + torsion), la flexion bassin-cuisse, la flexion buste/verticale.

Un chaînon particulier, associé au point œil (lui-même considéré comme une articulation), correspond à la direction du regard.

1.3.1.3 La modification de la posture

Revenons maintenant au niveau de la structure de base, composée de chaînons articulés. Nous nous intéressons ici à la question de sa manipulation. Toute modification d'angle associée à une articulation correspond à un changement d'attitude posturale. Une posture au sens de MAN3D est l'ensemble des valeurs pour chaque angle associé à la structure.

Pour modifier la posture d'un sujet, on peut donc accéder à un angle et lui donner une nouvelle valeur. Cette méthode pose deux problèmes : d'une part, cela constitue un exercice trop fastidieux pour l'utilisateur, d'autre part cela suppose une connaissance fine de la manière dont sont exécutés les gestes réels et la capacité de retranscrire cette connaissance au niveau des différents angles de la structure. D'où la nécessité de doter le modèle de capacités de détermination de géométries posturales. C'est pourquoi plusieurs niveaux de commande de la structure (qui correspondent à des calculs de postures) ont été modélisés.

Les mouvements élémentaires (décrits plus haut au §1.3.1.2), correspondent à un déplacement angulaire ne mettant en jeu qu'un seul axe de mobilité. Ils servent de base pour la définition de mouvements combinés, qui permettent à l'utilisateur d'obtenir des changements de posture typés (flexion-extension, orientation, pivotement, atteinte simple ou orientée). Ces changements de posture sont calculés par des algorithmes spécialisés qui fournissent en sortie les valeurs des angles, à partir de contraintes de position ou d'orientation spécifiées par l'utilisateur. Ces mouvements s'appliquent de manière non simultanée, soit à un membre (bras ou jambe, et aussi doigts), soit au tronc, soit au bassin. On nomme *segment mobile* chacune de ces parties composées de chaînons, sur lesquelles on peut appliquer des commandes de changement de posture.

La catégorie des mouvements combinés dits quelconques concerne les cas où le mouvement à figurer suppose des changements d'angles au niveau de plusieurs parties de la structure : par exemple, s'il s'agit de rehausser et d'avancer la position de l'œil par rapport au point talon, le mouvement entraîne des changements d'angle au niveau des chevilles, des genoux, des

hanches, de toutes les articulations de la colonne jusqu'à la tête et même des épaules et des coudes si les mains doivent rester à la même place. Dans ces mouvements, le calcul des valeurs d'angles à affecter aux différents axes est confié à un algorithme d'optimisation itératif. L'utilisateur doit spécifier avant le calcul les axes de mobilité participant aux mouvements, ainsi que les contraintes géométriques assignées aux différents segments corporels et représentant le but à atteindre.

1.3.1.4 Comment MAN3D est utilisé

MAN3D est actuellement utilisé principalement pour le test d'habitacles de voitures ou de poids-lourds en cours de conception.

Le principe d'utilisation est le suivant :

- au cours d'une session d'un logiciel de CAO, avec lequel on a construit une maquette du futur habitacle, on charge un sujet MAN3D[3]. Ce sujet a été défini préalablement (morphologie, posture) éventuellement par l'utilisateur, mais il existe aussi un ensemble de fichiers de sujets prédéfinis que l'on peut appeler.

- on peut, tout au long d'une session d'utilisation d'un sujet MAN3D, agir sur les options d'affichage. Il existe en effet différentes options : on peut choisir d'afficher ou de ne pas afficher le squelette; il en est de même pour les contours. Lorsque les contours sont affichés, on a accès aux modes filaire (points et segments de droites), volumique (solides délimités par des facettes quadrangulaires) ou surfacique (surfaces passant par les sommets des couronnes de contours). On peut choisir de ne faire afficher qu'une partie du mannequin, ce qui permet de réduire le temps de calcul lorsque les contours sont affichés (option particulièrement utile en mode surfacique).

- le sujet peut être manipulé : il peut être déplacé (indication d'une nouvelle position pour un point remarquable du mannequin), son orientation peut être modifiée (il faut choisir l'un des trois axes possibles et rentrer la valeur d'angle voulue).

- enfin, les différentes possibilités de mouvements (décrits plus haut aux § 1.3.1.2 et 1.3.1.3) permettent de modifier la posture du sujet et d'effectuer des tests d'atteinte.

Ainsi, le mannequin peut être utilisé pour tester l'habitacle en cours de conception :

- le chargement d'un sujet et son positionnement dans l'espace permet de vérifier que le futur conducteur pourra bien s'insérer dans la cabine et donc d'effectuer des tests d'encombrement. Cette vérification peut être effectuée pour des sujets de morphologies différentes grâces aux possibilités de dimensionnement des sujets MAN3D.

- l'utilisation des possibilités de mouvements permet d'effectuer des tests d'atteinte des différents éléments de l'habitacle, depuis une position. On peut contraindre un mouvement en agissant sur la désignation des points et chaînons fixes. On peut ainsi

[3] MAN3D est implémenté sous la forme d'une application Euclid (logiciel de MatraDatavision)

simuler par exemple une position de conduite, et tester l'atteinte des pédales, tout en conservant la position assise, le regard sur la route, et les mains sur le volant.

- en règle générale, la résultante principale d'une action commandée au mannequin est la modification de position, d'orientation générale ou de posture, visible sur l'écran. Lorsqu'un mouvement est impossible, on obtient à l'affichage soit aucun changement, soit la position limite obtenue. Par ailleurs, selon le cas, on obtient éventuellement le message *atteinte impossible* et (ou) le message *articulation en butée* (avec spécification de l'articulation concernée).

1.3.1.5 Aspects mathématique et informatique

Le modèle MAN3D s'appuie sur un certain nombre de techniques mathématiques, notamment la régression linéaire pour le calcul des dimensions corporelles et l'optimisation sous contraintes pour le calcul des postures.

L'implémentation est réalisée en langage Fortran. Il existe plusieurs versions de MAN3D, qui diffèrent du point de vue de l'usage des bibliothèques graphiques et de l'interaction avec l'utilisateur. La version principale (utilisée dans l'industrie) est interfacée avec Euclid, logiciel de CAO. Toute la partie interface est donc réalisée avec les possibilités de cet outil.

Les différentes commandes du système sont proposées sous la forme de menus déroulants. Outre le choix d'un item de menu, l'utilisateur peut être amené à fournir différentes informations : des noms de fichiers (pour sauvegarder ou charger des sujets ou bien des postures), des valeurs d'angles, et pour les commandes de mouvements la désignation de points de l'opérateur et d'axes de mobilité, ainsi que de points de l'espace (pour les points cibles à atteindre).

La
Figure 9 représente l'arborescence des menus proposés à l'utilisateur et fournit ainsi un résumé des fonctionnalités de MAN3D. Celles-ci sont regroupées en trois groupes principaux : la création d'un sujet, la gestion des paramètres d'affichage et les modifications de la posture.

1.3.1.6 Évolution de MAN3D

Nous avons présenté dans ce paragraphe (1.3.1) les bases du modèle MAN3D et les principes de son utilisation. Nous avons signalé dans le paragraphe précédent (1.3.1.5) qu'il existe plusieurs versions du logiciel, liées à son implantation. Il faut souligner aussi que le modèle lui-même évolue. Il s'agit en effet d'un produit de la recherche, dans lequel sont intégrés progressivement de nouveaux algorithmes, au fur et à mesure de l'évolution des connaissances du domaine. Son évolution est notamment liée à la progression des recherches biomécaniques effectuées sur des sujets humains. Ces recherches, basées sur des expérimentations lourdes et coûteuses, s'attachent par nature à des sous-ensembles très restreints de la connaissance du domaine. Nous reviendrons plus en détail dans le chapitre 4 sur certaines de ces recherches. Par ailleurs, le travail effectué dans le cadre de cette thèse s'appuie sur une des versions du modèle, disposant de commandes de mouvement associées à

chaque membre. D'autres ajouts ont été effectués au modèle en parallèle et il sera pertinent dans le futur de ré-examiner la question du rôle de MAN3D dans le système développé.

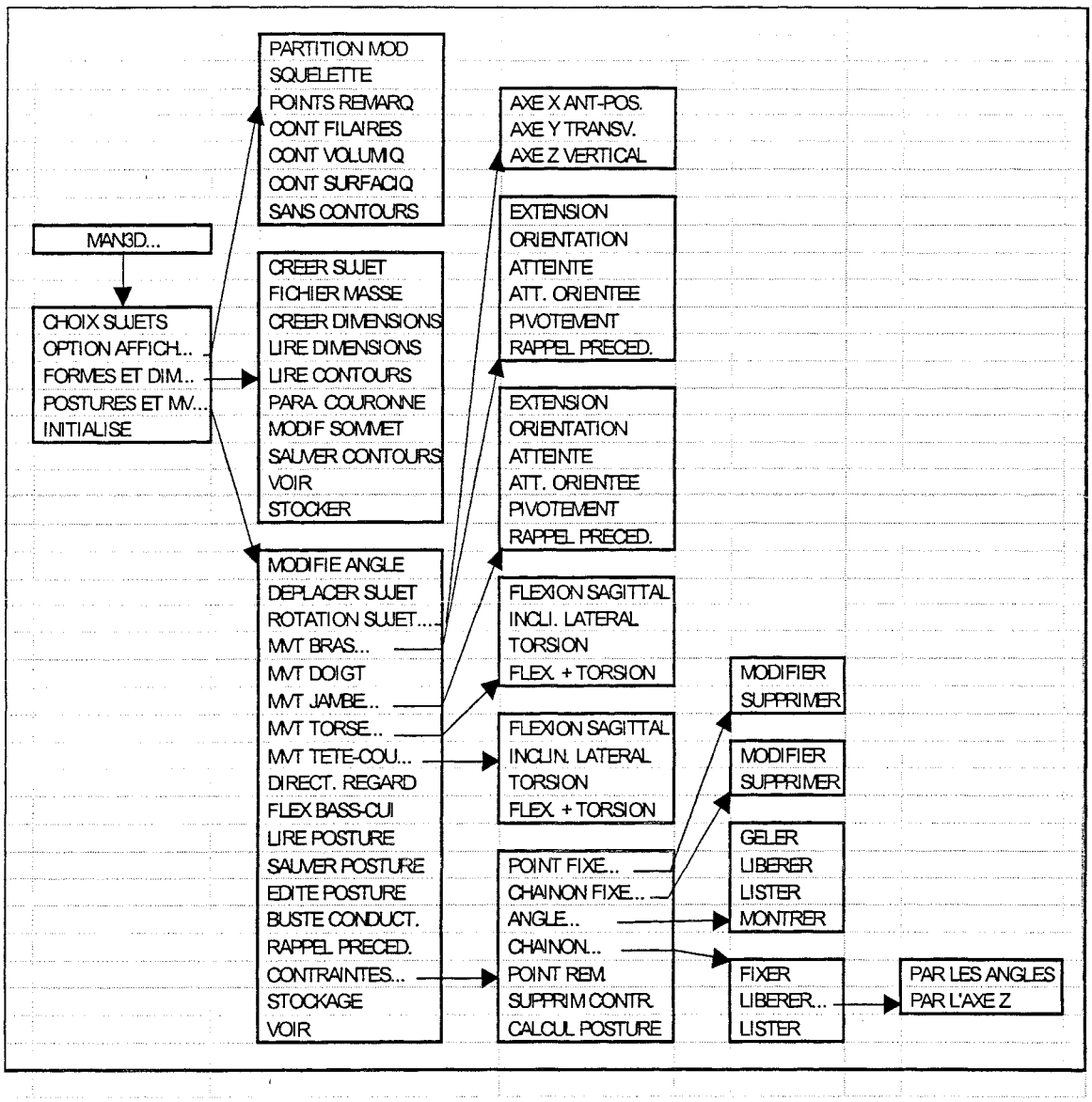

Figure 9 : Arborescence des menus de MAN3D

1.3.2 Les concepts clefs d'un outil d'aide à la conception centré sur un modèle d'opérateur humain

Nous analysons ici un certain nombre de concepts liés à la modélisation de l'opérateur humain, afin de pouvoir expliquer la complexité de cette thématique et nous permettre ensuite de mieux situer notre problématique. Ces concepts nous servent aussi à décrire l'existant en termes de modèles d'opérateur humain, dans le paragraphe suivant (1.3.3).

1.3.2.1 Le réalisme

Lorsque l'on conçoit un modèle graphique d'opérateur humain, on cherche à le rendre le plus réaliste possible. Ce réalisme peut concerner différentes composantes du modèle :

- le réalisme de la représentation du corps en terme de dimensions : il s'agit de s'écarter le moins possible des dimensions réelles de l'être humain. Pour prendre en compte la diversité des formes humaines, il faut proposer des outils permettant de moduler ces dimensions, tout en continuant à garantir le réalisme du résultat. Il faut donc avoir un modèle réaliste à la fois du point de vue des dimensions et des liens existant entre elles.

- le réalisme de la représentation des formes du corps : il s'agit là des aspects concernant les proportions de l'enveloppe corporelle. On peut appliquer ici les mêmes remarques que pour les dimensions.

- le réalisme de l'aspect : on peut chercher à produire un effet réaliste au niveau du rendu de l'image, par exemple pour les traits du visage, la régularité des contours, ...

- le réalisme en termes de capacités fonctionnelles : réalisme des butées articulaires, réalisme au niveau de la possibilité ou l'impossibilité de réaliser un geste ou d'être dans une posture particulière, réalisme sur la manière d'effectuer un geste donné (comment saisit-on un crayon par exemple ?, sachant qu'a priori, il existe une infinité de solutions possibles).

On peut chercher bien sûr à obtenir le maximum de réalisme sur tous ces aspects simultanément. Cependant, ceci est plus ou moins facile, plus ou moins coûteux, et surtout plus ou moins utile selon l'objectif que l'on poursuit. Il est clair que dans un objectif de simulation du comportement humain pour l'aide au diagnostic ergonomique en conception, les aspects liés à la qualité de l'image et aux traits du visage sont secondaires. Par contre, le réalisme de la représentation du corps et des capacités fonctionnelles est un objectif prioritaire. Par exemple, pour la conception d'espaces réduits tels que les cockpits ou les habitacles de voitures, le réalisme en terme de morphologie est un point crucial.

Le réalisme est donc un objectif pour la modélisation de l'opérateur humain, qui s'applique à différents niveaux, qui sont plus ou moins prioritaires selon les applications. Mais une fois cet objectif défini et des moyens mis en œuvre pour l'atteindre, il reste un problème important : celui de la mesure de la réalisation de cet objectif. La question est : comment juge-t-on du réalisme d'un modèle d'opérateur humain ? Cette question est plus ou moins difficile suivant le niveau dont on parle :

- pour les dimensions corporelles par exemple, on peut représenter de manière fiable un sujet donné, si l'on peut reproduire de manière exacte ses dimensions. Le degré de réalisme de la représentation dépendra alors des caractéristiques du modèle (le réalisme sera d'autant plus grand que le nombre de chaînons représentant le squelette se rapproche de la réalité), mais il est aussi lié à la qualité des données d'entrée (les mesures effectuées sur le sujet) qui peuvent poser certains problèmes.

- pour les postures, juger du réalisme est un problème difficile (si l'on essaie de lui trouver une solution analytique). Il existe des méthodes pour enregistrer des postures réelles de sujets, notamment à l'aide d'outils vidéo, mais la question de l'écart entre la posture observée et la posture simulée reste posée.

1.3.2.2 La prédiction de la posture

La posture d'un sujet humain au sens courant est l'attitude adoptée par l'ensemble du corps. On parle par exemple de la position ou posture assise, qui sous-entend des positions spécifiques de certaines parties du corps : buste vertical, jambes repliées ; notons que la position assise suppose aussi la présence d'un élément extérieur, tel que le sol ou un siège par exemple.

Les postures peuvent être associées à des activités particulières (conduire, dormir, courir, manger, ...). Un sondage effectué auprès de n'importe quelle population d'individus dans le but d'élaborer une description de la posture associée à une activité donnée fournirait certainement une forme de consensus. Par exemple, il faut porter la nourriture à la bouche pour manger. Si l'on demandait d'être plus précis dans la description, cela deviendrait plus complexe ; il faudrait introduire d'autres spécifications de départ : le contexte global (assis par terre, assis sur une chaise, debout, ...), les objets utilisés (un bol ou une assiette, une fourchette, ...), la localisation de la nourriture et sa distance par rapport à la bouche, ... En supposant que tous ces éléments soient spécifiés, le problème de la prédiction et de la description d'une attitude corporelle n'est pas résolu : par exemple, il peut exister plusieurs façons de porter une fourchette à sa bouche lorsque l'on est assis devant une table, ...

Ce problème de la description et de la prédiction de la posture est la difficulté majeure rencontrée pour la conception des modèles graphiques simulant le comportement moteur d'un opérateur humain. Avec un tel modèle, on doit représenter les postures, de manière réaliste, c'est-à-dire de la façon la plus ressemblante possible par rapport au sujet humain, à l'aide d'une chaîne de segments articulés. Et surtout, on doit être capable de faire en sorte que cette structure adopte des postures réalistes. Dans un contexte de simulation, le problème est de prédire une posture, en fonction de paramètres à définir.

Un premier pas vers le réalisme peut être réalisé grâce à la définition de limites pour les valeurs des angles associés aux articulations. Mais ceci n'est pas suffisant pour assurer l'obtention d'une posture réaliste dans tous les cas. En effet, même en respectant les limites angulaires représentant les butées articulaires, on peut obtenir une posture ayant très peu de chances de pouvoir être observée dans la réalité. Cela signifie donc qu'il faut se poser la question des paramètres qu'il faut prendre en compte pour expliquer la posture et en assurer le réalisme.

Sans prétendre avoir mené une étude approfondie sur le sujet, nous pouvons évoquer quelques paramètres qui semblent pertinents pour expliquer une posture donnée : le but du geste, la durée de la posture finale, le niveau de confort ou d'inconfort associé à la posture, le poids de l'objet (s'il s'agit de porter un objet par exemple), ou plus généralement les caractéristiques de l'objet (taille, forme, ...), la force exercée sur l'environnement, le respect de l'équilibre du corps, l'histoire de la personne (l'origine de son propre comportement postural, l'habitude, l'entraînement, ou l'expertise acquise) aussi bien que des paramètres plus culturels (il existe par exemple différentes façons d'effectuer le même geste selon les sociétés). Ces paramètres sont nombreux et il est difficile de modéliser leur influence sur la posture. De plus, la posture est sujette aux variations interindividuelles et il existe aussi une variabilité intra-individuelle, pour un même sujet au cours du temps. On trouvera notamment dans (8) une analyse de la

question de la simulation du mouvement humain pour la conception ergonomique, qui évoque cette question des déterminants de la posture.

En fait, le problème des déterminants de la posture est loin d'être résolu. Il nous semble que les approches jusque-là utilisées, basées sur des modèles purement biomécaniques et géométriques, ont contribué à limiter la recherche de ces déterminants.

Pour illustrer ce propos, nous exposons ci-dessous une manière de décrire le problème de la prédiction de la posture utilisée par les modèles géométriques d'opérateur. On pourra se référer par exemple à (9) pour des explications plus détaillées. Les données du problème sont :

- une partie du corps (par exemple le bras), dont la posture initiale est donnée par les valeurs des angles des articulations liant les différents segments de cette partie. Ces angles sont nommés θ_i, i variant de 1 à m (avec m le nombre d'axes concernés),
- la désignation d'un point de la partie concernée, qui est le point qui devra rejoindre le point cible,
- un point cible à atteindre, dont on fournit les coordonnées x, y et z dans le repère considéré.

Le problème est de déterminer la posture finale après l'atteinte du point cible par le point désigné, c'est-à-dire de déterminer les nouvelles valeurs des angles θ_i. C'est un problème de *géométrie inverse*. La structure est dite *redondante*, car il y a une infinité de solutions pour le problème défini ainsi.

Pour résoudre ce problème, il faut donc trouver des modèles, afin de le simplifier. Par exemple, un modèle simple peut résulter de la définition de relations entre les θ_i. Ainsi, le nombre d'inconnues à calculer diminue.

L'optimisation sous contraintes est une méthode fréquemment utilisée pour la résolution de ce problème. Il s'agit de minimiser une fonction des θ_i pour un critère donné (le confort par exemple). Si l'on définit plusieurs critères à prendre en compte simultanément, on cherche alors à minimiser la somme des fonctions associées aux critères. C'est la raison pour laquelle pour cette méthode les critères doivent être quantifiables. Les autres facteurs (non quantifiables) ne peuvent donc pas être pris en compte, même s'ils paraissent importants pour assurer le réalisme de la posture.

Outre les aspects purement géométriques, il est nécessaire de s'intéresser aux composantes d'effort associées à une posture. Il est possible de développer des modules de prédiction de la posture en fonction de l'effort à produire. Dans le même ordre d'idée, on peut réaliser un module de calcul de l'équilibre, associé à une posture donnée ; cet aspect (différent de la question de la prédiction de la posture, puisque le point de départ est une posture) rejoint la question de la qualification d'une posture, évoqué plus loin à propos du diagnostic ergonomique (paragraphe 1.3.2.5).

Sur les aspects réalisme et prédiction de la posture, on peut aussi s'intéresser à ce qui n'est pas de l'ordre des outils mathématiques, c'est-à-dire au savoir-faire humain. Il existe en effet (notamment dans le domaine du sport) des personnes possédant une expertise sur le réalisme d'une posture dans un contexte spécifique, capables par exemple de reconnaître une personne par son style (les postures qu'elle adopte dans sa pratique sportive), d'imaginer une posture efficace. Dans ce type de domaine, les outils de représentation graphique de l'homme fournissent une aide à la représentation des postures imaginées (ou permettent de calquer des

postures enregistrées sur des vidéos). Cela impose aux outils tels que MAN3D une nécessité de facilité de manipulation. Cet aspect est développé dans le paragraphe suivant.

1.3.2.3 La manipulation du mannequin

Les modèles graphiques d'opérateur humain posent donc un autre problème : celui de la manipulation de la structure articulée. La question est : comment l'utilisateur indique-t-il ce qu'il veut faire réaliser au mannequin ? Il s'agit d'un problème d'interface homme/machine.

Pour Phillips (10) le terme de manipulation géométrique s'applique au processus de composition interactive d'une scène, ou de spécification interactive de positions et de postures pour des représentations géométriques de l'homme. Ces actions sont généralement réalisées sur une base essai/erreur. La manipulation entraîne généralement le mouvement des figures manipulées, mais dans ce cas le mouvement obtenu n'est pas significatif en tant que séquence de mouvement réaliste. Plus exactement, le but de la manipulation est l'obtention d'une posture spécifique désirée. La manipulation a un caractère intrinsèquement temps réel : les objets doivent bouger en réponse directe aux actions de l'utilisateur.

Phillips a développé une méthode de manipulation directe pour des figures articulées en 3D. Cette méthode est implémentée dans Jack (voir notamment (11)), un système conçu pour modéliser, manipuler et analyser des figures articulées représentant des humains (ou des animaux par exemple). Le contrôle de la posture est réalisé sur la base du respect des limites articulaires, de l'équilibre, et des limites d'exertion de force. Avec Jack, l'utilisateur peut faire de la manipulation directe (avec la souris) en saisissant le sujet par l'une de ses poignées et en lui appliquant une traction pour le faire se déplacer. La structure se déforme alors dans la direction fournie, en respectant les limites articulaires et l'équilibre du corps. Cet atout de manipulation directe en temps réel apparaît dans un premier temps comme séduisant, mais il est cependant limité. En effet, le mouvement est limité à un seul plan et ne s'exerce qu'à partir d'un seul point à chaque opération, de sorte que la complexité de la manipulation persiste, du fait notamment de l'aspect 3D du mouvement réel.

Notons que les travaux réalisés pour Jack ont eu notamment pour cadre une problématique d'animation, alors que dans le contexte que nous évoquons, il s'agit de simulation. En animation, on cherche à créer des figures animées, dont certaines peuvent être réalisées dans le but de reproduire le comportement humain. Dans ce cas on s'attachera à obtenir un bon niveau de réalisme, à la fois pour l'apparence, et aussi pour la démarche par exemple. Mais les figures créées ne font que rejouer un scénario écrit d'avance, à la différence d'une problématique de simulation, ou les modèles sont dotés de mécanismes permettant de reproduire un comportement, en fonction de données de contexte variables. Dans le cas de la simulation, on est dans un contexte de prédiction du comportement, ce qui n'est pas le cas pour l'animation.

1.3.2.4 Niveau d'automatisation et détermination de la posture

Les questions de manipulation du mannequin et de détermination de la posture ne peuvent pas être déconnectées de la question du degré d'automatisation du modèle. On entend ici par degré d'automatisation, le caractère plus ou moins automatique de la simulation du comportement.

Au niveau d'automatisation maximum, l'intervention de l'utilisateur est limitée à l'envoi de la simulation, sur la base de la définition d'un objectif ou de spécifications de départ. Au niveau d'automatisation minimum, l'utilisateur doit manipuler la structure valeur d'angle par valeur d'angle. Entre ces deux extrêmes, il existe différents niveaux d'automatisation, correspondant à une prise en charge plus ou moins importante par le système de la détermination de la posture.

En effet, au niveau le plus bas, l'action du système est limitée au contrôle des valeurs d'angles fournies par l'utilisateur (pour qu'elles ne dépassent pas les limites articulaires). Au niveau au-dessus, on peut ajouter la prise en charge du calcul d'une posture pour une partie du corps (bras par exemple). Dans ce cas, où plusieurs articulations sont considérées simultanément, le problème se complexifie à différents titres. Tout d'abord, il faut envisager la question du couplage des valeurs d'angles pour les articulations proches : certaines articulations ont des fonctionnements dépendants. Cela suppose par ailleurs que l'on spécifie le problème, en terme d'entrées (valeurs d'angles initiales) et de résultat attendu (point cible atteint par un point désigné de la partie concernée). Ensuite il faut définir un mode de résolution pour ce problème non trivial, qui est celui de la prédiction de la posture (décrit au §1.3.2.2). Le mode d'intervention de l'utilisateur, lié à la question de la manipulation de la structure (évoqué au §1.3.2.3) doit être lui aussi défini. Il faut enfin spécifier les informations obtenues après la résolution du problème : visualisation du résultat obtenu, messages avertissant des problèmes rencontrés (atteinte impossible si le point est situé au-delà des limites d'extension du bras par exemple, articulation en butée). Dans ce cas, pour simuler le comportement postural, l'utilisateur doit encore prendre en charge l'enchaînement de l'appel aux différentes parties du corps. En effet, si l'atteinte avec le bras seul est impossible, il faut envisager de tenter d'incliner le torse (puis réessayer l'atteinte bras), ou bien avancer le corps entier, selon le cas, ...

Au niveau suivant, on peut donc ajouter la prise en charge de l'enchaînement des appels aux différentes parties du corps, c'est à dire le niveau stratégique permettant le choix du membre à activer en fonction du contexte, pour continuer à rechercher une solution au problème d'atteinte, lorsque celle-ci se révèle impossible dans le cas de l'activation d'un membre unique.

Le niveau d'automatisation le plus haut, niveau "presse-bouton" pour l'utilisateur, serait celui où le modèle permettrait d'obtenir une simulation prise en charge totalement : le système serait capable de prendre en compte tous les éléments de contexte, pour effectuer une tâche donnée et produire des commentaires associés. Dans ce cas, il faudra toujours spécifier la tâche que le modèle d'opérateur humain devra effectuer et ceci, en fonction du but poursuivi. La tâche peut être décrite en termes de buts à atteindre, définis en terme d'espace (marcher vers un point par exemple). Elle peut aussi être plus complexe, lorsqu'il s'agit de simuler l'utilisation d'objets : saisir des objets, les soulever, les déplacer, ... Le diagnostic associé à la simulation, obtenu en sortie, dépendra de la nature de la tâche demandée. Dans (11), on trouve l'exposé d'une tentative de conception d'un langage de description de tâches, permettant d'écrire des scripts d'animation. L'idée est d'utiliser des termes du langage naturel pour spécifier des tâches à faire réaliser par un mannequin graphique, de manière indépendante du cas particulier traité. Ainsi des termes tels que *look at*, ou bien *turn* sont définis pour désigner les actions à effectuer, tandis que les objets géométriques sont eux aussi dupliqués au niveau de la spécification des tâches dans une base visant à stocker de l'information symbolique.

1.3.2.5 Le diagnostic ergonomique

La question du diagnostic ergonomique appliqué à la simulation ergonomique recouvre différents aspects. L'objectif de fond est le diagnostic porté sur le produit en cours de conception, mais le diagnostic ergonomique peut s'appliquer à différents niveaux. Nous en distinguerons ici deux : le diagnostic porté sur les interactions avec le produit testé à l'aide du modèle d'opérateur humain, et le diagnostic associé à la posture générée par le modèle. Ces deux aspects du diagnostic ergonomique sont fortement liés et leur niveau de prise en charge conditionne la qualité des réponses fournies à l'utilisateur.

Critères d'accessibilité et d'encombrement

Dans le contexte de la simulation ergonomique (que nous avons défini au paragraphe 1.1), l'utilisateur d'un modèle d'opérateur humain a pour objectif de tester le produit en cours de conception. Dans le cas d'un habitacle de véhicule, le concepteur cherche à savoir si le produit convient à son futur utilisateur. Le problème est de savoir si ce dernier pourra exercer l'activité prévue dans de bonnes conditions de sécurité et de confort, c'est à dire entrer facilement dans l'habitacle, s'installer confortablement à son poste, utiliser correctement les différentes commandes, pouvoir regarder les afficheurs, détecter les alarmes, ...

Les modèles graphiques d'opérateur humain permettent d'obtenir des éléments de réponse à ce questionnement. En effet, la représentation graphique du "volume occupé" par l'opérateur permet de savoir si ce dernier "tient" dans l'habitacle ; les fonctionnalités issues de la modélisation des capacités motrices permettent d'effectuer des tests d'atteinte des commandes. La nature des réponses fournies par ce type de modèles est avant tout visuelle : le premier résultat obtenu est le positionnement (plus ou moins automatisé) du mannequin graphique dans la *scène* (c'est-à-dire la maquette du produit). En l'absence de fonctionnalités plus élaborées, le diagnostic ergonomique est donc implicite et laissé à la charge de l'utilisateur du modèle. Ainsi, le test d'encombrement sera validé par la perception visuelle que l'utilisateur aura des éventuelles interactions entre l'opérateur et les éléments de l'habitacle, le test d'atteinte sera dichotomique : atteinte possible si l'utilisateur a réussi à obtenir du modèle une position d'atteinte lui paraissant correcte, atteinte impossible sinon, ...

Le diagnostic ergonomique associé aux interactions entre l'opérateur et la scène

Un modèle graphique d'opérateur humain ne proposant que des capacités de représentation du corps humain et de simulation de postures et de gestes n'est donc pas suffisant pour fournir une aide au diagnostic ergonomique dans le contexte que nous décrivons. Un certain nombre de fonctionnalités peuvent être ajoutées au modèle pour fournir ce type d'aide. Par exemple une aide au test d'encombrement peut être proposée avec l'ajout d'un module de calcul d'intersections entre les éléments de la scène et le mannequin. Certains modèles proposent des fonctionnalités liées au champ visuel : délimitation graphique du champ visuel, représentation de la vue qu'aurait l'opérateur à partir d'une position donnée. On trouve aussi des modèles proposant la représentation de volumes d'atteinte.

Le diagnostic ergonomique associé à la simulation de postures

L'autre aspect du diagnostic ergonomique que nous évoquons ici est complémentaire de la simulation de la posture : il s'agit de fournir un diagnostic sur la posture obtenue. En effet,

lorsqu'une posture est obtenue et visualisée, sur un écran par exemple, par l'utilisateur du modèle, il faut pouvoir la qualifier.

Le premier niveau de qualification de la posture est celui du réalisme, qui peut être plus ou moins bien pris en charge par le modèle générant la posture. Or comme nous l'avons vu au paragraphe 1.3.2.1, la mesure du réalisme d'une posture est un problème difficile, et pour juger du réalisme, l'utilisateur doit se contenter de son propre jugement, au vu de la posture obtenue. Pour certains modèles, on pourrait cependant imaginer un degré de confiance à accorder à la posture obtenue, selon ce que le concepteur du modèle d'opérateur sait à propos des capacités de son modèle et selon les situations. Les problèmes de calcul de postures (pour un test d'atteinte par exemple) peuvent en effet être définis a priori comme étant plus ou moins durs à traiter, selon la localisation du point à atteindre, le nombre et la nature des segments corporels à utiliser, ...

Ce besoin de qualifier la posture obtenue concerne aussi le caractère plausible ou non de la posture simulée, le confort associé, le fait que la posture soit ou non adaptée à l'activité simulée. Sur cet aspect, il est possible de développer (comme nous l'avons évoqué plus haut) des modules d'aide au réalisme des postures, par exemple pour évaluer l'équilibre du corps pour une posture et proposer des corrections, ou bien pour évaluer les capacités d'effort à partir d'une posture donnée.

Il conviendrait donc de distinguer deux niveaux de qualification de la posture simulée : un premier niveau associé au degré de validité supposé de la posture simulée, lié aux limites du modèle mais aussi au cas particulier traité, et un autre niveau qualifiant la posture obtenue, en terme de confort par exemple. Ce deuxième niveau permet d'aboutir à un diagnostic ergonomique lié à l'interaction avec la scène : par exemple, un test d'atteinte aboutissant à une posture confortable permet de valider la position de l'élément de la scène testé.

1.3.3 Les modèles graphiques d'opérateur humain

Il existe de nombreux modèles similaires à MAN3D, permettant la représentation graphique d'un opérateur humain et de ses capacités motrices. On trouvera en annexe 1 une liste de modèles décrits dans la littérature du domaine, ainsi que quelques éléments sur l'origine et la nature de chacun d'eux. On en dénombre une vingtaine, dont la moitié aux Etats-Unis et seulement deux en France (dont MAN3D). Ces modèles sont principalement utilisés pour la conception de véhicules (12). La plupart sont conçus sur la base d'une chaîne polyarticulée plus ou moins complexe, qui modélise le squelette de l'homme. Des paramètres de forme sont ajoutés à cette structure, afin de représenter le corps. Ils proposent aussi des outils de manipulation de la structure à un niveau de base, et des commandes plus élaborées pour effectuer des simulations de mouvements d'atteinte et quelques tests ergonomiques.

Tous ces modèles ne fournissent pas tous le même niveau de réalisme. En ce qui concerne la représentation du corps humain, ils diffèrent selon :

- la complexité du squelette polyarticulé (le nombre de segments),
- le réalisme des contours (et le mode de calcul associé),
- la forme et la nature des données anthropométriques disponibles,
- la possibilité d'accéder à différentes morphologies prédéfinies,
- la possibilité de modifier individuellement les segments corporels (et la présence d'un éventuel contrôle sur ces modifications).

En ce qui concerne la manipulation du mannequin, la détermination de la posture et le degré d'automatisation, on ne trouve pas de modèle optimal sur ces critères. On trouve des modèles imposant par exemple à l'utilisateur d'indiquer les valeurs d'angle du modèle, ou bien de spécifier quels sont les angles à libérer pour une simulation. Le modèle le plus avancé sur ces aspects semble être le modèle Jack (11), que nous avons déjà évoqué plus haut.

La plupart des modèles comportent des fonctionnalités de base liées au diagnostic ergonomique. Elles sont plus ou moins automatisées. Pour les tests d'atteinte, certains modèles n'offrent que le test simple d'atteinte d'un point. D'autres proposent la production et la visualisation de volumes d'atteinte. En ce qui concerne l'interaction entre le mannequin et la scène, l'utilisateur est le plus souvent chargé de l'inspection visuelle. On trouve cependant des routines automatiques de détection d'interférences. Pour le champ de vision, les modules offerts vont du simple affichage de la localisation du regard, à l'affichage en perspective des vues depuis le mannequin. Certains modèles gèrent aussi la rétrovision.

Enfin, en ce qui concerne l'architecture informatique, les modèles différent selon qu'ils sont ou non couplés avec un logiciel de CAO et avec des bases de données anthropométriques. L'approche de conception utilisée est avant tout numérique. Les outils et langages utilisés sont ceux de l'informatique classique.

1.4 Conclusion

1.4.1 La théorie, l'offre et la demande, pour la simulation ergonomique

Nous avons décrit dans la section 1 de ce chapitre (1.1) la démarche de simulation ergonomique dans le cadre des projets de conception des systèmes de travail. Cette démarche consiste à simuler les conditions de réalisation du travail futur, ainsi que l'activité exercée par un opérateur virtuel représentant le futur opérateur réel. Cette simulation a pour objectif de fournir une aide à la conception des produits et espaces de travail, dans le but d'adapter le mieux possible ces derniers à leurs futurs utilisateurs.

La démarche repose sur la définition de 4 modèles :

- un modèle de l'environnement de travail,
- un modèle de l'opérateur,
- un modèle fournissant des scénarios de fonctionnement pour l'opérateur
- un modèle de la dynamique du système opérateur/environnement.

Dans la pratique, les modèles et outils utilisables pour la simulation ergonomique ne couvrent pas l'étendue de cette démarche ; ils sont partiels et spécifiques. Les modèles de simulation de l'activité n'intègrent pas de modèle d'opérateur. Par ailleurs, les modèles d'opérateur sont limités à des approches anthropométriques et biomécaniques, qui n'intègrent pas le concept d'activité dans sa globalité puisqu'elles s'arrêtent au niveau de la posture et des gestes.

La réalité au niveau des demandeurs potentiels de ce type d'outils pour la simulation ergonomique est encore autre. En effet, comme nous l'avons montré dans la section 2, la logique de conception des produits obéit à des objectifs liés au contexte industriel, laissant peu de place à l'intervention ergonomique, qui est loin d'être prioritaire.

Les modèles de l'homme qui ont été développés pour être utilisés dans les processus de conception industriels ont été conçus pour s'adapter à ce type de contexte ; ils sont utilisables directement avec les outils des concepteurs (les logiciels de CAO) et sont donc les mieux placés en tant qu'outils d'aide à l'intervention ergonomique en conception de produits. Par ailleurs, les efforts de développement de ces modèles ont été dictés par la demande : c'est ainsi que pour la conception des véhicules, les efforts ont plutôt porté sur la modélisation des aspects morphologiques et posturaux, tandis que pour l'évaluation des tâches de manutention, l'attention a été portée sur la modélisation des capacités d'effort.

En première conclusion, il apparaît donc qu'il existe un manque d'outils intégrés, réunissant les différents composants utiles à la simulation ergonomique. Pour être pertinents dans le contexte industriel, de tels outils devraient conserver le point d'entrée dans le processus de conception que constitue le couplage avec un logiciel de CAO.

1.4.2 Les limites des modèles d'opérateur humain pour l'analyse ergonomique

Bien que nés pour la plupart de la demande industrielle (industrie automobile et aérospatiale), et donc particulièrement adaptés aux besoins (à la différence de modèles à motivation de départ plus théorique), les modèles graphiques d'opérateur humain sont encore limités dans leur capacité à fournir une aide à l'évaluation ergonomique en CAO.

Tout d'abord, bien que s'adressant à des utilisateurs spécialistes de la manipulation d'objets graphiques, ils sont encore trop lourds à manipuler. Peu de modèles offrent de bonnes fonctionnalités de manipulation. D'ou la nécessité accrue de pouvoir proposer des fonctions de stockage/récupération de postures, et même d'enregistrement/reproduction de séquences de manipulation, ce qui n'est pas le cas pour la majorité des modèles. Mis à part quelques modèles tels que Jack ou SAFEWORK (13), qui proposent la manipulation directe faisant se mouvoir la totalité du corps, les modèles proposant les fonctionnalités les plus évoluées de manipulation s'arrêtent au niveau "membre" : l'utilisateur doit prendre en charge le niveau stratégique de choix du membre à activer et d'enchaînement de ces différents choix, pour une simulation d'atteinte.

Par ailleurs, nous avons vu que même lorsque l'utilisateur a franchi le cap de la manipulation, il reste la question de la validité de la posture obtenue. L'un des problèmes est celui des déterminants de la posture pris en compte par ces modèles. Par exemple, si l'on prend le cas où le calcul de la géométrie posturale à partir de la donnée d'un point cible s'effectue sur la base de la recherche du moindre écart par rapport à une posture de référence (posture de moindre inconfort), on sait que le résultat n'est pas valide dans tous les cas. En effet, certaines zones d'atteinte fournissent des résultats de postures apparemment incongrus, bien que les limites articulaires soient respectées. L'adoption d'une posture par un opérateur obéit donc à des règles qui ne sont pas toutes prises en compte par ces modèles.

Au-delà de la validité des résultats fournis par le modèle, un autre problème est celui de la qualification de la posture obtenue : les modèles ne fournissant pas d'autre résultat associé à la posture que la représentation graphique, rien ne permet de décider si la posture obtenue est confortable par exemple. Il n'existe pas de moyen de comparer différentes postures et au final l'utilisateur ne peut que tester la faisabilité de l'atteinte d'une cible. Rien ne lui permet de

décider laquelle de deux localisations accessibles est la meilleure, si ce n'est son propre jugement.

De manière générale, ces modèles ne prennent pas suffisamment en compte les paramètres liés au contexte de la tâche simulée. Citons ici un exemple de problème évoqué par un utilisateur de MAN3D, pour la conception d'habitacles de véhicules légers : le problème était de trouver l'écartement optimal entre les pédales ; rien ne permet de décider, au niveau du mannequin graphique, le bon écart entre les genoux. Ceci est dû tout d'abord à la question de la manipulation, évoquée plus haut : le système ne prend pas en charge la gestion simultanée des deux jambes. De plus, le modèle ne prend pas en compte la spécificité de la tâche simulée : on peut supposer que le bon écartement des genoux en position assise sera différent, selon que l'on s'intéresse à une activité de bureau ou à l'activité de conduite d'une voiture, ou à celle d'un poids-lourd. MAN3D est donc limité dans la mesure où il impose à l'utilisateur une expertise pour la prédiction de la posture. Son utilisation impose aussi une véritable expertise de manipulation.

Par ailleurs, nous avons évoqué dans le paragraphe 1.3.1.6 , le fait que MAN3D continue à faire l'objet de développements. L'un de ces développements concerne la production d'un algorithme de gestion des contraintes multiples pour les calculs de postures. L'approche algorithmique classique atteint ici ses limites, dans la mesure où le traitement de ce problème impose l'introduction dans le code de l'ensemble des cas de figure à traiter, sous formes d'instructions conditionnelles imbriquées de plus en plus complexes. Par contre, on rentre ici dans le champ des problèmes traités plus facilement avec les techniques d'Intelligence Artificielle.

Ces remarques soulignent quelques problèmes posés par ce type de modèles (et par MAN3D en particulier), qui sont à la base des objectifs que nous exposons dans le chapitre 3.

2 Paradigmes IA pour la simulation ergonomique

Ce chapitre est consacré à un bilan sur les outils de l'IA utiles dans le contexte de la simulation de l'activité de l'opérateur pour le diagnostic ergonomique. Nous évoquons tout d'abord les approches à bases de connaissances que nous avons trouvées dans la littérature du domaine (section 2.1), puis nous rappelons dans la section 2.2 quelques notions sur les systèmes multi-agents, qui est le domaine de l'IA auquel se rattache le paradigme que nous avons utilisé (le modèle de tableau noir), avec une description plus précise du système ATOME, qui est l'outil dont nous avons pu disposer.

2.1 Approches à bases de connaissances pour l'aide ergonomique à la conception

2.1.1 Revue des systèmes existants

Nous nous sommes intéressés aux approches à bases de connaissances pour la modélisation de l'homme et la conception des espaces de travail.

Järvinen et Karwowski ont recensé les applications des systèmes experts en ergonomie dans l'industrie (14). On trouve parmi ces applications quelques systèmes d'aide à la conception et à l'évaluation des espaces de travail. Le système ERGOSPEC (15) génère la liste des caractéristiques du poste de travail à partir des spécifications de l'utilisateur (qui doit fournir au minimum le sexe et la taille de l'opérateur). Le système ERGON-EXPERT (16) propose l'analyse et l'évaluation de séquences de travail pour des tâches de manutention. Jung et Freivalds (17) ont conçu un système expert pour aider l'utilisateur à définir les spécifications de conception appropriées pour une tâche donnée. Ces trois systèmes sont exclusivement fondés sur une approche à bases de connaissances ; ils ont une fonction d'assistance à la conception mais ne comportent pas de modèle de l'opérateur et ne permettent pas la simulation de l'activité. Nous détaillons le contenu des bases de connaissance de ces systèmes dans le paragraphe suivant (2.1.2).

Une autre façon d'explorer le domaine est de s'intéresser aux utilisations de techniques d'intelligence artificielle pour les systèmes basés sur des modèles graphiques d'opérateur humain tels que ceux évoqués au chapitre 1 (section 1.3), qui fournissent une aide à la conception des espaces de travail en simulant l'opérateur et ses gestes. La plupart de ces systèmes sont avant tout développés avec des techniques d'informatique classique. Les systèmes COMBIMAN (un modèle de pilote d'avion) et CREWCHIEF (un modèle de technicien de maintenance pour les avions) sont présentés comme des systèmes experts qui permettent à l'utilisateur d'obtenir une expertise ergonomique à partir de la définition de la tâche à accomplir (18). Mais l'auteur ne donne pas d'explication sur la manière dont ces systèmes sont conçus, pas plus que sur leurs architectures ou sur la nature de la connaissance experte intégrée, ni sur la manière dont cela est fait. Les seules techniques évoquées sont les équations de régression et les analyses multi-variées pour le dimensionnement du modèle humain. Il

semble que le terme de système expert est un peu abusif dans ce cas, ou bien que l'auteur ne veut pas dévoiler les techniques qui ont été utilisées.

Le système SAFEWORK (13) comporte un module d'analyse biomécanique composé de 14 modèles mathématiques gérés par un petit système expert, qui conduit l'utilisateur, à l'aide de questions simples, au choix du modèle approprié à la situation.

Le système SAMMIE (System for Aiding Man-Machine Interaction Evaluation) a été interfacé avec un système expert qui fournit à l'utilisateur une fonction de contrôle et de conseil, qui lui évite d'avoir a contrôler en détail les dimensions importantes pour les règles de conception (19).

Dans une publication plus récente de l'équipe développant le système Jack (11), on trouve l'évocation d'une base de connaissances utilisée pour fournir une description symbolique de l'environnement d'animation, en complément de sa description géométrique. Cette description des objets comporte ainsi des attributs tels que : le nom de l'objet, sa localisation (par exemple *tableau de bord*), son type (par exemple *commutateur*), son état (*on* ou *off*), ... Il s'agit apparemment plutôt d'une base de données.

Il existe donc dans la littérature portant sur les modèles graphiques d'opérateur humain, quelques cas peu documentés d'usage d'une approche à bases de connaissances, en complément d'une modélisation centrale s'appuyant sur des techniques de nature numérique. Par ailleurs, les systèmes experts pour l'aide à la conception des espaces de travail, conçus avec une approche de nature symbolique, ne sont pas connectés à un modèle de l'homme.

2.1.2 La connaissance experte dans les systèmes d'aide à la conception ergonomique

Nous nous intéressons ici à la nature de la connaissance experte intégrée dans les systèmes d'aide à la conception ergonomique évoqués au paragraphe précédent (2.1.1).

Le système ERGOSPEC (15) est utilisé pour obtenir la spécification d'un poste de travail idéal, à partir d'une description initiale simple du poste. Le parcours de la base de règles est effectué en chaînage avant. Le système comporte une base de données externe, qui contient des données anthropométriques, les dimensions optimales pour le siège et la table de travail, les caractéristiques des cadrans et les distances optimales de lecture correspondantes, les valeurs de capacités musculaires maximales pour les hommes et les femmes, les capacités d'effort maximales pour différentes postures de travail, ainsi que quelques recommandations ergonomiques. La base comporte 18 règles : deux règles lient le sexe et la taille de l'opérateur à diverses dimensions corporelles ainsi qu'à des paramètres de conception du siège et du poste de travail. Un autre groupe de règles se rapporte à la posture de l'opérateur lorsqu'il réalise des opérations nécessitant l'application d'une force musculaire, des opérations manuelles ou des tâches visuelles. Ce groupe permet aussi de déduire les capacités d'atteinte pour un profil d'opérateur donné, ainsi que les hauteurs acceptables pour les rangements. Cinq autres règles concernent l'environnement auditif et les messages sonores. Ces règles spécifient la durée et l'intensité maximales du bruit ainsi que les caractéristiques des messages sonores appropriés, pour différents types de signaux auditifs. Il y a aussi une règle concernant l'environnement visuel, qui spécifie les valeurs de luminance ainsi que les caractéristiques des afficheurs pour différents types de tâches. Deux autres règles se rapportent à la sélection et à

la localisation des commandes pour les opérations manuelles ainsi que les opérations impliquant l'usage des pieds. Enfin, les autres règles se rapportent à la communication entre les opérateurs et à la détermination des périodes de repos selon le type de travail effectué par l'opérateur. Sur la base des spécifications données par l'utilisateur, le système fournit des recommandations incluant les dimensions appropriées de l'espace de travail, les caractéristiques de conception des afficheurs et des commandes et les valeurs des paramètres environnementaux. Ces différents résultats sont fournis en format *texte*.

Le système ERGON-EXPERT (16) est un outil destiné à identifier les risques sur la santé liés à des tâches de manutention, par une analyse détaillée ainsi qu'une évaluation de séquences de ce type de tâches. Il analyse la conformité de la séquence de travail spécifiée avec les lois, les règlements, et les normes sur la sécurité et la santé pour les activités professionnelles.

Ces deux systèmes illustrent l'aspect multifactoriel de la connaissance utilisée pour l'évaluation ergonomique. La connaissance experte utilisée dans ce contexte provient de domaines différents et est divisée naturellement en modules spécifiques. Pour le système ERGON-EXPERT, Rombach et Laurig soulignent aussi le besoin de meta-règles permettant de décrire les relations entre les modules ainsi que leur hiérarchie. La question des éventuels conflits pouvant survenir entre les résultats produits par les différentes sources de connaissance n'est pas abordée. On peut cependant supposer que dans ce domaine il peut survenir des cas de conflits entre des règles applicables et fournissant des résultats contradictoires.

Jung et Freivalds (17) décrivent en effet la conception ergonomique d'un espace de travail comme un problème de formulation multi-critères et de prise de décision concernant des objectifs multiples pouvant être conflictuels. Une expertise est donc requise, à la fois pour sélectionner les classes de connaissances concernées et pour les combiner à l'aide de jugements heuristiques, selon leur importance. L'objectif des auteurs était de construire une architecture à bases de connaissances pour la conception et l'évaluation d'espaces de travail pour des tâches de manipulation de charges. Le système proposé comporte :

- une base de données pour les dimensions humaines,
- une base de connaissances séparée, contenant de l'expertise ergonomique sous forme de connaissance déclarative exprimant notamment des relations entre variables ergonomiques,
- de la connaissance à propos des règles et des méthodes pour trouver les faits pertinents,
- un mécanisme d'inférence permettant de fournir un raisonnement proche de celui d'un expert ; il comporte des heuristiques, des mécanismes de recherche et des stratégies de contrôle, et permet de guider l'utilisateur pour atteindre la solution optimale ou au moins une solution satisfaisante, en utilisant soit de l'information fournie par l'utilisateur, soit le raisonnement par défaut.

Un aspect intéressant de l'architecture conçue est son aspect fortement modulaire. La connaissance déclarative est stockée dans différents sous-modules selon le type d'information. Lorsque le module contient une grosse masse de données (comme les données anthropométriques), ces données sont stockées séparément dans une base de données. Les modèles analytiques comportant des calculs mathématiques sont implémentés en Fortran, dans des modules à part.

Dans la description du système SAMMIE (19), les auteurs soulignent le fait que l'approche système expert est particulièrement appropriée au contexte de la conception automobile, dans

la mesure où la résolution des problèmes s'appuie sur un mélange de lois, de normes, de pratiques, de principes ergonomiques, ainsi que de subtilités diverses provenant des besoins du marché et de l'image du produit. En pratique, l'interfaçage d'un modèle d'opérateur humain avec un système expert dans un tel contexte pose quelques problèmes. Tout d'abord, la connaissance intégrée dans la base est spécifique au domaine d'application (c'est-à-dire dans ce cas, à la disposition des différents équipements et commandes dans l'habitacle), ce qui oblige à redévelopper une nouvelle application si l'on change de domaine. Un autre problème important est lié à la connaissance de nature géométrique qui doit être intégrée dans le système expert. La solution consistant simplement à dupliquer le modèle géométrique dans la base de connaissance n'est pas intéressante en terme d'efficacité. Elle ne résout pas non plus les problèmes liés à la modélisation géométrique de l'espace, auxquels les modèles graphiques d'opérateur couplés avec les logiciels de CAO ne répondent pas. En effet, les questions simples liées aux dimensions des objets ou aux distances les séparant ne posent pas de problèmes, mais les questions qui requièrent une capacité de raisonnement géométrique dépassent les capacités de ces modèles. Par exemple, ces systèmes ne peuvent pas répondre à une question de nature géométrique comme « est-ce que le volant obscurcit une partie du compteur de vitesse et en gêne la vision ? ».

En conclusion on peut retenir que la connaissance experte nécessaire à l'évaluation ergonomique a un caractère distribué et qu'il est nécessaire de définir des niveaux stratégiques permettant le contrôle des différents modules de cette connaissance. Par ailleurs, le couplage entre modèle graphique d'opérateur et approche à base de connaissances peut poser certains problèmes. L'un de ces problèmes est la modélisation symbolique de l'espace géométrique, afin de pouvoir effectuer des raisonnements de nature géométrique.

2.2 Les systèmes multi-agents

Nous avons vu dans la section précédente (2.1), que l'expertise nécessaire à l'évaluation ergonomique pour la conception des produits tels que les habitacles de véhicules a un caractère multifactoriel et distribué. Or, il existe en Intelligence Artificielle un paradigme spécifiquement ciblé sur les problèmes de nature distribuée : c'est le paradigme multi-agents. Nous fournissons quelques précisions terminologiques tout d'abord (2.2.1), puis nous rappelons les fondements théoriques des systèmes multi-agents (2.2.2) avant d'évoquer des démarches multi-agents appliquées à des domaines connexes de celui que nous étudions (2.2.3).

2.2.1 Quelques précisions terminologiques

2.2.1.1 IA et systèmes à bases de connaissances

Les systèmes d'IA se caractérisent pour la plupart par une distinction entre les données, les opérations et le contrôle. Parmi ces systèmes, les systèmes à règles de production sont constitués de trois éléments principaux qui sont : une base de données globale ou base de faits, un ensemble de règles de production et une stratégie de contrôle. Les règles de production agissent sur la base de faits. Chaque règle a des préconditions qui sont ou ne sont pas satisfaites par la base de faits. Si les préconditions sont satisfaites, la règle peut être appliquée en chaînage avant. Cette application change la base de faits. La stratégie de contrôle choisit la règle applicable qui doit être appliquée et le processus se termine dès qu'une

condition d'arrêt est satisfaite par la base de faits. La connaissance liée au contrôle comprend des connaissances sur les divers processus, stratégies et structures utilisés pour coordonner le processus de résolution du problème. Le paradigme du système à règles de production a donné naissance au cas particulier des systèmes experts. L'ensemble des systèmes d'IA issus de ce paradigme (dont les systèmes multi-agents) est aussi connu sous le terme *systèmes à bases de connaissances*.

2.2.1.2 Intelligence artificielle distribuée et systèmes multi-agents

L'Intelligence Artificielle Distribuée (IAD) et les Systèmes Multi-Agents (SMA) constituent actuellement un domaine actif de l'IA. On peut citer notamment sur le sujet les conférences francophones JFIADSMA (Journées Francophones Intelligence Artificielle Distribuée et Systèmes Multi-Agents) ainsi que les conférences internationales ICMAS (International Conference on Multi-Agent Systems).

En 1980, Nilsson consacrait une petite section de son chapitre « perspectives de l'IA » aux systèmes parallèles et distribués (d'après la traduction française (20)). Parmi les utilisations du parallélisme, il présentait une approche visant à « imaginer une grande communauté de systèmes plus ou moins indépendants ». Ces systèmes communiqueraient entre eux dans le but de résoudre des problèmes en commun, ce qui impliquerait alors la spécification détaillée des protocoles de communication et des procédures de contrôle et de coopération. Dans le cas où chacun des systèmes serait lui-même un système d'IA complexe, alors la situation serait analogue à une société d'êtres humains ou d'autres êtres vivants évolués qui doivent planifier leurs propres stratégies de communication et de coopération. Cette description recouvre les différents travaux présentés aujourd'hui sous le vocable *systèmes multi-agents* se rapportant à différentes compositions d'agents plus ou moins "intelligents" et plus ou moins autonomes, œuvrant ensemble avec des buts communs.

Dans (21) les auteurs indiquent que la résolution distribuée de problèmes remonte au milieu des années 1970, avec d'une part les langages d'acteurs (22) et d'autre part le modèle d'architecture de tableau noir (23). Ils notent aussi que le concept d'IA distribuée recouvre la conception de systèmes multi-agents, mais aussi deux autres domaines :

- la parallélisation de systèmes d'IA visant à améliorer les performances des systèmes existants (utilisant les techniques et les modèles mis au point en informatique dans le domaine des algorithmes et des machines parallèles),
- ainsi que la distribution totale des connaissances et du raisonnement sur un très grand nombre de processeurs élémentaires fortement connectés, dotés chacun de capacités de traitements très limitées (l'intelligence naissant ainsi de la mise en commun de la multitude de traitements locaux).

Erceau et Ferber (24) présentent la démarche de l'IAD comme une réponse aux problèmes apparaissant avec une démarche IA "classique" s'appuyant sur une centralisation et une concentration de l'" intelligence" au sein d'un système unique. Ces problèmes sont liés aux difficultés d'intégrer dans une même base de connaissances, l'expertise, les compétences et les connaissances d'individus différents qui, dans la réalité communiquent et collaborent à la réalisation d'un but commun. Cette démarche conduit à la réalisation de systèmes multi-agents, « permettant de modéliser le comportement d'un ensemble d'entités plus ou moins expertes, plus ou moins organisées selon des lois de type social.».

2.2.1.3 Systèmes multi-agents et modèle de tableau noir

Une définition très générale des systèmes multi-agents peut être la suivante : les systèmes multi-agents sont des systèmes d'IA composés de plusieurs entités nommées agents, participant à la résolution d'un problème. Ces systèmes se différencient globalement selon la nature des agents, le mode de communication entre ces agents, et leur degré d'autonomie (qui rejoint la question du contrôle de la résolution). Dans ce cadre, les architectures multi-agents à base de tableau noir correspondent au cas où la communication est effectuée par l'intermédiaire d 'une zone commune : le tableau noir. Le modèle de tableau noir constitue donc une classe des systèmes multi-agents.

Dans (21) les auteurs distinguent en effet trois classes de systèmes multi-agents : les systèmes à tableau noir, les systèmes d'acteurs, et les systèmes physiquement distribués. Dans les systèmes à tableau noir, les éléments de la solution du problème sont constamment mémorisés dans le tableau noir, tandis que dans les systèmes d'acteurs, les éléments de la solution sont distribués entre les différents agents, chacun d'entre eux communicant avec les autres par envoi de messages. Les systèmes physiquement distribués sont composés d'agents plus complexes, dotés de capacités de raisonnement élaborées. Chaque agent est autonome et capable de résoudre seul le problème qui lui est affecté, tout en ayant le moyen de coopérer avec les autres.

Le statut du système à tableau noir vu comme un système multi-agents a évolué ces dernières années, comme on va le voir ci-dessous.

En 1988, Ferber et Ghallab (25) fournissaient une définition du terme d'agent, qui leur permettaient de distinguer deux paradigmes multi-agents différents :

> « On appelle agent une entité physique ou abstraite qui est capable d'agir sur elle-même et son environnement, qui dispose d'une représentation partielle de cet environnement, qui, dans un univers multi-agents, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de sa connaissance et des interactions avec les autres agents. »

A partir de cette définition, Ferber et Ghallab distinguaient deux classes de problèmes :

> « (i) - l'agent est une entité physique autonome, percevant et agissant sur l'environnement extérieur. Le paradigme est celui d'un robot dans un environnement multi-robots. L'agent possède une autonomie d'action, et donc de décision. L'interaction avec d'autres agents se fait au travers de systèmes de perception et de communication.
>
> (ii) - l'agent est une entité informatique (agent "computationnel") : il s'agit d'un module logiciel (et/ou matériel) plus ou moins élaboré interagissant avec d'autres modules selon une architecture particulière. Le paradigme est celui des systèmes multi-experts, ou de l'Intelligence Artificielle Distribuée en général. Chaque agent est une base de connaissance ou une procédure spécialisée dans une tâche qui lui est propre. »

Pour la classe (i), les notions d'action sur le monde extérieur et d'interaction entre agents sont premières, tandis que pour la classe (ii), ce sont des soucis de réduction de complexité, et de critères de modularité, de performances et de fiabilité qui conduisent à privilégier une approche multi-agents.
La classe (i) correspond au paradigme multi-robots, la classe (ii) au paradigme multi-experts.
Le modèle de tableau noir était alors présenté comme relevant du paradigme multi-experts.

Si l'on se réfère à cette définition, dans les travaux les plus récents de la communauté multi-agents française réunis sous le vocable multi-agents, il semble que cela soit plus le paradigme multi-robots qui soit privilégié. C'est le cas notamment pour les travaux évoqués dans le « Dossier IAD SMA » paru dans un récent numéro du bulletin de l'AFIA[4] (26). Cependant, les frontières entre les deux paradigmes de cette définition sont aujourd'hui plus floues.

Dans ce contexte, Ferber (27) souligne que les systèmes à base de tableaux noirs ne sont plus considérés aujourd'hui comme des systèmes d'IAD composés d'agents interagissant entre eux. Par contre, ces systèmes sont maintenant envisagés comme des architectures pratiques pour la réalisation de systèmes intelligents, au sens où ils constituent de bons outils de prototypage de systèmes. En effet, du fait de son caractère opportuniste et centralisé, ce type d'architecture procure une très bonne souplesse pour décrire des modules et articuler leur fonctionnement. La gestion fine du contrôle et le fait que celui-ci soit très explicite implique par contre une relative inefficacité (qui n'est pas un obstacle en phase de prototypage). Dans ce contexte, Ferber affirme qu'il est possible d'implémenter n'importe quelle structure d'agent en termes d'éléments de tableaux noirs et de sources de connaissances, ce qui lui permet de conclure que « le tableau noir se présente comme une sorte de méta-architecture, c'est-à-dire une architecture pour implémenter des architectures. »

2.2.2 Fondements théoriques

2.2.2.1 Systèmes multi-agents basés sur le paradigme multi-experts

Malgré l'évolution des définitions, la problématique des systèmes multi-agents au sens du paradigme "multi-experts" évoqué au paragraphe précédent (2.2.1.3) nous semble continuer à constituer une bonne base pour introduire les concepts manipulés.

Dans ce cadre, les systèmes multi-agents répondent au besoin de distribution de l'intelligence, soit pour la résolution d'un problème nécessitant la mise en commun de connaissances détenues par différents spécialistes, soit parce que le problème à résoudre peut être envisagé sous divers aspects (28). Les entités informatiques correspondantes (les agents) peuvent prendre la forme soit d'une procédure codant un algorithme, soit d'une base de connaissance avec son mécanisme de raisonnement associé.

La conception de ces systèmes pose principalement des problèmes de communication entre les agents, d'organisation générale et de définition des mécanismes de contrôles chargés de guider la résolution du problème.

Il existe deux grands mécanismes de communication (28) : le partage d'informations (communication implicite) et l'envoi de messages (communication explicite). Cette distinction est importante puisqu'elle conduit à deux types d'architecture de systèmes multi-agents.

Dans le cas du partage d'informations, les agents communiquent par l'intermédiaire d'une structure de données commune. Cette structure a une autre fonction puisqu'elle correspond en fait à la base de faits globale (au sens des systèmes experts), qui contient initialement les données et les faits décrivant un problème et qui s'enrichit progressivement des résultats

[4] Association Française pour l'Intelligence Artificielle

partiels jusqu'à contenir la solution globale du problème posé. Dans ce cas, les agents n'ont pas à se connaître, c'est en ce sens que la communication est implicite. Ce schéma de communication / gestion de la solution correspond typiquement au modèle de tableau noir (blackboard) que nous évoquerons plus en détail au paragraphe 2.2.2.2 .

Le cas de la communication par envoi de messages entre les différents agents nécessite la définition de protocoles d'échange. Dans ce cas, la communication est donc explicite entre l'agent émetteur et l'agent récepteur. Ce mode d'échange correspond par ailleurs au cas où tous les éléments de la solution sont distribués et se retrouvent au niveau de chaque agent : les connaissances, les techniques de résolution, les faits, ainsi que les résultats partiels. L'approche objets correspond à ce cas de figure, bien qu'elle ne soit généralement pas présentée en référence à la thématique des systèmes multi-agents.

Une fois résolues les questions du mode de communication ainsi que de la localisation des connaissances et des éléments de solution dans le système, reste la question du contrôle de la résolution. Il s'agit de décider de quelle manière les agents vont concourir ou coopérer pour la résolution du problème et d'assurer la synthèse et la cohérence des résultats.

2.2.2.2 Modèle de tableau noir

Nous l'avons introduit dans les paragraphes précédents : le modèle de tableau noir (blackboard), est donc un cas particulier de système multi-agents, basé sur le paradigme multi-experts, fonctionnant avec le principe de la communication entre les agents par partage de zone commune.

Le concept

Une image parlante pour décrire ce type de fonctionnement est celle du groupe d'experts, réunis pour construire ensemble un puzzle. Ils disposent en commun de l'affichage du cadre du puzzle (le tableau noir), mais n'ont pas le droit de communiquer. Chaque fois que l'un des experts pense pouvoir placer une pièce, il intervient dans la résolution.

Cette image correspond à une version simplifiée du modèle de tableau noir, caractérisée par trois éléments : la structure de données commune (le tableau noir), les experts (les sources de connaissance), ainsi que le processus de conduite de la résolution.

- le tableau noir est une structure de données qui contient l'état de la résolution du problème,
- les sources de connaissances sont indépendantes les unes des autres, elles contiennent les méthodes de résolution (expertise de résolution), elles interviennent dans la résolution par modification de l'état du tableau noir. Dans le cas de ce modèle simplifié, elles décident seules de leur intervention dans la résolution.
- le processus de conduite est coopératif (car les sources de connaissances coopèrent), il est opportuniste (car c'est l'évolution de l'état du tableau noir qui guide le processus de résolution).

Ce modèle simplifié peut être complété par l'ajout d'autres caractéristiques, liées d'une part au mécanisme de contrôle de la résolution, d'autre part à la structuration du tableau noir.

Un mécanisme de contrôle peut être réalisé par l'ajout d'un moniteur, chargé de gérer l'intervention des sources de connaissances. Ce moniteur réalise un cycle ; à chaque cycle il examine les sources de connaissances et fait la liste de celles qui souhaitent intervenir dans la résolution. Puis il choisit dans cette liste la source de connaissances qui va effectivement intervenir. Enfin, il fait intervenir cette source en lui donnant un ordre de déclenchement. Le processus de résolution est alors toujours coopératif et opportuniste (guidé par l'état du tableau noir), mais le contrôle de la résolution est ici centralisé dans le moniteur. Le processus de résolution se termine lorsqu'il n'existe plus de source de connaissances pouvant intervenir ou bien lorsqu'une condition d'arrêt (gérée par le moniteur) est vérifiée.

Le modèle de tableau noir peut aussi être complété par la structuration du tableau noir en niveaux composants la solution. La construction de la solution se fait de manière incrémentale, par ajouts de faits d'un niveau à partir de faits élémentaires du niveau précédent. Un exemple de décomposition en niveau est celui des applications de reconnaissance vocale, pour lesquelles on part du son (niveau le plus bas du tableau noir), pour arriver à la phrase entière (niveau le plus haut), en passant par des niveaux intermédiaires correspondant aux syllabes, aux mots, et aux séquences de mots. Cela nécessite que chaque source de connaissance soit spécialisée dans le passage d'un niveau à un autre. Ainsi, chaque source de connaissance émettra des hypothèses c'est-à-dire qu'elle ajoutera des faits dans un niveau, faits qu'elle aura construits à partir du niveau inférieur. Une telle décomposition en niveaux conceptuels peut être faite dans la plupart des domaines d'activité.

L'origine des systèmes de tableau noir

Le premier domaine dans lequel on a utilisé le modèle de tableau noir est celui de la reconnaissance vocale. Les travaux réalisés dans ce domaine à Carnegie-Mellon University ont aboutit au système Hearsay (23). D'après Carver et Lesser (29), ce sont les difficultés du problème traité et sa nature même (la reconnaissance vocale) qui ont conduit à développer une architecture de tableau noir. En effet, ce type d'application est caractérisé par la taille importante de l'espace de recherche, le caractère incomplet et/ou erroné des données d'entrée, ainsi que le caractère imprécis et/ou incomplet de la connaissance utile à la résolution du problème. Il est aussi caractérisé par la nécessité de faire intervenir des agents bien identifiés : acoustique, phonétique, lexical, syntaxique, etc. . Ces caractéristiques imposent un modèle de résolution de problème qui permette le développement incrémental de la solution ainsi que l'utilisation de différentes expertises, et qui puisse adapter la stratégie de résolution en fonction du cas particulier traité. La flexibilité dans la structuration du processus de résolution, le développement incrémental de la solution du problème, et la modularité sont des caractéristiques importantes du modèle de tableau noir.

Principes du modèle de tableau noir

Le modèle de base de tableau noir est composé de trois éléments principaux :

- le tableau noir,
- un ensemble de sources de connaissances ou agents,
- un mécanisme de contrôle/conduite.

Le tableau noir est une mémoire globale, partagée par les différentes sources de connaissances, qui contient les données et les hypothèses (c'est-à-dire les solutions partielles potentielles). Le tableau noir possède une structure hiérarchique de niveaux. A chaque niveau

est associé une classe particulière d'hypothèses, chaque hypothèse étant par ailleurs liée à des hypothèses des autres niveaux.

Les différentes sources de connaissances contiennent la connaissance nécessaire à la résolution du problème. D'après l'état du tableau noir, elles créent de nouvelles hypothèses, modifient des hypothèses existantes ou les suppriment, quand cela se révèle approprié. Les sources de connaissance ont un format condition-action. La partie condition décrit une configuration particulière d'hypothèses du tableau noir, tandis que la partie action spécifie ce que fera la source de connaissance considérée si elle est activée. Par ailleurs, lorsqu'elle est activée, une source de connaissance engendre des événements, qui mémorisent la nature des derniers changements survenus sur le tableau noir (création, modification ou suppression d'hypothèses).

Les événements sont adressés au mécanisme de contrôle qui gère l'intervention des sources de connaissance. Il s'agit, au cours de la résolution, de choisir la source de connaissance à appliquer, mais aussi de décider sur quel aspect du problème la résolution doit être focalisée. Ces différentes tâches nécessitent la mise en œuvre de meta-connaissances de contrôle. Plusieurs solutions ont été réalisées pour implanter un mécanisme de contrôle. On peut distinguer trois grandes classes de solutions (28) :

- le contrôle procédural (celui du modèle initial du système Hearsay-II), est réalisé sous la forme de simples heuristiques permettant de déterminer au cours de la résolution la source de connaissance prioritaire parmi celles qui sont applicables.
- le contrôle hiérarchique correspond au cas où la connaissance de contrôle est intégrée dans une source de connaissance. Les connaissances d'expertise et de contrôle ont alors la même forme, mais l'ensemble des sources de connaissance est organisé dans ce cas de manière hiérarchique : au niveau le plus "bas" se trouvent les sources de connaissances contenant l'expertise sur le domaine et accédant donc au tableau noir pour résoudre le problème, tandis que les autres niveaux contiennent la métaconnaissance de contrôle qui se trouve donc elle-même plus ou moins hiérarchisée (selon le nombre de niveaux définis au-dessus des sources de connaissances expertes). Ce cas est celui du système ATOME, que nous avons utilisé dans ce travail, et que nous évoquerons plus loin.
- le contrôle par tableau noir a été initialement proposé dans le système BB1 (30), qui modifiait l'architecture de contrôle de Hearsay-II. L'idée est de traiter le problème du contrôle comme une tâche de résolution de problème et d'utiliser un modèle de tableau noir pour gérer la solution de ce problème. Cela revient à regrouper les sources de connaissances de contrôle et à les faire communiquer par un tableau noir de contrôle, distinct de celui sur lequel travaillent les sources de connaissances expertes sur le domaine.

2.2.3 Le système ATOME

2.2.3.1 Principe général

Nous décrivons ici le système ATOME[5] (31), (32), qui permet de développer des applications avec le modèle de tableau noir.

[5] ATOME a été développé par l'équipe RFIA du CRIN/INRIA de Nancy

ATOME est un système multi-agents à base de tableau noir, avec un mécanisme de contrôle de type hiérarchique. Il permet de définir trois types de sources de connaissances : les spécialistes[6], qui sont les sources de connaissances du domaine, les tâches et la stratégie, qui contiennent la connaissance de contrôle.

La Figure 10 décrit l'architecture générale de ATOME.

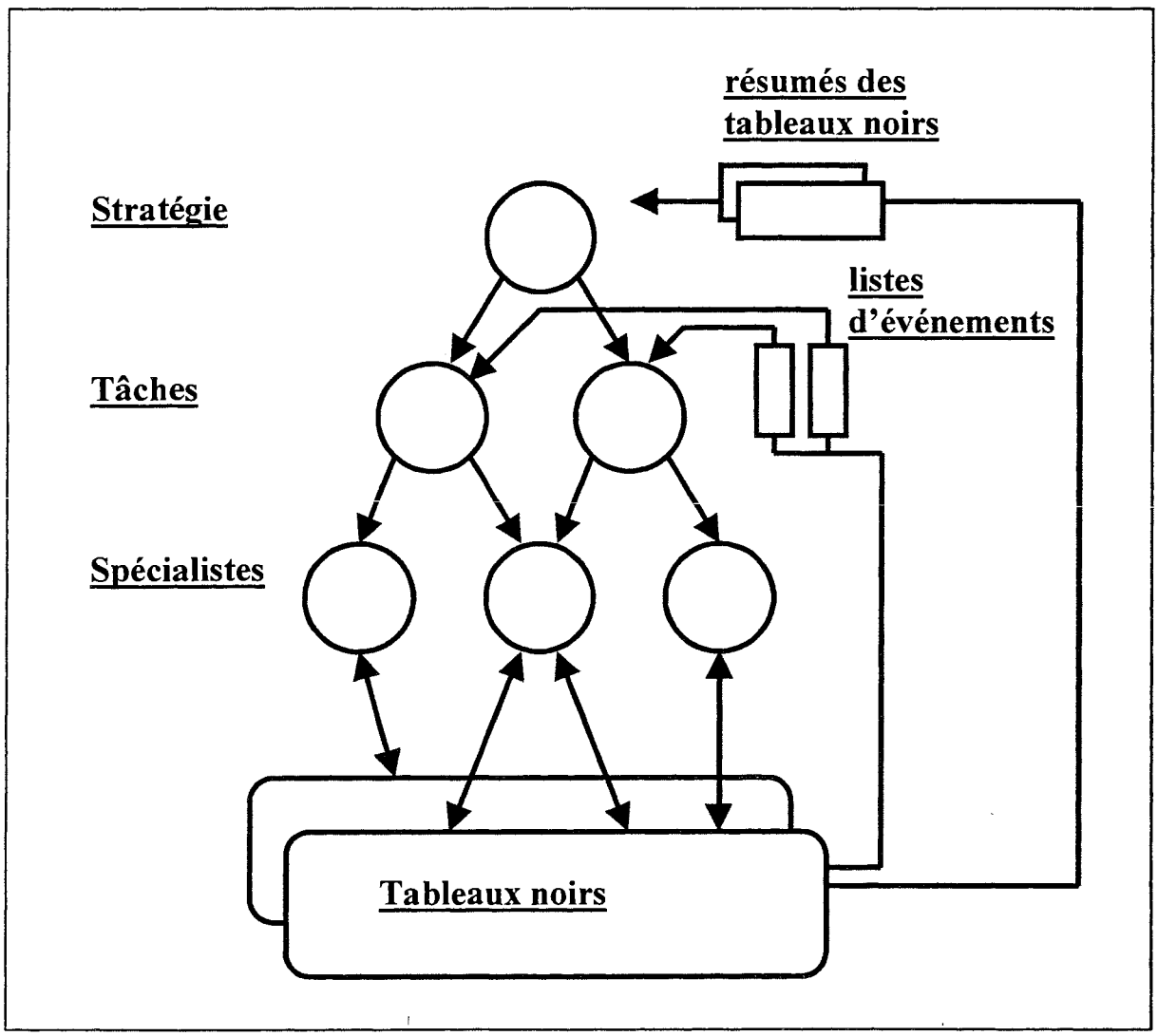

Figure 10 : Architecture générale de ATOME (d'après V. Chevrier (33))

Les spécialistes peuvent être constituées soit d'une base de règles, soit d'une procédure. Ce sont elles qui agissent sur le tableau noir en ajoutant, modifiant ou supprimant des hypothèses. Notons que l'on peut définir plusieurs tableaux noirs, ce qui permet de structurer davantage les données et les hypothèses du problème à résoudre (en plus des niveaux propres à chaque tableau noir).

Les tâches coordonnent les activités des différentes spécialistes. Elles disposent pour cela de listes d'événements signalant les dernières actions effectuées sur les tableaux noirs. Chaque tâche possède sa propre liste d'événements. C'est le contenu de sa liste d'événements qui permet à une tâche de sélectionner un ensemble de spécialistes à activer ainsi que leur ordre

[6] Le terme « spécialiste » est ici du genre féminin car il signifie « source de connaissance spécialiste »

d'activation. Notons que ce sont les spécialistes qui créent les événements associés à leurs propres actions sur les tableaux noirs.

La stratégie est chargée de l'activation des différentes tâches. Elle s'appuie pour cela sur les résumés des tableaux noirs qui ne contiennent que les informations importantes pour la suite de la résolution du problème au niveau le plus global.

2.2.3.2 Format des sources de connaissance

Schéma commun

Les trois types de sources de connaissance (stratégie, tâches et spécialistes) ont un schéma de base identique, décrit ci-dessous.

Chaque source de connaissance (SC) est définie par son *type* (stratégie, tâche ou spécialiste) et son *nom*.

On peut associer à toute SC des *variables locales*. Leur portée s'étend à toute SC que la SC considérée va activer.

Les SC ont un format condition-action (selon le modèle classique de tableau noir).

Chaque SC contient un *ensemble de règles*, qui correspond à sa *partie action* (sauf pour le cas d'une SC de type spécialiste ayant une forme procédurale).

Une *règle* est définie par son nom et la SC à laquelle elle appartient. Chaque règle est composée de prémisses et d'une partie action. Les prémisses définissent les conditions d'application de la règle ; la partie action contient la description des *actions* que la règle doit effectuer lorsqu'elle est déclenchée. Ces actions sont de différentes natures selon le type de SC considérée. Une action réalisable dans toute SC est la mise à jour des variables locales.

On peut associer à toute règle de l'ensemble de règles d'une SC un *coefficient d'importance*, permettant notamment de résoudre les conflits entre règles applicables.

Les paramètres *mode* et *conflit*, associés à chaque SC, permettent de définir la façon dont elle va fonctionner vis-à-vis de son ensemble de règles :

- le paramètre *mode* définit le mode de fonctionnement du moteur d'inférence pour la SC considérée (il est également possible de spécifier le mode de fonctionnement lors de l'activation de la SC).

- le paramètre *conflit* permet de définir la méthode de sélection d'une règle à déclencher parmi celles présentes dans l'ensemble de conflit. La fonction de résolution de conflit indiquée par ce paramètre peut correspondre par exemple à la première règle, ou bien une règle au hasard, ou bien la règle de meilleur coefficient d'importance, ou toute autre fonction. La fonction par défaut consiste à choisir une règle au hasard parmi celles ayant le plus grand coefficient d'importance.

Notons qu'il existe la possibilité de définir des *paquets de règles* à l'intérieur de l'ensemble des règles d'une SC. Chaque paquet de règles possède aussi les paramètres mode et conflit. Un paquet de règles appartenant à une SC donnée peut être activé par un ordre d'activation présent dans la partie action d'une règle de la même SC (cette règle fait alors office de meta-règle). Un paquet de règles peut être défini a priori ou au cours de la résolution. La sélection d'une règle pour l'inclure dans un paquet de règles s'effectue sur la base de la valeur de son coefficient d'importance (on peut aussi, dans le cas de la définition "dynamique", désigner les règles par leur nom).

La *partie condition* d'une SC est définie au niveau de la SC qui l'active. Elle est représentée en effet par les prémisses de la règle (de la SC appelante) dont la partie action contient l'ordre d'activation.

Caractéristiques propres aux spécialistes

Les spécialistes sont activées par les tâches. Notons qu'une même spécialiste peut être utilisée par plusieurs tâches.

La partie condition d'une SC spécialiste est représentée à deux niveaux : d'une part par les prémisses de la règle qui l'active, d'autre part, par la définition de *préconditions*. Ce paramètre permet de définir les conditions à vérifier pour valider l'ordre d'activation en provenance de la tâche appelante. Ces conditions peuvent s'appliquer aux variables locales.

Les prémisses des règles de spécialistes sont composées de deux éléments : une partie *conditions* et une partie *sélection*. La partie conditions contient une expression à résultat booléen, s'appliquant par exemple aux variables locales. La partie sélection s'applique aux tableaux noirs. Elle permet de sélectionner des faits (appelés *nœuds*) présents dans les tableaux noirs.

Les actions réalisables au niveau des règles de spécialistes sont de deux natures : actions sur les tableaux noirs (ajout, modification ou suppression de nœuds) et mise à jour des variables locales.

Les variables locales ont pour rôle de permettre aux spécialistes de déterminer leur contexte de travail, à chaque activation.

Caractéristiques propres aux tâches

Les tâches sont activées par la stratégie.

Toute opération effectuée sur un tableau noir crée un événement. Chaque tâche possède sa propre liste d'événements, l'informant des modifications survenues sur les nœuds des tableaux noirs qui l'intéressent.

La tâche activera donc une ou plusieurs spécialistes en fonction des événements qui lui parviennent. Il est cependant possible d'associer une autre manière de fonctionner à une tâche, à l'aide du paramètre *mode de raisonnement*. En effet, trois modes sont possibles : le mode *dirigé par les règles*, le mode *dirigé par les événements*, et le mode *opportuniste* :

- le mode dirigé par les règles entraîne un comportement identique à celui des autres sources de connaissance, à savoir le choix de la règle à déclencher selon la fonction de résolution de conflit.

- le mode dirigé par les événements recherche tout d'abord, dans la liste des événements de la tâche, le premier événement rendant vraie la partie gauche d'une ou plusieurs règles, puis utilise la fonction de résolution de conflit pour choisir celle à déclencher.

- le mode opportuniste combine événements et règles afin de déterminer les instances de spécialistes à activer, à l'aide d'heuristiques.

Les prémisses des règles de tâches peuvent donc s'appliquer aux listes d'événements (dans le cas du mode dirigé par les événements). Les règles de tâches peuvent effectuer trois types d'actions : activer des spécialistes, activer des sous-ensembles de règles internes à la tâche, ou bien mettre à jour des variables locales.

Notons que dans le cas des tâches, les variables locales n'ont pas le même rôle que pour les spécialistes. Elles permettent simplement de mémoriser une valeur ou un calcul intermédiaire mais ne constituent pas le contexte de travail de la tâche. En effet, le contexte de travail principal de la tâche est caractérisé par l'état de la liste locale d'événements au moment de son activation.

Caractéristiques propres à la stratégie

Afin de pouvoir évaluer l'état d'avancement dans la résolution du problème au niveau global, la stratégie dispose d'un résumé des tableaux noirs, qui constitue une vue ne lui fournissant que les nœuds importants de chaque niveau de tableau noir. Ces nœuds sont désignés a priori, lors de la définition des tableaux noirs.

Les prémisses des règles de la stratégie s'appliquent donc aux nœuds présents dans les résumés des tableaux noirs et aux variables locales. Une fois activée, une règle de stratégie peut effectuer trois types d'actions : activer des tâches, activer des sous-ensembles de règles internes à la stratégie, ou bien mettre à jour des variables locales. Les variables locales de la stratégie ont le même rôle que pour les tâches.

2.2.3.3 Principes de fonctionnement

Boucle de contrôle

Un cycle d'exécution de ATOME peut être décrit de la manière suivante (34) :

1. la stratégie choisit la règle la plus importante ayant sa partie gauche satisfaite et la déclenche. Elle consulte pour cela les résumés des tableaux noirs. L'exécution de cette règle provoque l'activation d'une séquence de tâches ou d'une unique tâche. La stratégie transmet alors, de façon séquentielle, le contrôle à chacune des tâches activées et se met en attente. Les nœuds sélectionnés par la stratégie sont également transmis, en guise de *première focalisation de contrôle*.

2. la tâche qui a le contrôle provoque l'activation d'une ou plusieurs spécialistes (toujours de façon séquentielle) et se met alors en attente. Là aussi, il y a transmission du contrôle aux spécialistes et d'une *seconde focalisation de contrôle* : les événements qui ont permis l'activation des spécialistes. Le but est ici de permettre à une spécialiste activée d'accéder aux nœuds qui sont associés à ces événements. Lorsque la tâche a fini de travailler, elle redonne le contrôle à la stratégie.

3. la spécialiste en cours d'activation consulte les régions des tableaux qui l'intéressent. Elle peut créer, modifier ou supprimer des hypothèses dans les divers tableaux. Ses interventions peuvent engendrer des événements ou, si leurs résultats sont importants, ceux-ci peuvent être notés dans les résumés des tableaux. C'est donc la spécialiste qui, en alimentant les listes d'événements et les résumés des tableaux, assure la poursuite du processus de résolution.

4. fin du cycle.

Le système s'arrête lorsque aucune règle de la stratégie n'est déclenchable.

Fonctionnement du moteur d'inférences

Le déclenchement d'une règle de SC s'effectue, en chaînage avant, selon le processus :

si partie-conditions vérifiée et partie-selection non-vide
alors exécuter partie-actions

Le moteur d'inférence de ATOME est paramétrable. Il fonctionne selon les étapes suivantes :

1. sélection dans la base de règles ou l'un de ses sous-ensembles (dans le cas de l'activation d'un paquet de règles) de celles ayant leurs prémisses vérifiées et constitution de l'ensemble de conflit.

2. choix d'une règle dans l'ensemble de conflit, à l'aide de la fonction de résolution de conflit.

3. déclenchement de la règle.

4. bouclage optionnel sur 3, tant que les prémisses de la règle sont vérifiées.

5. suppression de la règle déclenchée de la base de règles.

6. déclenchement optionnel de toutes les règles de l'ensemble de conflit (retour en 2).

7. reconstruction optionnelle de l'ensemble de conflit à partir de la base de règles initiale privée des règles déjà utilisées et passée au travers d'un filtre (retour en 1).

8. reconstruction optionnelle de l'ensemble de conflit à partir de la base de règles initiale privée des règles déjà utilisées (retour en 1).

Les étapes 4, 6, 7 et 8 sont optionnelles, elles dépendent de la valeur de paramètres modifiables[7] (ces paramètres ont été introduits dans Atome par Ferraris (35)) :

Le paramètre *rule* gouverne l'étape 4. Il détermine le comportement du moteur vis-à-vis de la règle à déclencher. Deux valeurs sont possibles :

- *simple* : la règle ne sera déclenchée qu'une seule fois (pas d'étape 4),
- *cyclic* : la règle sera déclenchée tant que ses prémisses seront vérifiées. Notons que cela suppose que la partie action de la règle alimente la partie prémisses et que le phénomène converge.

Le paramètre *conflict* s'applique à l'étape 6. Il détermine le comportement du moteur vis-à-vis de l'ensemble de conflit. Deux valeurs sont possibles :

- *simple* : une seule règle sera déclenchée (pas d'étape 6),
- *multiple* : toutes les règles seront déclenchées séquentiellement sans réévaluation de leur partie conditions. Cela suppose que l'application d'une règle ne va pas remettre en cause les conditions des règles suivantes.

Le paramètre *end-of-conflict* détermine le comportement du moteur à la fin d'un cycle de base (étape 7). Les valeurs possibles sont :

- *quit* : le cycle se termine,
- *again* : on poursuit par un nouveau cycle, les règles prises en compte pour la reconstruction de l'ensemble de conflit étant l'ensemble des règles utilisables privé de celles déjà utilisées, et filtré (selon le paramètre filter).

Le paramètre *filter* détermine les conditions que les règles doivent satisfaire pour être candidates à la reconstruction de l'ensemble de conflit. Ces conditions peuvent porter sur les caractéristiques des règles déclenchées lors de l'activation précédente. Un exemple de filtre est celui qui précise que les règles à sélectionner ne doivent pas posséder un coefficient d'importance supérieur à la dernière règle déclenchée.

Enfin, le paramètre end-of-cycle détermine le comportement du moteur lorsqu'un cycle est terminé (étape 8). Les valeurs sont :

- *quit* : l'activation est terminée,
- *again* : on recommence un nouveau cycle. Il y a reconstruction de l'ensemble de conflit à partir des règles non déclenchées (on ne tient pas compte du filtre).

2.2.3.4 L'utilisation de ATOME et ses limites

ATOME a été utilisé pour développer des applications dans différents domaines, tels que la compréhension de l'écriture (36), l'aide à la décision pour la maintenance de la voirie urbaine (37), (35), la planification de programmes de mesures en acoustique sous-marine (38), (33), l'aide au commandement (39), la télémanipulation spatiale (40), la conception d'une machine à dicter à entrée vocale (41), l'interprétation de dessins industriels en CAO (42), l'analyse des paysages agraires à partir de données satellitaires (34).

Certaines de ces utilisations ont conduit à apporter des modifications à ATOME. On peut citer notamment la transformation du moteur initial d'ordre 0+ en moteur d'ordre 1, et

[7] les détails fournis ici correspondent à la version C++ de ATOME

l'introduction de paramètres permettant de faire varier le mode de fonctionnement de ce moteur (35).

Par ailleurs, certaines directions de recherche ont aussi été explorées autour d'ATOME et ont conduit à l'élaboration de nouveaux systèmes. On peut citer notamment la gestion d'hypothèses (43), la représentation des informations incertaines et imprécises (44), la prise en compte des contraintes temps réel (45).

Nous avons utilisé dans ce travail la version "classique" de ATOME, dans sa dernière implémentation, réalisée en C++ (46), (47). On dispose avec ce système d'un outil de développement d'applications multi-agents qui offre beaucoup de souplesse à l'utilisateur pour représenter les connaissances et les raisonnements utiles à la résolution d'un problème.

Chevrier (33), puis Le Ber (34) ont cependant évoqué quelques problèmes rencontrés avec ATOME, concernant principalement la représentation des connaissances.

Chevrier souligne une difficulté liée à la représentation et à la gestion des données de base (dans son cas les mesures de bruit), dans les tableaux noirs. En effet, les nœuds des tableaux noirs offraient bien la possibilité de décrire les données par des attributs. Par contre, la définition d'opérateurs génériques sur ces attributs (génération, accès, test) a imposé l'introduction de code spécifique dans les règles. Cela a conduit à insérer de la connaissance "non experte" au milieu de la connaissance experte, ce qui a rendu les règles moins lisibles. Par ailleurs, il y avait duplication du code entre les différentes règles (car les instructions correspondant à ces manipulations étaient répétées plusieurs fois).

Le Ber a souligné elle aussi ce problème, qui l'a amenée à définir des spécialistes dont l'action se limite à calculer des valeurs d'attributs en fonction des valeurs d'autres attributs du même noeud.

Ces difficultés prêchent en faveur d'une représentation véritablement « objet » pour les nœuds des tableaux noirs.

3 Conclusion : problématique et objectifs

3.1 Les objectifs par rapport au domaine

3.1.1 Délimitation du champ

Nous évoquons ici différentes limites du problème traité.

Nous nous intéressons à la simulation ergonomique, au sens défini dans le chapitre 1 (1.1). Le contexte est donc celui de la constitution d'outils informatiques permettant d'appliquer une démarche visant à simuler a priori le comportement de l'homme, afin de participer aux choix de conception pour les produits industriels.

Nous avons vu au chapitre 1 que la constitution de modèles d'opérateurs humains a été initiée dans différents domaines. Nous nous limitons ici à celui de la conception de postes de travail et nous nous intéressons plus particulièrement à la conception des habitacles de poids-lourds.

Dans ce contexte, la simulation du comportement de l'homme est limitée à la simulation du comportement moteur ainsi qu'à la vision. Nous ne nous situons pas du tout au niveau des fonctions cognitives. Par ailleurs, la simulation du comportement moteur est ici limitée à la simulation de gestes et de postures, telle que la réalise le modèle MAN3D, que nous utilisons.

Nous allons expliquer plus loin que l'un de nos objectifs est de modéliser les interactions entre l'opérateur et l'environnement. Dans ce cadre, il n'est pas prévu de réaliser un couplage réel entre le mannequin graphique et la maquette CAO du produit. En effet, les actions du mannequin ne sont pas répercutées sur l'environnement : par exemple, si l'on simule l'action de pousser une porte, l'objet porte ne sera pas déplacé. De même, les déplacements d'objets dans l'environnement n'affectent pas la position du mannequin, même s'il y avait contact dans la situation initiale : par exemple si la main du mannequin est positionnée sur la porte, le fait de modifier l'ouverture de la porte n'entraîne pas un changement de position du mannequin.

Enfin, le problème traité a été limité par une contrainte importante, influant aussi bien sur la définition des objectifs que sur la modélisation du problème et l'implémentation : la décision d'utiliser MAN3D « en l'état », c'est-à-dire de ne pas en réécrire le code, de l'utiliser comme un module déjà conçu, sans possibilité de modifications.

3.1.2 Les objectifs

Nous avons décrit dans le chapitre 1 la démarche de simulation en ergonomie (section 1.1), telle qu'elle est définie de manière théorique. Or, comme nous l'avons vu, les outils existants ne permettent pas de couvrir l'étendue de cette démarche. Plus particulièrement, les modèles graphiques d'opérateur humain se limitent à la simulation de l'activité motrice, sans prendre en compte le contexte plus global dans lequel se déroule cette activité. En effet, c'est l'utilisateur de ce type de modèle qui « sait » quelle est l'activité qu'il simule (par exemple la conduite d'un poids-lourds) et qui ajoute sa connaissance de la tâche à réaliser aux capacités du modèle. Il devra par exemple décider de la combinaison de gestes élémentaires qui permettra de simuler une action spécifique. Par ailleurs, la plupart de ces modèles ne prennent pas en compte l'environnement dans lequel s'exerce l'activité simulée. Là encore, c'est l'utilisateur qui doit se charger de gérer l'interaction entre l'opérateur virtuel et le poste de travail en cours de conception : au niveau du modèle d'opérateur, l'environnement n'est qu'un ensemble indifférencié de points 3D pouvant servir de cibles pour l'atteinte.

Notre premier objectif est donc la modélisation de l'activité de l'opérateur, dans le but de pouvoir associer au modèle graphique d'opérateur (dans notre cas MAN3D), un modèle de l'environnement ainsi qu'un modèle des interactions entre l'opérateur et son environnement. L'idée est de pouvoir introduire la spécificité des objets de l'environnement comme un déterminant supplémentaire de l'activité posturale.

La définition d'un modèle de l'activité permet par ailleurs de prendre en charge les problèmes liés à la manipulation du mannequin. En effet cette manipulation est une tâche nécessitant une véritable expertise de la part de l'utilisateur. Nous avons vu dans le cas particulier de MAN3D (paragraphe 1.3.1), que l'utilisateur n'a à sa disposition que deux modes de manipulation : il peut soit modifier directement les angles de la structure, soit utiliser des commandes de mouvements, s'appliquant spécifiquement à une partie du corps à la fois (le tronc, la tête, un bras ou une jambe). La modification directe des valeurs d'angles est une méthode longue et fastidieuse pour l'utilisateur. Les commandes de mouvement associées aux différentes parties du corps permettent d'alléger la manipulation, mais elle laissent encore à la charge de l'utilisateur le choix de la partie à activer pour une simulation donnée. Par ailleurs, dans le cas d'une simulation nécessitant l'intervention successive (ou en parallèle) de plusieurs parties, c'est l'utilisateur qui doit gérer l'enchaînement des appels aux parties concernées. Par exemple si l'utilisateur simule l'atteinte d'un point cible, et que ce point se révèle hors d'atteinte en utilisant le bras, c'est à lui de décider s'il faut incliner le dos ou avancer les fessiers, et dans chacun des cas « de combien ». La définition d'un modèle de l'activité de l'opérateur permettra de réaliser des niveaux de commandes « au-dessus » de ceux existants. L'objectif est donc aussi d'intégrer la connaissance experte nécessaire au pilotage du mannequin et d'automatiser pour partie la manipulation du mannequin.

Enfin, nous avons vu (paragraphe 1.4.2) que les modèles graphiques d'opérateur laissent aussi à leur utilisateur la charge de décider au final si une posture obtenue par simulation est acceptable ou non, ou bien si une posture est préférable à une autre. Il manque à ces modèles une fonction de diagnostic ergonomique porté sur les postures simulées. L'ajout de la prise en compte du contexte de l'action simulée pourra nous permettre d'apporter des éléments de solution à ce problème. Par ailleurs, de manière plus générale, chaque test (atteinte, vision, ...) effectué par l'utilisateur d'un mannequin graphique a pour but de vérifier si l'activité prévue pour le produit peut effectivement être réalisée. Un dernier objectif est donc de rajouter un

module de diagnostic ergonomique, qui permette de fournir une aide à la décision dans ce contexte.

Nous souhaitons donc compléter le modèle MAN3D, avec les objectifs suivants :

- la modélisation de l'activité de l'opérateur,
- l'ajout de la prise en compte de l'environnement,
- l'ajout d'un niveau stratégique permettant le pilotage automatisé du modèle,
- l'ajout de fonctionnalités de diagnostic ergonomique.

Notre cadre étant celui de la conception d'habitacles de poids-lourds, l'idée est de concevoir un système :

- qui s'appuie sur la donnée d'objectifs fournis par l'utilisateur (tester l'utilisation d'un ou plusieurs éléments de la cabine, simuler une tâche, ...),
- qui fournisse une simulation automatisée de l'activité correspondante,
- qui fournisse un diagnostic ergonomique associé à la simulation.

Ceci permettrait à l'utilisateur du système de se focaliser sur les problèmes ergonomiques testés, sans voir sa tâche alourdie par la manipulation du mannequin. Par ailleurs l'idée est de lui fournir ainsi des éléments pour pouvoir comparer différentes solutions d'implantation de fonctions sur la maquette du produit.

Nous avons donc conçu une maquette de système, avec pour objectif de fournir un démonstrateur, permettant de montrer la faisabilité et l'intérêt d'une approche IA dans ce contexte de modélisation graphique de l'activité de l'opérateur pour l'assistance à l'évaluation ergonomique.

3.2 La problématique IA

3.2.1 Formaliser le qualitatif

Nous avons vu dans le chapitre 1 que les modèles graphiques d'opérateur humains sont des modèles numériques, basés sur des méthodes quantitatives. Or, ces méthodes peuvent avoir un caractère simplificateur dans le contexte de la simulation ergonomique, car elles ne permettent pas de prendre en compte les paramètres non quantifiables.

La solution que nous testons pour répondre aux objectifs fixés est l'appel aux méthodes de l'IA, permettant de formaliser le qualitatif.

Nous devons en effet définir :

- un formalisme pour modéliser l'environnement, afin de l'introduire comme paramètre du modèle,
- une façon de formaliser et d'introduire dans le modèle la connaissance sur la manière dont s'effectuent les interactions entre l'opérateur et l'environnement,
- une architecture permettant d'introduire un niveau stratégique, afin de gérer les appels aux différentes parties du corps (dans le cadre de la résolution de problèmes d'atteinte),
- un module formalisant la connaissance experte sur le diagnostic que l'on peut porter sur une posture, en termes de confort et d'acceptabilité, mais aussi pour les aspects liés à la vision, et plus généralement sur la possibilité d'exercer l'activité prévue pour le produit en cours de conception.

3.2.2 Choix d'une approche

L'objectif étant de concevoir plusieurs bases de connaissances autour de MAN3D, notre choix a rapidement été celui d'un environnement de type multi-experts.

Par ailleurs, les besoins de formalismes pour la modélisation de l'activité de l'opérateur nous ont conduit à adopter la souplesse de l'approche objet. De plus, la complexité du problème à traiter nécessitait de disposer d'un outil de conception suffisamment riche pour permettre la modélisation fine des concepts manipulés, de manière déconnectée de la réalisation. L'utilisation d'une méthode de conception orientée objet, dans notre cas la méthode OMT (48), répond à ce besoin.

L'utilisation d'une architecture de tableau noir (ici ATOME) nous permet d'avoir une démarche exploratoire pour la conception des différents modules de connaissance et de leur articulation.

Nous détaillons dans la partie 2 quels ont été les choix de conception effectués pour concilier une approche objet et l'usage du modèle de tableau noir.

Le système conçu a été nommé « ADELE », pour Aide au Diagnostic Ergonomique en Liaison avec des Bases Expertes.

PARTIE II : Le système ADELE

Introduction

Notre objectif est de concevoir un système capable de simuler l'activité motrice et d'émettre un diagnostic ergonomique à l'issue de cette simulation, dans le contexte de l'évaluation de solutions de conceptions de produits avec un logiciel de CAO.

Nous partons d'un système existant, MAN3D, modèle graphique d'opérateur humain, qui présente quelques manques du point de vue du système envisagé comme nous l'avons vu précédemment. L'objectif est d'enrichir ce modèle. Dans ce contexte, on pourrait imaginer atteindre l'objectif idéal d'un système complètement automatisé, c'est-à-dire d'un opérateur virtuel capable à l'appel de simuler l'utilisation de l'objet en cours de conception, d'afficher graphiquement ses actions et de fournir un diagnostic ergonomique sous forme de verbalisations. Le système conçu (ADELE) tend vers cet objectif, sans avoir pour prétention de l'atteindre. C'est pourquoi le chapitre 4, consacré à la modélisation des données et des connaissances nécessaires pour faire évoluer le modèle de départ vers un tel système, débute par une analyse globale du problème, visant à définir les limites que nous nous fixons. Nous identifions ensuite les domaines de connaissances nécessaires, au sein desquels nous délimitons et nous modélisons les différentes composantes du système (sections 4.2 à 4.4).

Le chapitre 5 développe les questions de nature conceptuelle relatives à l'architecture du système, au choix des outils (base de données objet, système multi-agents) et à leur mise en œuvre.

Ces phases de modélisation et de conception ont par ailleurs fait l'objet de communications (49), (50), (51).

4 Modélisation des données et des connaissances

4.1 Analyse globale du problème

4.1.1 Quelles limites pour la simulation ?

Partant de l'existant, MAN3D, mannequin 3D doté de segments mobiles[8] positionnables dans l'espace, il nous faut préciser comment se situe le système envisagé par rapport à l'objectif idéal d'une simulation complètement automatisée et donc définir :

- la "granularité" de la simulation de l'activité motrice,
- le niveau de prise en charge envisagé pour les aspects décisionnels et de diagnostic.

Du point de vue de l'intervention de l'utilisateur, dans le système idéal il suffit pour lancer la simulation, d'indiquer au système quelle est la solution CAO à évaluer et d'appeler l'opérateur virtuel. Cela suppose notamment que les différents tests ergonomiques à réaliser sont connus et définis et que l'opérateur virtuel peut enchaîner dans un ordre approprié les différentes séquences d'actions permettant de simuler l'utilisation du produit. Par exemple, pour le test d'un habitacle de poids-lourd, cela peut aller de l'entrée dans la cabine au test de l'usage de la boite à gants, en passant par la simulation de la tâche de conduite.

Dans le cas idéal, la granularité de la simulation est minimale : le point d'entrée est l'activité au sens large. La prise en charge des éléments de décision et de diagnostic est maximale : le système s'auto-pilote et le diagnostic sur la simulation est fourni.

Avec le système que nous avons conçu, nous n'atteignons pas encore l'automatisation maximale : l'utilisateur doit intervenir pour choisir les actions à simuler. Le reste de la simulation se déroule par contre en automatique. Le résultat est constitué d'une part de l'affichage à l'écran des modifications de posture engendrées, d'autre part de l'émission d'un compte-rendu associé à la simulation effectuée, fournissant des éléments de diagnostic.

Pour la modélisation de l'activité, nous avons choisi d'aller jusqu'à la définition de scénarios d'activité, composés eux-mêmes d'actions. Le Tableau 1 décrit les différentes couches nécessaires pour arriver au niveau scénario en partant des possibilités offertes par MAN3D. Chaque niveau utilise le niveau précédent. En "montant" de MAN3D vers ADELE, on augmente le degré d'automatisation de la simulation, en diminuant les interventions de l'utilisateur et en augmentant la prise en charge des aspects de décision et de diagnostic. Notre niveau de départ pour MAN3D est celui des segments mobiles : on ne dispose à ce niveau que des commandes associées à chacun des segments pris isolément.

[8] Rappelons qu'un segment mobile est une partie de la structure composée de plusieurs chaînons, à laquelle on peut appliquer des commandes de changement de posture : l'ensemble tête/cou, le buste, le bassin, les bras, les jambes, les doigts, sont des segments mobiles.

	Niveau	Intervention de l'utilisateur	Prise en charge par le système	Expertise intégrée
MAN3D	**Angles**	• Donnée d'une valeur d'angle Jugement sur le réalisme et les possibilités d'atteinte	• Contrôle : valeurs dans les limites (butées articulaires) • Modification de l'angle spécifié	• Butées articulaires
	Segments mobiles	• Donnée du point à atteindre • Choix du segment mobile à activer • Enchaînement des activations de segments mobiles Diagnostic final	• Modification des angles associés au segment mobile spécifié • Production d'un message (possible ou non + butée)	• Calcul de la position de moindre écart par rapport à une posture de confort
ADELE	**Enchaînement segments mobiles**	• Donnée du point à atteindre • Choix du segment mobile effecteur de départ	• Activation éventuelle d'autres segments mobiles et enchaînement	• Expertise sur l'enchaînement des segments mobiles (dont réaction au diagnostic MAN3D)
	Action opérateur objet	• Choix de l'objet à tester • Choix de l'action à simuler sur l'objet	• Enchaînement des activations de segments mobiles • Prise en compte des informations associées à l'action	• Spécification de l'action sur l'objet • Gestion du plan d'action • Diagnostic de conformité à la spécification initiale
	Scénario	• Choix du scénario	• Enchaînement des actions	• Spécification du scénario • Expertise sur l'enchaînement des actions

Tableau 1 : Niveaux imbriqués constituant MAN3D puis ADELE

4.1.2 Les connaissances à modéliser

Notre objectif est la conception d'un système d'assistance à l'utilisateur pour l'évaluation ergonomique, basé sur la simulation graphique de l'activité. L'enrichissement du système existant (MAN3D) consiste à définir une couche "symbolique" permettant de décrire et de manipuler l'opérateur, de décrire l'activité de l'opérateur de manière dédiée à la simulation et en intégrant des éléments nouveaux (tels que l'environnement). Par ailleurs, la simulation doit être orientée vers le diagnostic ergonomique.

Nous pouvons donc distinguer trois domaines de connaissances devant être analysés :

(1) - les connaissances liées au modèle d'opérateur et à la modélisation de l'activité,
(2) - les connaissances liées au pilotage du mannequin,
(3) - les connaissances liées à l'expertise ergonomique.

Il s'agit pour le premier domaine de fournir une description symbolique de l'activité : de l'opérateur et de ses postures, de l'environnement, des interactions entre l'opérateur et l'environnement. Cette description a plusieurs objectifs. Elle doit d'une part constituer la base d'un outil pour l'utilisateur futur de ADELE, lui permettant de spécifier les objectifs de la simulation (actions ou scénario), d'autre part servir de données d'entrée pour le système, pour les aspects simulation et diagnostic ergonomique. Cette description doit être générique et permettre d'intégrer la connaissance concernant la manière dont l'opérateur interagit avec son environnement, tout en permettant de prendre en compte différents types d'environnements. Enfin, elle doit être associée au modèle MAN3D.

Le deuxième domaine concerne les connaissances utiles au pilotage du mannequin. Il s'agit d'identifier et de formaliser la manière dont s'effectuent les gestes réels, c'est-à-dire les stratégies de mouvement relatives à l'enchaînement des différents segments corporels pour l'atteinte d'un point cible.

Le troisième domaine, celui de l'expertise ergonomique, concerne l'ensemble des éléments de raisonnement identifiables pour la production d'un diagnostic ergonomique à l'issue de la simulation d'une ou plusieurs actions. Ces éléments peuvent être de nature générique (par exemple l'évaluation du confort d'une posture par rapport à la durée) ou spécifique à l'environnement en cours de conception (par exemple l'évaluation de la possibilité d'effectuer une tâche telle que le passage de vitesses dans un habitacle de poids-lourd).

Nous développons les aspects liés à l'identification et à la formalisation ou la modélisation des connaissances pour ces trois domaines dans les sections suivantes (4.2 à 4.4). Les aspects liés à l'utilisation de ces connaissances par la mise en œuvre de raisonnements ainsi qu'à leur localisation dans une architecture de système sont évoqués dans le chapitre 5.

4.2 La modélisation de l'activité

L'objectif est de fournir un modèle permettant de décrire l'activité d'un opérateur humain, au sens de l'activité motrice exercée dans un environnement contenant des objets ou outils utilisés pour réaliser une tâche prédéfinie. Ce modèle est conçu pour servir à la simulation ergonomique à l'aide d'un mannequin graphique sous logiciel de CAO.

La conception de ce modèle a été faîte à l'aide de la méthode de conception objet OMT (48). On trouvera en annexe 2 un bref descriptif de la méthode, qui permettra notamment de pouvoir lire les schémas produits dans les sections qui suivent.

Les diagrammes d'instances, facilitant la compréhension des concepts, sont fournis au chapitre 6 (paragraphe 6.2.3).

4.2.1 Modélisation de l'environnement

L'environnement est "ce qui entoure l'opérateur". Pour la simulation ergonomique, on choisit de définir l'environnement comme un ensemble d'objets utilisables par l'opérateur ou avec lesquels l'opérateur va pouvoir être en contact. On ne retiendra donc, pour la description des objets constituants l'environnement, que les éléments pertinents pour la simulation et l'émission d'un diagnostic ergonomique. Ainsi, par exemple, la couleur d'un objet ne sera

retenue comme élément de sa description que si elle influe sur la manière d'utiliser cet objet, ou bien si elle modifie le diagnostic porté sur la simulation de l'utilisation de l'objet.

La Figure 11 introduit la description de l'environnement.

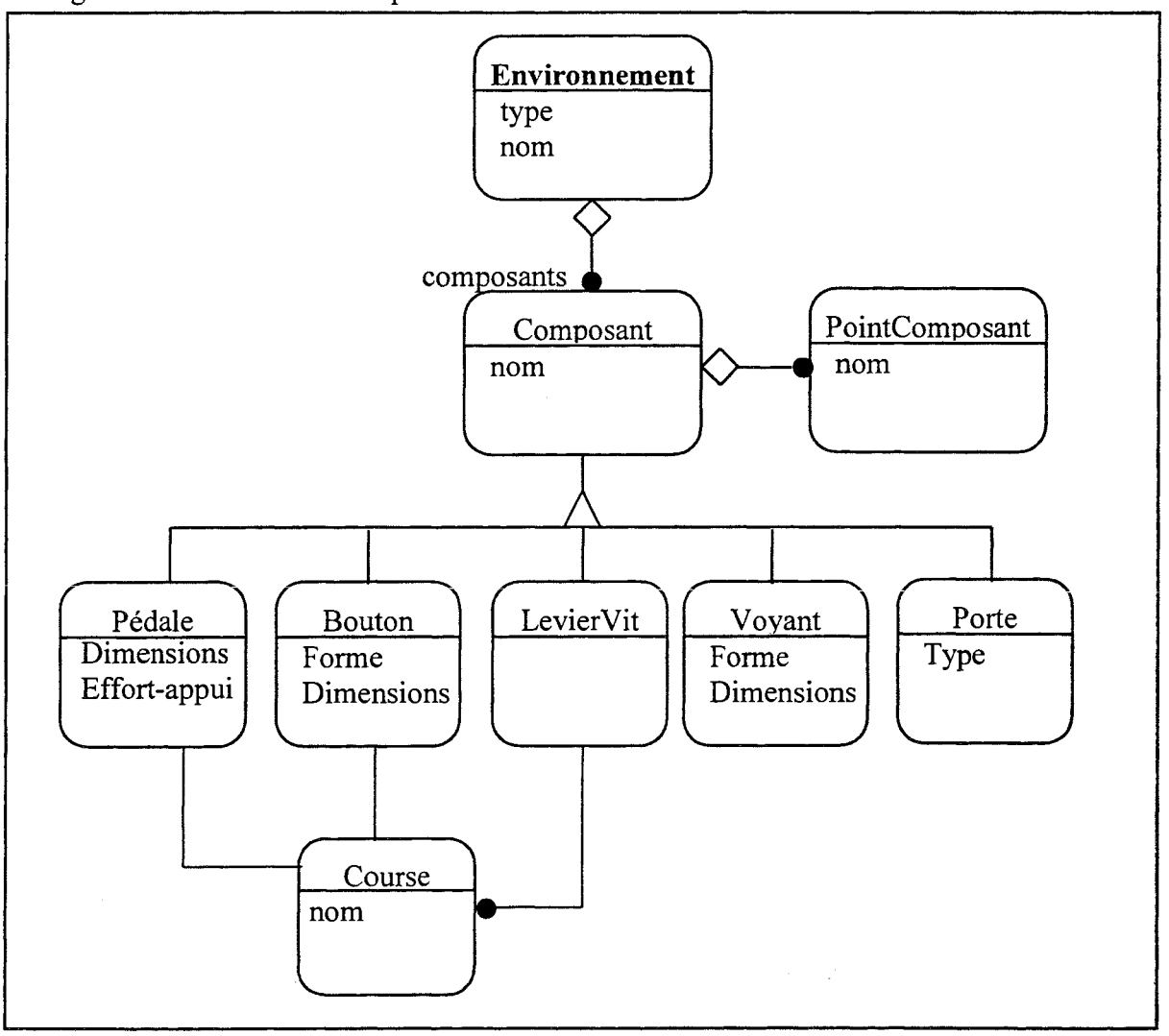

Figure 11 : Modélisation de l'environnement

Un environnement est défini par son nom et son type. Le type indique la nature de l'environnement (cabine de poids-lourd ou de véhicule léger, bureau, ...). On nomme *composant* un objet de l'environnement. Un environnement peut contenir plusieurs composants. A chaque composant on associe un ensemble de points particuliers (par exemple le milieu d'un bouton, ou bien certains points du cercle du volant). On verra plus loin que ces points servent à la modélisation des interactions de l'opérateur avec l'environnement.

On définit plusieurs sortes de composants, ayant des caractéristiques différentes. On associe à certains composants une (ou plusieurs) *course.* C'est le cas pour les composants de type *pédale*, ou *levier de vitesses*, ou *bouton.* Il s'agit de représenter la trajectoire du composant lors de son utilisation.
On ne retient donc dans cette modélisation de l'environnement et de ses composants que les éléments utiles à la simulation de l'activité. Notons que l'on ne crée pas ici d'élément de

description se rapportant à la localisation du composant dans l'environnement. Cet aspect est en effet pris en charge au niveau du mannequin graphique et du système de CAO.

L'environnement est donc modélisé de manière graphique dans le logiciel de CAO. A ce niveau, les objets sont des ensembles de points, repérables par des noms. Le lien avec la maquette CAO sera effectué en associant les noms des composants (et/ou trajectoires et/ou noms de points) définis dans le modèle d'environnement et les noms des objets graphiques correspondants dans la base d'objets du logiciel de CAO.

4.2.2 Modélisation de l'opérateur

On définit d'une part une description de l'opérateur en termes symboliques (opérateur générique), d'autre part une description de l'opérateur graphique (associé à MAN3D), puis on décrit les liens entre ces deux modèles. Ces deux modèles fournissant une description de nature "structurelle", il faut ajouter le concept d'opérateur actif, pour réunir des éléments de description d'un opérateur associé à un objectif de simulation.

4.2.2.1 Modèle symbolique

On définit le concept d'*opérateur générique*, permettant de fournir une description d'un opérateur de nature générique, c'est-à-dire indépendante de sa représentation graphique.

La Figure 12 fournit la description d'un opérateur générique pour le modèle symbolique.

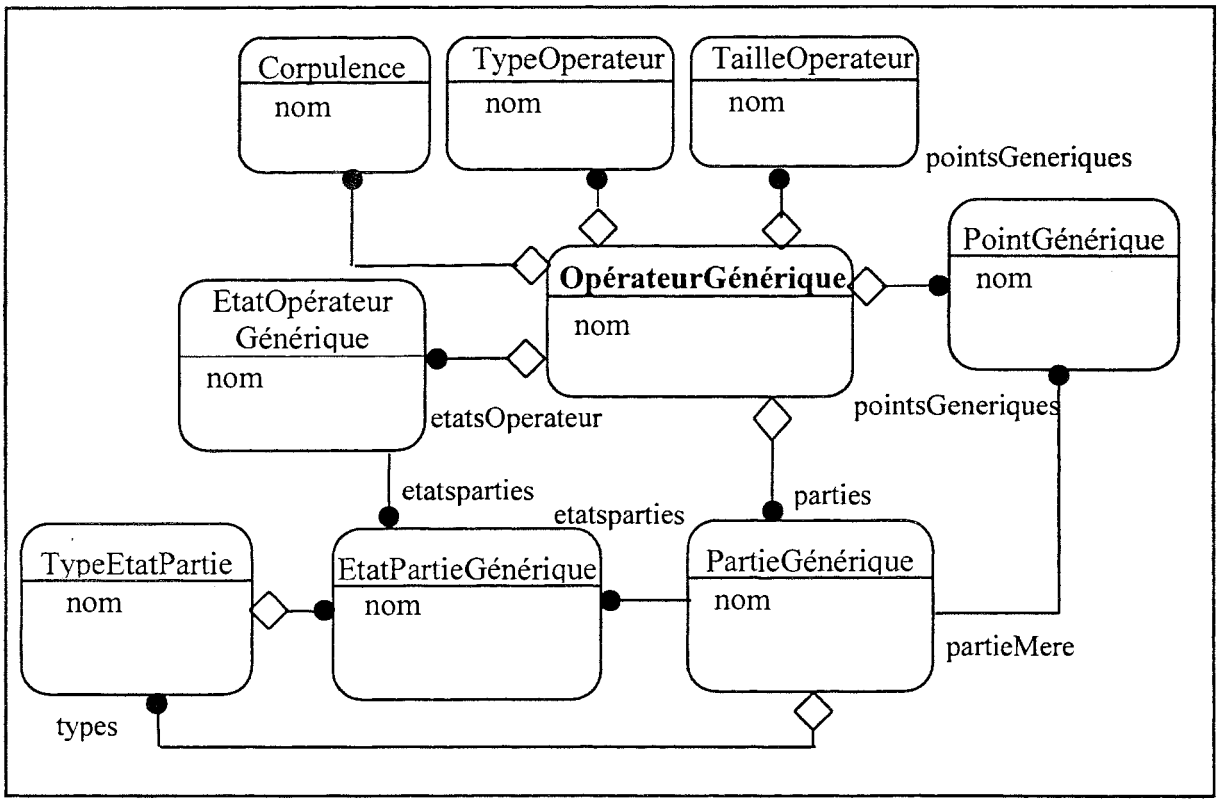

Figure 12 : Opérateur - modèle symbolique

L'opérateur générique (par opposition à l'opérateur graphique décrit plus loin) peut être décrit par un ensemble de parties génériques (par exemple la tête est une partie générique) qui le composent. Par ailleurs, certains points spécifiques (points génériques) sont identifiés : par exemple la paume d'une main. Chaque point générique est associé à la partie de l'opérateur

sur laquelle il se trouve. A chaque partie on peut associer plusieurs descriptifs d'états possibles, eux-mêmes associés à un type. Il s'agit de pouvoir décrire l'état postural d'une partie (nous reviendrons sur ces descriptifs au paragraphe 4.2.4). On peut aussi associer à l'opérateur générique des descriptifs d'état postural du corps tout entier (par exemple *assis*). Chacun de ces descriptifs est lui-même associé à des états posturaux de parties du corps. Enfin, on peut conserver les descriptifs possibles pour la taille, le type (homme, femme ou enfant) et la corpulence.

Comme nous le verrons au chapitre 6, il existe une seule instance d'opérateur humain générique. Par ailleurs, les noms fournis pour les concepts liés à l'opérateur génériques sont les symboles qui sont utilisés en prémisses de règles dans le système multi-agents.

4.2.2.2 Modèle graphique

Le modèle graphique permet de décrire l'opérateur au sens du modèle graphique d'opérateur tel que celui fourni par MAN3D. Il est décrit par la Figure 13.

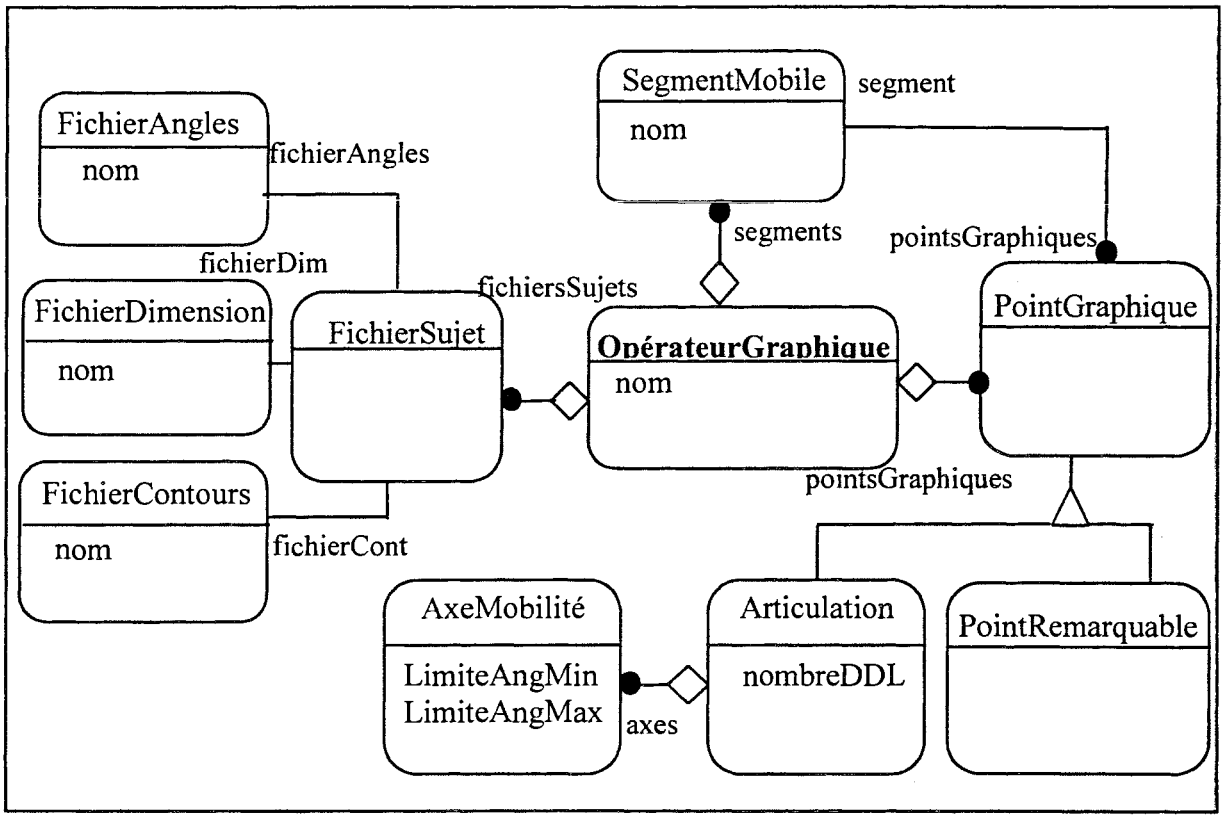

Figure 13 : Opérateur - modèle graphique

L'*opérateur graphique* est défini par un certain nombre d'éléments structurels. Il s'agit tout d'abord des segments mobiles, qui le composent. A chacun de ces segments mobiles sont associés des points graphiques, qui peuvent être soit des points remarquables (au sens de MAN3D, par exemple le bout d'un doigt), soit des articulations. Si le point graphique est une articulation, on lui associe plusieurs axes de mobilité (en nombre égal au nombre de Degrés De Liberté (DDL) de l'articulation). Chaque axe de mobilité est défini par une limite angulaire minimale et une limite angulaire maximale. Ces limites correspondent aux butées articulaires. Notons que dans les cas de couplages entre articulations, ces maxima peuvent varier.

Par ailleurs, on associe à l'opérateur un ensemble de réalisations, les sujets (au sens de MAN3D, voir §1.3.1). Ces sujets sont décrits par plusieurs fichiers de structures différentes.

4.2.2.3 Liens entre l'opérateur générique et l'opérateur graphique

Les liens entre l'opérateur générique et l'opérateur graphique sont réalisés au niveau de la description des états des parties. Ils sont décrits sur la Figure 14.

On associe à un type d'état partie donné (par exemple la *flexion* pour la partie *bras*), un ou plusieurs axes de mobilité de la structure graphique (dans l'exemple de la flexion du bras il s'agit d'un axe de mobilité associé à l'articulation du coude). Par ailleurs, on associe à un état partie donné (par exemple *bras fléchi*) les limites (minimale et maximale) correspondantes, pour le ou les axe(s) de mobilité concernés par le type d'état partie considéré. Par exemple, on fournit pour *bras fléchi* deux valeurs délimitant un intervalle inclus dans les limites angulaires pour l'axe de mobilité du coude concerné.

On ajoute de plus un lien entre un point générique de l'opérateur générique (par exemple la paume d'une main), avec le point graphique correspondant, appartenant à la structure de l'opérateur graphique (par exemple le point paume peut être associé à un point situé au niveau d'une articulation entre phalanges).

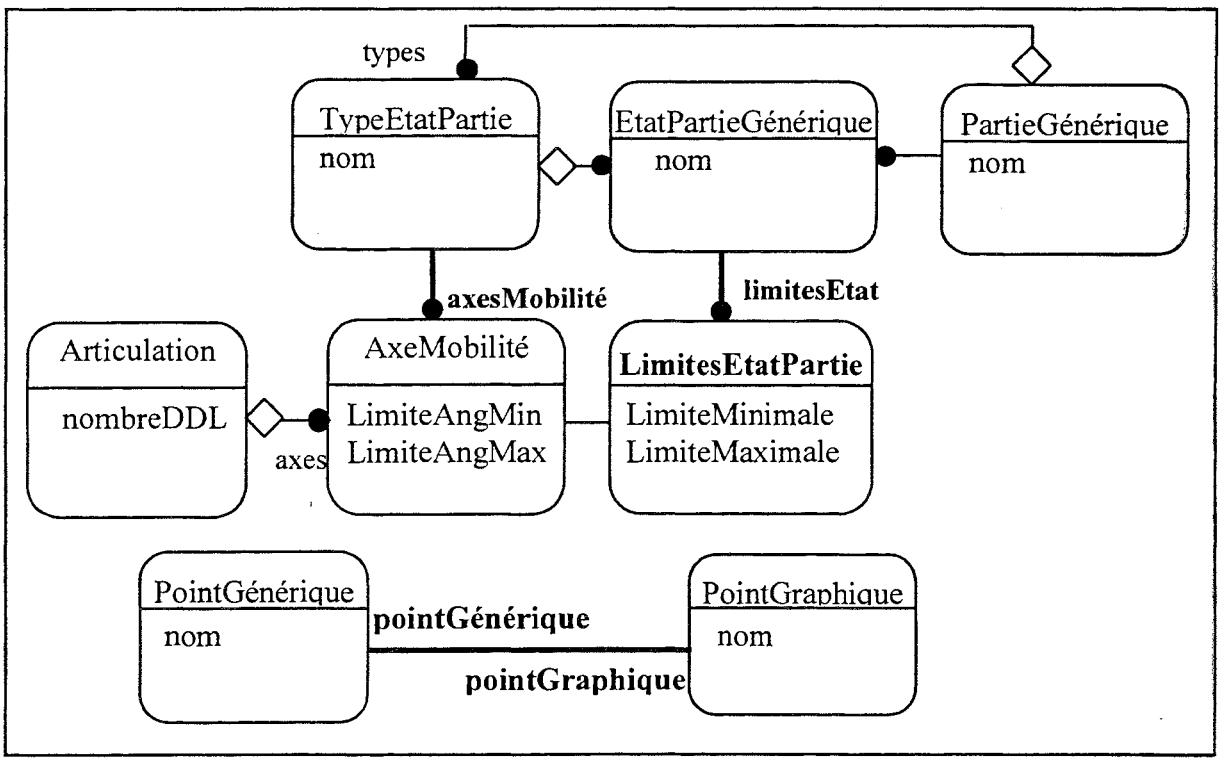

Figure 14 : Liens modèle symbolique / modèle graphique pour l'opérateur

4.2.2.4 Opérateur actif

On définit le concept d'opérateur actif, afin de pouvoir décrire les différents paramètres associés au choix d'un opérateur pour une simulation : les caractéristiques

anthropomorphiques d'une part, et les paramètres d'état d'autre part. Ces paramètres sont décrits sur la Figure 15.

Ainsi, pour une simulation, le système travaillera avec une instance d'opérateur actif donnée. Cette instance fournit un type d'opérateur (par exemple *Femme*) une taille (par exemple *Grand*), une corpulence (par exemple *Moyen*), ces trois paramètres fournissant une description en termes symboliques associée aux valeurs numériques fournies dans le fichier dimensions associé au fichier sujet[9]. Par ailleurs, à cette instance d'opérateur actif sont associés des descriptifs d'état : état de l'opérateur (*assis* par exemple) et états de certaines de ses parties (par exemple un bras *fléchi*). Là encore, les noms utilisés pour décrire l'état d'une partie se retrouveront en prémisse de règles du système multi-agents. Au niveau du modèle objet, une méthode de la classe EtatPartieGénérique permet de calculer la valeur symbolique d'un état de la partie considérée, à partir des valeurs d'angles fournies par le fichier angles associé au fichier sujet.

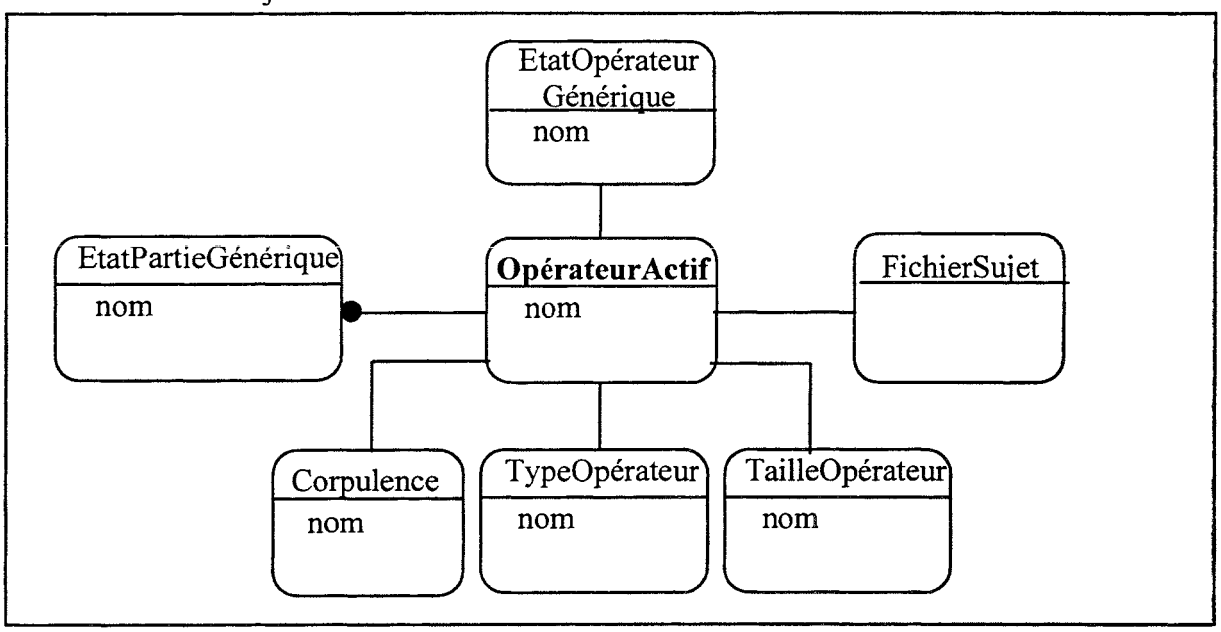

Figure 15 : Opérateur actif

4.2.3 Modélisation des interactions de l'opérateur avec l'environnement

Pour la modélisation des interactions de l'opérateur avec l'environnement, on s'appuie sur le concept de base de contact entre l'opérateur et l'environnement (Figure 16). Un contact est l'association entre un point d'un composant et un point de l'opérateur (par exemple le bout de l'index avec le point situé au centre d'un bouton, ou bien le milieu de la paume avec un point situé sur le pommeau du levier de vitesses, ou encore un point du dos avec un point du dossier du siège, ...).

[9] Le passage du numérique au symbolique impose un choix de découpage en classes de tailles pour la variable taille exprimées en centimètres (en fonction de la distribution de cette variable sur la population ciblée). Le cas de la corpulence peut être résolu par une analyse multivariée (si l'on prend en compte plusieurs variables).

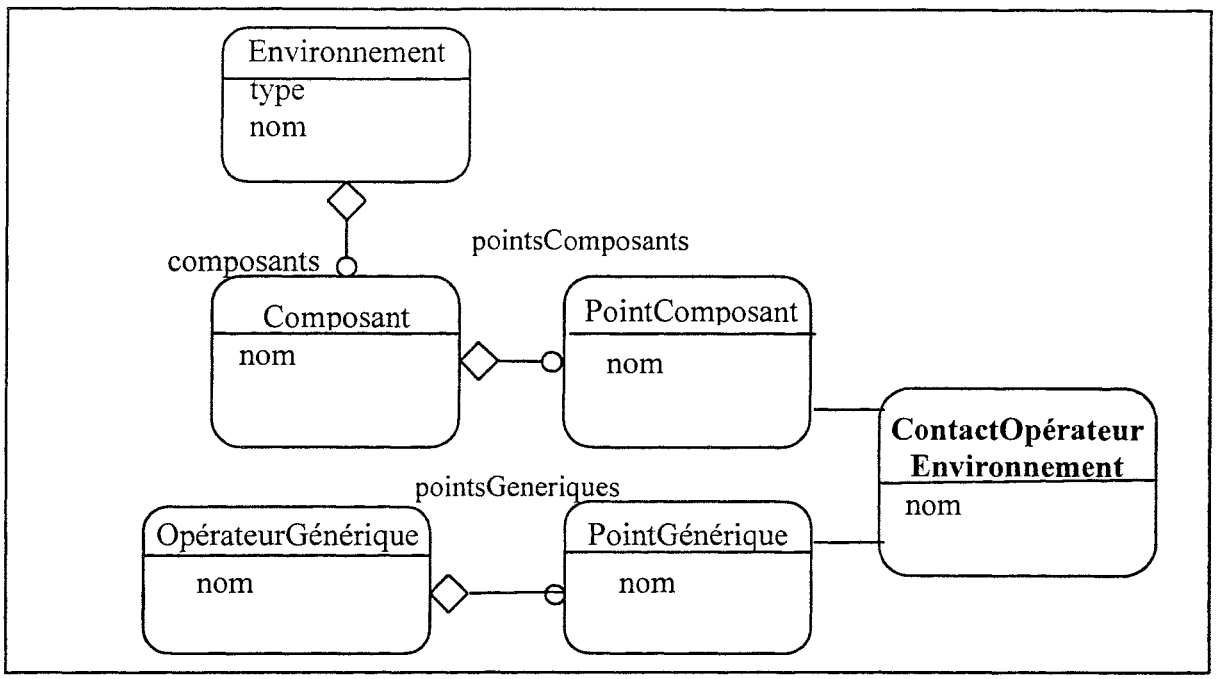

Figure 16: Contact Opérateur Environnement

De manière similaire à la définition de l'opérateur actif, pour une session de simulation, on définira l'ensemble des contacts actifs entre l'opérateur et l'environnement.

Au concept de contact entre l'opérateur et l'environnement, on associe le concept d'action élémentaire, qui désigne les actions motrices de base pouvant être effectuées (Figure 17).

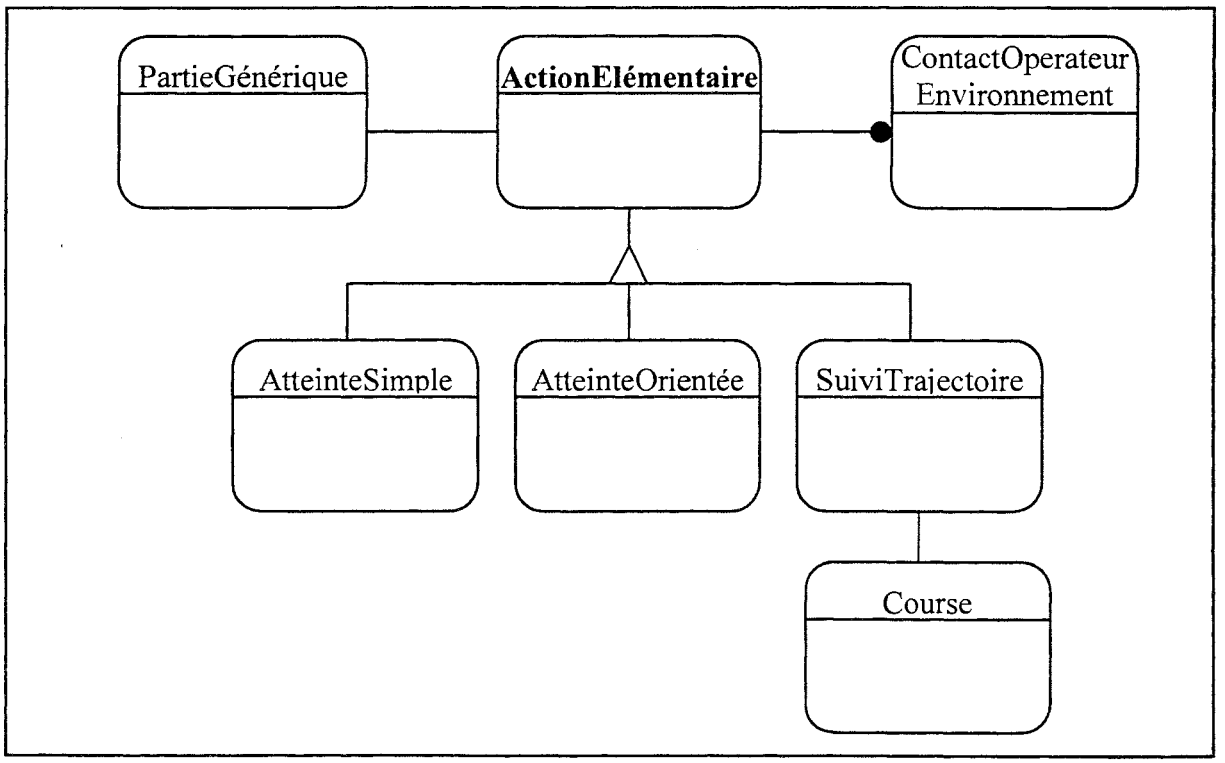

Figure 17 : Action élémentaire

Toute action élémentaire est associée à un segment mobile effecteur, c'est-à-dire à une partie de l'opérateur (un pied ou une main, mais aussi toute autre partie comme le dos par exemple).

Trois sortes d'actions élémentaires sont définies : l'atteinte simple, l'atteinte orientée et le suivi de trajectoire. Elles correspondent aux actions élémentaires de MAN3D et permettent de représenter tout type d'interaction motrice avec l'environnement. L'atteinte simple correspond à la réalisation d'un contact opérateur/environnement. L'atteinte orientée correspond au cas où l'on indique l'orientation à suivre par le membre effecteur pour l'atteinte. Ce sont donc deux contacts entre l'opérateur et l'environnement qui sont associés à l'atteinte orientée. Le suivi de trajectoire correspond au suivi d'une course d'un objet par le membre effecteur, à partir d'un contact opérateur environnement initial.

Afin de pouvoir représenter toutes les actions possibles sur un composant, on ajoute la composition d'actions élémentaires. On définit donc le concept plus général d'action (Figure 18). Deux modes de composition sont possibles : la succession et la simultanéité.

La succession est utile par exemple pour décrire les actions *ouvrir* et *fermer* associées au composant *porte*. Ces actions ne sont pas élémentaires. L'action *ouvrir la porte* peut être décomposée en plusieurs sous-actions successives : *tourner la tête vers la poignée*, *atteindre la poignée*, *actionner la poignée*, *pousser la porte*.

La simultanéité sert dans deux cas : soit pour décrire une action nécessitant plusieurs points de contact entre l'opérateur et l'environnement, soit pour décrire une action complexe, impliquant plusieurs sous-actions simultanées. L'action *coller le dos au siège* peut ainsi être décrite par un ensemble d'actions élémentaires simultanées, dont chacune concerne un point de contact particulier entre le dos et le dossier du siège. L'action globale *conduire* peut s'appliquer au composant *habitacle* et concerner plusieurs actions élémentaires simultanées : tourner le volant, actionner les pédales, regarder la route.

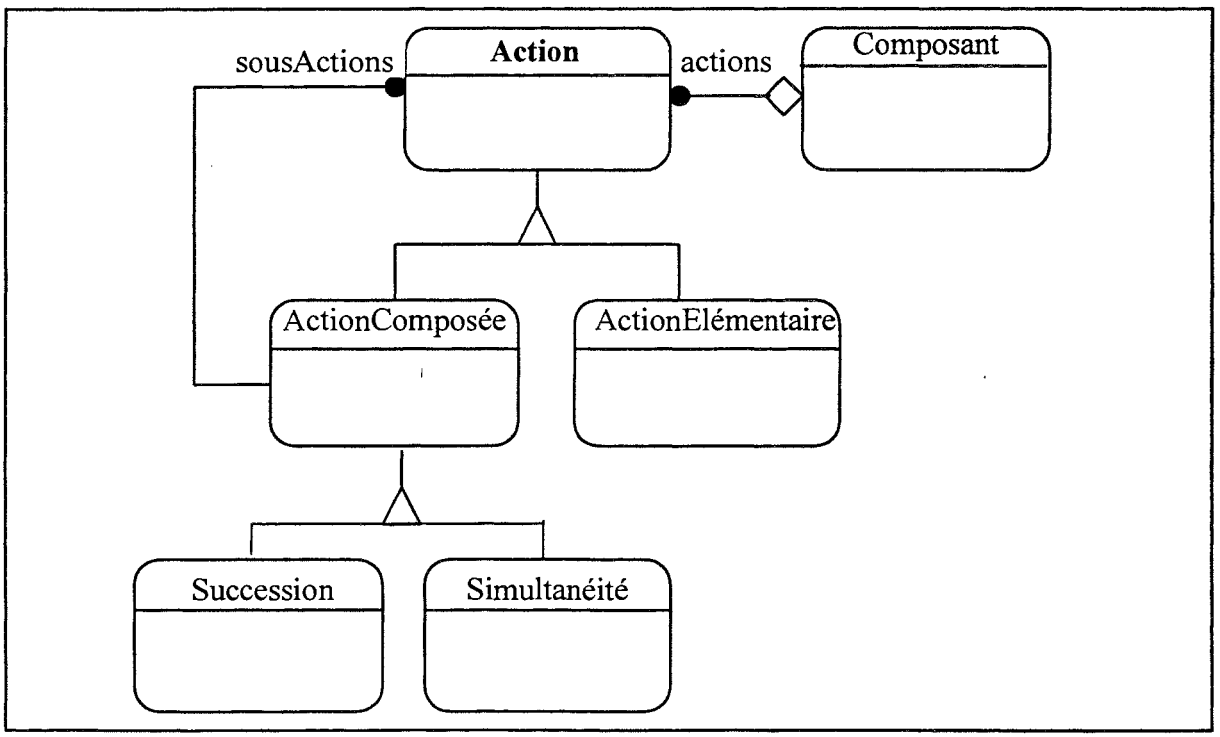

Figure 18 : Action sur un composant

Une action composée correspond à l'agrégation d'un ensemble d'actions pouvant être elles-mêmes composées ou élémentaires (la définition est réflexive). Une succession correspond à un enchaînement de sous-actions, c'est une sorte particulière d'action composée, à laquelle est

associé un ordre sur l'ensemble des sous-actions qui la composent. La simultanéité correspond au cas général où les actions ne sont pas ordonnées.

Enfin, il faut compléter le modèle par le concept de scénario. Un scénario est composé d'actions. En ce sens il est une sorte d'action composée. Deux éléments distinguent un scénario d'une action composée. Tout d'abord, un scénario d'activité n'est pas attaché à un composant particulier, puisqu'il peut concerner l'ensemble de l'environnement considéré. Il est donc logique de le rattacher à l'objet environnement. Par ailleurs, avec un scénario, la notion de temps apparaît : il a une durée totale, qui peut être calculée à partir des durées des actions qui le composent. Cette notion de durée des scénarios et des actions introduit une nouvelle dimension dans le modèle. Du point de vue de l'objectif de diagnostic ergonomique, qui est la raison d'être de ce modèle de l'activité, avec les actions successives ou simultanées, l'idée est de tester la possibilité pour l'opérateur d'effectuer une action motrice donnée, après une autre, ou en même temps qu'une ou plusieurs autres. L'orchestration temporelle de ces actions permet de se rapprocher d'une simulation plus réaliste de l'activité de l'opérateur et de s'intéresser à des tests portant sur des capacités à la fois motrices et cognitives de l'opérateur. En effet, en affectant des durées aux actions, et en introduisant des connaissances sur les temps de réaction (par exemple détection d'une alarme visuelle en fonction de sa localisation), on peut fournir des bases pour effectuer des tests portant sur la pertinence de la localisation des composants, du point de vue de la sécurité. Le test de la capacité d'un opérateur à réagir à un événement extérieur, alors qu'il est occupé à un ensemble de tâches simultanées est un autre exemple de ce type de tests. Ceci nécessiterait de coupler le modèle d'activité motrice à un modèle cognitif, permettant d'intégrer les connaissances sur la charge mentale, les situations de double (ou multiple) tâche, ... Dans un premier temps, on prévoit donc simplement de stocker des dates et des durées, afin de pouvoir plus tard élargir le modèle.

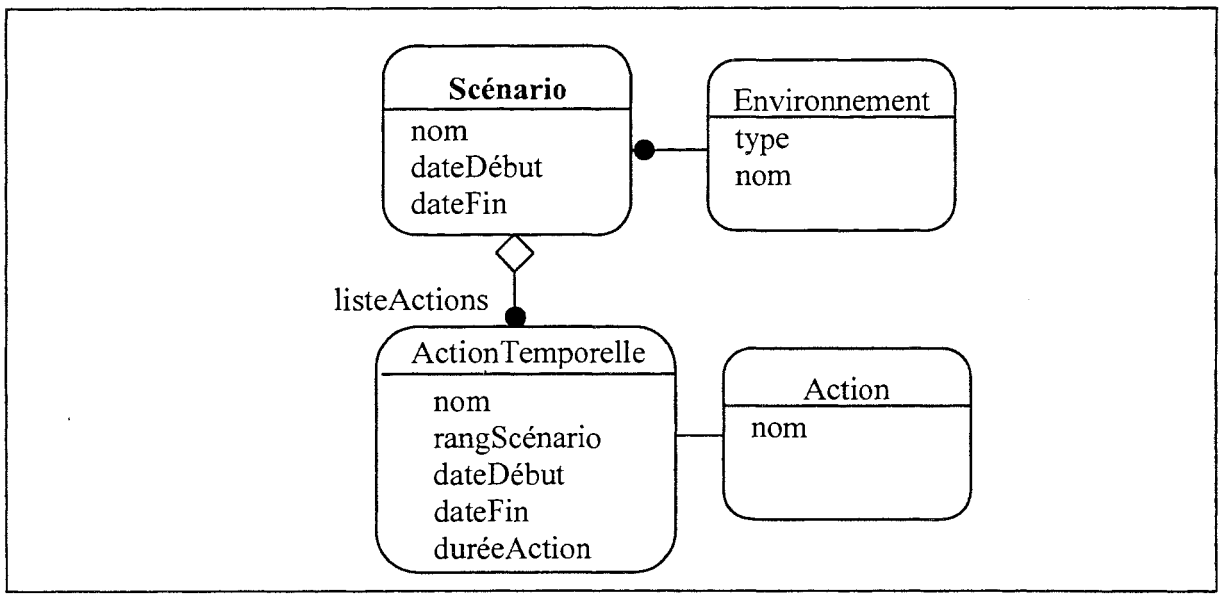

Figure 19 : Scénario d'activité

La Figure 19 décrit un scénario. On stocke les dates de début et de fin, permettant de calculer la durée totale lorsque cela est nécessaire. Un scénario est composé d'actions temporelles elles-mêmes associées chacune à une action. Pour une action temporelle, les dates de début et de fin permettent de positionner l'action dans le temps. L'attribut durée correspond à la durée effective de l'action (durée de l'appui sur un bouton par exemple), tandis que l'écart entre la date de début et la date de fin correspond à l'addition de la durée nécessaire pour réaliser

l'action (le ou les contacts spécifiés) et de la durée associée au but atteint. Il faut souligner en effet que notre modèle initial correspond à des actions ponctuelles (éventuellement ordonnées dans le temps pour le cas des actions successives). On ne gère pas la continuité, c'est-à-dire le passage d'une action à une autre. L'ajout au modèle que nous faisons avec la notion de scénario temporel implique la prise en compte du temps nécessaire pour passer d'une action (correspondant à une posture et à un type d'interaction avec l'environnement) à une autre.

4.2.4 Description symbolique des postures

Nous avons vu précédemment (chapitre 1), qu'une posture au sens d'un modèle graphique d'opérateur est un ensemble de valeurs d'angles décrivant la position de chacune des articulations entre les différents segments composants le squelette. Nous avons défini au paragraphe 4.2.2.1 le concept d'opérateur générique, permettant de disposer d'une description symbolique de l'opérateur, afin de pouvoir désigner les parties de l'opérateur et leurs états de manière indépendante de leur réalisation au niveau du modèle graphique. Nous avons indiqué au paragraphe 4.2.2.3 de quelle manière on définit un état de partie. Nous fournissons ici l'illustration d'une application possible de ces principes, avec un essai de nomenclature des types d'états parties et des différents états des parties possibles.

Le tableau 2 fournit les éléments de cette nomenclature, pour chaque partie du corps.

Ainsi, la posture de la main est décrite par trois types d'états. Le premier exprime l'attitude générale de la main (en fait l'attitude des doigts). Ce paramètre permet de spécifier des "types de prise". Les deux autres paramètres décrivent la position de la main par rapport au bras et concernent donc l'articulation du poignet, avec deux degrés de liberté, la flexion/extension et la rotation.

La posture du bras est décrite par quatre paramètres. Le premier décrit la flexion/extension de l'articulation du coude. Les trois autres paramètres décrivent la position du bras par rapport au reste du corps et concernent donc l'articulation de l'épaule.

La posture de la jambe est décrite de manière similaire, avec l'articulation du genou pour l'attitude générale, et trois axes de mobilité de la hanche pour la position par rapport au reste du corps.

Pour l'ensemble tête/cou, on ne retient que 3 paramètres : la flexion du cou, la torsion, et la latéro-flexion.

La posture du buste est aussi décrite d'une part par une attitude générale, d'autre part par sa position par rapport au bassin, avec 3 autres paramètres.

Les dessins (Figure 20 à Figure 22) illustrent quelques-unes des différentes modalités associées à ces paramètres de description.

Cette nomenclature est fournie à titre d'ébauche. Elle est encore incomplète et sujette à révisions. Il faut souligner notamment qu'elle n'est pas exhaustive, dans la mesure où certaines postures ne pourraient être décrites par cette nomenclature. Par ailleurs, son utilisation implique la définition précise des axes de mobilité concernés et l'identification des zones d'angles associées à chaque item de description d'un état de partie. Ceci permettra de

réaliser le module de traduction des postures, ayant pour tâche de traduire des valeurs d'angles en items posturaux.

Partie				
Main (D ou G)	**Attitude générale** ouverte repliée fermée appuiIndex prisePince	**Flexion poignet** prolongementBras flexionPoignet extensionPoignet	**Rotation poignet/bras** paumeEnBas penchéeInterieur verticale penchéeExterieur paumeEnHaut	
Bras (D ou G)	**Flexion coude** tendu fléchi replié	**Position % buste** derrièreBuste niveauBuste ouvertureLatérale devantBuste opposition	**Position Hauteur** verticalBas diagonalBas horizontal diagonalHaut verticalHaut	**Rotation Epaule**
Jambe (D ou G)	**Flexion genou** tendue pliée repliée	**Position % buste** derrièreBuste niveauBuste ouvertureLatérale devantBuste opposition	**Position Hauteur** verticaleBas diagonaleBas horizontale diagonaleHaut verticaleHaut	**Rotation Hanche**
Tête/Cou		**Flexion latérale** oreilleEpauleD pencheéD milieu penchéeG oreilleEpauleG	**Flexion centrale** flexionComplète inclinéeAvant droite mentonRedressé extensionComplète	**Rotation** rotationCompG rotationG milieu rotationD rotationCompD
Buste	**Attitude générale** droit dosRond	**Position % bassin** avantBassin milieu arrièreBassin	**Position Hauteur** inclinéBas diagonalBas horizontal diagonalHaut vertical	**Rotation Taille**

Tableau 2 : Description symbolique des postures

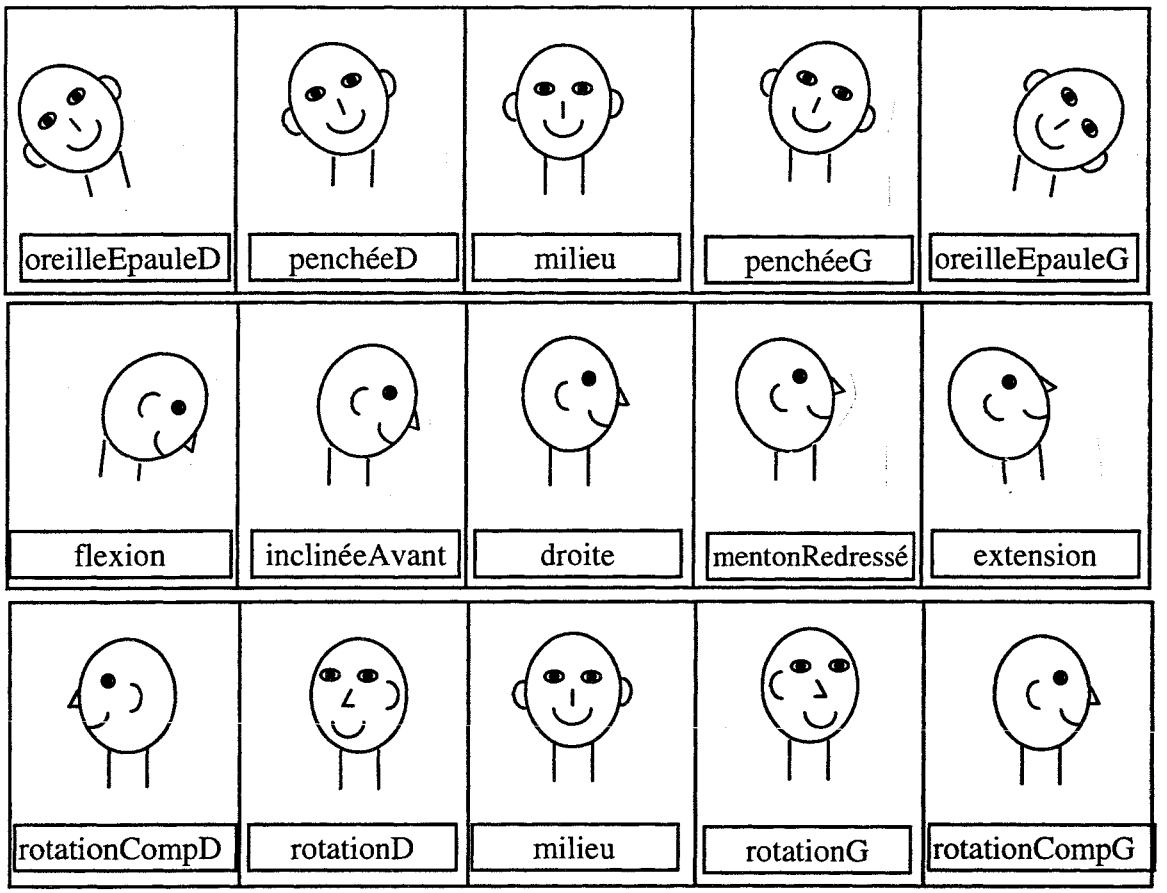

Figure 1 : Postures tête/cou

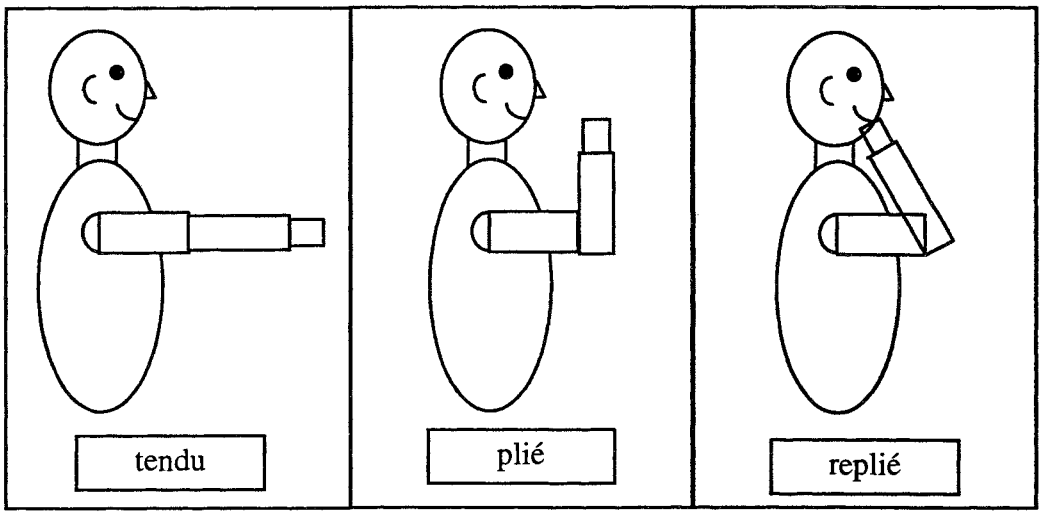

Figure 2 : Attitude générale Bras

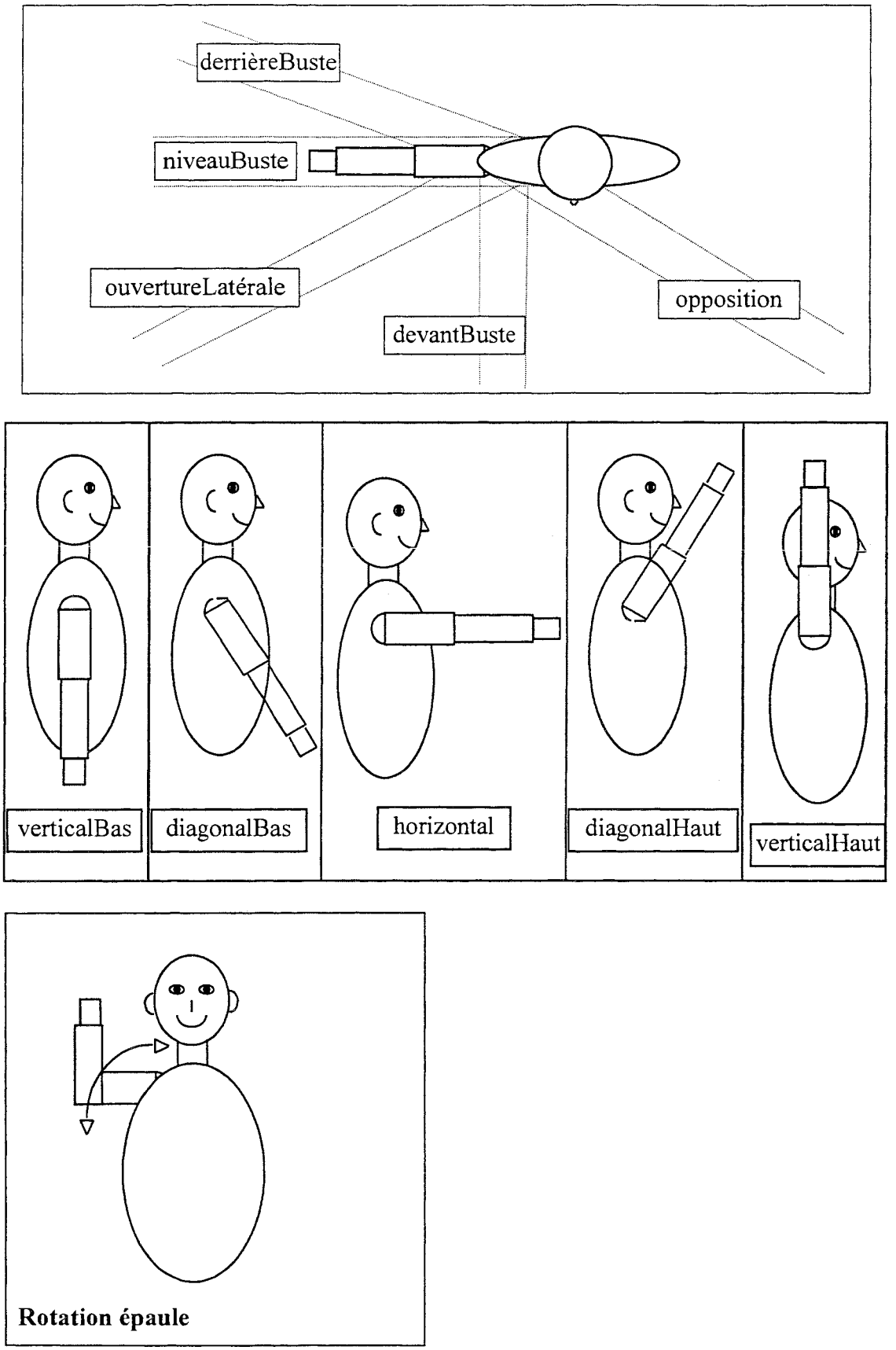

Figure 22 : Positionnements Bras

4.3 Les stratégies de mouvement

L'étude des stratégies de mouvement dans le contexte de la simulation de l'activité pour l'aide à l'évaluation ergonomique concerne principalement les problèmes d'atteinte et de déplacement. Par exemple dans le cas de la conception d'une cabine de poids-lourd, on doit pouvoir simuler l'accès à la cabine (atteindre une poignée pour se hisser, monter des marches, ouvrir la porte, entrer dans l'habitacle), l'atteinte des commandes en situation de conduite, la circulation dans l'habitacle.

La posture globale (assis, debout, couché) est un élément déterminant pour les stratégies de mouvement. En effet, elle peut influer sur les stratégies d'atteinte : par exemple pour se rapprocher d'un point lorsque l'on est debout, on peut éventuellement plier les genoux (alors que ceci n'a pas de sens en position assise). La stratégie porte ici sur le choix de la zone corporelle à activer. La posture globale influe aussi sur les possibilités articulaires, c'est-à-dire sur les limites acceptables : par exemple la limite maximale d'inclinaison du buste pour l'atteinte n'est pas la même en position debout ou assise.

Enfin, que l'on soit assis, debout, ou couché, les stratégies de mouvement adoptées pour atteindre (un lieu, un objet) dépendent à l'évidence principalement de la localisation du point à atteindre.

En ce qui concerne la littérature du domaine, il existe de nombreuses approches s'intéressant à l'analyse de mouvements simples impliquant une ou deux articulations, notamment pour le bras. On pourra se reporter par exemple à (52) qui rassemble de nombreuses contributions sur le sujet. Par contre, le cas des mouvements complexes impliquant le corps entier ou même seulement la coopération de différents membres a peu fait l'objet de publications. Il existe cependant quelques tentatives d'approches plus globales. Dans (53) on trouve un regroupement de différentes contributions à l'analyse du mouvement humain, avec l'objectif de rassembler des résultats de disciplines diverses.

Nous allons évoquer ici les deux cas dont le traitement a été abordé dans le cadre du travail exposé ici : les postures debout et assise. Notons qu'en l'absence d'étude approfondie sur le sujet (à notre connaissance), et d'expert en "stratégies de mouvements" à interroger, l'approche adoptée a été de nature heuristique. L'idée était de produire des règles heuristiques, au sens où elles puissent fournir une aide pour la simulation du mouvement, sans pour autant prétendre être toutes "prouvées" en amont. Nous évoquons brièvement ci-dessous les travaux plus formels menés au LESCO dans le but d'identifier des attitudes posturales précises, dans des cas spécifiques définis dans le cadre de procédures expérimentales et/ou d'approches numériques.

Un autre aspect a été abordé : celui des stratégies de mouvement lorsque plusieurs actions doivent être réalisées en même temps. Il s'agit d'identifier les cas possibles de conflit et d'établir des règles de priorité. Le travail d'analyse sur cet aspect n'étant qu'ébauché, il n'en sera pas fait état ici.

4.3.1 Le cas de la position assise

Un certain nombre de règles intuitives peuvent être imaginées lorsque l'on cherche à identifier des stratégies pour l'atteinte. Par exemple on peut définir les règles suivantes :

1. *Si l'objet à atteindre est à portée de main, ou bien si je n'arrive pas à évaluer son éloignement, alors j'essaie de l'attraper en tendant le bras et sans bouger le reste.*
2. *Si je n'ai pas le bras assez long, j'incline le buste et je tend le bras, sans bouger le reste.*
3. *Si en me penchant et en tendant le bras je n'arrive pas à atteindre l'objet, alors je m'avance sur mon siège.*

Ce groupe de règles illustre tout d'abord la nécessité de définir des priorités entre zones corporelles activables. On décide donc que l'atteinte avec le bras seul est prioritaire par rapport à l'inclinaison du torse, et que l'inclinaison du torse est prioritaire par rapport à l'avancée des fessiers.

Par ailleurs, la localisation relative de l'objet par rapport aux capacités fonctionnelles liées à l'atteinte détermine le choix de la zone corporelle à activer : l'objet à atteindre est ou non "à portée de main", le bras est "assez long" ou non. Dans un contexte de simulation, où l'on connaît la longueur du bras (et du torse) et où l'on peut mesurer la distance entre la main et le point à atteindre, l'information du caractère "atteignable" de l'objet peut être calculée à priori : si la distance entre le centre de l'épaule et le point à atteindre est supérieure à la longueur du bras déplié, alors le point ne peut pas être atteint en n'activant que le bras ; si la distance entre le bassin et le point à atteindre est supérieure à la longueur du bras déplié plus le torse, alors le point ne peut pas être atteint en n'utilisant que le bras et le torse. Ceci suppose l'accès à une représentation géométrique de l'environnement.

On pourrait donc imaginer une stratégie générale basée sur la connaissance géométrique de la localisation de l'objet et sur le calcul de la distance à "parcourir". Mais si l'on se replace dans le contexte du travail exposé ici, il faut tenir compte des possibilités offertes par MAN3D. L'architecture du système ADELE ne comporte pas la représentation géométrique de l'environnement. Cette représentation existe au niveau du modèle MAN3D. On pourrait imaginer alors que MAN3D fonctionne pour ADELE comme un "serveur" d'informations géométriques. Cette solution n'est pas celle qui a été adoptée ici, car elle supposait trop de transformations au niveau de MAN3D.

Le raisonnement conçu pour mettre en œuvre des stratégies de mouvement a donc été bâti en fonction de ce qu'est MAN3D : un automate doté d'un langage de communication, auquel on peut demander d'activer une zone corporelle à la fois (bras, jambe, torse, bassin, tête) et de réaliser une atteinte de point, et qui produit un diagnostic sur l'action réalisée.

Les messages produits par MAN3D concernant les actions d'atteinte sont : « point hors d'atteinte » et « articulation *nom_articulation* en butée », pouvant être émis éventuellement simultanément comme résultats d'une action.

Le message « articulation *nom_articulation* en butée » impose le choix d'une stratégie alternative pour réaliser l'action. En effet, il indique qu'une butée articulaire est atteinte mais pas nécessairement que le point est hors d'atteinte. Ce message est associé aux articulations de l'épaule, du coude, du poignet, de la hanche, du genou, de la cheville, du cou, du bassin et des doigts.

Le cas de l'épaule a été étudié de manière approfondie au LESCO ((54), (55), (56)), où deux thèses ont même été consacrées à ce sujet. Le problème posé est celui de la détermination du degré de pivotement[10] de l'articulation nécessaire à la réalisation de l'atteinte d'un point spécifié. En effet, dans les mouvements de faible amplitude se déroulant dans la partie centrale de la zone d'atteinte, il est rare que l'on observe un pivotement de l'épaule. En revanche, dans certains cas, le pivotement s'avère indispensable pour éviter que le bras soit bloqué par la butée articulaire de l'épaule. Le problème est de déterminer ces cas, et de concevoir une méthode permettant de calculer le degré de pivotement nécessaire. Plusieurs approches menées au LESCO dans ce but ont en commun l'aspect expérimental : des conditions expérimentales sont conçues, afin de pouvoir faire réaliser à un échantillon de sujets un ensemble de tests d'atteinte. Les positions réalisées par les sujets sont enregistrées par des moyens vidéo permettant à posteriori de stocker numériquement (et très précisément) les postures du bras adoptées, et donc le degré de pivotement associé à chaque geste d'atteinte. Sur ce type de recueil, plusieurs méthodes de détermination du pivotement ont été envisagées. La régression linéaire utilisée dans un premier temps (55) n'a pas été jugée complètement satisfaisante, car elle ne permettait pas d'expliquer de manière fiable le pivotement pour l'ensemble du volume d'atteinte du bras. Une approche plus systématique a ensuite été proposée (56). Elle consiste à faire réaliser par les sujets des tests d'atteinte sur la totalité d'un maillage assez fin de la sphère d'atteinte du bras. Ceci constitue une base de cas, sur laquelle on peut se baser pour calculer un degré de pivotement pour un nouveau point test. Une autre approche, de type algorithmique, a été étudiée (57) : elle consiste lors de la simulation d'un geste d'atteinte, à introduire (et donc à calculer) un degré de pivotement uniquement en réponse à un problème de butée du bras (par exemple butée du bras avec le torse lors d'un essai d'atteinte de l'épaule gauche par la main droite).

Dans l'attente de l'intégration d'une méthode fiable de calcul du pivotement au modèle MAN3D, il faut prévoir que ADELE gère la réponse au problème de l'articulation en butée. Plusieurs stratégies sont envisageables :

- à partir des cas de Brunel, découper la sphère d'atteinte en zones, et affecter une valeur moyenne de pivotement à chaque zone, puis depuis ADELE, identifier la zone dans laquelle se trouve le point à atteindre et utiliser la commande de pivotement de MAN3D avec la valeur adéquate. Cette méthode pose différents problèmes : tout d'abord la détermination des zones n'est pas un problème trivial car il apparaît que les frontières de changement de stratégies posturales en termes de pivotement ne sont pas du tout régulières ; par ailleurs, cette méthode s'éloigne un peu de la réalité puisqu'elle fournit une valeur moyenne (et qu'elle risque donc de produire des valeurs non réalistes dans certains cas).
- récupérer directement une valeur de pivotement calculée en amont à partir des cas de Brunel, à l'aide d'une méthode (en cours de développement) garantissant plus le réalisme du résultat.
- en cas de butée articulaire sur l'épaule, replacer le bras dans une position intermédiaire (en terme de pivotement de l'épaule), depuis laquelle on sait que tous les points de la sphère d'atteinte du bras sont atteignables, puis refaire la commande d'atteinte. Cette méthode est efficace en terme de levée de la butée articulaire, mais le degré de réalisme de la posture obtenue n'est pas connu. Elle permet néanmoins de sortir de l'impasse.

[10] Le pivotement correspond à la rotation du bras entier autour d'une droite reliant le point atteint par l'extrémité du bras au centre de l'épaule.

Les deux premières méthodes imposent que ADELE puisse travailler sur la représentation géométrique de l'espace. Compte tenu des contraintes évoquées plus haut (en terme de transformations de MAN3D), nous adoptons donc pour le prototype la troisième méthode.

On prévoit dans un premier temps de gérer de la même manière les butées articulaires concernant les autres articulations.

Le message « point hors d'atteinte » renvoie à la question du choix du membre à activer, que nous choisissons de résoudre à l'aide de la définition d'un ordre de priorité sur l'ensemble des membres activables. Notons qu'il faut aussi gérer la possibilité d'activer ou non un membre, en fonction des contraintes associées à l'action que l'utilisateur veut simuler (par exemple, dans le cas d'une action qui doit être effectuée avec le dos collé au siège, le torse n'est pas activable).
La méthode que nous adoptons ici consiste donc, pour une atteinte avec le bras, à activer le torse (pour l'incliner en direction du point) lorsque l'atteinte n'est pas possible avec le bras tendu (message « point hors d'atteinte » obtenu après une tentative d'atteinte avec le bras). Ceci implique que l'on fait l'hypothèse que lorsque le point est dans la sphère d'atteinte du bras, il n'y a pas de participation du torse (qui participerait donc seulement à partir du moment où le point se trouve hors de la sphère d'atteinte du bras seul). Cette hypothèse n'est pas vérifiée et est intuitivement sujette à caution.

En parallèle avec le travail que nous exposons ici, un autre travail de thèse a été entrepris [Santos 97] dans le but de déterminer notamment (entre autres) les participations respectives du torse et du bras pour un geste d'atteinte. Là encore, l'approche est expérimentale. Un échantillon de sujets (de morphologies différentes) a été placé en conditions expérimentales : un ensemble de cibles, disposées sur plusieurs plans selon différentes directions, à atteindre en position assise sans contrainte sur le torse (mais en conservant les fessiers collés au siège). Le recueil des postures obtenues permet d'étudier la participation du torse et du bras au mouvement d'atteinte, pour des cibles situées hors de l'atteinte du bras seul. Le traitement de ces données est encore en cours. On peut néanmoins dégager quelques tendances. Tout d'abord, le fait qu'il n'y ait pas eu de tests sur des cibles situées à l'intérieur de la sphère d'atteinte du bras ne permet pas de répondre directement à la question de la participation du torse au mouvement d'atteinte dans le cas où cela n'est pas indispensable. (Ceci pourra faire l'objet d'autres expérimentations). Les résultats obtenus sur les cibles situées hors de la sphère d'atteinte du bras seul tendent cependant à laisser supposer que le torse participe au mouvement avant d'atteindre la limite de la sphère d'atteinte du bras seul. En effet, on observe pour ces cibles que le bras n'est jamais tendu.

La stratégie d'activation des membres pour une atteinte en position assise doit commencer par une identification du cas, avec un premier essai d'atteinte. Si le point est "hors d'atteinte", c'est qu'il se trouve en dehors de la sphère d'atteinte du bras. Il faut donc faire appel au torse (en supposant que l'inclinaison du torse est prioritaire sur l'avancée des fessiers). Si par contre le point est dans la sphère d'atteinte, il faut vérifier que l'angle du coude est bon. Or pour l'instant, nous ne disposons pas de valeurs d'angles références permettant de juger du réalisme d'un geste d'atteinte. Le problème est similaire à celui du degré de pivotement de l'épaule. Il est vraisemblable qu'une valeur d'angle moyenne ne pourrait pas être satisfaisante pour l'ensemble de la sphère d'atteinte. Il existe cependant peut-être des zones sur lesquelles l'écart-type sur les valeurs d'angle pour la flexion du coude est faible, et où il est donc pertinent de travailler avec la moyenne. Ce travail d'identification des bonnes valeurs de

flexion du coude reste à faire. Dans un premier temps, nous utiliserons une valeur d'angle qui sera appliquée systématiquement, de manière à éviter le bras complètement tendu.

Le problème est le même pour l'inclinaison du torse : en l'absence de plages de valeurs de référence, on commencera par utiliser de petites valeurs de flexion, de manière incrémentale.

4.3.2 Le cas de la station debout

Dans le cas de la position debout, les stratégies adoptées pour se rapprocher d'un point diffèrent nécessairement de la position assise, puisque dans ce cas les jambes font partie des membres activables (au contraire de la position assise pour un geste d'atteinte avec le bras). Notons que l'action « se déplacer jusqu'à ... » peut être formalisée comme un geste d'atteinte, cette fois avec la plante du pied par exemple, et en imposant comme contrainte la position debout.

Il faut préciser aussi que la manière dont le déplacement est effectué ne fait pas partie de notre problématique (au sens par exemple de la simulation de la marche). Ce qui est important dans le cadre de l'évaluation ergonomique, ce sont éventuellement les obstacles présents sur le parcours. Par exemple, dans le cas de la cabine de poids-lourd, on peut être amené à tester la possibilité pour le conducteur de se déplacer à l'intérieur de l'habitacle, l'une des questions que se pose le concepteur pouvant être par exemple : l'utilisateur ne va-t-il pas être gêné par le volume situé au centre de la cabine ? Il s'agit donc, au niveau de la simulation, de pouvoir identifier les collisions possibles entre le mannequin et les objets de l'environnement. En l'absence de module géométrique de calcul de collision, ce problème ne peut pas être pris en charge.

Au-delà de cette question, la gestion des stratégies de mouvements pour la position debout impose des choix d'activation de membres en fonction de l'éloignement du point à atteindre, ainsi qu'en fonction de sa hauteur par rapport au sol. En effet, si le point est trop éloigné, il faut s'en rapprocher tout d'abord, puis, en fonction de la hauteur, il faut éventuellement tendre le bras et incliner le torse et/ou plier les genoux, ou bien au contraire se hisser sur la pointe des pieds. Plusieurs questions se posent donc : sur quel critère évaluer l'éloignement du point ? quelles sont les actions alternatives réalisables en fonction de l'éloignement ?, si l'on se trouve dans le cas où le point a été jugé trop éloigné pour une atteinte directe sans déplacement du corps entier, jusqu'où doit-on se rapprocher ? enfin, lorsque le rapprochement adéquat a été réalisé, de quelle manière les jambes, les pieds, les bras, le torse doivent-ils participer au mouvement ?

Afin de tenter de trouver des réponses à ce type de question, une expérimentation a été réalisée dans le cadre du travail de thèse évoqué ci-dessus (58). Il s'agissait pour les sujets d'atteindre des points matérialisés sur un mur, à partir d'une position de départ éloignée du mur (c'est-à-dire imposant un rapprochement). Les cibles à atteindre étaient réparties depuis le sol jusqu'à 2 mètres de hauteur. Chaque cible faisait l'objet de tests séparés (avec retour préalable au point de départ). Pour chaque cible ont été effectués des tests instantanés (consigne de simple atteinte) et des tests avec durée (atteinte avec position à conserver, dans un cas pendant 10 secondes, dans un autre pendant 20 secondes). Bien que cette étude ait été plutôt exploratoire (elle n'a été réalisée que sur trois sujets), quelques tendances ont pu être observées : pour les tests instantanés, la tendance est plutôt d'incliner le torse que de plier les jambes, la position accroupie n'étant utilisée que pour les points les plus bas, dans le cas de

tests avec durée. Pour ce qui est du placement par rapport au mur, pour les points les plus hauts, il y a un rapprochement tout près du mur ; on observe par ailleurs un décalage plus ou moins important des pieds l'un par rapport à l'autre.

4.4 Le diagnostic ergonomique

Le diagnostic ergonomique comporte plusieurs volets. Il s'applique tout d'abord à une action spécifique, à l'issue d'une tentative de simulation de cette action. Il s'agit de porter un diagnostic sur la posture obtenue, en précisant tout d'abord si l'action est possible dans les conditions spécifiées par le cahier des charges ergonomique (par exemple atteindre une commande en conservant le dos collé au siège). Il faut aussi qualifier la posture en terme de confort, car si une posture est possible physiologiquement, elle peut aussi être inconfortable et donc non envisageable. Ceci peut aussi fournir au concepteur un critère de décision pour le choix entre plusieurs implantations concernant un même composant.

Le diagnostic ergonomique peut aussi porter sur l'ensemble de la maquette. Cet aspect est beaucoup plus spécifique au produit conçu. Par exemple pour le cas de l'habitacle de poids-lourd, on peut fournir des indicateurs sur un certains nombre de rubriques, telles que l'accessibilité, l'habitabilité, le confort, la manœuvrabilité, la conduite ... .

Les paragraphes suivants évoquent ces différents aspects.

4.4.1 Le respect du cahier des charges ergonomique

Le « cahier des charges ergonomique » désigne l'ensemble des spécifications et contraintes liées à la simulation de l'utilisation d'un composant.

Le "mode d'utilisation" du composant comporte différents aspects :

- la désignation du membre à utiliser (main par exemple pour un bouton, regard pour un voyant, pied pour une pédale, ...),
- l'attitude posturale du membre concerné,
- l'attitude générale du corps (position assise en général pour les composants liés à la conduite).

Les contraintes portant sur les attitudes posturales sont traduites ici par l'introduction de conditions dans la définition d'une action sur un composant, portant soit sur l'état de l'opérateur, soit sur l'état des parties de l'opérateur.

Le cahier des charges ergonomique peut aussi comporter d'autres contraintes posturales liées au mode d'utilisation de l'objet : ainsi certains composants doivent être utilisés sans décoller le dos du siège. Ces contraintes sont aussi traduites par l'ajout de conditions à l'action, portant cette fois sur les contacts entre opérateur et environnement.

La Figure 23 décrit le concept d'action ainsi complété.

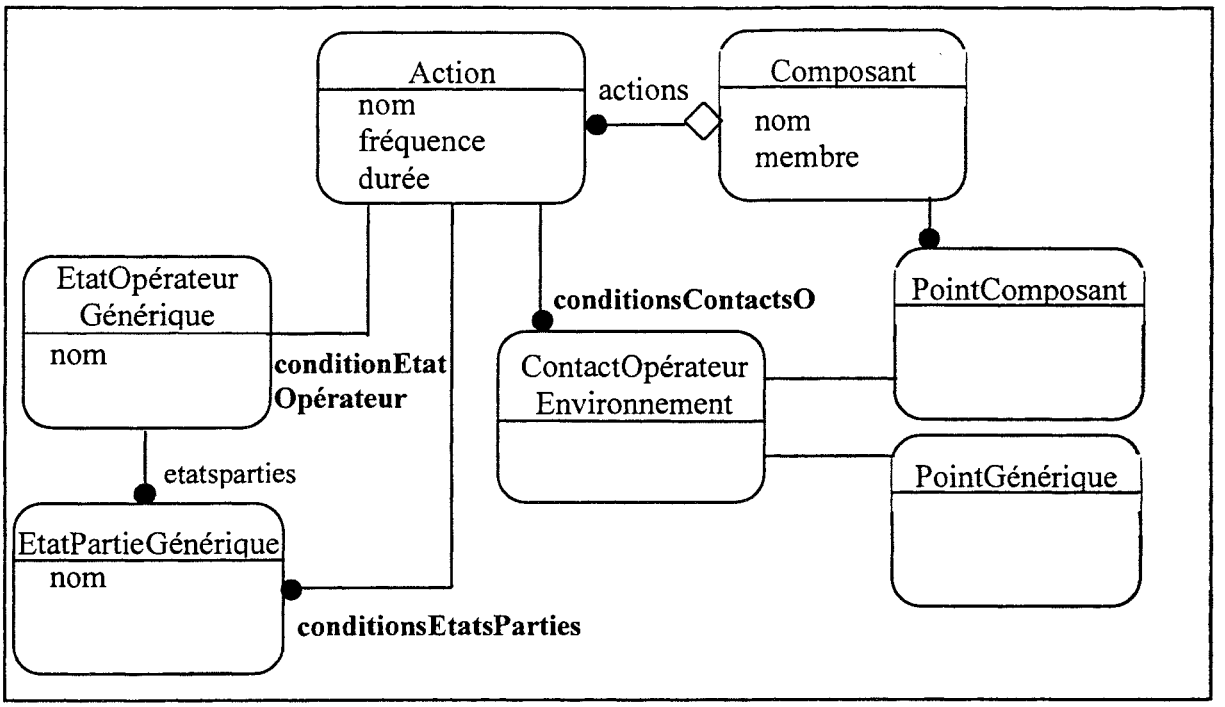

Figure 23 : Conditions associées à une action

On voit aussi sur la Figure 23 que deux autres attributs ont été rajoutés à l'action effectuée avec un composant : il s'agit de la fréquence de l'action et de sa durée moyenne. Ces attributs participent à l'élaboration du qualificatif associé à la posture obtenue : par exemple une posture inconfortable pour une action donnée sera d'autant moins acceptable que cette action est fréquente.

4.4.2 Le confort de la posture

Le thème du confort postural est un des axes de recherche du LESCO. Dans le cadre de recherches visant à élaborer des outils d'aide à la conception ergonomique d'habitacles de véhicules, un premier travail de thèse a été mené (59). Nous exposons ci-dessous ce travail, de manière relativement détaillée, afin de dégager quelques tendances utilisables dans le cadre de notre travail et afin aussi d'expliciter la complexité du problème de la détermination du confort postural, et de souligner l'importance des moyens mis en œuvre pour répondre à des questions sur ce thème.

L'objectif du travail de Serge Ribouchon était d'élaborer une méthode d'analyse de la posture assise, afin de recueillir des paramètres objectifs permettant d'expliquer le confort. Bien que partant du constat que la sensation de confort résulte de la combinaison de nombreux facteurs (physiques, sociologiques, situationnels, psychologiques, physiologiques, ...), l'étude a été centrée sur les facteurs physiques, et plus particulièrement sur les facteurs spatiaux. En effet, dans un véhicule, les facteurs physiques influent fortement sur la sensation de confort. Ces facteurs incluent les facteurs dynamiques tels que les accélérations et le changement de vitesse du véhicule, les facteurs ambiants (pression, niveau sonore, température, humidité, ventilation, ...) et les facteurs spatiaux (géométrie du poste de conduite, forme et réglage du siège).

L'étude a donc été centrée sur la posture assise et sa variabilité chez un même individu et entre les individus. Une expérimentation sur siège multi-réglable a été réalisée. La situation reproduite était celle de la conduite. La consigne aux sujets était de rechercher à l'aide des réglages une position la moins inconfortable possible, avec les mains sur le volant, et le regard fixé sur un point symbolisant la route. Puis les sujets devaient de nouveau appliquer la même consigne, mais sous contrainte de hauteur d'assise, et en cotant la qualité de la posture obtenue par une note (comprise entre 0 et 10). Il faut souligner que la contrainte sur la hauteur d'assise était posée afin de se rapprocher des conditions réelles de conduite, pour lesquelles il existe des contraintes pesant sur ce paramètre : hauteur du plafond pour les véhicules légers, imposant des hauteurs d'assise inférieures à 300 mm, visibilité pour les poids-lourds, imposant des hauteurs d'assise supérieures à 450 mm. Chaque cas issu de l'expérimentation a été décrit par trois ensembles de variables : les variables anthropométriques (40 variables), les variables décrivant la géométrie du poste de conduite (inclinaison du dossier, inclinaison de l'assise, hauteur du siège, hauteur, inclinaison et avance du volant, inclinaison et avance du repose-pieds, hauteur et avance de l'appui lombaire, hauteur et avance de l'appui dorsal) et les variables décrivant la posture (positions de mires externes et de palpeurs, et variables calculées : points articulaires et angles posturaux). Ces résultats issus de l'expérimentation (585 postures au total, pour 45 sujets, 4 cas de posture optimale ainsi que 9 cas de posture sous contrainte par sujet) ont été stockés dans une base de données relationnelle.

Les résultats de l'expérimentation ont été analysés tout d'abord dans le but de déterminer la posture optimale (dite « de moindre inconfort ») et sa variabilité au sein de la population. Outre la production des moyennes et des écarts-types des différentes variables (pour l'ensemble des sujets ainsi que par sujet), cette analyse a été effectuée à l'aide de techniques d'analyse de données :

- analyse en composantes principales (ACP) sur 10 angles de flexion, qui a mis en évidence la variation de la posture au sein de l'échantillon étudié, et qui a permis de définir des groupes indépendants d'angles corrélés entre eux et de les hiérarchiser,
- classification ascendante hiérarchique (CAH) à partir des mêmes angles, afin d'obtenir des classes de postures et de les décrire ensuite à l'aide des variables anthropométriques. Cette analyse fournit des pistes intéressantes sur la correspondance entre les mesures anthropométriques d'un individu et la posture adoptée, mais la faiblesse de l'échantillon n'a pas permis de tirer des conclusions précises et définitives.

Sur les postures libres, la tendance est à la faible variabilité pour un même sujet.

L'analyse des situations obtenues sous contrainte de hauteur d'assise a permis tout d'abord de fournir les valeurs moyennes (et les écarts-types) pour les paramètres recueillis, notamment les moyennes des réglages du poste de conduite obtenus pour les différentes hauteurs d'assise testées, ainsi que les moyennes des angles posturaux observés pour chaque hauteur d'assise. Des techniques d'analyse de données ont également été utilisées :

- une CAH, effectuée sur les variables angulaires pouvant être affectées par la hauteur d'assise (identifiées à partir des moyennes). Cette analyse a fait apparaître les différences posturales entre les individus (comme l'analyse effectuée sur les postures libres), mais n'a pas permis de déterminer les différentes formes de compensation induites par la hauteur d'assise. Elle a également montré que les variations posturales interindividuelles étaient plus fortes que les variations intra-individuelles induites par la hauteur d'assise.
- une autre CAH, effectuée cette fois non pas sur les valeurs angulaires brutes, mais sur les écarts avec la posture moyenne sans contrainte (par sujet), permettant cette fois

d'analyser le lien entre le comportement postural et la variation de la hauteur d'assise (en éliminant l'effet inter-sujets). Cette analyse fait apparaître une forte distinction entre les postures hautes et basses, tandis que les hauteurs intermédiaires ne font pas apparaître de stratégies particulières d'adaptation à la hauteur d'assise. On constate notamment que la contrainte de hauteur d'assise haute engendre en moyenne une variation importante des flexions de la hanche et du genou.

Enfin, d'autres analyses ont été effectuées, dans le but de définir un modèle permettant de représenter l'évolution de la posture en fonction de la hauteur d'assise :

- une ACP effectuée sur les écarts calculés entre les angles posturaux observés sous contrainte de hauteur d'assise et les angles observés sur les postures libres, avec une nouvelle variable pour la hauteur d'assise : l'écart entre la hauteur fixée pour chaque situation sous contrainte et les moyennes des 4 hauteurs d'assise libre,
- l'analyse des notes fournies par les sujets pour qualifier les postures obtenues ; en croisant cette note avec la hauteur d'assise, on constate que l'éloignement de la posture optimale provoque une insatisfaction : en partant d'une note de satisfaction maximale (note proche de 9, pour une hauteur d'assise proche de 300 mm), on voit que la note diminue fortement lorsque la hauteur d'assise augmente et qu'elle diminue aussi, mais moins fortement lorsque la hauteur d'assise diminue. On observe cependant un fort écart-type entre les individus pour les notes exprimées. Par ailleurs, l'étude de la corrélation entre la note exprimée et la somme des valeurs absolues des écarts angulaires entre situation sous contrainte et situation libre s'est révélée intéressante : il semble qu'il existe une relation linéaire entre la sensation de confort postural et la somme calculée.

Nous pouvons tenter d'utiliser les résultats du travail de Serge Ribouchon, en essayant de dégager les thèmes sur lesquels il est possible de dégager des tendances, afin de fournir des indicateurs de confort (même s'il faut considérer que certains résultats doivent faire l'objet d'approfondissements).

Il faut préciser tout d'abord que les résultats obtenus sur les situations libres ont déjà été intégrés dans un module du modèle MAN3D, le module de détermination de la géométrie posturale sous contraintes. Ce module sert à calculer une posture, à partir de la donnée de contraintes de position et/ou d'orientation des éléments du squelette ou des contours (par exemple, on peut indiquer comme contraintes les positions simultanées des mains sur le volant, le regard sur la route, les pieds sur les pédales, ...). Le module calcule la posture finale du corps entier, de manière itérative, à l'aide d'un algorithme d'optimisation sous contraintes. Le critère optimisé est l'écart entre la posture obtenue et une posture de moindre inconfort. C'est cette posture moyenne de moindre inconfort qui a été déterminée à partir du travail de Ribouchon. Ce module n'est donc valable que pour la posture assise. De plus, il fonctionne bien dans des configurations pas trop éloignées de la posture de référence. Enfin, le temps de calcul se révèle parfois beaucoup trop important. Ce module ne fait donc pas partie des modules de MAN3D utilisés dans notre travail (comme nous l'avons précisé plus haut, nous n'utilisons que les modules de calcul de MAN3D attachés aux segments mobiles (bras, jambes, tête, torse)).

Un autre travail a été mené sur le thème du confort postural, dans le cadre de la thèse déjà évoquée dans la section 4.3 (58). Dans le cadre de l'expérimentation réalisée sur la position assise (voir paragraphe 4.3.1), il a été demandé aux sujets de qualifier la gêne éventuelle ressentie lors du test d'atteinte de chacune des cibles, par une note d'inconfort (sur une

échelle en 11 points). Les premières analyses montrent que les notes d'inconfort augmentent avec l'éloignement des cibles. Par ailleurs, la durée tend à augmenter l'inconfort : le maintien de la posture pendant une durée de 10 secondes majore la note d'inconfort en moyenne de 2 points. Les résultats sont encore en cours de traitement. La direction explorée est la recherche de corrélations entre l'inconfort exprimé et des paramètres tels que l'écart de la posture obtenue par rapport à une posture de référence, et le calcul des moments gravitaires par rapport aux articulations de l'épaule et du bassin. Il pourrait être utile d'essayer de relier l'inconfort exprimé avec le degré de participation au mouvement du bras et du torse, séparément tout d'abord (comment la note d'inconfort varie-t-elle avec l'inclinaison du torse, avec la flexion du bras ?), puis en examinant le couple bras/torse (le couplage flexion bras / inclinaison torse permet-il d'expliquer la note d'inconfort ?).

Ces travaux menés sur le confort de la posture sont encore exploratoires. Il conviendra d'approfondir l'analyse des données disponibles dans le but d'identifier des règles sur le confort intégrables dans la spécialiste de ADELE en charge de l'évaluation du confort d'une posture obtenue par la simulation. Dans un premier temps, il est possible de proposer des règles qui, sans produire une note de confort, permettraient d'identifier les cas d'inconfort probable. Par exemple :

- *Si l'épaule est en limite de pivotement vers l'arrière, alors la posture est inconfortable,*
- *Si l 'inclinaison du torse est trop importante, alors la posture est inconfortable.*

Ce type de règles peut aussi être modulé par la durée de l'action, son type, sa fréquence, et/ou le type de composant concerné. Par exemple, une forte inclinaison du torse sur le côté peut être tolérée pour un accès bref (du type "saisir un objet dans une boite à gants"), elle devient moins acceptable s'il s'agit d'actionner un levier de vitesse. Une position acceptable si elle est brève ne le sera plus si elle dure. De même, on pourra accepter une position si elle concerne une action rare mais la refuser s'il s'agit d'une action à faire fréquemment. Contrairement aux prédictions de confort basées uniquement sur la géométrie posturale, ces règles sont spécifiques à un type d'activité et/ou à un type d'environnement. Cet aspect rejoint celui du bilan ergonomique, développé dans le paragraphe suivant (4.4.3).

4.4.3 Le bilan ergonomique

Dans le paragraphe précédent on s'intéressait à la qualification des postures en terme de confort. Fournir une indication de confort à propos d'une posture obtenue sur un test d'utilisation de composant, c'est déjà fournir un indicateur de qualité sur l'implantation de ce composant. Plus généralement, toute implantation de fonction peut être qualifiée en termes de confort, de visibilité et éventuellement d'accessibilité et de manœuvrabilité.

On peut établir un certain nombre de préconisations, qui font partie du cahier des charges ergonomique (évoqué au paragraphe 4.4.1). Ainsi, en terme de visibilité, on peut spécifier si un composant doit être visible dans le champ de vision principal (défini en position de conduite), dans le champ périphérique ou au-delà (c'est-à-dire en tournant la tête). Nous avons déjà évoqué des préconisations portant sur l'accessibilité, que nous avons traduites sous forme de contraintes pour la simulation d'une action (exemple d'une commande qui doit être accessible en gardant le dos collé au siège). La manœuvrabilité est traduite dans notre cas par le test de la possibilité d'atteinte de tous les points d'une trajectoire représentant la course du composant, lors de son utilisation (exemple du levier de vitesse).

Un même composant peut faire l'objet de plusieurs types de préconisations. Ainsi, certaines commandes d'un habitacle de véhicule peuvent être fortement contraintes, à la fois en termes de visibilité (champ de vision principal) et d'accessibilité (dos collé au siège). Pour une implantation de composant, il faut pouvoir disposer de règles permettant de juger de la qualité, lorsqu'une ou plusieurs préconisations ne sont pas respectées, c'est-à-dire de règles réagissant à la "dégradation des critères ergonomiques". Dans cette optique, il est possible de fournir a priori des règles de compromis entre critères (par exemple si l'accessibilité et la visibilité sont nécessaires pour une commande, il est possible dans certains cas de spécifier si l'on privilégie l'un ou l'autre des critères lorsqu'il n'est pas possible de les respecter en même temps).

Si l'on est capable de qualifier chaque composant selon des critères fournis a priori, il est alors possible de qualifier globalement l'environnement conçu, pour chacun de ces critères. Il faut définir la manière de synthétiser l'information issue de la qualification de chaque implantation de composant. On peut également s'appuyer sur des règles de priorité, ou bien sur la définition de poids, dans l'hypothèse ou certains composants sont plus importants que d'autres (avec une modulation possible en fonction des critères considérés).

L'établissement du bilan ergonomique peut faire l'objet d'une application de simulation particulière. En effet, à partir des préconisations de départ (sur la visibilité, l'accessibilité et la manœuvrabilité), le système peut générer les actions à réaliser afin de tester ces préconisations :

- un geste d'atteinte pour l'accessibilité,
- un suivi de trajectoire pour la manœuvrabilité,
- une orientation du regard pour la visibilité.

Cette utilisation est plus spécifique et plus restreinte que les possibilité offertes par la définition des actions sur les composants par l'utilisateur, mais elle offre l'avantage d'une automatisation complète.

4.5 Conclusion

Nous avons exposé dans ce chapitre l'état du travail mené dans le but d'intégrer au système ADELE les connaissances nécessaires à la simulation du comportement moteur pour le diagnostic ergonomique. Nous proposons une modélisation objet permettant de décrire l'activité dans ce contexte, incluant une tentative de description symbolique des postures. En ce qui concerne les stratégies de mouvement, un certain nombre de pistes sont explorées, permettant d'extraire quelques règles heuristiques permettant la prise de décision pour le pilotage du mannequin pour la simulation de l'activité. De même, nous proposons une approche permettant d'associer des éléments de diagnostic ergonomique à la simulation des postures.

Ainsi, le système ADELE constitue donc un outil permettant l'intégration de connaissances provenant de différents domaines d'expertise, pour la simulation ergonomique

5 Aspect conceptuels du système ADELE

5.1 Conception d'une architecture

Nous avons analysé et modélisé dans le chapitre 4 les différentes connaissances à intégrer dans le système ADELE pour atteindre les objectifs fixés au chapitre 3. Nous recentrons maintenant l'analyse sur la mise en œuvre de ces connaissances dans une optique de résolution de problème. L'étude des aspects de raisonnement (paragraphe 5.1.1) et la différenciation des données selon leur rôle dans le système (paragraphe 5.1.2) nous conduisent au choix des outils et à la structuration du système (paragraphe 5.1.3).

5.1.1 Le raisonnement dans ADELE

On parle ici du raisonnement au sens de l'Intelligence Artificielle, c'est-à-dire « un enchaînement d'énoncés ou de représentations symboliques, conduit en fonction d'un but. » (21). Dans notre contexte, il s'agit d'un enchaînement de découvertes d'éléments de connaissances, s'appuyant sur des informations connues, menant au but recherché.

Les étapes d'une simulation avec ADELE sont :

- la définition du scénario ou de l'action à simuler,
- la mise en œuvre de la simulation,
- l'émission d'un diagnostic ergonomique.

En ce qui concerne la description de l'activité, l'expertise nécessaire à la définition des actions et des scénarios est celle d'un expert humain chargé de remplir la base. L'aspect définition des actions et des scénarios ne fait donc pas partie du raisonnement intégré dans ADELE dans un premier temps. On fournit à l'utilisateur les moyens de spécifier des actions qui vont être simulées. Il est possible d'envisager ultérieurement un module de raisonnement associé à la spécification des actions, afin de fournir une aide à la constitution de scénarios. Dans le même ordre d'idée, on peut envisager d'intégrer ultérieurement un module de raisonnement conduisant au choix de l'action à simuler (qui doit actuellement être effectué par l'utilisateur de ADELE qui décide du test d'une ou plusieurs actions associées aux composants).

Par contre, l'étape de simulation nécessite la mise en œuvre d'un raisonnement. En effet, la simulation proprement dite consiste à piloter le mannequin, c'est-à-dire à décider, en fonction des actions à simuler puis en fonction des retours fournis par le mannequin quelles sont les parties du corps à activer et donc quels sont les ordres à donner au mannequin. L'aspect génération de commandes comporte un aspect décisionnel, qui est l'identification de la commande à générer, et un aspect de traduction, non direct (il faut rechercher les informations nécessaires à l'élaboration de la commande), mais déterministe.

De même, l'émission du diagnostic ergonomique fait l'objet d'un raisonnement, s'appuyant sur le compte-rendu du déroulement de la simulation, sur son résultat, sur l'écart entre le

résultat obtenu et les spécifications de départ, sur les éléments de connaissance permettant de qualifier la posture obtenue en terme de confort et d'acceptabilité pour la situation simulée.

5.1.2 Nature des informations manipulées

La Figure 24 rassemble tous les éléments du modèle objet dont la description a été développée au chapitre 4. Les informations qu'il comporte sont de deux natures :

- des informations de nature purement statique, décrivant le domaine, dont l'évolution est indépendante de la simulation,
- des informations décrivant l'état du système mannequin/environnement. Ces informations sont destinées à être stockées entre deux simulations d'actions. Elles permettent de sauvegarder, conserver et réutiliser des contextes de simulation.

Un autre ensemble d'informations du système est constitué par les données utilisées au cours d'une simulation, qui constituent une troisième catégorie :

- des informations de nature dynamique, qui décrivent, pour un objectif de simulation donné, l'état de l'évolution de la résolution du problème. Ces informations dynamiques ont une durée de vie limitée à la durée de la simulation d'une action.

Ces trois catégories d'informations recouvrent l'ensemble des données manipulées par le système ADELE. Nous reprenons ci-dessous, en les détaillant, chacune de ces trois catégories.

Informations statiques, à évolution indépendante de la résolution

Il s'agit des données décrivant le domaine, selon trois axes : l'opérateur, l'environnement et les interactions entre l'opérateur et l'environnement.

Les éléments concernant l'opérateur sont de deux types. On dispose d'une description générique de l'opérateur en terme de structure (représentation des différentes parties du corps et des liens entre elles) et de postures (états possibles associés à chacune des parties). A cette description générique, est associée la description de la structure MAN3D. La description générique de l'opérateur permet de disposer d'un ensemble de termes (du langage courant) qui peuvent être utilisés en prémisses de règles (« Si le *coude* est *plié* ... »). Elle permet d'implémenter la connaissance portant notamment sur les stratégies de mouvement, de manière indépendante du modèle d'opérateur utilisé pour la simulation. Pour les besoins de la simulation, il faut associer au modèle générique le modèle graphique, qui en constitue une sorte d'instance, une traduction, qui comporte elle aussi ses termes propres. On empile donc les modèles : l'approche consiste à rajouter entre l'opérateur réel et la structure MAN3D (qui est un modèle d'opérateur réel) un autre modèle.

L'environnement et les interactions opérateur/environnement constituent les éléments d'un modèle de l'activité motrice.

En effet, la description de l'environnement est faite par l'intermédiaire de la description des objets utilisables par l'opérateur. Les objets sont décrits par leur forme ainsi qu'à l'aide d'autres informations liées a leur utilisation.

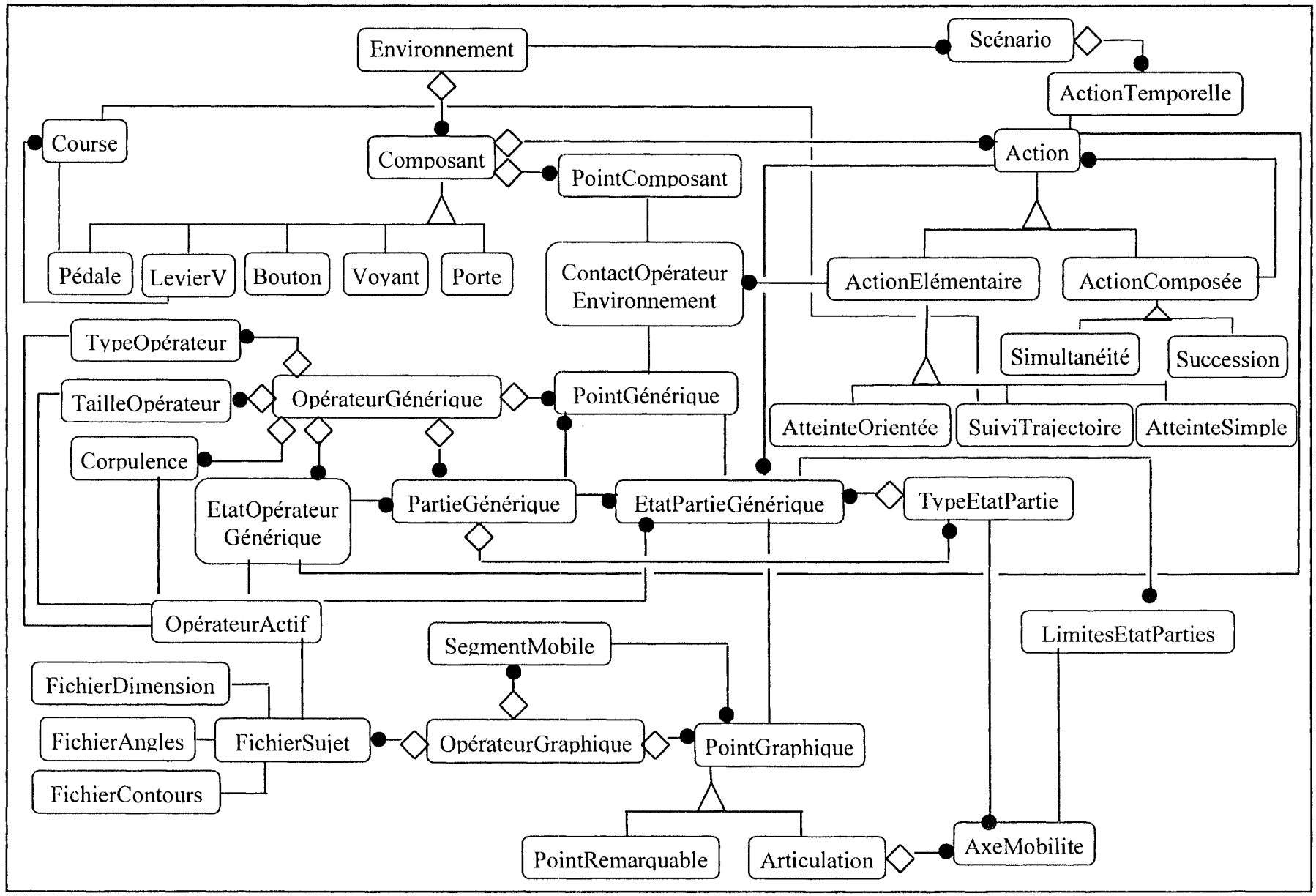

Figure 24 : Le modèle objet de ADELE

Le lien entre l'opérateur et son environnement est décrit sous la forme d'actions prédéfinies. Elles décrivent des actions réalisées par l'opérateur avec un objet de l'environnement. Il s'agit de décrire la manière dont doit s'effectuer l'action, en termes de posture (posture globale, ainsi qu'attitudes particulières de parties du corps), de durée. C'est là aussi que l'on décrit le "cahier des charges" d'utilisation de l'objet : on associe des contraintes à l'action (du type : le dos doit être collé au siège, pour un appui sur bouton).

Les actions prédéfinies sont des données d'entrée pour la simulation : une fois l'action à simuler connue, ses caractéristiques constituent les spécifications de départ de la simulation.

Informations évoluant au cours de la simulation d'une action et sauvegardées en sortie

Ces informations concernent la description d'un contexte de simulation, en terme d'état du système opérateur/environnement. Elles recouvrent donc :

- le choix d'un opérateur donné,
- l'état postural de cet opérateur en terme de posture globale et par parties,
- les contacts actifs entre l'opérateur et l'environnement.

L'état postural et les contacts actifs ont une valeur en début de simulation et évoluent au cours de celle-ci. Les autres éléments liés à l'opérateur actif concernent sa morphologie. On peut envisager de changer d'opérateur en cours de session de simulation (si l'on souhaite par exemple tester l'utilisation d'un composant, implanté à une position fixe mais pour différents sujets).

On peut également envisager de stocker les résultats liés au diagnostic ergonomique porté sur une situation a l'issue d'une simulation.

Informations dynamiques, limitées à la simulation d'une action :

Il s'agit des données décrivant l'état de la résolution du problème de simulation et d'émission du diagnostic ergonomique.

Elles comportent :

- la donnée initiale du problème (une ou plusieurs actions à simuler),
- la séquence d'actions calculée par le système pour atteindre le but envisagé,
- les résultats associés à chaque essai d'action (donnant lieu initialement à une commande MAN3D),
- l'état du mannequin associé à chaque exécution d'action,
- le diagnostic de simulation (état de la résolution),
- le diagnostic ergonomique associé à la simulation.

5.1.3 Structuration du système

Ce sont les besoins en terme de gestion et de manipulation des informations, ainsi que pour le type de raisonnement à mettre en œuvre, qui ont guidé le choix des outils, et donc la structuration finale du système.

Nous avons choisi de gérer les informations statiques dans un SGBD objet. Le choix d'utiliser un SGBD répond :

- à la nécessité de stocker ces informations de manière pérenne,
- au besoin de gérer un certain nombre de données utiles à la simulation indépendamment d'une session de pilotage du mannequin : le stockage de la description des objets de l'environnement, des interactions, des scénarios d'activité, doit être effectué préalablement à une session de pilotage du mannequin,
- au besoin d'utiliser un outil préexistant, spécialisé dans la gestion des données, et permettant donc aisément leur manipulation (opérations de consultation, de modification, de suppression, ...).

La conception du modèle objet nous a conduit à la préférence pour un SGBD de type objet, afin de conserver la richesse du modèle, notamment en ce qui concerne la capacité à exprimer la sémantique des liens entre objets. En effet, avec un SGBD objet, les relations complexes entre objets sont stockées en tant que partie de la structure de la base de données. Ce n'est pas le cas par exemple avec un SGBD de type relationnel, où les données sont stockées dans des tables ; les liens entre les données contenues dans ces tables sont exprimés en dehors de la base de données, sous la forme d'instructions SQL, permettant de réaliser des jointures entre tables.

Par ailleurs, la nécessité d'implémenter différents niveaux d'expertise, sur les stratégies de mouvement, sur le pilotage du mannequin, sur le diagnostic ergonomique, imposait le choix d'une plate-forme logicielle IA supportant l'expertise multiple. Le choix d'une architecture de tableau noir nous permet d'intégrer l'aspect multi-expertise mais aussi de développer de manière incrémentale une application dont la complexité réside plus dans les données. Les aspects raisonnement et contrôle associés à la simulation sont donc intégrés dans un système multi-agents à tableau noir.

La Figure 25 décrit l'architecture globale à laquelle conduisent ces choix.

Compte tenu de la contrainte imposant de ne pas réécrire MAN3D, il était nécessaire de concevoir un système capable d'utiliser les ressources fournies par MAN3D en communiquant avec le modèle depuis l'extérieur. Les ressources de MAN3D sont :

- les commandes de mouvement,
- les diagnostics sur l'exécution des commandes,
- les postures conservées (sous forme de fichiers).

Les connaissances nécessaires à la description de l'activité, au pilotage du mannequin et à l'émission du diagnostic ergonomique sont réparties entre les tableaux noirs du système multi-agents et la base de données objet. La base de données *Activité* correspond au modèle objet d'activité décrit dans le chapitre 4.

La base de données *Dimensions et postures* contient les données sur les dimensions et les postures produites par MAN3D (qui permet la sauvegarde et le rechargement de sujets et de postures). Ces données sont reliées aux éléments de description de l'activité au niveau de la base objet.

La structuration de la base de données objet est conforme au modèle objet décrit au chapitre 4. Son contenu est décrit dans le chapitre 6 (paragraphe 6.2.3). Nous décrivons le système multi-agents dans la section suivante (5.2). La question de l'utilisation de la base de données objet

par le modèle de tableau noir est évoquée ensuite, dans la section 5.3. Enfin, les détails liés à l'implémentation de cette architecture sont décrits dans le chapitre 6, avec notamment les aspects de communication et de partage de données, d'une part entre le système multi-agents et le SGBD objet, d'autre part entre le système multi-agents et MAN3D.

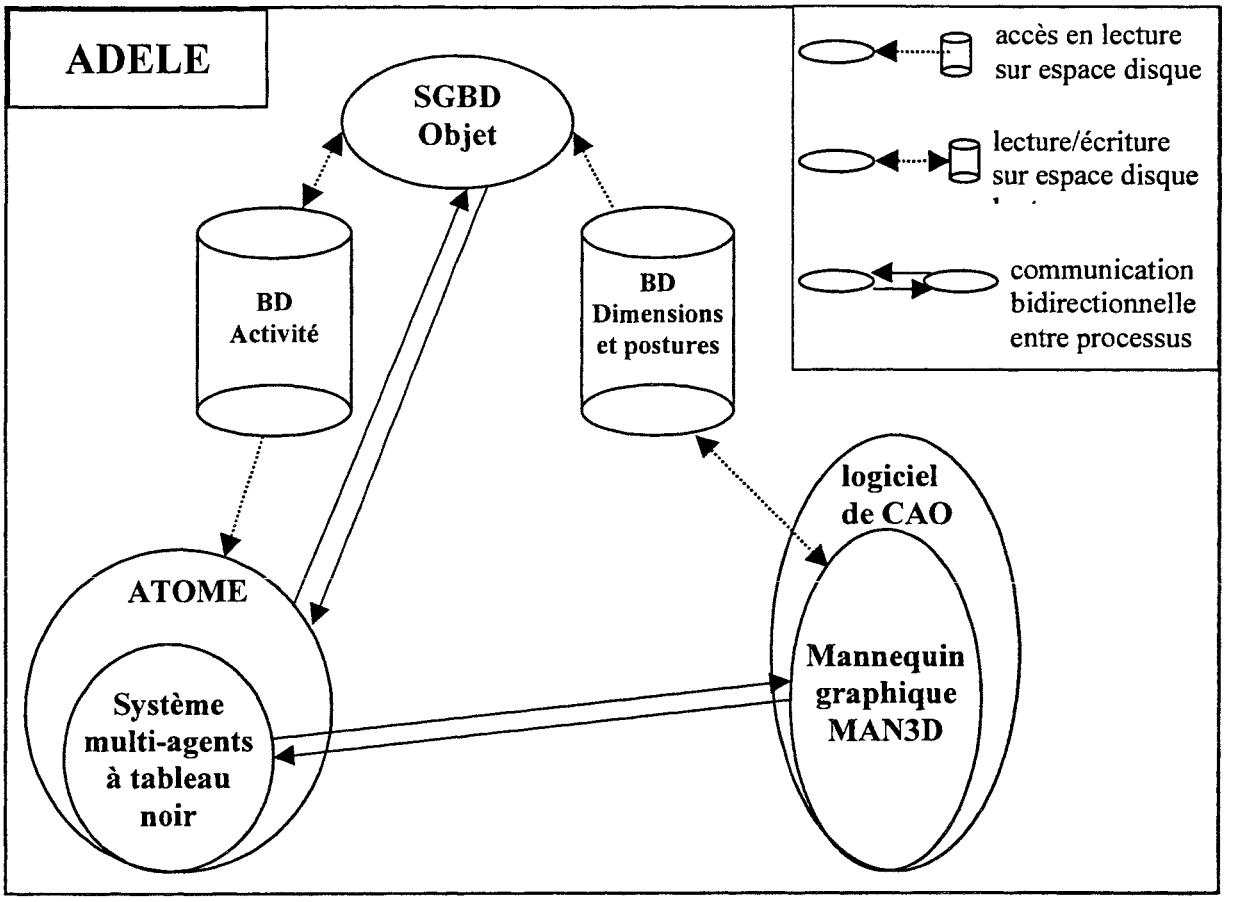

Figure 25 : Architecture globale de ADELE

5.2 Le système multi-agents

5.2.1 Le rôle de la partie multi-agents dans ADELE

Rappelons que le rôle du système multi-agents est double : piloter le mannequin selon les spécifications décrites dans l'action choisie et émettre un diagnostic ergonomique.

Le pilotage du mannequin pour simuler une action donnée consiste à :

- recueillir les paramètres de l'action à simuler,
- recueillir les éléments de contexte : opérateur actif, état postural de l'opérateur, état du système opérateur/environnement,
- définir le plan d'exécution pour l'action à simuler,
- mettre en œuvre le plan d'exécution, c'est-à-dire définir les commandes à adresser au mannequin, et modifier ce plan d'exécution en fonction des retours fournis par le modèle graphique.
- décider d'arrêter le processus en fonction du résultat obtenu.

Le raisonnement s'appuie sur :

- des éléments de connaissance permettant de choisir le membre à activer (permettant de déterminer, par exemple, dans le cas où un geste d'atteinte avec le bras est impossible, quelle est l'action à faire : incliner le dos ou avancer les fessiers, et si cette action est possible a priori),
- des éléments de connaissance permettant de déterminer l'action à effectuer en cas d'impasse,
- les informations sur l'état postural de l'opérateur à l'issue des commandes de mouvements.

L'émission du diagnostic ergonomique vise à qualifier le résultat postural obtenu (en terme de plausibilité et de confort), en relation avec la nature de l'action spécifiée au départ.

Nous décrivons dans les paragraphes suivants la conception de la partie multi-agents de ADELE.

5.2.2 Les tableaux noirs

Nous choisissons de commencer par la description des tableaux noirs du système. Comme nous l'avons rappelé aux paragraphes 2.2.2.2 et 2.2.3, le concept de tableau noir correspond à une structure de données commune aux différents agents (les spécialistes dans le cas de Atome) et permettant de stocker l'évolution de la résolution du problème. Chacun des agents est spécialisé dans une partie de la résolution du problème, et contient donc l'expertise permettant de "passer d'un niveau à un autre", c'est-à-dire de créer des éléments d'un niveau à partir d'éléments des autres niveaux. La description des tableaux noirs de ADELE (Figure 26 et Figure 27), permet d'introduire la description des spécialistes dans le paragraphe suivant.

La décomposition en plusieurs tableaux noirs correspond essentiellement à un découpage sémantique. Trois aspects principaux sont distingués : l'évolution de la simulation, le diagnostic ergonomique (tableau noir DIAGNOSTIC), et les aspects marginaux du dialogue avec MAN3D non relatifs à la simulation proprement dite (tableau noir GestionMAN3D). En ce qui concerne l'évolution de la simulation, on distingue : les éléments relatifs à l'élaboration du plan d'actions à partir de l'action de départ à simuler (tableau noir ACTION), les éléments relatifs à l'environnement (tableau noir ENVIRONNEMENT), et les éléments correspondants à des états de l'opérateur ou du système opérateur/environnement (tableau noir ETATS-EN-COURS). Le tableau noir OPERATEUR conserve les caractéristiques non évolutives de l'opérateur en cours.

Pour chaque tableau noir on décrit les différents niveaux (dont les noms sont écrits en caractères gras). Chaque niveau contient un type de nœuds. Les liens entre les nœuds sont bidirectionnels (ce qui implique que l'on peut atteindre un nœud à partir de n'importe quel lien associé à un autre nœud). Les flèches dessinées sur la droite des tableaux noirs indiquent ici le "sens de création des nœuds" : le départ de la flèche indique le niveau lu, la pointe de la flèche indique le niveau où sont créés de nouveaux nœuds, sur la base des nœuds consultés. Chacune de ces flèches correspond donc à une action de spécialiste.

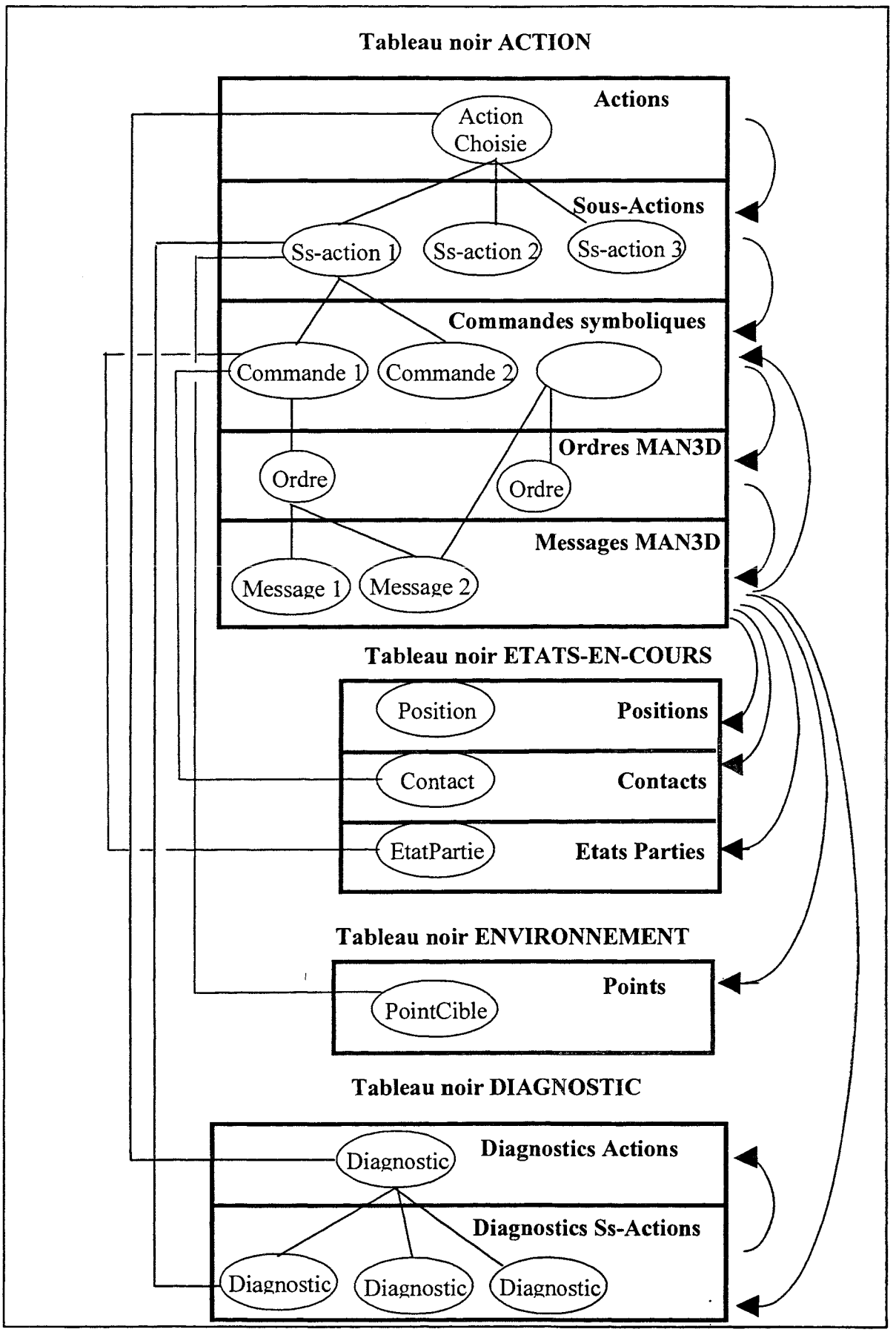

Figure 26 : Les tableaux noirs de la simulation et du diagnostic

Le niveau *Actions* du tableau noir ACTION contient des nœuds qui correspondent au concept d'action (élémentaire ou composée) défini dans le cadre de la modélisation de l'activité (voir paragraphe 4.2.3). A un nœud du niveau *Actions* est associée une liste ordonnée de nœuds du niveau *Sous-Actions*, correspondant aux sous-actions composant l'action choisie. A partir d'un nœud du niveau *Sous-Actions*, un nœud du niveau *Commandes symboliques* peut être créé. Une commande symbolique correspond à un premier niveau de traduction d'une sous-action en commande de pilotage de l'opérateur graphique. Le deuxième niveau de traduction est celui du niveau *Ordres MAN3D*, qui contient des nœuds fournissant des commandes MAN3D prêtes à être envoyées.

Le niveau *Messages MAN3D* contient les messages résultant de l'envoi de commandes à MAN3D. Le nombre important de flèches partant de ce niveau (flèches correspondant à des créations de nœuds sur les autres niveaux induites par l'évolution de ce niveau) souligne l'importance des informations recueillies lors du dialogue avec MAN3D. Ainsi, à l'issue de l'envoi d'une commande à MAN3D, de nouvelles commandes symboliques peuvent être créées, et les nœuds des tableaux noirs ENVIRONNEMENT et ETATS-EN-COURS peuvent être mis à jour. Lorsque la simulation est terminée, les messages obtenus servent aussi de base à l'établissement du diagnostic associé à chacune des sous-actions.

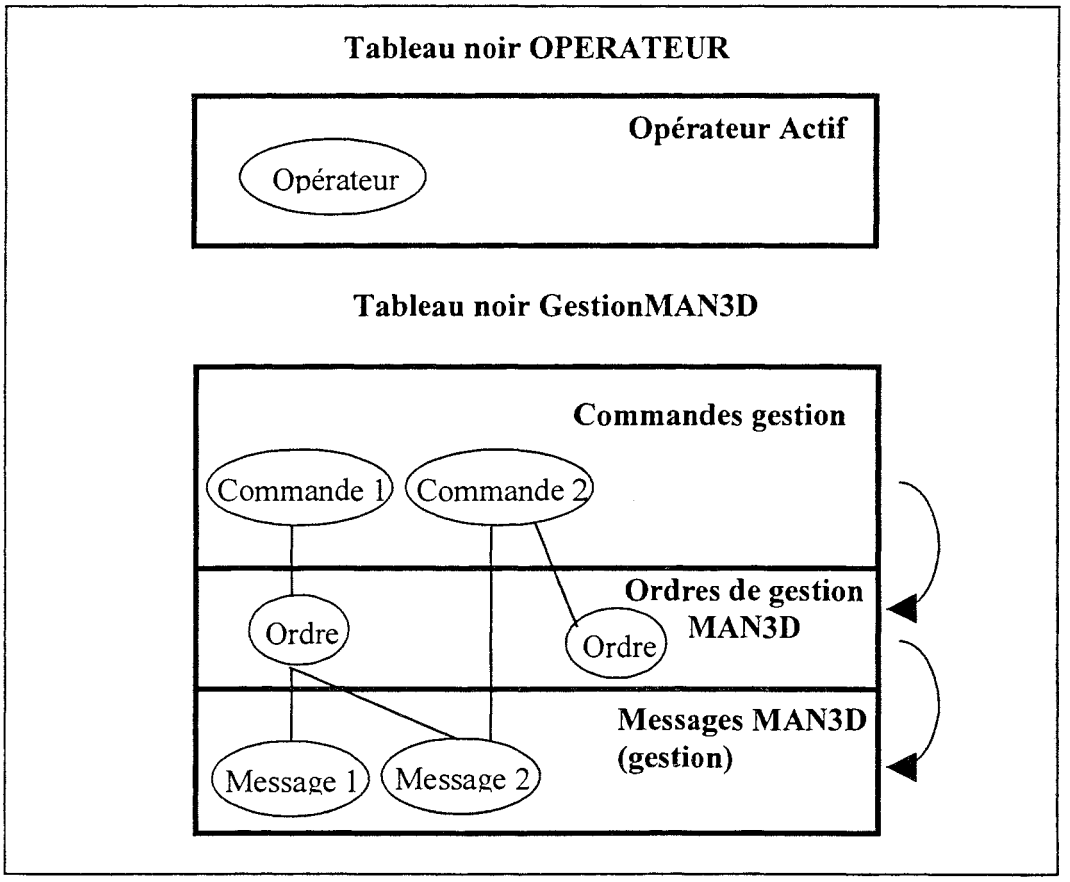

Figure 27 : Autres tableaux noirs

Le tableau noir ENVIRONNEMENT permet de stocker les éléments concernant la maquette de produit testée. On s'intéresse ici à la localisation du point cible pour une atteinte. Le point cible est une information de départ, liée à la sous-action. Outre le nom de ce point (qui est l'information dont on dispose au niveau de la sous-action), on doit disposer d'informations

sur sa localisation, permettant de faire des choix de mouvement. C'est pourquoi il est identifié de manière complémentaire dans les données de tableau noir locales, dans la mesure où les informations sur sa localisation sont mises à jour lors du dialogue avec MAN3D (nous y reviendrons dans le paragraphe 5.2.3, avec la description des spécialistes).

Le tableau noir ETATS-EN-COURS permet de stocker les éléments issus de la résolution concernant l'opérateur. Ainsi les états des parties de l'opérateur seront stockés au début de la simulation, puis à l'issue de commandes de mouvement, après l'intervention de la spécialiste de traduction des postures. La position de l'opérateur (correspondant à la position du centre du bassin) sera stockée en début de simulation et éventuellement modifiée si des mouvement de déplacement sont effectués. De même l'état du système opérateur/environnement est stocké par l'intermédiaire des contacts devenus actifs suite à des commandes d'atteinte (ou actifs dès le début). C'est pourquoi les nœuds des niveaux *Contacts* et *Etats Parties* sont liés aux commandes symboliques qui ont mené à leur création ou à leur mise à jour.

Le tableau noir DIAGNOSTIC permet de stocker les diagnostics émis, associés aux sous-actions, puis à l'action choisie.

La Figure 27 complète la description des tableaux noirs de ADELE. Le tableau noir OPERATEUR permet de stocker la définition de l'opérateur actif. En effet les informations concernant la morphologie de l'opérateur et à sa posture globale sont utilisées pour la résolution. Le tableau noir GestionMAN3D concerne un autre ensemble de données, relatif à des aspects de communication avec MAN3D ne concernant pas directement les ordres liés à la simulation d'une action. Il s'agit de commandes de gestion (telles que la sauvegarde de postures par exemple) intervenant au cours de la résolution d'un problème de simulation d'action. Elles donnent lieu, comme les commandes de mouvement, à des ordres transmis à MAN3D.

5.2.3 Les sources de connaissances spécialistes

Avec le modèle de tableau noir de Atome (voir paragraphe 2.2.3), les agents, ou sources de connaissance, sont soit des spécialistes, qui accèdent directement aux tableaux noirs pour les consulter et éventuellement créer ou modifier des nœuds, soit des tâches ou une stratégie, dans lesquelles on répartit le contrôle de la résolution du problème, qui est fonction de l'évolution des tableaux noirs.

Afin de poursuivre notre schéma de conception de système multi-agents, nous évoquons ici tout d'abord les spécialistes (les tâches et la stratégie sont évoquées au paragraphe 5.2.4). La Figure 28 liste ces différentes spécialistes, en référence à leur action sur les tableaux noirs (représentée par une flèche).

Nous indiquons le rôle de chacune de ces spécialistes :

- la spécialiste *Création plan d'actions* a en charge l'élaboration du plan initial des actions à simuler. Il s'agit simplement de créer les nœuds sous-actions, en décomposant l'action choisie pour la simulation.
- la spécialiste *Création plan d'exécution* crée les commandes symboliques nécessaires à la réalisation de chaque sous-action.

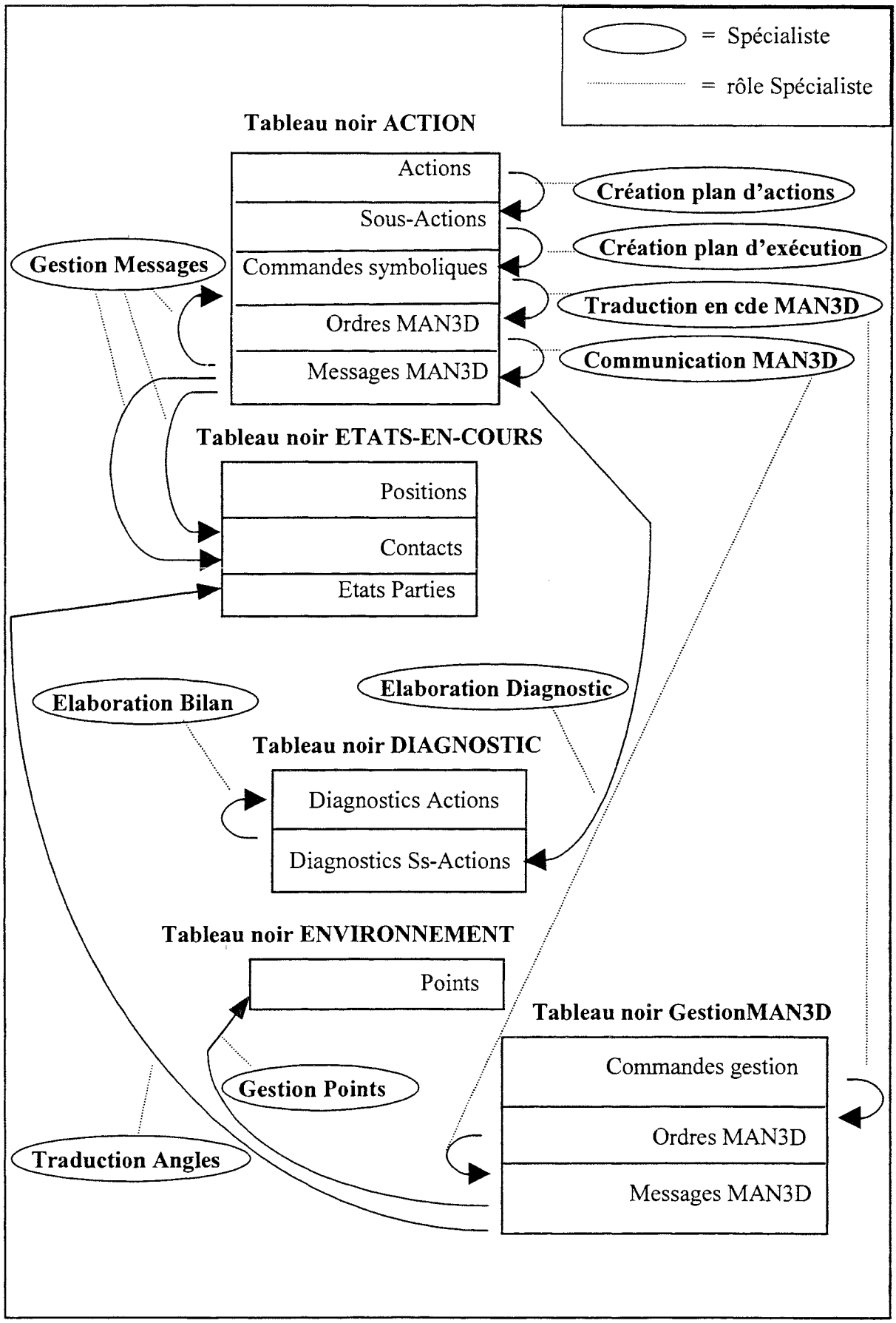

Figure 28 : Les Spécialistes de ADELE

- la spécialiste *Traduction en commande MAN3D* élabore la traduction d'une commande (commande symbolique de mouvement ou commande de gestion), et la stocke dans le niveau Ordres MAN3D correspondant.
- la spécialiste *Communication MAN3D* envoi une commande à MAN3D, récupère le retour fourni par MAN3D et crée les messages correspondants.
- la spécialiste *Gestion Messages* est chargée de lire les messages associés aux commandes. Elle crée éventuellement de nouvelles commandes symboliques, met à jour les contacts opérateur/environnement et les états des parties de l'opérateur, si besoin.
- la spécialiste *Traduction Angles* à pour objet la traduction des valeurs angulaires concernant l'état du mannequin graphique en items posturaux. En pratique, il s'agit d'envoyer à MAN3D une commande de sauvegarde de posture dans un fichier d'angles, et de lire les valeurs d'angles désirées dans ce fichier. Nous reviendrons dans la section 5.3 sur le choix possible concernant l'implantation de la procédure permettant de traduire les angles en items posturaux.
- la spécialiste *Gestion Points* s'occupe de la mise à jour des informations concernant la localisation des points cibles. Au cours du raisonnement, pour choisir une commande de mouvement, il peut être nécessaire de connaître la position relative du point par rapport à l'opérateur. Les coordonnées du point cible sont obtenues auprès de MAN3D par une autre commande de gestion ("décode entité"). La position relative du point (devant ou derrière l'opérateur par exemple) pour un geste d'atteinte donné peut être obtenue à partir de la position de l'opérateur, des coordonnées du point cible, et des angles posturaux (par exemple l'angle du bassin pour identifier la position relative du point par rapport au buste).
- les spécialistes *Elaboration Bilan* et *Elaboration Diagnostic*, sont chargées d'établir le diagnostic ergonomique, à partir du résultat des essais d'exécution d'actions.

Une autre spécialiste, *Communication Base*, est chargée de la communication avec la base de données. Elle doit permettre de gérer la connexion à la base, ainsi que le dialogue entre le système multi-agents et la partie de ADELE qui gère les accès à la base. La création d'une tâche et d'une spécialiste consacrées à la communication avec la base de données permet d'isoler cet aspect et de simplifier les modifications à apporter au système en cas de changement d'outil SGBD.

5.2.4 Le contrôle de la résolution

Après avoir spécifié les informations à gérer par le système (dans les tableaux noirs), puis identifié les différentes spécialistes chargées de mettre à jour ces informations, il faut définir la manière dont la résolution doit se dérouler. Avec Atome, le contrôle de la résolution, c'est-à-dire la gestion de l'activation des différentes spécialistes pour la mise en œuvre du raisonnement, est réparti entre la stratégie et les tâches, qui s'appuient respectivement sur des résumés des tableaux noirs et des listes d'événements provenant des actions effectuées sur ces tableaux noirs (voir paragraphe 2.2.3).

Nous avons précisé plus haut que la complexité du système ADELE réside plutôt dans la gestion des données. En effet, le contrôle de la résolution est simple. Quatre tâches ont été identifiées : la communication avec la base de données (*ComBD*), l'initialisation (*Init*), le déroulement du plan d'action (*DéroulePlan*), et l'émission du diagnostic ergonomique (*Diagnostic*).

Le schéma global du système multi-agents est décrit sur la Figure 29.

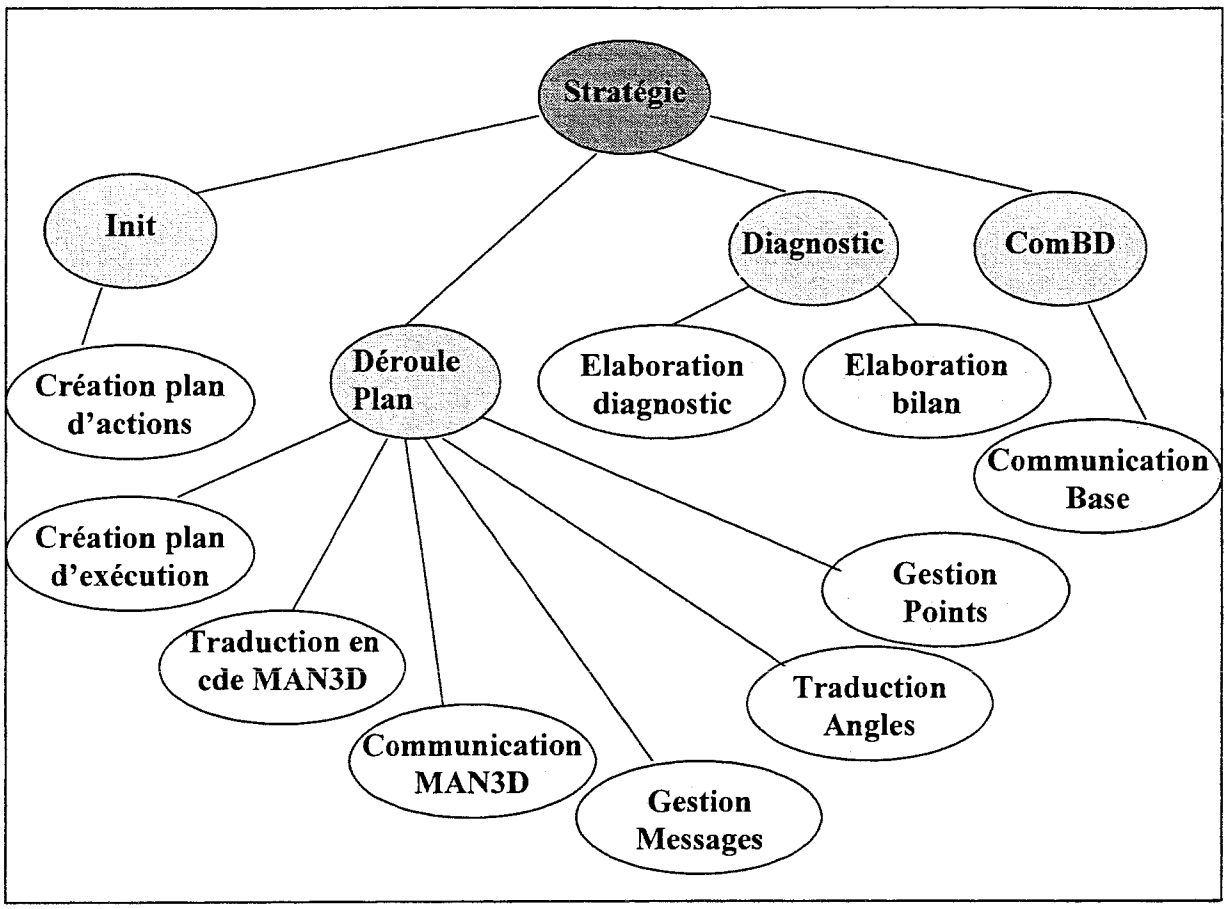

Figure 29 : Organisation de la partie multi-agents de ADELE

La tâche *ComBD* concerne tous les aspects de la communication avec GemStone. Elle active la spécialiste *Communication Base*.

La tâche *Init* est chargée de l'initialisation des tableaux noirs. Elle active la spécialiste *Création plan d'actions*.

La tâche *DéroulePlan* concerne la simulation proprement dite. Elle est chargée de l'activation des spécialistes *Création plan d'exécution*, *Traduction en commande MAN3D*, *Communication MAN3D*, *Gestion Messages*, *Traduction Angles*, et *Gestion Points*.

La tâche *Diagnostic* concerne l'établissement du diagnostic qui fait suite à la simulation. Elle active donc les spécialistes *Elaboration Bilan* et *Elaboration Diagnostic*.

5.3 L'intégration de données du tableau noir dans une BD objet

Nous avons fait l'inventaire des données manipulées par le système dans les paragraphes précédents. Nous avons vu au paragraphe 5.1.2 qu'elles peuvent être pour le système soit statiques, soit évolutives, soit dynamiques. Les besoins de manipulation de ces informations nous ont conduit à choisir deux outils disjoints pour leur intégration : une base de données objet et un système multi-agents. Dans la description des tableaux noirs du système multi-

agents (paragraphe 5.2.2) nous avons fait abstraction du problème de la localisation des informations utilisées. C'est cette question que nous abordons ici.

Les informations statiques sont par définition persistantes. Elles sont donc conservées dans la base de données. Par contre, elles servent de données pour la simulation. Les informations relatives à la résolution du problème de simulation, que nous avons appelées dynamiques, constituent par définition le contenu des tableaux noirs. Elles n'ont pas de caractère persistant. D'autres informations, telles que les résultats de la simulation, que nous avons appelées évolutives, sont destinées à être conservées dans la base, à l'issue de la simulation et de l'émission du diagnostic ergonomique. Il existe donc un besoin de partage de données entre le système à base de tableau noir et le système objet[11].

Une première approche envisageable est de prévoir la sélection des informations utiles dans la base de données à l'initialisation de la simulation, et de les recopier dans les tableaux noirs. De même, en fin de simulation, les informations des tableaux noirs à conserver sont recopiées dans la base. Ceci présente deux inconvénients. Tout d'abord, les informations sont dupliquées, ce qui n'est pas optimal. Par ailleurs, cela conduit à créer une spécialiste ayant un statut particulier, lié à la gestion des données, c'est-à-dire ayant un rôle fonctionnel, générique et non spécifique à la résolution du problème .

Une solution satisfaisante est la localisation de données du tableau noir dans la base de données elle-même. Elle permet en effet d'éviter la duplication des données et conserve aux spécialistes leur rôle consistant à faire évoluer la solution du problème traité. C'est cette approche que nous avons adoptée.

Dans notre application, plusieurs objets (au sens du modèle objet) font partie des informations communes :

- l'action choisie et les sous-actions qui la composent : l'action à simuler doit être choisie parmi les actions définies dans la base et fournie comme donnée d'entrée de la simulation; puis, au cours de la simulation, le système doit accéder aux caractéristiques de l'action choisie, c'est-à-dire aux sous-actions qui la composent éventuellement, au composant auquel elle s'applique, ...
- de même les caractéristiques de l'opérateur actif, stockées dans la base, doivent être accessibles en tant qu'éléments de tableau noir, pour disposer par exemple des informations relatives à la morphologie de l'opérateur, utilisables en prémisses de règles portant sur les stratégies de mouvement.
- les états initiaux des parties de l'opérateur ainsi que les contacts actifs sont aussi des éléments de tableau noir et des instances d'objets de la base, mis à jour par la simulation et stockés à la fin dans la base.

Un aspect intéressant de l'implantation du tableau noir dans la base de données objet est que cette approche permet pour certains cas de choisir le lieu ou l'on va décrire la manipulation des données. Prenons le cas de la traduction des angles en postures symboliques. La procédure de traduction pourra indifféremment être localisée dans une méthode d'objet de la base ou bien faire l'objet d'une spécialiste.

[11] Nous utilisons le terme de *système objet* pour désigner l'application objet définie par la base de données objet (qui contient à la fois les données et leur utilisation).

Cette solution permet donc d'enrichir Atome et de répondre aux observations faites lors de ses précédentes utilisations (voir paragraphe 2.2.3.4). En fournissant vraiment un statut d'objet aux nœuds des tableaux noirs, on offre la possibilité d'implanter dans les méthodes des objets des opérations de calcul qu'il fallait auparavant intégrer dans les spécialistes, ce qui obligeait soit à créer des spécialistes supplémentaires, soit à introduire ces calculs dans les règles des spécialistes existantes, en dupliquant parfois le code de calcul.

L'implémentation de cette approche, qui consiste à offrir la possibilité de définir certains nœuds de tableaux noirs comme faisant partie de la base de données objet, est décrite dans la section 6.4. La solution adoptée fournit la possibilité de définir des nœuds soit locaux à l'application multi-agents, soit présents dans la base, comme instances d'objets, tout en garantissant la transparence dans leur utilisation, quel que soit leur type. A partir du moment où certains nœuds sont localisés dans la base de données, leurs mises à jour sont directement effectuées dans la base (en pratique, il faut simplement veiller à sauvegarder l'image de la base sur laquelle travaille la session de simulation, en sortie, ce qui est de l'ordre des informations globales gérées au niveau stratégie).

Cette nouvelle version de Atome ouvre aussi d'autres possibilités concernant le suivi de la résolution du problème. En effet, si l'on choisit d'implanter tous les nœuds de tableau noir dans la base de données (y compris des informations prévues a priori comme n'étant pas persistantes), on peut envisager de développer des applications de la base objet permettant la visualisation du raisonnement. Ceci suppose d'avoir développé comme base d'objets la totalité des nœuds du tableau noir considéré.

PARTIE III : Le prototype - son implémentation et son évaluation

Introduction

Nous consacrons cette partie aux aspects liés à l'implémentation du système ADELE. Nous avons en effet développé un premier prototype, ayant pour objectifs d'une part d'établir la faisabilité de la modélisation et de la structure conçues, et d'autre part de servir de démonstrateur à destination des usagers futurs du système d'aide à la conception. Nous décrivons donc tout d'abord le système développé (chapitre 6) en commençant par ses principes généraux (section 6.1), puis en décrivant les aspects propres à la base de données (section 6.2). Dans la section suivante (section 6.3) nous décrivons les développements relatifs à la communication entre les trois gros modules du système : MAN3D, le système multi-agents développé avec Atome (nommé SMA-Adele) et l'application développée sous le SGBD orienté objet GemStone (comportant une base de données nommée BaseAdele et une partie interface, nommée InterfaceAdele gérant les aspects de communication). Un autre point, celui du partage de données entre Atome et GemStone, a fait l'objet d'une réflexion de nature plus conceptuelle, dont nous faisons état dans la section 6.4. Le chapitre 7 concerne l'évaluation du prototype. Nous définissons la nature de l'évaluation, puis nous exposons la phase de test proprement dite, pour finir par un bilan de cette évaluation.

6 Architecture logicielle du système

6.1 Principes du système

6.1.1 Architecture générale

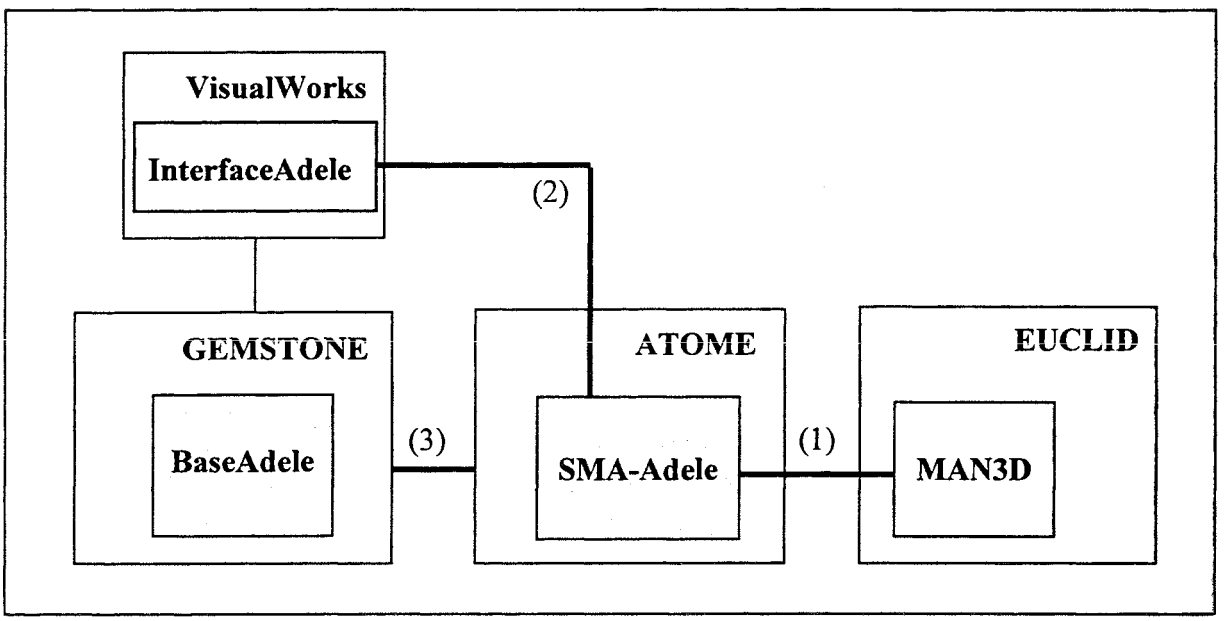

Figure 30 : Architecture générale du prototype de ADELE

La Figure 30 décrit les éléments logiciels du système et leurs interconnexions. Les développements propres au travail exposé ici sont :

- *SMA-Adele*, le système multi-agents développé avec Atome,
- *BaseAdele*, la base d'objets implémentant le modèle objet conçu,
- *InterfaceAdele*, l'application gérant les interfaces graphiques avec GemStone,
- la communication entre MAN3D et SMA-Adele (1),
- la communication entre SMA-Adele et InterfaceAdele (2),
- le partage de données entre Atome et GemStone (3).

Nous décrivons brièvement le SGBDOO GemStone et son environnement (incluant VisualWorks) dans le paragraphe 6.2.1.

Le système est développé sur station de travail Sun sous Unix. Les trois processus (GemStone/InterfaceAdele, Atome/SMA-Adele, Euclid/MAN3D) sont lancés en parallèle, puis ils communiquent entre eux :

- par une communication spécifique entre MAN3D et SMA-Adele (voir paragraphe 6.3.1),
- par une communication spécifique entre SMA-Adele et InterfaceAdele (voir paragraphe 6.3.2), plus des accès à la base de données depuis Atome (voir section 6.4).

Il n'existe pas de communication spécifique entre MAN3D et GemStone/BaseAdele, mais certains fichiers stockés par MAN3D sont référencés dans la base (les fichiers sujets, angles, dimensions).

6.1.2 Principes d'utilisation

6.1.2.1 Cohérence entre Euclid/MAN3D et GemStone/BaseAdele

Le fonctionnement du système implique une cohérence à deux niveaux :

- entre les modèles (MAN3D et BaseAdele) : ceci implique que soient initialisées (en amont de toute utilisation du système) les données sur l'opérateur graphique (voir paragraphe 4.2.2.2),
- à propos des situations manipulées : il faut que l'opérateur actif (défini au paragraphe 4.2.2.4) corresponde au sujet graphique chargé par l'utilisateur. De même, il faut que l'environnement en cours corresponde à la maquette CAO chargée, et que les noms des composants correspondent aux noms des entités graphiques présentes sur la maquette. On ajoute donc le concept de session de simulation pour permettre à un utilisateur de décrire les paramètres d'une situation, de les sauvegarder et les retrouver pour une utilisation ultérieure.

La Figure 31 décrit les concepts ajoutés au modèle dans cette optique. Un utilisateur peut être identifié, et il peut stocker sous son nom des descriptions de composants, des descriptions d'environnements, ainsi que des sessions de simulation. Une session de simulation associe un environnement en cours, un opérateur actif et l'ensemble des contacts actifs entre l'opérateur et l'environnement.

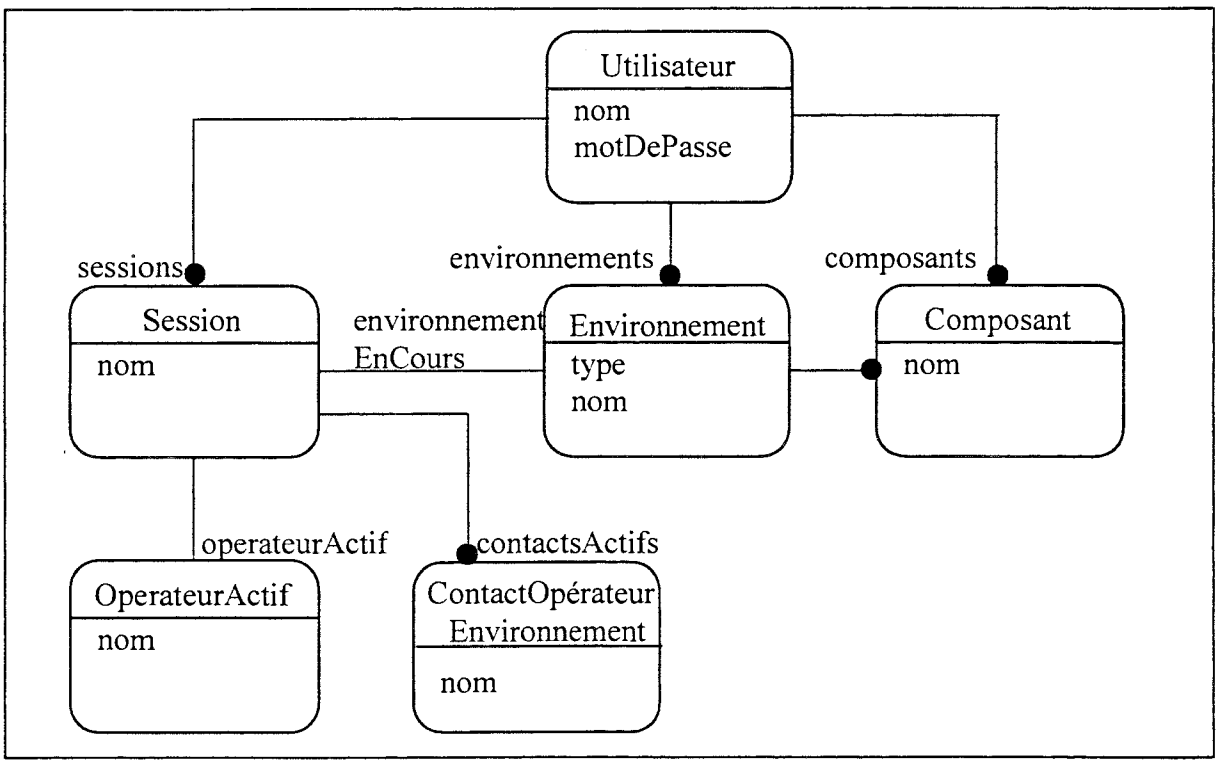

Figure 31 : Gestion des sessions de simulation

6.1.2.2 Déroulement d'une session de simulation

Le système sert à obtenir un diagnostic ergonomique à partir de la simulation de l'activité de l'opérateur, sur une maquette de produit sous CAO.

La Figure 32 décrit le déroulement d'une session de simulation (selon le temps, en vertical).

GemStone / InterfaceAdele	Atome / SMA-Adele	Euclid / MAN3D
		lancement -> boucle d'attente du fichier provenant de SMA-Adele
	lancement -> initialisation comme session GemStone -> boucle d'attente d'un message provenant de InterfaceAdele	
lancement -> initialisation comme session GemStone -> envoi d'un message à SMA-Adele pour lancer la simulation -> attente du signal de fin de la simulation		
	Réception du message de InterfaceAdele -> lancement de la simulation	
	BOUCLE DE SIMULATION **-> Création du fichier de communication avec MAN3D** **-> attente du fichier retour**	
		détection du fichier provenant de SMA-Adele **-> traitement des commandes contenues dans le fichier** **-> création du fichier retour avec les messages obtenus** **-> attente du fichier provenant de SMA-Adele**
	Lecture du fichier retour MAN3D **-> traitement des messages**	
	Envoi du signal de fin de la simulation à InterfaceAdele	
Réception du signal de fin de la simulation		

Figure 32 : Déroulement d'une session de simulation

L'utilisation du prototype implique d'une part le lancement d'une session Euclid/MAN3D, c'est-à-dire :

- une session Euclid avec une maquette de produit en cours de conception (par exemple un habitacle de poids-lourd), incluant un certain nombre de fonctions prépositionnées (des composants de l'habitacle par exemple),
- un sujet MAN3D chargé sous Euclid.

Par ailleurs, la partie système multi-agents (SMA-Adele) doit être lancée et en attente d'un signal de la part de GemStone.

Enfin, une session de simulation doit être lancée depuis GemStone/InterfaceAdele.

En utilisation courante, l'utilisateur modifie sa maquette CAO sous Euclid. Lorsqu'il souhaite simuler une action, il doit utiliser l'interface de InterfaceAdele permettant de choisir l'action à simuler. SMA-Adele devient alors le processus maître : il dialogue avec MAN3D et fait des accès à BaseAdele. Il rend la main lorsque la simulation est terminée et le diagnostic final émis.

6.1.3 Limites du prototype

Le prototype a été réalisé afin de démontrer la faisabilité de l'approche conçue pour développer un outil d'assistance au concepteur de maquettes sous CAO, pour l'évaluation ergonomique. Il comporte l'implémentation de l'architecture conçue, mais les bases de connaissances intégrées ne sont qu'ébauchées, notamment en ce qui concerne les stratégies de mouvement et l'émission du diagnostic ergonomique.

6.2 La base de données objets

6.2.1 Quelques mots sur GemStone et son environnement

GemStone est un système de stockage à objets avec une architecture de type client/serveur, qui permet l'accès concurrent de plusieurs utilisateurs à une base d'objets commune. Le système comporte trois éléments :

- un espace de stockage (repository) pour stocker des objets persistants et partagés,
- un moniteur (Stone) permettant le contrôle des accès et assurant l'intégrité des objets stockés,
- un serveur d'objets (Gem) attaché à chaque application accédant aux objets, qui lit les objets dans l'espace de stockage et qui les met à jour pour les besoins de l'application.

Une session d'accès à la base (qu'elle soit ou non interactive) dispose d'une vue de la base de données, initialisée au démarrage de la session. Afin de maintenir la cohérence de la base, GemStone rassemble les modifications effectuées par une session dans une *transaction*. La validation de ces modifications doit être demandée explicitement par la session (utilisateur en interactif ou application externe) par un *commit*. Cette opération échoue si GemStone détecte un conflit entre les modifications de la base de données proposées par la session considérée et les modifications soumises par d'autres utilisateurs. Dans le cas où la validation réussit, les ajouts et les modifications proposées sont rendues visibles aux autres utilisateurs et la vue de la base de données associée à la session considérée est mise à jour.

La définition des objets, la manipulation des données et les interrogations se font avec Smalltalk (un Smalltalk qui permet l'interrogation, la gestion de transactions et la persistance). L'organisation et la syntaxe du langage Smalltalk de GemStone sont semblables à celles de ParcPlace\Smalltalk. Les objets sont définis par des classes, basées sur une

structure commune. Les classes sont organisées en une hiérarchie d'héritage (du type « sorte-de »), dont la racine est la classe *Object*.

Pour accéder aux objets stockés sous GemStone (pour développer des interfaces graphiques par exemple), on peut utiliser VisualWorks, qui est un environnement de programmation orientée objet avec le langage ParcPlace\Smalltalk. GemStone propose une interface avec VisualWorks (GemStone Smalltalk Interface for VisualWorks), qui est un ensemble de classes et de primitives, permettant le développement d'applications VisualWorks qui travaillent sur des objets stockés sous GemStone. Dans ce contexte, GemStone agit comme un serveur d'objets pour l'application Smalltalk cliente. On peut ainsi :

- stocker les objets de l'application Smalltalk cliente sous GemStone,
- importer des objets GemStone dans VisualWorks (en tant qu'objets ParcPlace\Smalltalk),
- dupliquer et maintenir de manière transparente des objets entre VisualWorks et GemStone, ou bien travailler avec des objets résidant uniquement sous GemStone mais apparaissant comme des objets ParcPlace\Smalltalk.

VisualWorks fournit donc à GemStone un environnement de programmation et un outil pour créer des interfaces graphiques.

De la même façon, GemStone fournit des interfaces avec des langages tels que C, C++, ... Nous évoquerons cet aspect dans le paragraphe 6.4.2, à propos du partage d'informations entre GemStone et Atome.

6.2.2 Interface interactive avec la base de données : InterfaceAdele

Les accès à la base de données ainsi que l'envoi de la simulation se font grâce au module InterfaceAdele. La Figure 33 liste les interfaces développées dans ce module, avec leur enchaînement. La session de l'item de menu *Création Session* (de *Gestion de BaseAdele*) correspond au concept de session de simulation, décrit au paragraphe 6.1.2.1 .

Nous décrivons de manière plus détaillée ces interfaces au chapitre 7 (section 7.1).

Pour le prototype, la différenciation des utilisateurs n'est pas faite et n'est donc pas répercutée au niveau des interfaces et des autorisations d'accès aux informations contenues dans la base. Il existe en effet plusieurs modes d'utilisation possibles (regroupés sur la Figure 33) :

- une utilisation se limitant au choix des actions à simuler, qui pourrait être celle d'un dessinateur/concepteur travaillant sur sa maquette CAO (accès unique à *Choix Action à simuler*),
- une utilisation impliquant la définition des actions possibles sur les composants ; cette utilisation pourrait être par exemple soit réservée à l'expert ergonome, soit possible pour un dessinateur/concepteur à partir de modèles d'actions déjà définis et récupérables pour des modifications de détail (accès possible à *Gestion de BaseAdele*),
- une utilisation portant sur la mise à jour des caractéristiques du mannequin graphique et des liens entre le modèle d'opérateur graphique et le modèle générique d'opérateur, qui permet notamment le suivi de l'évolution du modèle graphique, indépendamment du modèle générique et sans nécessité de mise à jour de l'ensemble du système ADELE (accès à *Gestion Opérateur Graphique*) ; cette utilisation est utile dans le cadre de la maintenance du système,

- une utilisation portant sur la mise à jour des caractéristiques de l'opérateur générique, par exemple rajouter une morphologie (accès à « Gestion Opérateur Unique») . Cette utilisation est de l'ordre de la mise au point du modèle, puisqu'elle implique aussi l'ajout de règles dont les prémisses porteraient sur ces nouvelles caractéristiques.

Notons que ces utilisations concernent les informations stockées dans BaseAdele, et que bien sûr seule la simulation d'une action nécessite l'appel de MAN3D et de SMA-Adele. La mise à jour des informations contenues dans la base, quel qu'en soit le niveau, se fait en amont et indépendamment d'une session de simulation. Le fait que l'envoi de la simulation se fasse au niveau de InterfaceAdele (et non de SMA-Adele par exemple) est un choix que nous commenterons plus loin en évoquant les questions portant sur la communication entre les modules et le partage de données (paragraphe 6.3.2 et section 6.4).

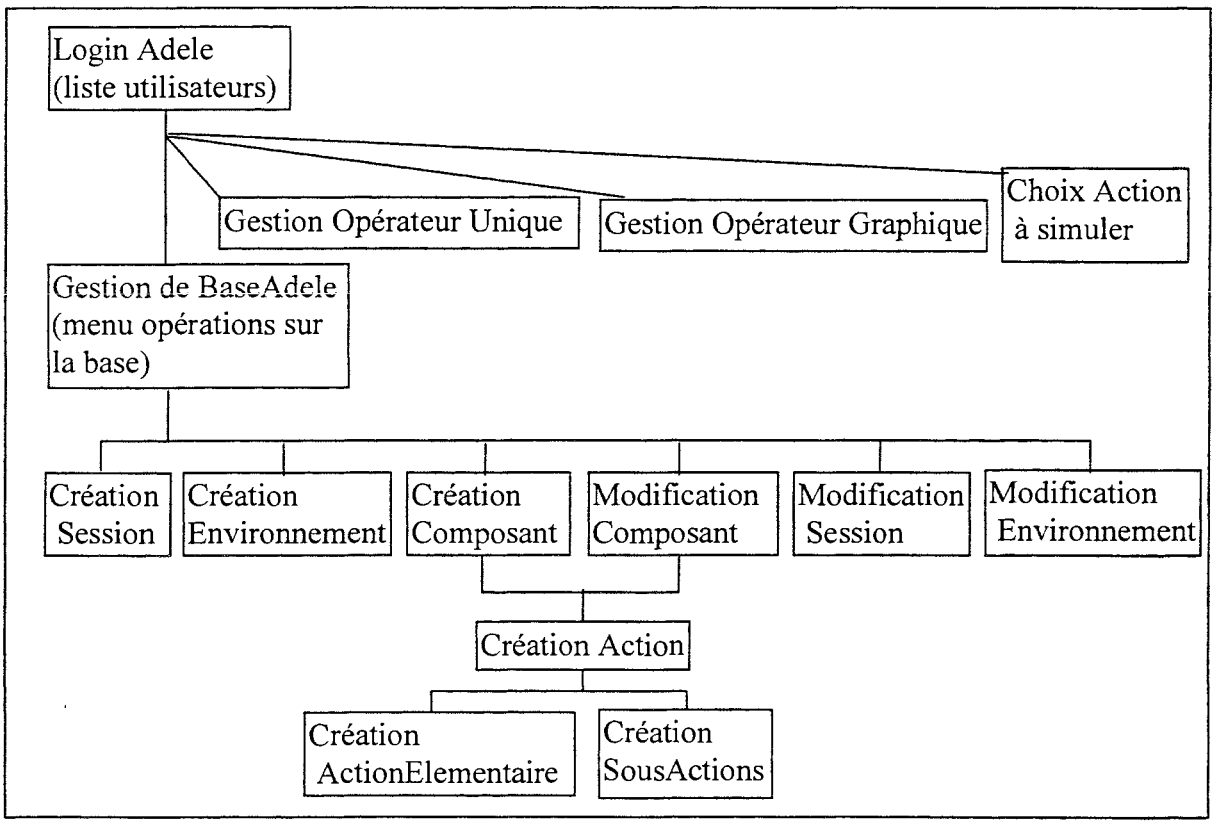

Figure 33 : Enchaînement des interfaces de BaseAdele

6.2.3 La base de données : BaseAdele

Le modèle objet décrit au chapitre 4 a été implémenté sous GemStone, dans une base nommée BaseAdele, qui contient environ 60 classes. La définition de ces classes est faite en Smalltalk. On trouvera en annexe 5 un extrait de la base.

Les figures suivantes décrivent les instances des objets dont la modélisation a été décrite au chapitre 4.

Il existe une instance d'opérateur générique (Figure 35), nommée « OpérateurUnique », permettant de décrire un opérateur avec des termes du langage courant. Ces termes sont reliés

aux termes qui décrivent la structure de MAN3D, qui sont stockés dans une instance d'opérateur graphique.

Une dizaine d'instances de composants ont été rentrées dans la base, avec 2 ou 3 actions associées à chacun des composants. La figure 34 fournit quelques exemples.

Le contenu de la base sera décrit plus en détail au chapitre 7, avec l'évocation des interfaces permettant de manipuler le système

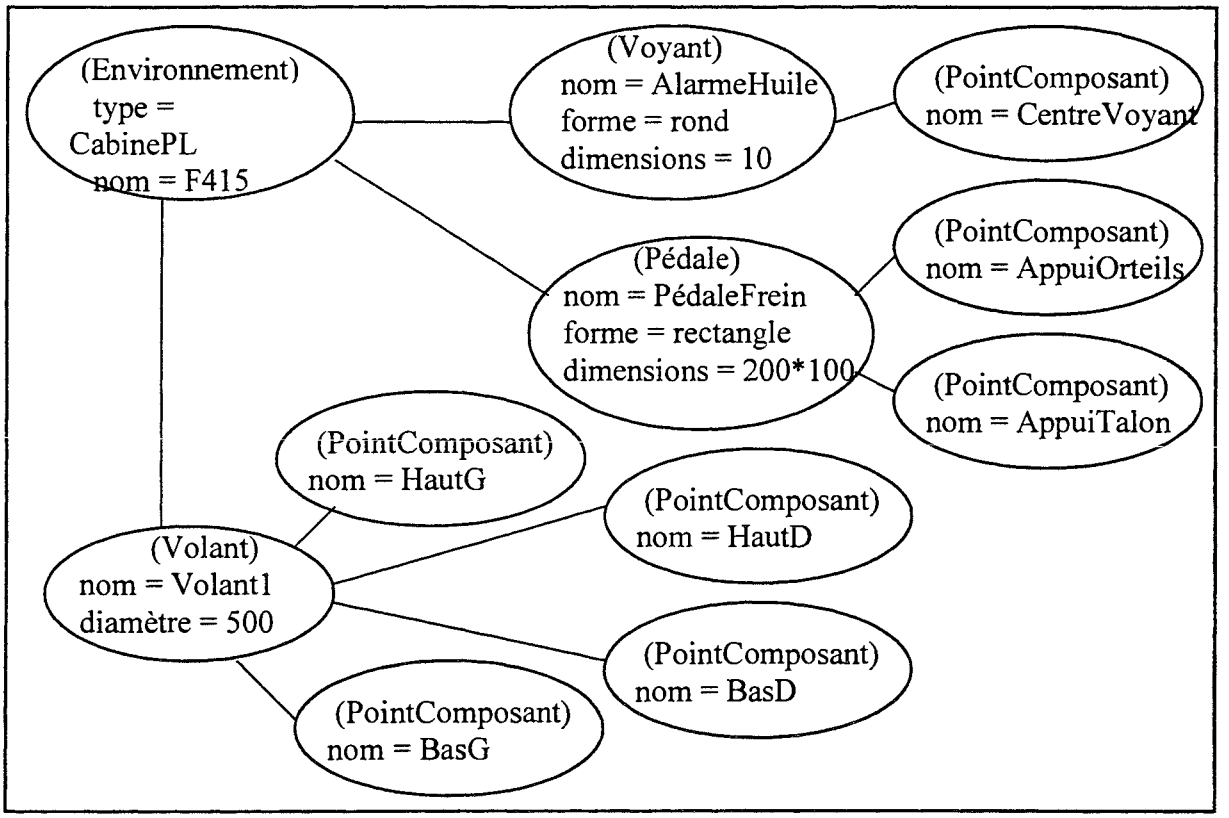

Figure 34 : Exemples d'instances (environnement, composants, points composants)

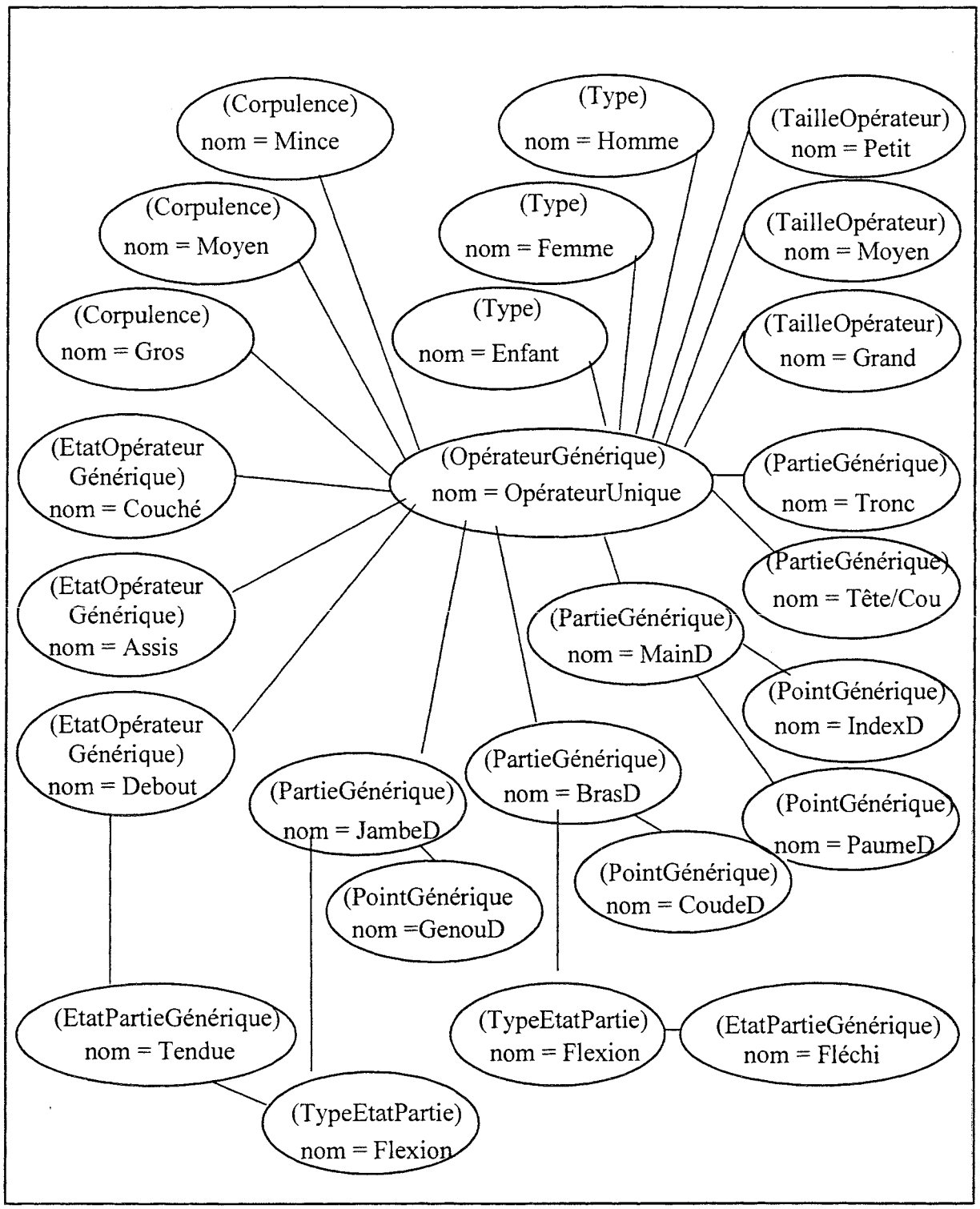

Figure 35 : Exemples d'instances (opérateur générique, types, corpulences, tailles, parties génériques, points génériques, états opérateur, états parties génériques)

6.3 Communication entre les différents modules du système

6.3.1 Communication entre SMA-Adele et MAN3D

La première des conditions nécessaires pour permettre le pilotage de MAN3D par un module externe était la conception d'un mode de communication dédié à cet usage. En effet, comme nous l'avons décrit plus haut (voir §1.3.1.5), MAN3D a été conçu initialement pour n'être utilisé qu'en mode interactif : l'utilisateur rentre les informations au clavier, les retours lui sont fournis à l'écran. Il a donc fallu concevoir un langage de commandes ainsi qu'un protocole pour la communication entre MAN3D et un module externe.

Une syntaxe a été définie et un équivalent-commande a été créé pour chacune des commandes de MAN3D. On dispose ainsi d'un ensemble que l'on peut classer en 4 catégories principales :

- les commandes liées à la définition du sujet et à la lecture/sauvegarde de postures,
- les commandes de positionnement général du sujet,
- la modification d'angle,
- les commandes de mouvement (impliquant la donnée d'un ou plusieurs points objectif).

À ces 4 catégories liées à la simulation proprement dite, s'ajoutent :

- les commandes de gestion de l'affichage,
- les "meta-commandes" (par exemple la lecture d'un fichier de commandes).

On trouvera en annexe 4 un récapitulatif des commandes disponibles à l'issue de l'implémentation du mode commandes (60).

L'envoi d'une commande donne lieu à au moins un message en retour. Il existe trois natures de retours attendus, répondant aux trois types d'interrogations : la commande a-t-elle été lue ?, sa syntaxe est-elle correcte ?, quel est le résultat de la commande ?. La première de ces interrogations relève du protocole de communication, les deux autres, bien que de natures différentes, ont trait au contenu du message envoyé initialement par le module qui communique avec MAN3D.

En ce qui concerne le protocole de communication, il fallait définir les éléments suivants :

- le mode de communication choisi,
- la nature du dialogue (qui interroge, qui répond),
- qui est maître du dialogue,
- les tâches que doivent réaliser les deux processus en parallèle avec le dialogue.

C'est la communication par fichiers qui a été choisie pour le prototype, en raison de sa portabilité (et malgré sa lourdeur). Les commandes envoyées à MAN3D doivent donc se trouver dans un fichier spécifique (de nom et de localisation prédéfinis). Après la lecture de ce fichier, MAN3D crée en réponse un fichier contenant les retours associés à chacune de ces commandes.

La réponse à la question de l'initiative du dialogue dépend du mode d'utilisation envisagé. En effet, bien qu'a priori l'envoi d'un fichier de commandes puisse être considéré comme

premier, la symétrie de fonctionnement des deux modules communicants autorise deux possibilités :

- soit l'initiative du dialogue part de MAN3D, ce qui suppose d'une part de définir le contexte de cette initiative, d'autre part de prévoir une commande spécifique partant de MAN3D pour initier le dialogue avec SMA-Adele,
- soit l'initiative du dialogue est prise au niveau de SMA-Adele.

Chacun des deux modules communicants fonctionne de la manière suivante : chacun peut avoir deux états possibles, un état de fonctionnement propre, qui ne nécessite pas de communication, et un état de communication. En mode communication, il s'agit soit d'envoyer un fichier (il s'agit ici de sa création), soit d'attendre l'arrivée d'un fichier, c'est-à-dire de boucler sur un test de présence de ce fichier.

Par ailleurs, il faut rappeler que MAN3D est interfacé avec un logiciel de CAO (Euclid dans le cas de la version avec laquelle nous avons travaillé) ; MAN3D est appelé par SMA-Adele au cours d'une session Euclid. La question est : faut-il ou non autoriser le fonctionnement en interactif de MAN3D en parallèle avec la communication avec ADELE ? Pour répondre à cette question, il faut définir le mode d'utilisation de ADELE. Dans une version très aboutie, ADELE pourra être un assistant intelligent qui délègue certaines tâches telles que l'affichage et les calculs géométriques à MAN3D, mais qui remplace celui-ci dans ses fonctions de dialogue avec l'usager. L'utilisateur pourra alors, au cours de sa session CAO, utiliser ADELE (comme il utilise actuellement MAN3D) pour répondre à son questionnement sur les positionnements des différents composants du produit dont il est en train de réaliser la maquette. Le prototype ne réalise pas cette version aboutie possible, puisque MAN3D et ADELE sont conçus comme deux modules ayant leur existence propre.

Le prototype, réalisé dans le but de tester avant tout la faisabilité des concepts, a donc été implémenté avec les spécifications suivantes :

- c'est l'utilisateur qui, au cours d'une session Euclid dans laquelle MAN3D est inclus, décide de faire appel à ADELE pour la simulation d'actions. Cet appel est effectué de l'extérieur, en lançant d'une part SMA-Adele, d'autre part Interface-Adele (ceci est détaillé dans le paragraphe suivant, 6.3.2).
- la première tâche réalisée par SMA-Adele, une fois que l'utilisateur a choisi une action à simuler, est le blocage de MAN3D en mode communication avec ADELE. Il existe donc une commande spécifique ("BOUCLE"), qui demande à MAN3D de se mettre en attente de commandes extérieures, ainsi qu'une autre commande ("RETOUR") permettant de remettre MAN3D en mode interactif. Ceci permet de réaliser l'asservissement envisagé pour la version finale de ADELE[12].
- le reste du dialogue est constitué d'envois de commandes par SMA-Adele à MAN3D (qui les réceptionne, puis les traite) et d'envois de messages de retour par MAN3D à SMA-Adele.

En ce qui concerne le contenu des messages provenant de MAN3D, nous avons vu plus haut qu'ils sont de deux natures : ils concernent soit la syntaxe, soit le contenu. Dans le prototype, il n'existe qu'un seul message concernant la syntaxe ("Syntaxe incorrecte"). Pour les messages concernant le contenu, deux cas se présentent :

[12] En pratique, il faut aussi lancer MAN3D depuis Euclid, pour qu'il détecte le fichier de commandes dans lequel se trouve la commande BOUCLE.

- la syntaxe est bonne, mais la commande n'a pas pu être traitée. Par exemple, dans le cas d'une commande d'atteinte d'un point cible, le nom du point n'est pas reconnu comme un nom de point existant sur la maquette CAO en cours (message "Point inconnu"),
- la syntaxe est bonne, la commande a pu être traitée. Dans ce cas, le résultat de la commande est fourni (pour une commande de mouvement messages "OK", "Point hors d'atteinte", "Articulation *xxxx* en butée"). Pour tous ces cas, le mannequin est affiché dans la position obtenue après exécution de la commande. Le message "OK" indique que la commande a été réalisée sans problème. Le message "Point hors d'atteinte" indique que l'exécution de la commande n'a pas permis au point de l'opérateur désigné d'atteindre le point cible désigné. Dans ce cas, les coordonnées du point atteint par le point de l'opérateur sont fournies. Enfin, le message "Articulation *xxxx* en butée" indique qu'une articulation est en butée et fournit le nom de cette articulation. Chacun de ces deux messages peut être obtenu isolément. Les deux messages peuvent aussi survenir simultanément.

6.3.2 Communication entre Interface-Adele et SMA-Adele

Comme nous l'avons évoqué plus haut (paragraphes 6.1.2.2 et 6.2.2), une session de simulation commence par le choix de l'action à simuler et ce choix est effectué au niveau de InterfaceAdele. Nous avons en effet choisi d'utiliser les fonctionnalités de création d'interface offertes par GemStone/VisualWorks pour gérer tout le dialogue avec l'utilisateur. Une autre raison justifiant la localisation du lancement de la simulation dans InterfaceAdele concerne la nature des échanges d'informations entre modules. En effet, pour ce dialogue avec l'utilisateur portant sur le choix de l'action à simuler, il y avait deux possibilités :

- soit l'utilisateur était interrogé depuis SMA-Adele : cela simplifiait l'envoi de la simulation, puisqu'il s'agit du module qui gère la simulation. Par contre, cela imposait aussi d'obtenir des informations non gérées au niveau du module de simulation (les listes d'actions contenues dans la base, parmi lesquelles l'utilisateur doit choisir celle qu'il veut simuler). Pour obtenir ces informations, il fallait alors soit passer par le dialogue entre SMA-Adele et InterfaceAdele, soit utiliser le partage d'informations entre les modules (évoqué en section 6.4).
- soit l'utilisateur était interrogé depuis InterfaceAdele, en effectuant directement son choix à partir d'une liste d'actions stockées dans la base. Cette solution ajoute la nécessité du dialogue pour l'envoi de la simulation.

Dans les deux cas, il fallait dialoguer, sachant que dans le deuxième, le dialogue est moins lourd (simple signal d'envoi). C'est cette deuxième possibilité qui a été choisie.

GemStone permet le dialogue entre sessions, à l'aide de primitives permettant l'envoi de messages. Nous avons donc choisi d'identifier SMA-Adele au niveau de la base comme une session GemStone. Finalement, SMA-Adele est donc une session GemStone pour deux raisons : pour que le dialogue puisse se faire avec InterfaceAdele, et parce que ce module doit accéder aussi directement aux informations contenues dans la base (aspect développé en section 6.4).

Nous avons donc conçu le concept de session communicante, et créé sous GemStone un objet correspondant dans la base.

L'objet *SessionCommunicante* a la définition suivante :

attributs :
numéro de session
temps maximum boucle d'attente
correspondant (du type SessionCommunicante)
méthodes :
retourneCorrespondant
retourneNum
changeValeurCorrespondant(nouveauCorrespondant)
login
logout
wait(temps)
envoiMessageAcorrespondant

InterfaceAdele et SMA-Adele sont donc identifiées sous GemStone comme deux instances de SessionCommunicante (comme elles ne sont rattachées à aucun autre objet de BaseAdele, ce sont deux racines de persistance, nommées respectivement SessionInteractive et SessionAdele). Elles sont initialisées comme étant correspondante l'une de l'autre. Les sessions sous GemStone sont identifiées par leur numéro. La création d'une instance de l'objet SessionCommunicante permet de réaliser un lieu pour stocker ce numéro : au moment ou le *login* (de SessionInteractive) est fait, *numéro de session* est initialisé avec le numéro donné à la session par GemStone (au moment de la connexion à GemStone). Ce numéro de session est un paramètre nécessaire pour l'envoi de messages entre sessions. Si les numéros des deux sessions n'étaient pas stockés dans SessionInteractive et SessionAdele, il serait difficile de reconnaître la bonne session à laquelle envoyer un message, parmi l'ensemble des sessions gérées par GemStone.

Le *login* de SessionInteractive est fait au moment ou l'utilisateur a choisi l'action a simuler (puisque c'est le point de démarrage du dialogue). Puis SessionInteractive envoie un message à SessionAdele, pour signaler qu'il est possible de démarrer la simulation.

Le *login* de SessionAdele est réalisé directement au moment où SMA-Adele est lancé (puisque sa première fonction est la simulation). Puis SMA-Adele/SessionAdele est en attente d'un message provenant de InterfaceAdele/SessionInteractive.

Outre le dialogue permettant d'échanger entre les deux sessions le signal d'envoi de la simulation, il existe une information à partager : l'action à simuler, choisie par l'utilisateur. C'est un objet beaucoup trop complexe pour qu'il puisse être transmis par un message. Il s'agit donc d'une information commune, à partager. Une fois que l'utilisateur a choisi l'action à simuler, il faut disposer au niveau de SMA-Adele, d'un "pointeur" sur cette action. Comme nous avons choisi de ne pas transmettre de données dans les messages, ceci a donc été réalisé par la création d'une autre racine de persistance, nommée *ActionChoisie* (de type Action au sens du modèle décrit dans la partie 4), mise à jour par InterfaceAdele, et lue par SMA-Adele (selon un procédé que nous décrivons dans la section 6.4) après l'envoi du signal de début de la simulation.

6.4 Le partage de données entre Atome et GemStone

6.4.1 Le problème à résoudre

Nous avons évoqué plus haut (section 5.3) la nécessité du partage de données entre le SGBD objet et Atome. Il s'agit de faire en sorte que les informations contenues dans BaseAdele soient utilisables par SMA-Adele. Plus précisément, SMA-Adele étant un système multi-agents à base de tableau noir, l'objectif est de concevoir un mécanisme permettant d'indiquer que certaines données du tableau noir sont dans la base de données. Cet objectif nécessitait une transformation structurelle, du point de vue de Atome, puisqu'il fallait concevoir une nouvelle manière de définir les nœuds des tableaux noirs.

Concrètement, le problème est le suivant : dans une application Atome, les règles des spécialistes sont activées à partir de faits présents dans les tableaux noirs. Avec la version C++ de Atome, que nous utilisons, les nœuds des tableaux noirs correspondent à des déclarations d'objets C++, dont la déclaration est locale à l'application multi-agents. On veut que les règles des spécialistes d'une application Atome puissent aussi être activées à partir de faits présents dans une base de données externe (située dans notre cas sous GemStone). Pour conserver l'intégrité du paradigme multi-agents, il faut que ces faits aient le même statut que les autres, à savoir des nœuds de tableaux noirs. Il faut donc concevoir la possibilité de spécifier que certains nœuds sont externes lorsque l'on conçoit une application Atome, et fournir les primitives permettant d'accéder à ces nœuds.

Ainsi, dans le cadre du développement de ADELE, une nouvelle version de Atome a été conçue. Son implémentation a fait l'objet d'un stage (61).

Nous évoquons ci-dessous tout d'abord les caractéristiques de GemStone permettant l'accès aux données depuis un module externe (paragraphe 6.4.2), puis nous décrivons brièvement la nouvelle version de Atome permettant l'utilisation de GemStone pour la définition de nœuds des tableaux noirs (paragraphe 6.4.3), enfin nous décrivons comment cette nouvelle version de Atome a été utilisée pour le système ADELE (paragraphe 6.4.4).

6.4.2 Interface C++ avec GemStone

GemStone a son propre langage SmallTalk, mais les applications GemStone peuvent aussi être écrites dans d'autres langages, tels que d'autres SmallTalk, C, C++, ou même tout langage pouvant faire des appels en C (comme COBOL ou Fortran). Pour transférer des objets entre un programme application écrit en C++ (cas qui nous intéresse) et la base de données, GemStone fournit des librairies de classes et de fonctions C++ (rassemblées sous le nom GC++I, pour GemStone C++ Interface). Grâce à cette interface on peut :

- créer des objets C++ dans une application externe et les stocker dans GemStone,
- importer des objets GemStone dans une application externe pour les manipuler comme des objets C++,
- exécuter des fonctions systèmes telles que la validation de transactions, le démarrage ou la terminaison de sessions GemStone.

GC++I fournit donc à la fois un stockage d'objets persistants pour des applications C++ et un accès à des objets persistants stockés dans GemStone par des applications écrites avec d'autres langages.

Tout objet manipulé dans ce contexte doit remplir les conditions suivantes :

- il ne peut être rendu persistant que par un *commit*,
- il doit être rattaché à un dictionnaire de symboles GemStone,
- il doit être d'une classe persistante qui peut être : soit une classe GemStone prédéfinie, soit une sous-classe d'une classe GemStone, définie par l'utilisateur.

Par ailleurs, l'accès aux objets persistants se fait à travers des pointeurs. En effet, une fonction membre (c'est-à-dire une méthode) qui retourne un objet persistant doit en fait retourner un pointeur GemStone (GPTR). Les fonctions membre ne peuvent pas retourner les objets persistants eux-mêmes ni des pointeurs C++ sur les objets persistants.

Pour créer des classes persistantes, il faut utiliser un utilitaire de GC++I, le *Registrar*. Il pré-analyse et il stocke dans la base de données GemStone les définitions de classes persistantes. L'enregistrement d'une telle classe nécessite de fournir au *Registrar* sa définition sous la forme d'un fichier d'extension *.hxx* pour la déclaration de la classe et d'un fichier d'extension *.cxx* pour l'implémentation des fonctions membre.

Le *Registrar* est en fait lui-même une application, qui se connecte à GemStone et qui crée, puis stocke les nouvelles définitions de classes dans le dictionnaire de symboles approprié. Il peut être utilisé en interactif ou lancé en procédure à distance.

6.4.3 Une nouvelle version de Atome

Nous ne décrirons pas ici de manière complète l'implémentation de la nouvelle version de Atome (pour plus de détails on se reportera à (61)). L'idée est ici d'indiquer la nature des modifications, d'en fournir le principe, et enfin de décrire le résultat constitué par les nouvelles possibilités offertes pour le développement d'applications avec Atome.

Atome fournit au programmeur un ensemble de primitives et de mots-clefs lui permettant de décrire un ou plusieurs tableaux noirs, de créer une stratégie, des tâches, des spécialistes et d'écrire les règles associées à chacune de ces sources de connaissance. Les prémisses des règles des spécialistes contiennent des références aux faits contenus dans les tableaux noirs. Le corps de la partie action des règles est constitué de code C++ et/ou de mots-clefs et primitives Atome.

Un tableau noir Atome est composé de niveaux. Chaque niveau correspond à un type de nœud. Un nœud est constitué d'un ensemble d'attributs et de liens. Les liens permettent de créer des relations bijectives entre différents nœuds, quelle que soit leur nature. Plusieurs opérations sont possibles sur un tableau noir : créer un nœud, détruire un nœud, modifier un nœud, et sélectionner les nœuds vérifiant un critère donné.

La modification souhaitée pour Atome consiste à étendre les possibilités de définition des nœuds, afin que l'utilisateur (le programmeur d'une application Atome) puisse définir des nœuds contenant soit des objets GemStone, soit des objets C++. Mais la manipulation des nœuds doit rester la même quel que soit le type d'objet manipulé.

Les modifications apportées à Atome ont été de deux natures :

- la modification des classes Atome : les classes relatives au tableau noir ont été modifiées en profondeur (la hiérarchie a notamment été redéfinie) et certaines classes relatives à la gestion des entrées/sorties avec GemStone ont été rajoutées.
- la modification de l'automate d'analyse, qui, à partir des fichiers de définitions fournis par l'utilisateur, génère des fichiers sources C++ composant le système à base de tableau noir. En effet, les nouvelles primitives offertes à l'utilisateur imposaient la modification de l'analyseur syntaxique et lexical.

Dans la nouvelle version de Atome, il existe une classe *nœud*, à laquelle sont associés d'une part un *objet* et d'autre part une liste de *liens*. L'objet peut être de deux natures : soit un pointeur sur un objet C++, soit un pointeur sur un objet GemStone. Pour définir ses tableaux noirs, l'utilisateur-programmeur doit définir le type de l'objet associé à chaque nœud. Il a finalement trois possibilités pour définir les nœuds des tableaux noirs :

- il peut définir un nœud ayant un objet de type "classe définie par Atome" : il fournit le type de chaque attribut et la définition de classe correspondante est générée automatiquement par Atome. Cette possibilité correspond à l'ancienne définition de nœud. La définition des nœuds est dans ce cas complètement interne à Atome.
- il peut définir un nœud ayant un objet de type "classe définie par l'utilisateur" : il doit fournir le nom de la classe, ainsi que le nom du fichier d'extension *.h* contenant sa définition. On peut ainsi attacher à un tableau noir Atome, des nœuds dont la définition a été faite extérieurement, en C++, éventuellement dans le cadre d'une autre application.
- il peut définir un nœud ayant un objet de type "classe définie pour GemStone" : il doit fournir le nom de la classe, ainsi que le nom du fichier d'extension *.hxx* contenant sa définition. Cela permet d'attacher à un tableau noir de Atome un objet défini en C++ et intégré sous GemStone.

6.4.4 L'accès aux données de BaseAdele par SMA-Adele

La modification de Atome évoquée au paragraphe précédent ne constitue qu'une base pour la résolution de notre problème, qui est de rendre accessibles les objets contenus dans BaseAdele, pour SMA-Adele. En effet, l'utilisation de la définition de nœuds avec objet de type "classe définie pour GemStone" est limitée aux classes créées en C++, dont on doit intégrer la définition dans GemStone, à l'aide du Registrar (voir paragraphe 6.4.2). Or dans notre cas, nous travaillons avec des classes définies avec GemStone, en Smalltalk. Pour pouvoir travailler directement avec ces classes et éviter d'en dupliquer la définition en C++, nous partons donc de la deuxième des possibilités nouvelles (décrites au paragraphe 6.4.3) : la définition du type de l'objet d'un nœud à partir d'une définition de classe C++ quelconque. L'idée est, en utilisant ce mécanisme, de créer une classe spéciale, qui constituera une sorte de "moule" permettant de véhiculer tout objet GemStone, quelle que soit sa nature.

Nous avons ainsi conçu la classe C++ *Liaison_Base*. Cette classe a pour seul attribut *obj*, qui a pour type *GS_Object_GPTR* (le type des pointeurs sur les objets GemStone). Elle comporte une ensemble de méthodes :

- des méthodes de création d'objets de type *Liaison_Base*, à partir de différents types d'objets (chaîne de caractères, entier, réel, pointeur GemStone),
- des méthodes de transformation, permettant de traduire un entier, un réel, ou une chaîne GemStone en entier, réel ou chaîne C++,
- une méthode permettant d'imprimer l'objet,
- une méthode fournissant la classe de l'objet,

- une méthode permettant d'envoyer un message à l'objet GemStone, sous la forme d'une chaîne de caractères composée d'un ou plusieurs paramètres. Cette méthode permet d'envoyer à l'objet GemStone n'importe quel message exécutable sous GemStone et donc de lui appliquer ses méthodes (ou celles des classes dont il hérite).

Ainsi, dans SMA-Adele, les nœuds dont la partie *objet* est située dans GemStone sont déclarés avec le type *Liaison_Base*. C'est le cas notamment pour les deux premiers niveaux du tableau noir ACTION : Actions et Sous-Actions. A l'exécution de ADELE, lorsque le signal d'envoi de la simulation a été reçu, le premier nœud créé est un nœud du niveau Actions, qui pointe sur un objet GemStone, la racine de persistance *ActionChoisie*, dans laquelle a été stocké le pointeur sur l'action choisie par l'utilisateur pour la simulation (comme cela a été décrit au paragraphe 6.3.2). Les sous-actions composant cette action sont récupérées, au moment de la création des nœuds du niveau Sous-Actions, grâce à l'envoi de messages (par les méthodes de *Liaison_Base*) à l'objet pointé par le nœud de type Action. Lorsqu'il est nécessaire d'accéder aux caractéristiques d'une sous-action donnée (par exemple pour élaborer la ligne de commande MAN3D correspondante), ce même mécanisme sert à parcourir la hiérarchie des objets.

7 Manipulation et évaluation du système

7.1 Utilisation de ADELE

ADELE sert donc à simuler des actions sur une maquette de produit en cours de conception. Ces actions doivent être décrites dans BaseAdele. Lorsque la description de l'environnement est instanciée dans la base, avec les composants présents et les actions associées, l'utilisateur peut lancer la simulation d'une action, sur une maquette présente sous Euclid.

La Figure 36 est la copie-écran de l'interface permettant d'accéder à BaseAdele.

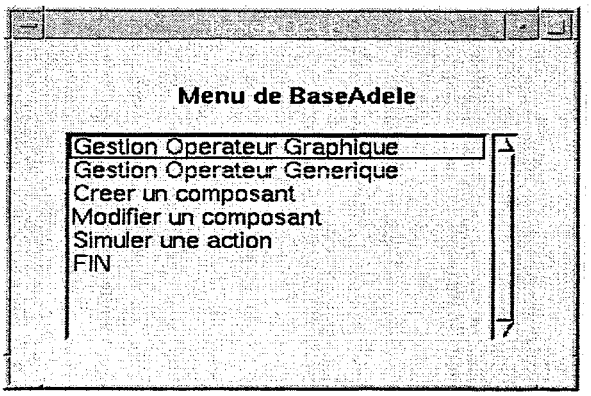

Figure 36 : Menu d'accès à BaseAdele

Sur ce menu sont réunis plusieurs modes d'accès à la base, correspondant à différents niveaux d'utilisation.

L'item *Gestion Opérateur Graphique* permet d'accéder à une interface permettant de rentrer dans la base les noms des différents éléments de la structure du modèle MAN3D (Figure 37). Cette interface permet aussi de définir ou de modifier les liaisons entre les éléments de la structure graphique et leurs correspondants définis dans le modèle générique d'opérateur. Cette utilisation est réservée à un utilisateur expert capable de réaliser ces liens : par exemple capable de décider à quel point de la structure MAN3D doit correspondre la paume de la main. La description de l'opérateur graphique ainsi que la liaison OpérateurGénérique/ OpérateurGraphique sont donc paramétrées, elles sont modifiables en utilisation courante (sans nécessité de réécriture et recompilation du système). Le système ADELE est donc ouvert aux modifications éventuelles du modèle MAN3D.

L'item *Gestion Opérateur Générique* du menu de BaseAdele permet d'accéder à l'unique instance d'opérateur générique : Opérateur Unique (Figure 38). La définition et la modification des noms désignant la structure de l'opérateur et ses caractéristiques correspondent à une phase de définition du modèle utilisé par ADELE. Ces noms sont ceux qui sont utilisés en prémisses de règles, dans SMA-Adele. L'ajout de nouveaux noms ou la

modification de noms existants impose le report des modifications dans les règles (modification des noms ou ajout de nouvelles règles) et donc la recompilation du système.

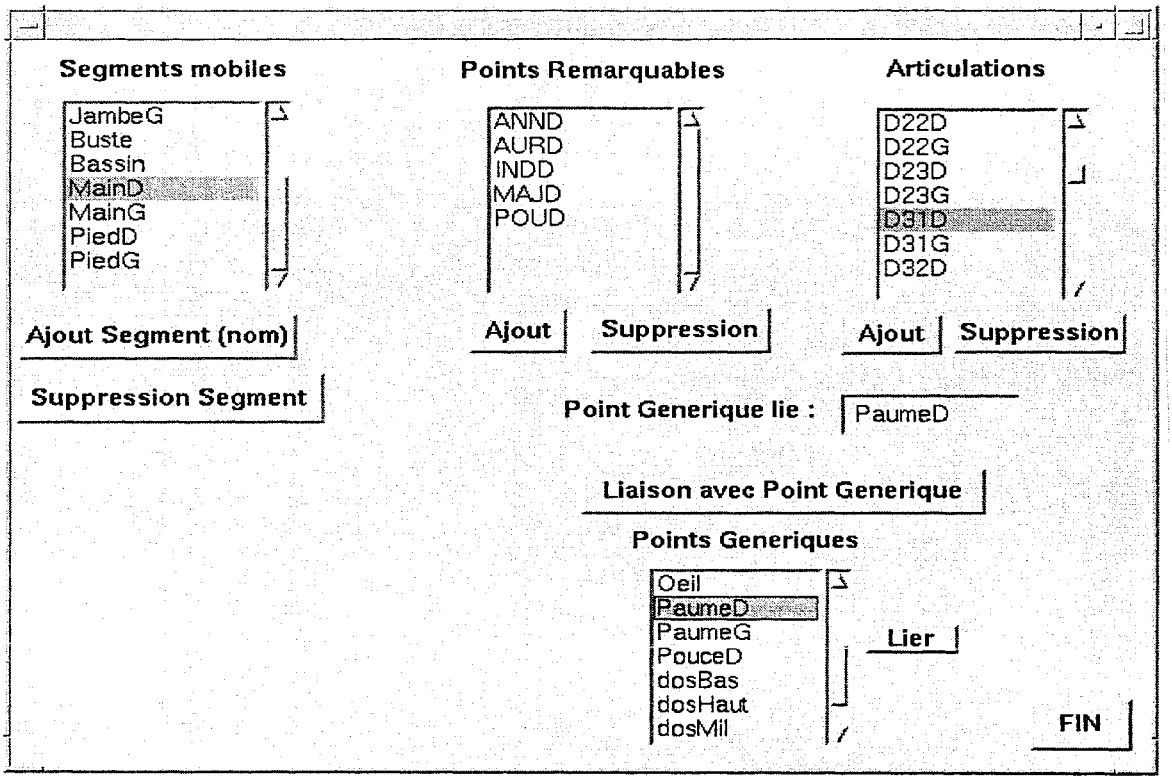

Figure 37 : Gestion de l'opérateur graphique

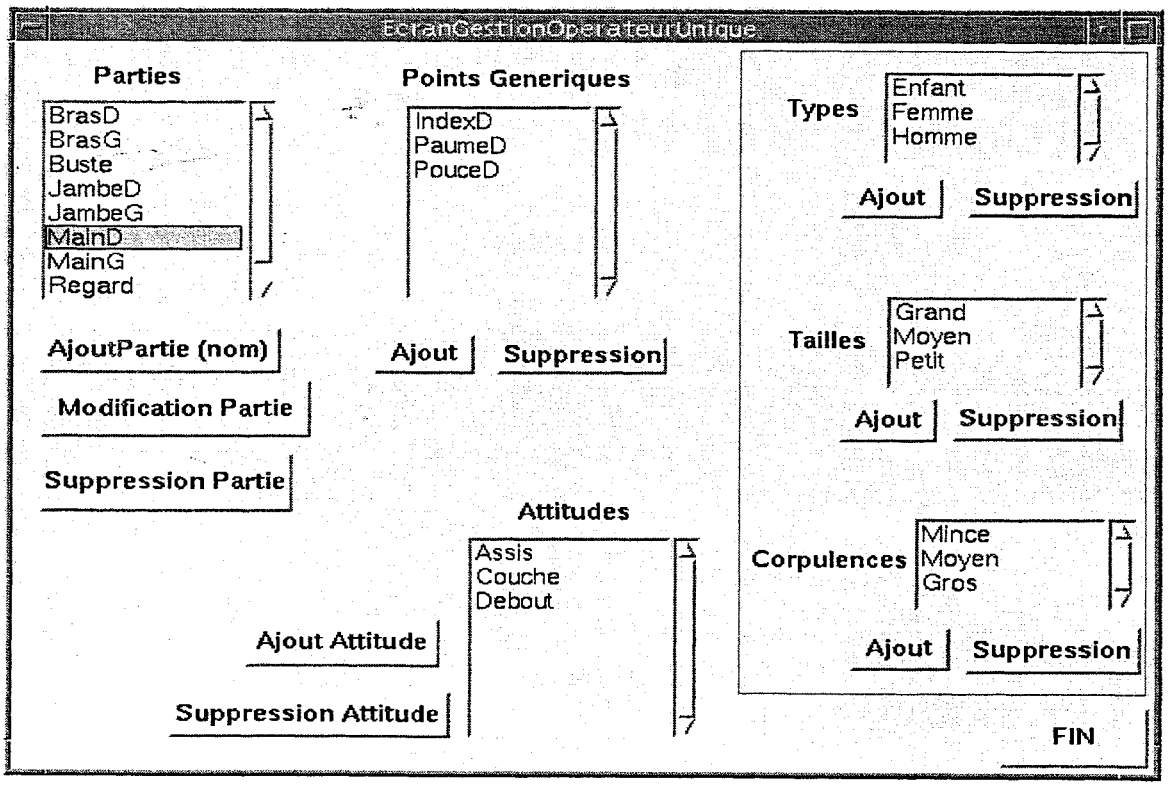

Figure 38 : Gestion de l'opérateur générique

Les autres items du menu d'accès à BaseAdele correspondent à une utilisation plus courante : la mise à jour de la base (créer ou modifier un composant et ces caractéristiques) et l'envoi de la simulation d'une action.

L'interface de création ou de modification d'un composant (figure 39), permet de définir ou de modifier de nouvelles actions associées au composant. Lorsque l'on définit une nouvelle action (figure 40), on doit spécifier si elle est élémentaire (accès à l'interface de définition d'une action élémentaire, figure 41) ou bien composée (accès à l'interface de définition d'une action composée, figure 42).

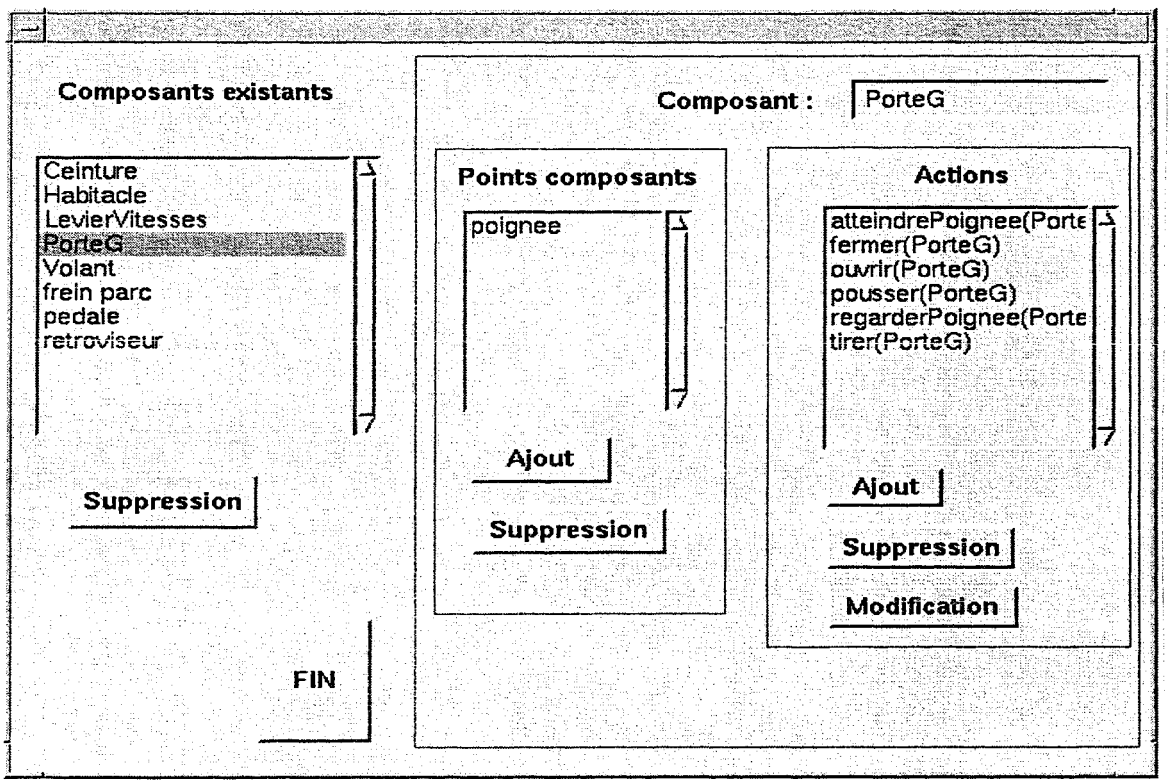

Figure 39 : Accès aux caractéristiques d'un composant

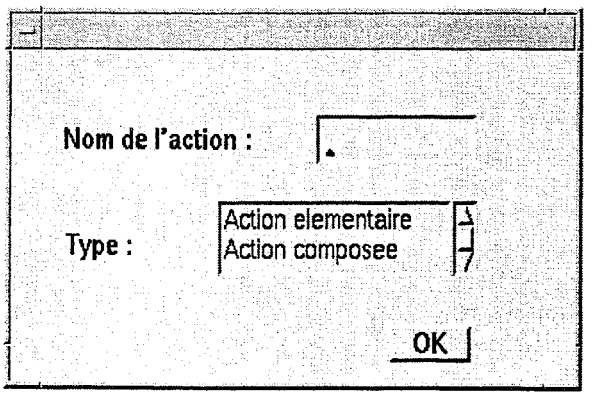

Figure 40 : Création d'une action associée à un composant

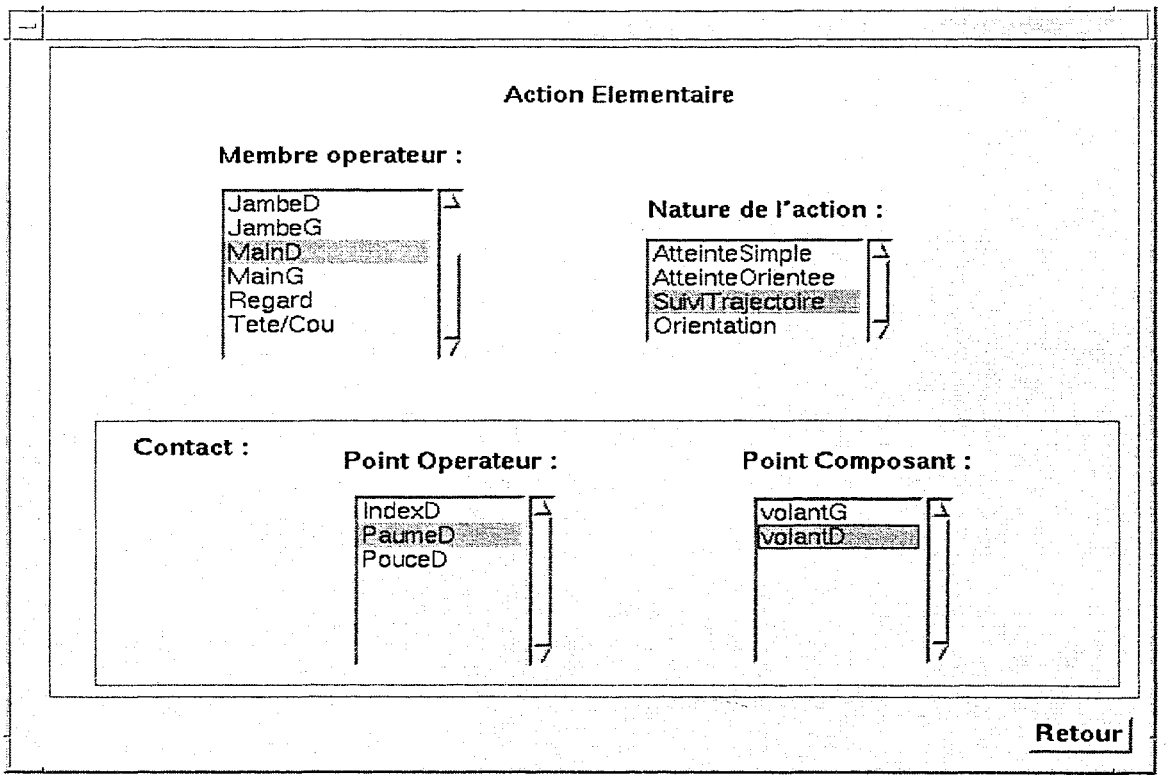

Figure 41 : Définition d'une action élémentaire

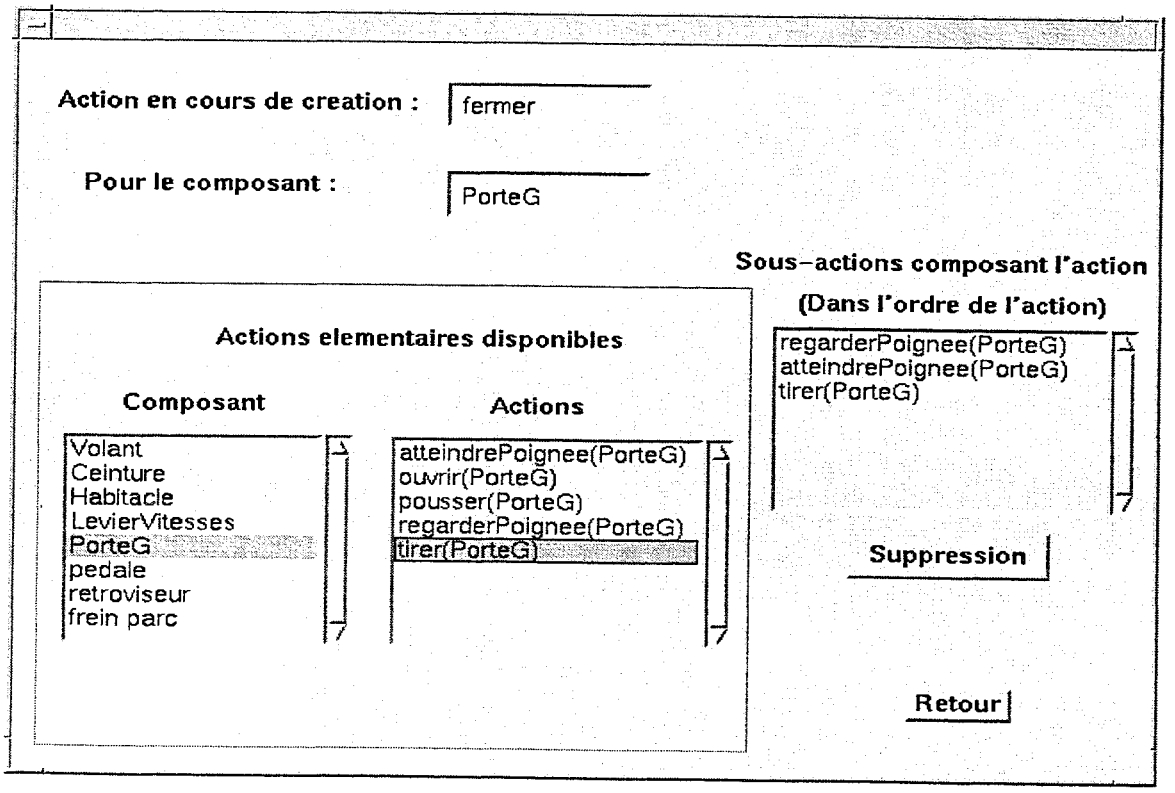

Figure 42 : Définition d'une action composée

L'item *Simuler une action* du menu d'accès à BaseAdele permet d'accéder à l'interface de choix d'une action pour la simulation (Figure 43).

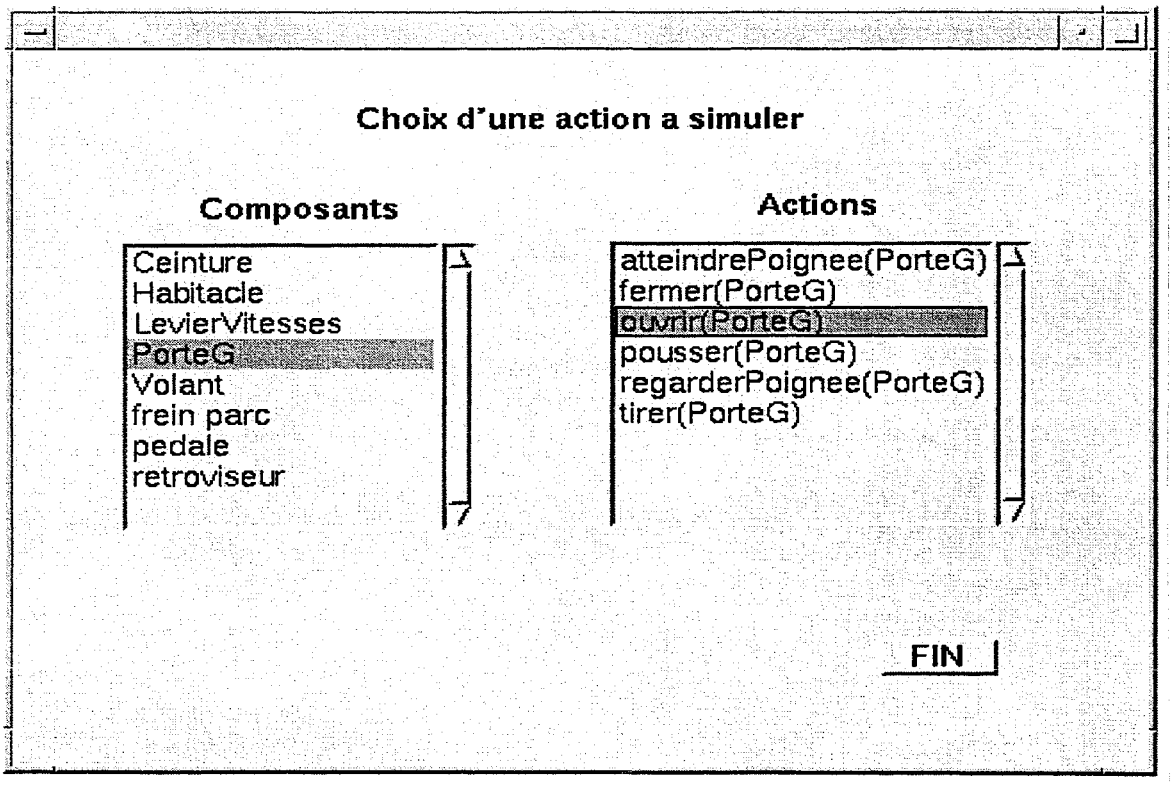

Figure 43 : Interface de choix d'une action pour la simulation

La validation du choix de l'action entraîne le lancement de la simulation : le module de raisonnement/pilotage (SMA-Adele) est alors activé et le dialogue s'engage avec MAN3D, entraînant la modification de la posture du mannequin affiché à l'écran. L'utilisateur a alors la possibilité de conserver la posture obtenue (soit pour demander une autre simulation, soit pour poursuivre son utilisation de MAN3D sous Euclid), ou bien de revenir à la posture initiale (existant avant l'enchaînement dicté par ADELE conduisant à une nouvelle posture).

C'est cette version du prototype qui a été utilisée pour l'évaluation, décrite dans la section suivante.

7.2 Objectifs de l'évaluation

L'évaluation d'un système à base de connaissances consiste en général essentiellement à évaluer la connaissance modélisée. Lorsque cette connaissance est celle d'un expert humain, la validation consiste à s'assurer de l'adéquation entre la connaissance modélisée dans la base de connaissance et la connaissance réelle que l'expert a dans la tête. Deux méthodes de validation sont envisageables dans cette optique : une validation par l'examen des résultats obtenus à partir de la base de connaissances et une validation par l'étude de la cohérence de la base de connaissances [Ayel et Rousset 90].

Dans notre cas, nous avons deux ensembles de connaissances : les connaissances portant sur la simulation de l'activité et celles portant sur le diagnostic ergonomique. Le cas du diagnostic

ergonomique est proche du cas classique : l'expertise à implémenter doit se rapprocher de celle de l'ergonome en charge des questions ergonomiques dans les équipes de conception. Par contre, pour la simulation de l'activité, et notamment pour les connaissances portant sur les stratégies de mouvement, il n'y a pas d'expert qui ait cette expertise "dans la tête". Nous avons vu que la recherche des règles portant sur les stratégies de mouvement passe notamment par la mise en œuvre d'expérimentations ayant pour objectif d'observer les gestes réels. La validation de ces règles dans un objectif de simulation consisterait donc à confronter la simulation obtenue avec le cas réel. Cela imposerait la définition d'un ensemble de problèmes permettant de constituer un jeu de tests et surtout la connaissance a priori de la solution à chacun de ces problèmes dans le cas réel, ce qui est donc loin d'être trivial. Ce mode de validation, qui correspond à la première des deux méthodes évoquées ci-dessus, ne peut être envisagé qu'à long terme, et dépasse le cadre de ce travail.

Nous nous intéressons ici à l'évaluation du prototype du système ADELE que nous avons développé. Nous avons précisé lors de la description du prototype que celui-ci ne constitue qu'une ébauche du système envisagé, notamment en ce qui concerne les bases de connaissances intégrées. Les objectifs de ce prototype sont d'une part de prouver la faisabilité de la mise en place des concepts, d'autre part de servir de démonstrateur à destination des usagers futurs d'un système dont il faudra définir les orientations de développement. Compte tenu de la nouveauté de l'approche proposée pour le modèle de simulation, il était nécessaire de passer par une phase de prototype ayant pour objet de faire comprendre les possibilités nouvelles se dessinant afin de démarrer véritablement ensuite la phase de constitution de l'expertise. L'objectif de l'évaluation est donc de s'assurer que le prototype répond bien à sa finalité d'explication et à son statut de démonstrateur.

7.3 Mise en œuvre de la phase de test

La phase de test a donc été conçue dans le but de présenter les nouvelles possibilités offertes par le système ADELE par rapport au système existant (MAN3D). Rappelons que les nouveautés apportées par le système ADELE sont les suivantes :

- c'est un générateur de commandes MAN3D, sur la base d'une spécification exprimée "en langage courant",
- c'est un générateur et un gestionnaire d'une succession d'actions (ou d'actions simultanées),
- il est capable de piloter MAN3D en fonction du contexte initial et en fonction de ce qui se passe à l'exécution de la commande, ainsi que de générer éventuellement de nouvelles commandes pour atteindre l'objectif initial,
- enfin, toutes les conditions sont réunies pour permettre l'émission d'un diagnostic ergonomique associé à une ou plusieurs actions, prenant en compte l'état postural obtenu, et comparant le résultat de la simulation avec les préconisations du cahier des charges ergonomique.

Pour l'évaluation du prototype plusieurs aspects ont été différenciés. Pour l'évaluation en tant que démonstrateur d'un futur système d'assistance pour l'évaluation ergonomique, nous avons tout d'abord soumis le système à notre interlocuteur de Renault Véhicules Industriels (RVI), ergonome. Il s'agit d'une personne connaissant MAN3D, qui l'utilise dans le cadre de son activité d'ergonome intégré dans l'équipe de conception. Son travail consiste notamment à répondre aux demandes des dessinateurs/concepteurs chargés d'implanter les fonctions dans

l'habitacle du véhicule en cours de conception (poids-lourd, car ou bus). Pour la nouveauté de l'approche utilisée pour la simulation de l'activité motrice, nous avons soumis le prototype à un biomécanicien (en cours de thèse sur l'analyse du pivotement du bras, au LESCO). Enfin, pour la pertinence de l'approche dans son ensemble, notre interlocuteur a été le concepteur et développeur de MAN3D (biomécanicien au LESCO).

Pour le test proprement dit, le déroulement a été le suivant :

- une phase de rappel des objectifs du système et de ses fonctionnalités,
- une phase de démonstration, incluant :
  - la démonstration de l'enchaînement BaseAdele/SMA-Adele/MAN3D : Sélection d'une action dans la base (correspondant à une commande simple) - Activation de la simulation avec appel automatique de MAN3D et exécution de l'action,
  - la démonstration de l'enchaînement de plusieurs commandes successives (à partir d'une action composée),
  - la démonstration de la réaction à un message de MAN3D, avec une prise de décision et la génération d'une nouvelle commande (à partir d'une action portant sur un point à atteindre situé hors de la sphère d'atteinte du bras seul),
  - la démonstration de la prise en compte du contexte (avec une différenciation sur la génération des commandes selon la localisation du point devant ou derrière l'opérateur),
  - la démonstration de l'utilisation de la base de données pour créer des définitions de composants et spécifier les actions possibles sur ces composants,
- une phase d'utilisation libre,
- une phase de verbalisation, commencée lors de la phase d'utilisation libre par une verbalisation simultanée et complétée par un entretien post-expérimentation.

7.4 Bilan de l'évaluation

La présentation de ADELE à ses utilisateurs futurs a permis de montrer de quelle manière les objectifs annoncés pouvaient être atteints. Dans cette optique, le prototype remplit bien son rôle de démonstrateur. Par ailleurs, les personnes consultées ont réagi différemment suivant leur position.

Notre interlocuteur ergonome de RVI a été particulièrement intéressé par les aspects pilotage automatisé du mannequin, et base de données d'actions. En effet, la nouveauté permettant de commander le mannequin "en langage clair" ouvre la possibilité de faire utiliser l'outil par les utilisateurs novices et/ou non experts, sans phase d'apprentissage lourde. Par ailleurs, cela permet de proposer aux dessinateurs une check-list prédéfinie des tests à effectuer pour chaque composant, l'enjeu étant de rendre les dessinateurs autonomes sur certains aspects des tests ergonomiques. Sur l'aspect simulation proprement dit, l'outil reste dépendant de MAN3D et de ses capacités à être "représentatif du geste réel".

Suite à ce test, on peut donc envisager de proposer le développement "en vraie grandeur" d'un outil d'assistance au diagnostic ergonomique, dédié aux dessinateurs RVI. Ceci implique une coopération visant au développement de cet outil, avec d'une part une réflexion de nature technique, sur le choix des composants logiciels notamment (par exemple quel SGBD utiliser) et d'autre part la mise en place d'une collaboration pour le développement effectif de la base

de connaissances portant sur l'évaluation ergonomique. Il faut en effet affiner la liste des critères d'évaluation et décider des modèles ou des heuristiques à implémenter. Par ailleurs, le mode de fonctionnement proposé avec le prototype peut être complété par un mode plus automatique. En effet, actuellement, c'est l'utilisateur qui décide d'une ou plusieurs actions à simuler pour le test d'un composant. On peut envisager d'automatiser le test de chaque composant (comme cela a déjà été évoqué au paragraphe 4.4.3), en générant les actions nécessaires au tests associés aux préconisations de départ concernant chaque composant (portant sur la visibilité, l'accessibilité, la manœuvrabilité).

Les biomécaniciens du LESCO ont eux des centres d'intérêt différents. Leur objet d'étude étant la modélisation du geste, ils sont plus concernés par l'aspect simulation que par celui du diagnostic ergonomique. L'enjeu du démonstrateur est donc de montrer quel type d'enrichissement peut apporter un outil basé sur des techniques qualitatives, pour la problématique de la simulation. Les réactions obtenues sont de deux natures, et illustrent la "séparation des cultures" à laquelle on doit se confronter. Pour une personne n'ayant pas de sensibilité aux techniques de l'Intelligence Artificielle mais ayant une culture informatique classique et étant de plus habituée aux outils de modélisation numériques, la nouveauté proposée est très difficile à appréhender. Sur la démonstration de ADELE, nous avons en effet assisté à des réactions d'incompréhension. Pour notre interlocuteur biomécanicien en thèse au LESCO, l'aspect habillage des commandes est perçu, mais avec l'idée que sous chaque action en langage réel, il y a simplement un fichier prédéfini listant les commandes MAN3D à réaliser. En insistant sur les aspects génération de commandes et réactions différentes du système suivant le contexte, l'incrédulité persiste. Le problème est celui de la compréhension du "comment". Ne disposant pas des outils conceptuels permettant de comprendre comment ceci est implémenté, notre interlocuteur ne peut pas percevoir l'intérêt de la démarche. Nous sommes dans un tout autre cas de figure avec le concepteur de MAN3D. Biomécanicien et donc de culture proche de la personne précédente, il a par contre été sensibilisé à l'Intelligence Artificielle par des stages d'initiation. Par ailleurs, ce travail de thèse a été l'occasion (pendant plusieurs années !) d'échanges permettant de "raccourcir les distances entre les problématiques". Pour cette personne, le prototype permet de boucler le travail de conception et démontre la faisabilité réelle de ce qui avait été simplement annoncé jusque-là. De plus, l'existence du prototype est la condition sine qua non pour faire avancer la recherche commune des règles portant sur les stratégies de mouvement.

Dans cette optique, le prototype de ADELE peut être perçu comme un outil "d'aide à la recherche". En effet, la proposition de règles heuristiques sur les stratégies de mouvement, intégrées dans le modèle de simulation produit un premier niveau de résultat, qui peut être évalué de manière visuelle, à l'exécution. Une règle apparaissant comme satisfaisante peut servir d'hypothèse de base à tester avec une expérimentation. Par ailleurs, l'observation du réel (dans un cadre expérimental ou non) peut elle-même conduire à l'intégration de nouvelles règles sur les stratégies de mouvement. On peut ainsi à la fois enrichir l'outil et produire une sorte de dynamique en fournissant des objectifs à la recherche sur les stratégies de mouvement.

Conclusion générale

Notre travail se situe à l'interface entre trois disciplines : l'informatique, l'ergonomie et la biomécanique. La prise en compte des éléments de l'environnement pour la simulation des actions motrices de l'Homme élargit cette problématique de simulation, jusqu'ici limitée aux aspects biomécaniques. Plus généralement, nous avons proposé ici une approche de nature symbolique complétant l'approche traditionnelle essentiellement numérique. De plus, le fait de reformuler la question de la simulation des mouvements moteurs en rapport avec l'objectif poursuivi, qui est ici le diagnostic ergonomique, nous a permis d'élaborer un modèle associant des connaissances des deux champs disciplinaires que sont l'ergonomie et la biomécanique.

Concrètement, nous avons proposé :

- un modèle de l'activité motrice de l'opérateur,
- une architecture de système associant un modèle de simulation graphique de l'activité motrice, un système à base de tableau noir et une base de données objet,
- l'implémentation de ces aspects dans un prototype servant de démonstrateur.

Le prototype ne constitue qu'une ébauche de système, sa faiblesse résidant dans la pauvreté des bases de règles développées, tant du point de vue des stratégies de mouvement que du diagnostic ergonomique. En effet, le problème est celui de la disponibilité de l'information. Dans le cas des stratégies de mouvement, la connaissance fait l'objet de recherches en cours en biomécanique. Par contre, pour ce qui est du bilan ergonomique, l'information existe, mais l'élaboration de l'outil impose la mise en place d'un cadre formel de coopération pour le développement de la base de connaissances.

Deux types différents d'outils sont donc envisageables à partir de ce travail :

- un outil d'assistance au diagnostic ergonomique, à développer en coopération avec les utilisateurs,
- un outil d'aide à la recherche, permettant d'orienter les investigations à propos des stratégies de mouvement et d'en tester les résultats en les intégrant au modèle de l'activité, par des aller-retour entre implémentation et recherche de connaissance.

Du point de vue informatique, deux aspects peuvent être soulignés :

- nous avons développé une approche basée sur les techniques d'Intelligence Artificielle (modélisation objet et architecture à base de tableau noir), sur un problème nouveau (celui de la simulation de l'activité motrice),
- nous proposons un enrichissement du modèle de tableau noir, avec la définition d'un tableau noir "objet" intégré dans une base de données.

Bibliographie

1 **J. Maline**, *Simuler le travail - Une aide à la conduite de projet*, Anact, 1994.

2 **K.R. Laughery**, *Micro Saint : a Tool for Modeling Human Performance in Systems*, in Applications of Human Performance Models to System Design, edited by G.R McMillan, D. Beevis, E. Salas, M.H. Strub, R. Sutton and L. Van Breda, Plenum Press, New York, 1989, pp. 219-230.

3 **G.R McMillan, D. Beevis, E. Salas, M.H. Strub, R. Sutton and L. Van Breda** (Eds), *Applications of Human Performance Models to System Design*, Plenum Press, New York, 1989.

4 **B. Peacock, W. Karwowski** (Eds) *Automotive Ergonomics*, Taylor & Francis, 1993.

5 **K.H.E. Kroemer**, *Modeling the human-equipment interface*, in Automotive Ergonomics, edited by B. Peacock and W. Karwowski, Taylor & Francis, 1993, pp. 1-9.

6 **W.B. Rouse, W.J. Cody**, *Designers' Criteria for choosing Human Performance Models*, in Applications of Human Performance Models to System Design, edited by G.R McMillan, D. Beevis, E. Salas, M.H. Strub, R. Sutton and L. Van Breda, Plenum Press, New York, 1989, pp. 7-14.

7 **J.P. Verriest**, *MAN3D - Mannequin graphique en 3 dimensions - Notice d'utilisation*, INRETS, Septembre 1993.

8 **J.P. Verriest**, *The Simulation of Human Movement for the Ergonomic Design of Numerical Mock-up*, Proc. 77th Annual Meeting of the Transportation Research Board, Washington D.C., Preprint CD-ROM, TRB and Mira CD-ROM Publishing, January 1998.

9 **F.X. Lepoutre**, *Human Posture Modelization as a problem of Inverse Kinematics of Redundant Robots*, Robotics, 11, 1993, pp. 339-343.

10 **C.B. Phillips**, *Interactive Postural Control Of Articulated Geometric Figures*, Technical Report (PhD thesis) MS-CIS-91-82, Department of Computer Information Science, University of Pennsylvania, 1991.

11 **N.I. Badler, C.B. Phillips, B.L. Webber**, *Simulating humans - Computer Graphics Animation and Control*, Oxford University Press, Oxford, 1993.

12 **J.M. Porter, K. Case, M.T. Freer, M.C. Bonney**, *Computer-aided ergonomics design of automobiles*, in Automotive Ergonomics, edited by B. Peacock and W. Karwowski (Taylor & Francis), 1993, pp. 43-77.

13 **R. Gilbert, R. Carrier, J. Schiettekatte, C. Fortin, B. Dechamplain, H.N. Cheng, A. Savard, C. Benoit, M. Lachapelle**, *Safework : Software to Analyse and Design*

Workplaces, in Applications of Human Performance Models to System Design, edited by G.R McMillan, D. Beevis, E. Salas, M.H. Strub, R. Sutton and L. Van Breda, Plenum Press, New York, 1989, pp. 389-396.

14 **J. Järvinen, W. Karwowski**, *Applications of knowledge-based expert systems in industrial ergonomics : a review and appraisal*, in Computer applications in ergonomics, occupational safety and health (CAES'92), M. Mattila, W. Karwowski, (Eds), Elsevier science publishers, 1992, pp. 45-54.

15 **L. Brennan, L. Farrell, D. McGlennon**, *ERGOSPEC : a prototype expert system for workstation design*, in Computer-aided ergonomics, edited by W. Karwowski, A.M. Genaidy and S.S. Asfour (Taylor & Francis), 1990, pp. 117-127.

16 **V. Rombach, W. Laurig**, *Ergon-Expert - a knowledge-based approach to the design of workplaces* , in Trends in Ergonomics/human Factors V, edited by F. Aghazadeh, Amsterdam, North-Holland, 1988, pp. 53-61.

17 **E. Jung, A. Freivalds**, *Development of an expert system for designing workplaces in manual materials handling jobs*, in Computer-aided ergonomics, edited by W. Karwowski, A.M. Genaidy and S.S. Asfour (Taylor & Francis), 1990, pp. 279- 298.

18 **J. McDaniel**, *Models for ergonomic analysis and design : COMBIMAN and CREW CHIEF*, in Computer-aided ergonomics, edited by W. Karwowski, A.M. Genaidy and S.S. Asfour (Taylor & Francis), 1990, pp. 138-156.

19 **K. Case, J.M. Porter, M.C. Bonney**, *SAMMIE : a man and workplace modelling system*, in Computer-aided ergonomics, edited by W. Karwowski, A.M. Genaidy and S.S. Asfour (Taylor & Francis), 1990, pp. 31-56.

20 **N.J. Nilsson**, *Principes d'intelligence artificielle*, Cepadues, Toulouse, 1988.

21 **J.P. Haton, N. Bouzid, F. Charpillet, M.C. Haton, B. Lâasri, H. Lâasri, P. Marquis, T. Mondot, A. Napoli**, *Le raisonnement en intelligence artificielle - Modèles, techniques et architectures pour les systèmes à bases de connaissances*, InterEditions, Paris, 1991.

22 **C. Hewitt**, *Viewing Control Structures as Patterns of Passing Messages*, in Artificial Intelligence, vol. 8 n°3, 1977, pp. 323-364.

23 **V. Lesser, R. Fennel, L. Erman, D. Reddy**, *Organization of the Hearsay-II Speech Understanding System*, IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-23, 1975, pp. 11-24.

24 **J. Erceau, J. Ferber**, *L'intelligence artificielle distribuée*, La Recherche n°233 vol. 22, Juin 1991, pp. 750-758.

25 **J. Ferber, M. Ghallab**, *Problématique des univers multi-agents intelligents*, Actes des 2èmes journées du GRECO-PRC IA, Toulouse, 1988, pp. 295-320.

26 **A. Collinot, A. Drogoul**, *Intelligence Artificielle Distribuée et Systèmes Multi-Agents*, Dossier du Bulletin de l'AFIA n°28, Janvier 1997, pp. 23-54.

27 **J. Ferber**, *Les systèmes multi-agents - Vers une intelligence collective*, InterEditions, 1995.

28 **J.P. Haton**, *Panorama des systèmes multi-agents*, Actes des 11èmes journées francophones sur l'informatique : architectures avancées pour l'Intelligence Artificielle, EC2 Editeur, Nancy, 1989.

29 **N. Carver, V. Lesser**, *Evolution of Blackboard Control Architecture*, Expert Sytems with Applications, Vol. 7, pp. 1-30, 1994.

30 **B. Hayes-Roth**, *A Blackboard Architecture for Control*, in Artificial Intelligence, vol. 26, 1985, pp. 251-321.

31 **H. Lâasri, B. Maître, T. Mondot, F. Charpillet, J.P. Haton**, *ATOME : A Blackboard Architecture with Temporal and Hypothetical Reasoning*, Proc. 8th ECAI, Munich, 1988, pp. 1-5.

32 **H. Lâasri, B. Maître**, *Coopération dans un univers multi-agent basée sur le modèle du blackboard : Etudes et réalisations*, Thèse de doctorat, Université de Nancy 1, Février 1989.

33 **V. Chevrier**, *Étude et mise en oeuvre du paradigme multi-agents : De Atome à Gtmas*, Doctorat de l'Université de Nancy 1, Juin 1993.

34 **F. Le Ber**, *Modélisation des connaissances et raisonnements pour l'analyse des paysages agraires à partir de données satellitaires*, Doctorat de l'Université de Nancy 1, Décembre 1993.

35 **C. Ferraris**, *Acquisition des connaissances et raisonnements dans un univers multi-agents : application à la prise de décision en génie civil urbain*, Doctorat de l'Université de Nancy 1, Février 1992.

36 **A. Belaïd, Y. Chenevoy, H. Lâasri et B. Maître**, *Vers un système à bases de connaissances pour la compréhension de l'écriture*, Actes du 7ème congrès AFCET-RFIA, Paris, 1989.

37 **C. Ferraris, M.C. Haton**, *Assisting the Specialist for Decision Making about the Road Maintenance : A Multi-expert Approach*, Actes de la 2ème Conférence Européenne sur les Applications de l'IA et de la Robotique en Architecture et en Génie Civil, EuropIA 90, Liège, 1990, pp. 244-251.

38 **S. Brunessaux, V. Chevrier et J.P. Haton**, *A Blackboard-based Expert System for Noise Measurement Planning*, Proc. IASTED International Conference on Artificial Intelligence Applications and Neural Networks, Zurich, Switzerland, 1990.

39 **F. Desard, M. Collet, S. Bertuzzi, T. Joubert, J.C. Le Mentec, P. Lalanda, F. Charpillet et J.P. Haton**, *Etude d'un système d'aide au pilote pour l'autoprotection d'un avion de combat par des techniques multi-bases de connaissances*, Actes du 10ème congrès d'Avignon, Conférence IA et Défense, 1990, pp. 15-31.

40 **N. Mathé**, *Design of a Space Telerobotics Assistant System*, Actes du 10ème Congrès d'Avignon, Conférence générale Outils, Techniques et Applications, 1990, pp. 471-485.

41 **K. Smaïli**, *Conception et réalisation d'une machine à dicter à entrée vocale destinée aux grands vocabulaires. Le système MAUD*, Doctorat de l'Université de Nancy 1, 1991.

42 **P. Vaxivière**, *CELESSTIN : A System for Conversion of Mechanical Engineering Drawings into CAD Format*, IEEE Computer Magazine, July 1992.

43 **A.-C. Boury-Brisset**, *Maintien de la vérité dans un système à architecture de blackboard*, Doctorat de l'Université de Nancy 1, Février 1992.

44 **N. Doggaz, J.P. Haton**, *L'incertitude et l'imprécision dans les systèmes à multi-sources de connaissances : le système ATOMIC*, Proc. of the 2nd Maghrebine Conference on Software Engineering and Artificial Intelligence, Tunis, 1992, pp. 132-141.

45 **J.C. Le Mentec, S. Brunessaux**, *ATOME-TR : A real-time control for blackboard scheduling*, Actes des 12èmes Journées Internationales sur les Systèmes Experts et leurs Applications, Avignon, Juin 1992.

46 **F. Warin**, *ATOME - Version C++ - Dictionnaire des fonctions et variables*, Rapport Technique CRIN/INRIA Lorraine, 1992.

47 **F. Warin** *ATOME - Version C++ - Manuel Utilisateur*, Rapport Technique CRIN/INRIA Lorraine, 1992.

48 **J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen**, *Object-oriented Modelling and Design*, Prentice-Hall International Editions, 1991.

49 **A. Alauzet**, *Motor Behavior Simulation for Ergonomic Evaluation in Product Design Context : Use of Artificial Intelligence Techniques*, Proccedings of CESA'96 (Computational Engineering in Systems Applications), Symposium on Robotics and Cybernetics, ed. Gerf EC Lille, 1996, pp. 815-819.

50 **A. Alauzet**, *Une approche multi-agents pour la simulation de l'activité motrice d'un opérateur humain*, Actes des 3èmes Rencontres des Jeunes Chercheurs en Intelligence Artificielle, T.Schaub, L. Siklossy, P. Lamarre (Eds), 1996, p. 254.

51 **A. Alauzet**, *Computer Aided Ergonomic Design : Use of Artificial Intelligence Techniques*, EXPERSYS-96, Expert systems applications and artificial intelligence, IITT International, 1996, pp. 153-158.

52 **J.M. Winters, S.L-Y. Woo** (Eds) *Multiple Muscle Systems*, Springer-Verlag, New-York, 1990.

53 **P. Morasso, V. Tagliasco** (Eds) *Human Movement Understanding*, Elsevier Science Publishers B.V. (North-Holland), 1986.

54 **J.P. Verriest, M. A. Rezgui, X. Wang**, *Experimental Validation of Arm Reach Movement simulation*, Proc. 12th Congress of Int. Ergonomics Ass., IEA'94, Toronto, Vol. 2, 1994, pp. 342-344.

55 **M. A. Rezgui**, *Contribution à la modélisation du geste d'atteinte du membre supérieur chez l'homme*, Doctorat de l'Ecole Centrale de Lyon, Janvier 1994.

56 **N. Brunel**, *Recherche d'une méthode de simulation du mouvement volontaire de l'opérateur humain,* Rapport d'activités à la Région Rhône-Alpes, Octobre 1993 - Mai 1996, INRETS, 1996.

57 **X. Wang, J.P. Verriest**, *A Geometric Algorithm to Predict the Arm Reach Posture for Computer-aided Ergonomic Evaluation*, The Journal of Visualization and Computer Animation, Vol. 9, 1998 (à paraître).

58 **R. Santos**, *Capacités fonctionnelles motrices - Mouvements combinés du membre supérieur et du tronc*, dans Rapport d'activités LESCO 1996, INRETS, 1997.

59 **S. Ribouchon**, *Confort postural du conducteur assis : étude des situations de confort optimal et des situations observées sous contrainte de hauteur d'assise*, Doctorat de l'Université Paris XI Orsay, Décembre 1991.

60 **B. Astier**, *Communication entre un système expert d'aide à la conception et MAN3D, mannequin graphique 3D,* Rapport LESCO n°9414, INRETS, 1994.

61 **C. Carrez**, *Développement d'un système de tableau noir sur un SGBD objet*, Rapport LESCO n°9604, INRETS, 1996.

62 **M. Ayel, M.-C. Rousset**, *La cohérence dans les bases de connaissances*, Cepadues Editions, Toulouse, 1990.

63 **H. Tattegrain**, *Description de quelques méthodes d'analyse et de conception orientées objets,* Rapport MAIA, INRETS, Mars 1994.

ANNEXES

Annexe 1 : Modèles graphiques d'opérateur humain

USA :

ATB (Articulated Total Body model)
[National Highway Traffic Safety Administration (NHTSA), puis US Air Force's Armstrong Aerospace Medical Research Laboratory (AAMRL)]

BOEMAN (modèle d'opérateur assis pour l'évaluation et la conception des cockpits)
[Boeing Corporation, Seattle (Washington)]

BUFORD (modèle de représentation et positionnement d'un opérateur)
[Rockwell International, Downey (Californie)]

CAPE (Computerized Accomodated Percentage Evaluation Model, modèle pour l'évaluation de la conception des cockpits d'avion) et **CAR** (Crewstation Assessment of Reach model)
[Naval Air Developpement Center]

COMBIMAN (COMputerized BIOmedical MAN model, modèle d'un opérateur assis) et CREW CHIEF (modèle d'un technicien de maintenance aérienne)
[Armstrong Aerospace Medical Research Laboratory (AAMRL)]

CYBERMAN (CYBERnetic MAN model, modélisation des conducteurs et passagers ainsi que leurs activités dans et autour de la voiture)
[Chrysler Corporation, Detroit (Michigan)]

JACK (système d'aide à la conception ergonomique)
[Université de Pennsylvanie]

MANNEQUIN
[Biomedical Corporation of America]

PLAID et **TEMPUS** (conception et évaluation des interfaces homme/machine pour la navette spatiale)
[National Aeronautics and Space Administration (NASA)]

Allemagne :

ANYBODY et **ANTHROPOS**
[Inter Somacad Team (IST), Gernsheim]

FRANKY
[GIT, Essen]

HEINER (modèle anthropométrique)
[Institute of Ergonomics, Université de Technologie de Darmstadt]

WERNER
[Institute of Occupationnal Health, Université de Dortmund]

Finlande :

ErgoSHAPE et **ErgoSPACE**
[Institute of Occupationnal Health]

MINTAC
[Kuopio Regional Institute of Occupationnal Health et Université de Oulu]

France :

ERGOMAN (application basée sur un modèle humain, interfacée avec EUCLID, pour la conception des espaces de travail)
[Laboratoire d'Anthropologie Appliquée, Université René Descartes]

MAN3D (modèle humain 3D interfacé avec EUCLID]
[INRETS-LESCO]

Angleterre :

SAMMIE (System for Aiding Man-Machine Interaction Evaluation, modèle général pour des analyses anthropométriques)
[Université Technologique de Loughborough]

Canada :

SAFEWORK
[Ecole Polytechnique de Montréal, Department of Industrial Engineering]

Autres pays :

APOLIN (man-model based on the APproximation by Ovals and LINes)
[Institute of Organisation and Management, Technical University of Wroclaw, Pologne]

TADAPS
[University of Twente, Pays-Bas]

Annexe 2 : Notions sur la méthode OMT

(d'après H. Tattegrain (63))

OMT (Object Modeling Technique, (48)) est une méthode de conception orientée objet. Elle permet de réaliser des modèles pour les trois axes de la conception : statique (structure du système), dynamique (comportement temporel du système et de ses objets) et fonctionnel (description des dépendances entre les valeurs des objets).

Les produits de la méthode sont :
- le modèle d'objets, contenant des diagrammes de classes et des diagrammes d'instances,
- le modèle dynamique, composé de diagrammes d'états, du diagramme global des flots d'événements et de traces d'événements pour les scénarios,
- le modèle fonctionnel, composé d'un ensemble de diagrammes de flots de données.

Nous nous limitons ici à la description du modèle d'objets, en fournissant une brève description des concepts et de la méthode. Les deux notions de base sont :
- les **classes**, qui sont les moules des objets, avec leurs réalisations, les instances,
- les **relations entre classes**.

Chaque classe est décrite par ses attributs et ses opérations (qui permettent d'accéder aux valeurs des attributs ou d'obtenir des valeurs calculées sur la base des valeurs des attributs). Une classe peut être *dérivée* (classe redondante car pouvant être complètement déduite à partir d'un ensemble d'autres classes, mais aidant à la compréhension du modèle). Les attributs permettent de caractériser les instances. Un *attribut de classe* est un attribut très particulier puisqu'il est partagé par toutes les instances d'une classe (par exemple, un attribut de classe peut être le nombre d'instances de la classe, mis a jour à chaque création ou destruction d'instance). De même, une *opération de classe* est une opération spéciale, envoyée à la classe et non à une de ses instances (par exemple, créer est une opération de classe et détruire est une opération d'instance). Un *attribut dérivé* est un attribut pouvant être déduit d'autres attributs.
Une *opération abstraite* correspond à une déclaration de l'opération dans une super-classe sans implémentation, celle-ci devant être réalisée dans chaque sous-classe (voir plus loin pour les notions de superclasse et de sous-classe). La figure A1 présente les principales notations pour les classes et les instances.

Les relations entre classes peuvent être de différentes natures. Nous évoquons ici uniquement (voir figure A2 pour les notations):
- les *liens de généralisation*, qui correspondent au concept d'héritage. Les sous-classes héritent des attributs et des opérations de la super-classe.
- les *liens d'agrégation* ou de composition. Une agrégation sera posée s'il existe des opérations du composé s'appliquant automatiquement sur le composant (par exemple imprimer un texte implique imprimer ses paragraphes), s'il existe des valeurs d'attributs propagées du composé aux composants (par exemple style du texte).
- les *associations* entre classes. L'association de base (binaire) relie deux classes et comporte deux indications de rôle. Un rôle est placé à l'extrémité du trait vers la classe où il définit une instance (ou un ensemble d'instances).

Une notation permet de plus d'indiquer la cardinalité des associations et des agrégations.

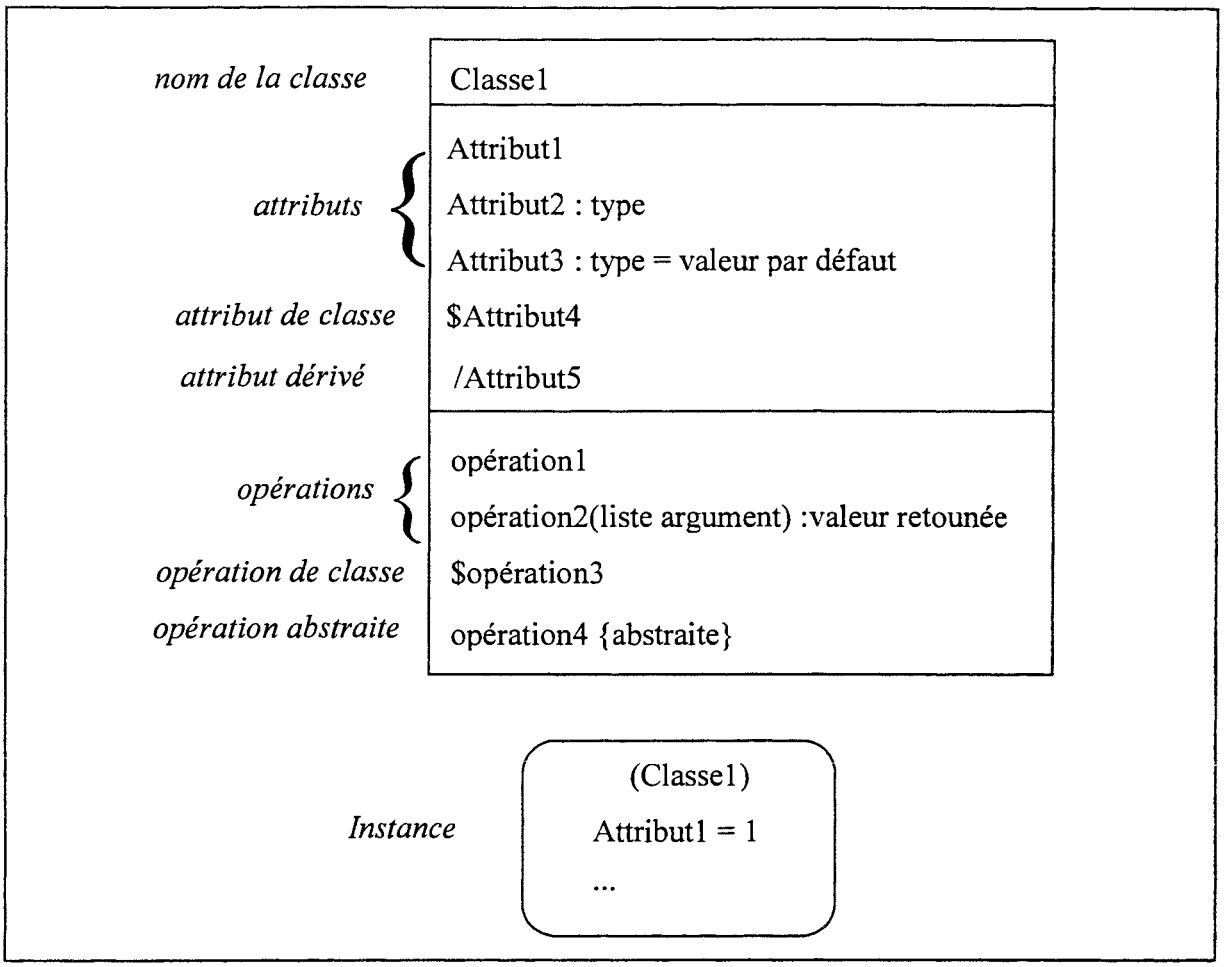

Figure A1 : principales notations pour les classes et leurs instances

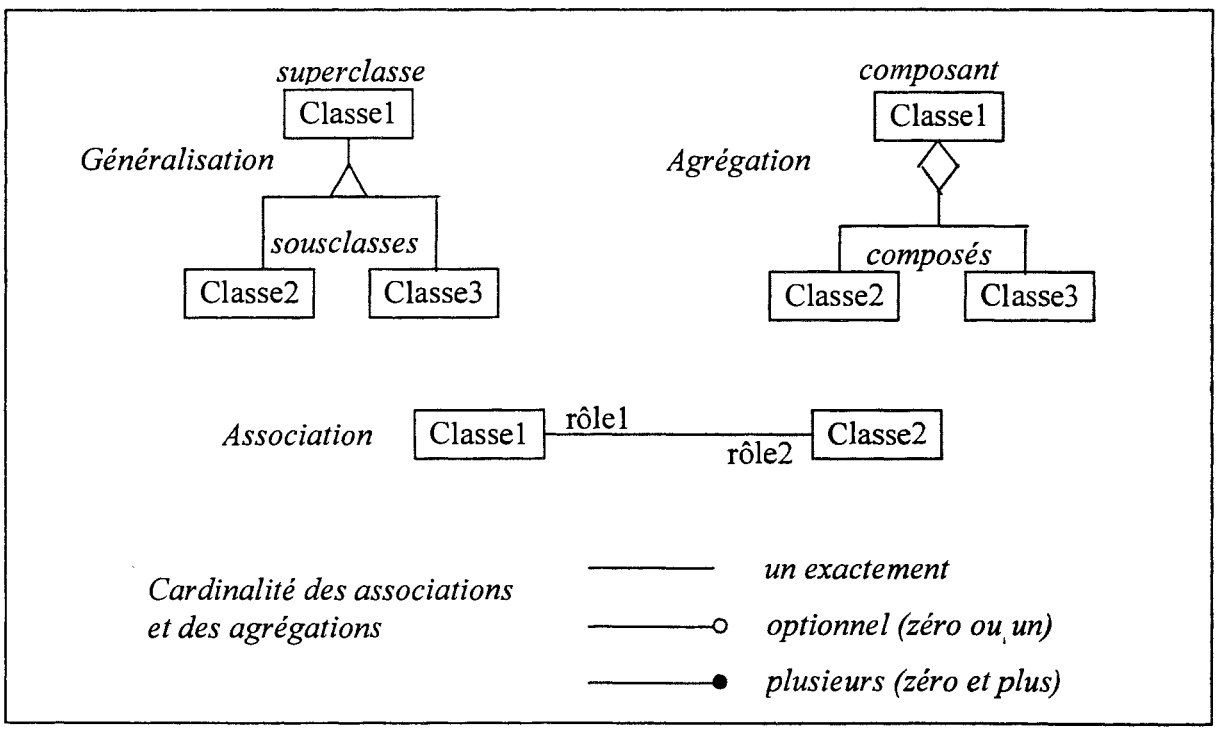

Figure A2 : principales notations pour les relations entre classes

Annexe 3 : Commandes MAN3D

Mnémonique de la Commande	Numéro du format associé	Fonction réalisée par la commande
FINI	3	Sortie du mode commande
LSCO	3	**LiSte** les **COmmandes** disponibles sur écran alpha
FCOM	6	Lecture d'un **F**ichier de **COM**mandes
TEMP	7	**TEMP**orisation entre deux commandes
		Gestion d'affichage
VISU	2	**VISU**alise le sujet courant (OUI/NON)
RAPP	3	**RAPP**elle la posture précédente
SQUE	3	Affichage **SQUE**lette (bascule)
PREM	3	Affichage Points **REM**arquables (bascule)
CFIL	3	Affichage Contour **FIL**aire du mannequin
CVOL	3	Affichage Contour **VOL**umique du mannequin
CSUR	3	Affichage Contour **SUR**facique du mannequin
SCTR	3	Affichage Sans ConTouR

Mnémonique de la Commande	Numéro du format associé	Fonction réalisée par la commande
CSUJ	6	Choix SUJet
		Posture du Sujet
LPOS	6	Lecture d'un fichier de POSture
SPOS	8	Sauvegarde d'une POSture
EPOS	3	Edite POSture
		Définition du Sujet
LDIM	6	Lecture d'un fichier DIMensions
LCTR	6	Lecture d'un fichier ConTouR
LPAR	6	Lecture d'un fichier de PARtition d'un modèle
		Positionnement du Sujet
DSUJ	4	Déplace SUJet
RSUX	1	Rotation du SUjet suivant X
RSUY	1	Rotation du SUjet suivant Y
RSUZ	1	Rotation du SUjet suivant Z
		Modifier Angle
MODA	5	MODifie la valeur d'un Angle d'une articulation

Mnémonique de la Commande	Numéro du format associé	Fonction réalisée par la commande
		Mouvements du Bras
EXTB	4	**EXT**ension Bras
ATLB	4	**AT**teinte Libre du Bras
ATOB	10	**AT**teinte Orientée du Bras
ORIB	4	**ORI**entation du Bras
PVAB	11	**PiV**otement avec un **A**ngle (**B**ras)
PVPB	4	**PiV**otement avec **P**oint objectif (**B**ras)
TRJB	14	Suivi **TRaJ**ectoire Bras (pivotement nul)
TRCB	14	Suivi **TRajeC**toire Bras (pivotement Calculé)
		Mouvements de la jambe
EXTJ	4	**EXT**ension Jambe
ATLJ	4	**AT**teinte Libre de la **J**ambe
ATOJ	10	**AT**teinte **O**reintée de la **J**ambe
ORIJ	4	**ORI**entation de la **J**ambe
PVAJ	11	**PiV**otement avec un **A**ngle (**J**ambe)
PVPJ	4	**PiV**otement avec un **P**oint objectif (**J**ambe)
TRJJ	14	Suivi **TRaJ**ectoire **J**ambe (pivotement nul)

Mnémonique de la Commande	Numéro du format associé	Fonction réalisée par la commande
		Mouvements du Torse
FLTO	4	**FL**exion sagittale du **TO**rse
ILTO	4	Inclinaison **L**atérale du **TO**rse
TRTO	4	**To**R**sion du TO**rse
FGTO	4	**F**lexion **G**énérale du **TO**rse
		Mouvement Tête/Cou
ILTC	4	Inclinaison **L**atérale **Tê**te/Cou
TRTC	4	**To**R**sion Tê**te/Cou
FGTC	4	**F**lexion **G**énérale **Tê**te/Cou
OCHA	15	**O**riente un **CHA**înon
DREG	12	**D**irection du **REG**ard
MDGT	4	**M**ouvement d'un **DoiGT**
FLBC	4	**FL**exion **B**assin - **C**uisse
BUCO	4	**BU**ste **CO**nducteur

Mnémonique de la Commande	Numéro du format associé	Fonction réalisée par la commande
		Gestion des Axes
SLAX	3	**SéLection des AXes**
SVAX	3	**SauVegarde des AXes**
		Spécificités EUCLID
DECO	13	**DECO**de une entité EUCLID
STOK	9	**STOcKe** le sujet courant en SD EUCLID

Annexe 4 : Extrait de la base de données BaseAdele, développée sous GemStone (les objets Action, ActionElementaire et ActionComposée)

```
!
! Class 'Action'
!

Object subclass: 'Action'
  instVarNames: #( 'nom' 'type' 'frequence'
                   'duree' 'composant' 'conditionEtatOperateur' 'conditionsEtatsParties'
                   'conditionsContactOE')
  classVars: #()
  classInstVars: #()
  poolDictionaries: #[]
  inDictionary: BaseAdele
  constraints: #[ #[ #nom, String],
                  #[ #type, String],
                  #[ #frequence, String],
                  #[ #duree, String],
                  #[ #composant, Composant],
                  #[ #conditionEtatOperateur, EtatOperateur],
                  #[ #conditionsEtatsParties, TabEtatPartie],
                  #[ #conditionsContactOE, TabContactOperateurEnv] ]
  instancesInvariant: false
  isModifiable: false
%
!
! Instance Category 'Updating'
!
category: 'Updating'
method: Action
nom: newValue

   "Modify the value of the instance variable 'nom'."
   nom := newValue
%
method: Action
composant: newValue

   "Modify the value of the instance variable 'composant'."
   composant := newValue
%
method: Action
conditionsContactOE: newValue

   "Modify the value of the instance variable 'conditionsContactOE'."
   conditionsContactOE := newValue
%
method: Action
type: newValue

   "Modify the value of the instance variable 'type'."
   type := newValue
%
method: Action
init

   "Modify the value of the instance variables"
   composant := Composant new.
   composant init.
   nom := String new.
   type := String new.
   conditionEtatOperateur := EtatOperateur new.
   conditionEtatOperateur init.
   conditionsContactOE := TabContactOperateurEnv new.


%
method: Action
conditionEtatOperateur: newEtatOperateur

   "Modify the value of the instance variable 'conditionEtatOperateur'."
```

```
   conditionEtatOperateur := newEtatOperateur.

%
!
! Instance Category 'Comparing'
!
category: 'Comparing'
method: Action
<= anAction

^(self nom <= anAction nom).

%
!
! Instance Category 'Accessing'
!
category: 'Accessing'
method: Action
nom

   "Return the value of the instance variable 'nom'."
   ^nom
%
method: Action
composant

   "Return the value of the instance variable 'composant'."
   ^composant
%
method: Action
type

   "Return the value of the instance variable 'nom'."
   ^type.

%
method: Action
conditionsContactOE

   "Return the value of the instance variable 'conditionsContactOE'."
   ^conditionsContactOE.

%
method: Action
displayString

^nom + '(' + composant + ')'.
%
method: Action
conditionEtatOperateur

   "Return the value of the instance variable 'conditionsEtatOperateur'."
   ^conditionEtatOperateur.

%
method: Action
printString

^nom + '(' +(composant nom)+ ')'.
%
method: Action
asString

^nom + '(' + (composant nom) + ')'.
%
```

```
!
! Class 'ActionElementaire'
!

Action subclass: 'ActionElementaire'
  instVarNames: #( 'membre' 'nature' 'contact')
  classVars: #()
  classInstVars: #()
  poolDictionaries: #[]
  inDictionary: BaseAdele
  constraints: #[ #[ #membre, Partie],
                  #[ #nature, String],
                  #[ #contact, ContactOperateurEnvironnement] ]
  instancesInvariant: false
  isModifiable: false
%
!
! Instance Category 'Accessing'
!
category: 'Accessing'
method: ActionElementaire
nature

^nature.
%
method: ActionElementaire
membre

^membre.

%
method: ActionElementaire
contact

^contact.
%
method: ActionElementaire
displayString
^nom+'(' + (composant nom) + ')'.
%
method: ActionElementaire
actionsElem

"Retourne l'action comme element unique d'un tableau d'actions.
 (besoin = existence d'une methode actionsElem quel que soit le
  type de l'action (composee ou elementaire))"

|T|
T := TabAction new.
T add: self.
^T.
%
!
! Instance Category 'Updating'
!
category: 'Updating'
method: ActionElementaire
membre: aString

membre := aString.
%
method: ActionElementaire
nature: aString

nature := aString.
%
method: ActionElementaire
init
```

```
   "Modify the value of the instance variables."

 super init.
 membre := Partie new.
 membre init.
 nature := String new.
 contact := ContactOperateurEnvironnement new.
 contact init.
%
```

```
!
! Class 'ActionComposee'
!

Action subclass: 'ActionComposee'
  instVarNames: #( 'sousActions')
  classVars: #()
  classInstVars: #()
  poolDictionaries: #[]
  inDictionary: BaseAdele
  constraints: #[ #[ #sousActions, TabAction] ]
  instancesInvariant: false
  isModifiable: false
%
!
! Instance Category 'Updating'
!
category: 'Updating'
method: ActionComposee
init

   "Modify the value of the instance variables"
 super init.
 sousActions := TabAction new.
%
method: ActionComposee
sousActions: newTabAction

   "Modify the value of the instance variable 'actions'."
   sousActions := newTabAction.

%
!
! Instance Category 'Accessing'
!
category: 'Accessing'
method: ActionComposee
displayString
^nom+'(' + (composant nom) + ')'.
%
method: ActionComposee
actionsElem

| T Tp TAE tailleT tailleTp |

TAE := TabAction new.
T := TabAction new.
T := self sousActions.
tailleT := T size.
1 to: tailleT do: [ :each |
         ( ((T at: each) type) = 'Action composee')
            ifFalse: [ TAE add: (T at: each)]
                ifTrue: [ Tp := TabAction new.
                Tp := (T at: each) actionsElem.
                tailleTp := Tp size.
                1 to: tailleTp do: [:chq | TAE add: (Tp at: chq).] ]
                ].
^TAE.
%
method: ActionComposee
sousActions

^ sousActions.
%
```

université HENRI POINCARÉ
NANCY 1
FACULTÉ DES SCIENCES

Mademoiselle **ALAUZET Aline**

DOCTORAT de l'UNIVERSITE HENRI POINCARE, NANCY-I

en INFORMATIQUE

VU, APPROUVÉ ET PERMIS D'IMPRIMER

Nancy, le 27 MAI 1998 n° 24

Le Président de l'Université

J.P. FINANCE

3C-F9

ADELE : Une architecture de tableau noir pour la simulation ergonomique

Résumé

L'évaluation ergonomique de maquettes de produits (habitacles de voitures, poids-lourds, ...) réalisées sous logiciel de CAO nécessite des outils de simulation permettant de tester l'adéquation du produit pour ses futurs utilisateurs. Les mannequins graphiques 3D sont les outils de simulation utilisés pour effectuer des tests d'encombrement, d'atteinte ou de vision. Bien que ces outils remplissent bien leur fonction d'aide, leur manipulation nécessite une véritable expertise. L'utilisateur (concepteur de bureau d'étude) doit aussi posséder l'expertise lui permettant de décider, au vu de la position obtenue avec le mannequin, si l'implantation choisie pour les constituants du produit est bonne. Sur la base de l'analyse des outils d'aide à la simulation ergonomique existants, nous proposons une nouvelle approche permettant de remédier à ces défauts en automatisant la manipulation du mannequin et en fournissant des aides à l'évaluation ergonomique. Cette approche est basée sur la constitution d'un modèle de l'activité motrice et l'élaboration de bases de connaissances sur les stratégies de mouvement et sur le diagnostic ergonomique. Le modèle d'activité, conçu à l'aide d'une modélisation objet, constitue un outil de spécification des tâches que le mannequin doit effectuer. L'architecture conçue, basée sur un modèle de tableau noir, nous permet la modélisation des aspects de contrôle liés au pilotage du mannequin graphique et à la production d'un diagnostic ergonomique. Le prototype du système, nommé ADELE, associe trois éléments : le modèle graphique d'opérateur humain MAN3D (développé à l'INRETS/LESCO), une base de données objet (développée avec le SGBD GemStone) et un système à base de tableau noir (développé avec l'outil Atome, du LORIA à Nancy). La liaison entre Atome et le SGBD utilisé a fait l'objet d'un développement particulier, permettant l'implantation du tableau noir dans la base de donnée objet.

ADELE: A Blackboard-based Architecture for Ergonomic Simulation

Abstract

The ergonomic evaluation of computer assisted design prototypes of systems like cars or trucks requires simulation tools, in order to assess the convenience of the product for its future users. 3D graphic mannequins are such simulation tools, which provides facilities for fit, vision and reach tests. These tools prove to be very useful, however the use of the tools is in fact not easy at all due to the large amount of expertise needed to manipulate them. Furthermore, the user must possess a great amount of expert knowledge to decide, on the basis of the mannequin posture, whether he has chosen a good implantation for the elements of the product, or not. We analyse the existing ergonomic simulation tools, then we propose a new approach, based on the automation of the mannequin handling and the add of ergonomic assessment tools. We propose a motor activity model (which provides a tool for the specification of the tasks to be simulated by the mannequin) and knowledge bases (for movement strategies and ergonomic diagnosis). The system is based on a blackboard architecture which helps to model the control aspects (mannequin handling and production of an ergonomic diagnosis). The prototype of our system, ADELE, has three components : a graphical human model, MAN3D (developed at INRETS/LESCO), an object database (developed with GemStone DBMS) and a blackboard system (developed with Atome, a LORIA-Nancy tool). A specific link between Atome and GemStone has been developed, which allows to implement a blackboard located in the object database.

Discipline

Informatique (Intelligence Artificielle)

Mots-clés

Intelligence artificielle, Architecture de tableau noir, Modélisation objet, Modélisation de l'activité, Simulation de l'activité motrice, CAO, Modèle humain 3D, Diagnostic ergonomique.

Laboratoire

LESCO (Laboratoire Ergonomie Santé Confort), 25 av. F. Mitterrand, Case 24, 69675 Bron cedex, France.