Par Margaux LHOMMET

REPLICANTS : Humains virtuels cognitifs, émotionnels et sociaux.
De l'empathie cognitive à l'empathie affective.

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC

Soutenue le 9 novembre 2012
Spécialité : Technologie de l'Information et des Systèmes

REPLICANTS : Humains virtuels cognitifs, émotionnels et sociaux.

De l'empathie cognitive à l'empathie affective.

Thèse soutenue le 9 novembre 2012 devant le jury composé de :

J.P. Barthès	Professeur Émérite Laboratoire Heudiasyc, UTC ***Co-directeur de thèse***
D. Bazalgette	Direction de la Stratégie Direction Générale pour l'Armement, Bagneux
D. Lourdeaux	Maitre de Conférences HDR Laboratoire Heudiasyc, UTC ***Co-directeur de thèse***
P. Morizet-Mahoudeaux	Professeur Laboratoire Heudiasyc, UTC ***Président***
C. Pelachaud	Directrice de Recherche LTCI Télécom ParisTech, Paris ***Rapporteur***
S. Pesty	Professeur IMAG, Laboratoire LIG, Grenoble
N. Sabouret	Professeur LIMSI-CNRS, Orsay ***Rapporteur***

Thèse réalisée au Laboratoire Heudiasyc
UMR CNRS 7253
Université de Technologie de Compiègne

Sous la direction de Domitile Lourdeaux domitile.lourdeaux@hds.utc.fr
Jean-Paul Barthès jean-paul.barthes@hds.utc.fr

Financement Mission pour la Recherche et l'Innovation Scientifique
Direction Générale pour l'Armement
Contrat doctoral
Octobre 2009 à Octobre 2012

Contact m.lhommet@gmail.com

Illustration de couverture par Arian Noveir ©

Résumé

Les humains virtuels sont de plus en plus présents dans les environnements virtuels. Les travaux en informatique affective visent à doter ces personnages autonomes de capacités émotionnelles.

Nous souhaitons modéliser des humains virtuels dont le comportement est cohérent, adaptable et explicable. Nous définissons la cohérence d'un humain virtuel comme l'adéquation entre sa situation, son état mental et son comportement. L'humain virtuel doit pouvoir s'adapter à de nouvelles connaissances et les intégrer dans son raisonnement. Enfin, pour que l'apprenant comprenne l'impact de ses actions, le comportement de l'humain virtuel doit être explicable.

Nous utilisons des modèles issus de la psychologie qui explicitent les dimensions humaines et leur dynamique. Notre humain virtuel dispose ainsi d'une personnalité, d'émotions et sont liés par des relations sociales.

Afin de garantir l'adaptabilité de l'humain virtuel, nous définissons un ensemble de processus indépendants du domaine permettant de gérer la dynamique de ces dimensions humaines et leur impact sur le comportement au sein d'une architecture cognitive. Un langage de description des entités, des actions et des activités permet de représenter les connaissances spécifiques du domaine d'application. Ce formalisme est suffisamment simple pour être renseigné par des individus sans connaissances en programmation et suffisamment expressif et robuste pour être interprété directement par les humains virtuels.

La prise en compte des interactions affectives entre les humains virtuels pose un problème de passage à l'échelle. Nous proposons un modèle d'empathie affective qui prend en compte les dimensions humaines et permet de résoudre ce problème.

Pour produire de tels humains virtuels, nous proposons REPLICANTS, un moteur d'intelligence artificielle décisionnelle basé sur des modèles de connaissances. Des cas d'usage montrent comment l'humain virtuel combine ses processus cognitifs de haut niveau à des connaissances du domaine afin de se comporter de manière cohérente et explicable.

Mots clés : Agent cognitif, informatique affective, empathie, représentation des connaissances.

Replicants: emotional, social and cognitive virtual humans. From cognitive to affective empathy.

Abstract

VIRTUAL HUMANS are more and more common in virtual environments such as simulations, training softwares, serious-games or video games. Affective computing aims at giving those artificial characters emotional capabilities.

We aim at generating virtual humans whose behavior is coherent, adaptative and explainable. We define coherence as the adequacy between the situation, the virtual human's mental state and her behavior. Adaptability is the capacity to adapt to new knowledge an reason about it. This knowledge may be specified by people without computer programming skills and therefore be incomplete. Finally, the virtual human's behavior must be explainable in order for the learner to understand the impact of her actions.

Using models from psychology that explicitly address the components and their dynamics, our virtual human model is given a personality, an emotional state and is linked to others via social relationships. In order to ensure the adaptability of our virtual human, she is given a set of domain-independent processes to take care of the dynamics of those human components and their impact on behavior. Those processes are integrated on a cognitive architecture. Domain-dependent knowledge such as entities, actions and activities are designed using a description language inspired by ergonomy methodology. This formalism is simple enough to be used without any computer programming skill, and expressive enough to be directly used by the high-level processes of our virtual human. An affective empathy model based on individual characteristics is proposed to model affective relations between virtual humans.

To generate such virtual humans, we propose REPLICANTS, a decisional artificial intelligence engine. Some examples are presented and show how the virtual human can combine her generic set of cognitive rules with domain specific knowledge in order to adapt to her environment as well as behaving rationally in pursuing goals.

Keywords : Cognitive agent, affective computing, empathy, knowledge representation.

Remerciements

Au moment d'écrire ces remerciements, mes premières pensées vont à Domitile Lourdeaux. Merci de m'avoir convaincue (oserai-je dire forcée ?) à te suivre dans cette superbe aventure pleine de rebondissements. J'ai particulièrement apprécié la confiance et le soutien que tu m'as accordés lorsque j'ai décidé de partir, tête baissée, vers des délires en psycho-socio, et ta capacité à me rassurer dans les moments de doute. Tu m'as supportée sans jamais t'énerver ni imposer ton point de vue et je pense que bien d'autres auraient craqué face à mon obstination ; à l'inverse, j'aurais abandonné si tu n'avais pas été là (mais aurai-je même commencé ?). Bien plus qu'une directrice, j'ai trouvé en toi une collègue et une amie et je souhaite à tout doctorant d'être encadré par quelqu'un qui a ton talent humain. Alors que je m'apprête à m'envoler vers de nouveaux horizons, je repense à tout le chemin parcouru depuis notre rencontre en IA01 il y a 6 ans ; nos longues discussions dans le train me manqueront cruellement.

Merci à Jean-Paul Barthès pour avoir co-encadré cette thèse et ne pas m'avoir tenu rigueur de ne pas programmer en LISP.

Je tiens à remercier chaleureusement Pierre Morizet-Mahoudeaux d'avoir accepté de présider mon jury de thèse ainsi que pour l'intérêt porté à mes travaux tout au long de cette thèse.

Mes excuses à Sylvie Pesty pour l'avoir forcée à retourner sur les lieux de sa soutenance de thèse quelques années plus tard. Et mes remerciements pour ses conseils d'orientation avisés.

Un grand merci à Catherine Pelachaud et à Nicolas Sabouret pour leur relecture attentive de ce mémoire de thèse, leurs critiques constructives, leurs conseils et leurs encouragements.

Enfin, cette thèse n'aurait pu voir le jour sans le support financier de la Direction Générale de l'Armement. Merci à Didier Bazalgette pour son soutien et la confiance qu'il nous a accordés pendant ces trois années.

Merci à vous qui avez constitué mon jury de thèse, pour vos questions pertinentes et votre bienveillance qui ont fait que ce moment n'était finalement pas si désagréable :)

Merci aux enseignants de l'UTC qui m'ont permis de découvrir les joies de l'enseignement ainsi qu'à tous les étudiants que j'ai vu passer ; vous n'avez pas réussi à me dégouter de l'enseignement, bien au contraire ! Un merci tout particulier aux étudiants de TX que j'ai encadrés avec un grand plaisir. Votre aide a été précieuse. Merci à l'ensemble du laboratoire Heudiasyc et en particulier à l'équipe ICI pour leur accueil.

Au cours de ma thèse, j'ai vu se former l'équipe HUMANS. La soutenance de thèse de Kahina, celle de Lydie et son départ, l'arrivée de Camille, Kevin et Vincent. La famille s'est agrandie autour de Domitile et d'un beau projet... J'encourage sans réserve tout étudiant qui lirait ces lignes à rejoindre cette équipe formidable. Vous allez vraiment beaucoup me manquer.

Merci à tous les doctorants du domaine ICI pour les rires quotidiens, les journées déguisées et les bières au Shamrock. Une pensée spéciale pour Kiwi et Ben, la fin (le début ?) est proche.

Pour entretenir l'éternel débat entre l'inné et l'acquis, je remercierai les responsables de mon patrimoine génétique.

Côté Lhommet, merci à Papa, Manou, Tata Bargeot et à mon frérot Blondinet.

Merci aux ritals du côté Bonacina pour leur présence réconfortante à ma soutenance de thèse (heureusement que j'avais prévu double ration de champ') ; et pour tout le reste :).

Bien que leur responsabilité ne soit pas engagée, merci à ma belle-famille ; Robert pour tous ces repas magiques du mercredi soir, Jiejie & Jojo pour m'avoir accueillie dans votre famille comme une sœur, et tous les Lor-Lim mais vous êtes bien trop nombreux pour être tous cités.

Merci à mes amis non doctorants qui m'ont permis de maintenir ma santé mentale à un niveau presque acceptable et en particulier à Motie pour avoir supporté mes répétitions de soutenance sans broncher ni s'endormir.

Merci à Steve Reich, Eurythmics, Tom Petty, The Rolling Stones, America, Peaches, Kelly Hogan et The Cliks pour la note musicale.

Enfin, j'espère qu'on m'accordera maintenant de sauver deux personnes si le bateau coule. Merci à Bubu et à Emilie.

À Maman et à Mamie

Table des matières

Résumé i

Abstract v

Remerciements vii

Table des matières ix

Table des figures xv

Liste des tableaux xix

Introduction 1

État de l'art : Modélisation de l'humain virtuel 7

1 L'humain virtuel 11

1.1 Personnalité 12

1.1.1 Approche topologique 12

1.1.2 Approche par traits 13

1.2 Émotions et Humeur 17

1.2.1 Théories de l'évaluation cognitive 18

1.2.1.1 Ortony, Clore et Collins 20

1.2.1.2 Smith &Lazarus 21

1.2.1.3 Scherer 24

1.3 Architectures cognitives 26

1.3.1 Définition 26

1.3.2 Principales architectures cognitives 27

1.3.2.1 Principe de fonctionnement 29

1.4 Implémentations informatiques 31
1.4.1 Le modèle de Marinier 32
1.4.2 EMA 36
1.5 Conclusion 40

2 Interactions émotionnelles entre humains virtuels 43
2.1 Empathie cognitive et théories de l'évaluation cognitive 45
2.1.1 Empathie cognitive et OCC 46
2.1.2 Empathie cognitive et théories de l'évaluation cognitive . . . 47
2.1.3 Conclusion 51
2.2 Empathie affective, contagion émotionnelle 51
2.2.1 Modèles catégoriels de contagion émotionnelle 53
2.2.2 Modèle ASCRIBE 55
2.2.3 Conclusion sur les modèles de contagion émotionnelle 59
2.3 Conclusion 60

3 Modèles d'action et d'activité 61
3.1 Modèle d'actions 61
3.1.1 Sans abstraction, un modèle ad hoc 63
3.1.2 L'utilisation de variables, verbosité contre complexité 63
3.2 Modèle d'activité 67
3.2.1 Approche cognitiviste 67
3.2.2 Approche sociocognitiviste 68
3.2.3 HAWAI-DL : Modélisation de l'activité pour les humains virtuels 70
3.3 Conclusion 74

Proposition 77

4 Connaissances 81
4.1 Intégration des connaissances dans l'humain virtuel 83
4.2 DOMAIN-DL, le modèle d'action 84
4.2.1 Principaux concepts 84
4.2.2 Exemple 84
4.2.2.1 Modèle du monde 84
4.2.2.2 Règle d'action 86
4.2.2.3 Application 86
4.3 ACTIVITY-DL, le modèle d'activité 87
4.3.1 Principaux concepts 87
4.3.1.1 Tâches 88

4.3.1.2 Conditions 90
4.3.2 Marqueurs personnalisés 91
4.4 Couplage entre DOMAIN-DL et ACTIVITY-DL 92
4.5 Arbre d'activité 93
4.6 Conclusion 93

5 Humain virtuel 97
5.1 Dimensions humaines 98
5.1.1 Personnalité 98
5.1.2 Relations sociales 98
5.2 Cycle de l'humain virtuel 99
5.2.1 Perception et construction de la situation 99
5.2.2 Évaluation Cognitive 104
5.2.3 Mise à jour des relations sociales 108
5.2.4 *Coping* centré problème 110
5.3 Conclusion 113

6 Contagion émotionnelle 115
6.1 Caractéristiques individuelles 116
6.2 Proposition d'implémentation 118
6.3 Dynamique 119

Implémentation et résultats 121

7 La plateforme HUMANS 123
7.1 Présentation de la plateforme 124
7.1.1 Architecture globale 124
7.1.1.1 HUMANS>>COMMUNICATOR, l'interface de communication 124
7.1.1.2 HUMANS>>WORLD-MANAGER 126
7.1.1.3 HUMANS>>MONITOR 127
7.1.1.4 HUMANS>>SELDON 127
7.2 HUMANS>>REPLICANTS - Les humains virtuels de HUMANS . 128
7.2.1 Architecture générale de REPLICANTS 128
7.2.2 Cycle des replicants 130
7.2.2.1 Perception 131
7.2.2.2 Évaluation cognitive 131
7.2.2.3 Contagion émotionnelle 132
7.2.2.4 *Coping* et action 132
7.2.2.5 Action 133

7.3 HUMANS-GEN 134
7.3.1 Processus général 134
7.3.2 NGS 134
7.3.2.1 Object 137
7.3.2.2 Considerable 137
7.3.2.3 Conditions 138
7.3.2.4 Goals 138
7.3.3 ACTIVITY-NGS 139
7.3.4 Conclusion 141
7.4 Conclusion 141

8 Cas d'usage et résultats 143
8.1 Contagion émotionnelle 143
8.1.1 Personnalité et relations interindividuelles 143
8.1.2 Contagion émotionnelle et caractéristiques individuelles . . . 144
8.1.3 Résultats 145
8.2 Contagion émotionnelle, évaluation cognitive et *coping* simple. . . . 146
8.2.1 Évaluation cognitive 146
8.2.1.1 Création de l'*appraisal-frame* 146
8.2.1.2 Évaluation de la désirabilité 147
8.2.2 Contagion émotionnelle 147
8.2.3 Prise de décision et coping 148
8.2.4 Résultats 149
8.3 Utilisation des arbres d'activité lors de la phase de *coping*. 150
8.3.1 Arbre d'activité 150
8.3.2 Marqueurs et préférences 152
8.3.3 Actions possibles 153
8.3.4 Exécution et résultats 154
8.3.4.1 Instant t_0 154
8.3.4.2 Instant t_1 155
8.3.4.3 Instant t_2 157
8.3.4.4 Instant t_3 157
8.3.5 Conclusion 158

Conclusion et perspectives 159

Annexes **165**

A Règles Soar génériques **167**
A.1 Évaluation Cognitive 167
A.1.1 Création des *appraisal frames* 167
A.1.2 Élaboration des dimensions d'une *appraisal frame* 167
A.1.2.1 Calcul de la désirabilité d'un évènement liée à un arbre d'activité 167
A.1.2.2 Transfert des dimensions d'appraisal 169
A.1.3 Attribution des catégories émotionnelles OCC 171
A.2 Mise à jour des relations sociales 173
A.2.1 Modifications des relations sociales dues à une émotion causée par autrui 173
A.2.2 Modifications des relations sociales dues à la perception des émotions d'autrui 175
A.3 Coping 176
A.3.1 Création d'une *coping-frame* 176
A.3.2 Évaluation des dimensions d'une *coping-frame* 177
A.3.2.1 Évaluation des marqueurs personnalisés (impact de la personnalité) 177
A.3.2.2 Évaluation de l'avancement dans l'arbre d'activité 177
A.3.3 Aggrégation des dimensions d'appraisal 179

B Macros Humans-Gen **181**
B.1 Création des tâches mères (*Task)* 181
B.2 Création des tâches feuilles (*Task_Leaf)* 181
B.3 Création des conditions 182

C Implémentation du 3ème cas d'usage **183**
C.1 Arbre d'activité en ACTIVITY-DL 183
C.2 Règles d'action 188

Glossaire **191**

Bibliographie **193**

Table des figures

0.0.1 Formation à la gestion de crise avec *Mission Rehearsal Exercise* (Institute for Creative Technologies) (à gauche), formation aux entretiens annuels avec *Entretien Annuel Manager* (Daesign) (à droite) 1
0.0.2 Exemples d'outils auteurs pour la spécification de comportements : *Spirops Editor* (à gauche), *AI Bench* (à droite) 2

1.1.1 Théorie des Five Factors (d'après [McCrae 03]) 17
1.2.1 Théories de l'évaluation cognitive - Processus général 19
1.2.2 Types d'émotions de la théorie OCC (d'après [Ortony 88, p. 19]) . 22
1.3.1 Représentation de la mémoire de travail Soar sous forme de graphe sémantique (d'après [Laird 98]) 29
1.3.2 Représentation de la mémoire de travail Soar sous forme de triplets (d'après [Laird 98]) 30
1.3.3 Cycle Soar (d'après [Laird 98]) 30
1.3.4 Architecture fonctionnelle de Soar 9 31
1.4.1 Marinier - Cycle PEACTIDM et évaluation cognitive (d'après [Marinier 09]) 33
1.4.2 Marinier - Relation entre émotion, humeur et sentiment 35
1.4.3 EMA - Capture d'écran de *Mission Rehearsal Exercise* 40

2.0.1 Classification empathisation-systémisation (d'après [Baron-Cohen 09]) 44
2.1.1 Modélisation de l'empathie cognitive avec deux individus. 49
2.1.2 Thespian - Projection dans le futur lors de la prise de décision ([Si 09]) 50
2.2.1 Contagion - modèles SIR ([Kermack 27]) 54
2.2.2 Contagion - modèles à seuil 55
2.2.3 ASCRIBE - Vidéo source (à gauche) et capture d'écran de la simulation (à droite) [Bosse 11] 58

3.1.1 Photographie du jeu des tours de Hanoï avec 8 disques (Ævar Arnfjörð Bjarmason, license creative commons) 63
3.1.2 Modélisation des objets du monde pour le jeu de Hanoï 65
3.2.1 HAWAI-DL - Principaux concepts, tiré de [Edward 11] 71
3.2.2 HAWAI-DL - Extrait d'un modèle d'activité 72
3.2.3 Capture d'écran de V3S 73

4.1.1 Modèles de connaissances 83
4.2.1 Concepts principaux de DOMAIN-DL (en haut) et extrait d'un modèle du monde (en bas) 85
4.2.2 Exemple d'application de la règle d'action décrite par l'algorithme 4.1 sur une instance de l'environnement 87
4.3.1 ACTIVITY-DL - Concepts principaux 88
4.4.1 HUMANS-DL - Concepts communs à DOMAIN-DL et ACTIVITY-DL 92
4.5.1 ACTIVITY-DL - Modèle d'arbre d'activité 94
4.5.2 ACTIVITY-DL - Modèle des conditions de l'arbre d'activité . . . 94

5.2.1 Cycle classique des agents cognitifs (à gauche), cycle défini par les théories de l'évaluation cognitive (à droite) 100
5.2.2 Cycle décisionnel d'un humain virtuel 101
5.2.3 Exemple d'écriture des entités perçues dans la mémoire de travail 103
5.2.4 Dynamique des émotions sur la relation d'appréciation (extrait de [Ochs 09]) 108

6.3.1 Cycle d'un humain virtuel - prise en compte de la contagion émotionnelle 120

7.1.1 HUMANS - Principaux modules 125
7.1.2 COMMUNICATOR - Classes principales 126
7.1.3 COMMUNICATOR - Interfaçage avec les modules 127
7.2.1 Architecture fonctionnelle de REPLICANTS 129
7.2.2 Capture d'écran de REPLICANTS (à gauche) et du Debugger Soar connecté (à droite) 130
7.2.3 Cycle des replicants 131
7.3.1 HUMANS-GEN, le module d'opérationnalisation des modèles de connaissances 135
7.3.2 HUMANS-GEN - Processus d'opérationnalisation 136
7.3.3 NGS - Concepts 136
7.3.4 NGS - États d'un *Considerable*, *Conditions* et *Actions* 137
7.3.5 NGS - Lien entre les *Goals* et les *Operators Soar* 140

8.1.1 Évolution de l'humeur dans le temps 145
8.2.1 Captures d'écran de REPLICANTS montrant l'évolution de la situation 149
8.3.1 Arbre d'activité et utilités initiales 151
8.3.2 *Coping-frames* et comportement de l'agent 154

Liste des tableaux

1.1 OCEAN - Facettes du trait *Openness* 14
1.2 OCEAN - Facettes du trait *Conscientiousness* 15
1.3 OCEAN - Facettes du trait *Extravertion* 15
1.4 OCEAN - Facettes du trait *Agreeableness* 16
1.5 OCEAN - Facettes du trait *Neuroticism* 16
1.6 Scherer - Dimensions d'*appraisal* 25
1.7 Comparaison des différentes implémentations d'humains virtuels émotionnels 32
1.8 EMA - Dimensions d'appraisal 37
1.9 EMA - Dimensions de coping 38
1.10 EMA - Stratégies de coping 39

2.1 OCC - Émotions de la classe *Fortunes-of-others* 46
2.2 ASCRIBE - Paramètres régulant la contagion émotionnelle d'un émetteur E vers un receveur R 56

3.1 Actions de mouvement nécessaires pour la résolution du jeu des tours de Hanoï avec 3 disques 64

4.1 Relations temporelles d'Allen 89
4.2 Constructeurs temporels d'ACTIVITY-DL 90

5.1 Dimensions d'*appraisal* 104
5.2 Correspondance entre les catégories OCC et les dimensions des *appraisal frames* 107
5.3 Impact des émotions ressenties sur la relation sociale pour une émotion causée par j (d'après [Ochs 09]) 109
5.4 Impact des émotions affichées de j et des émotions ressenties de i (d'après [Ochs 09]) 109
5.5 Dimensions d'une *coping-frame* 111

6.1 Influence des différences individuelles sur la contagion émotionnelle 118

7.1 NGS - Types de *conditions* 139

8.1 Relations inter-personnelles 144
8.2 Force du canal de communication 145

Introduction

Ces dernières années, les environnements virtuels pour l'apprentissage humain (EIAH) ont connu un essor considérable. Sous la forme de serious games ou de simulations, ils visent à confronter un (ou plusieurs) apprenant(s) à des situations observées sur le terrain. Parce qu'un individu travaille rarement seul, il est indispensable de le former également aux situations dans lesquelles il évolue au sein d'une équipe (collaboration, coactivité). Les humains virtuels sont donc de plus en plus présents dans ces environnements virtuels afin de représenter les autres participants de la situation. Ils prendront par exemple le rôle de collègues ou de clients dans le cadre d'une formation managériale, ou celui de victimes d'attaques terroristes dans le cadre d'une simulation pour la formation à la gestion de crise.

FIGURE 0.0.1: Formation à la gestion de crise avec *Mission Rehearsal Exercise* (Institute for Creative Technologies) (à gauche), formation aux entretiens annuels avec *Entretien Annuel Manager* (Daesign) (à droite)

Cohérence de l'humain virtuel

La cohérence des comportements des humains virtuels est une composante indispensable dès lors que des interactions entre humains réels et humains virtuels sont nécessaires. À la base de cette cohérence se trouve la **rationalité**, c'est-à-dire la capacité à sélectionner l'action la plus efficace pour s'approcher de buts explicites [Bratman 87], car elle permet d'attribuer de l'intentionnalité à l'humain virtuel.

L'intégration de composantes « humaines » telles que les **émotions** et la **personnalité** améliore également cette cohérence [Bates 94, Ortony 03, Niewiadomski 10]. Enfin, l'apprenant doit pouvoir interagir avec ces humains virtuels de manière isolée, par exemple pour soigner un blessé, ainsi qu'avec un ensemble d'individus, par exemple pour donner des ordres à une équipe. Il est donc nécessaire d'offrir à l'apprenant la possibilité d'interagir finement avec les humains virtuels, en proposant différents niveaux d'action. Dans chaque cas, les humains virtuels ciblés doivent **réagir de manière cohérente avec la situation** et les résultats de cette interaction doivent avoir des incidences sur le comportement général du groupe. La **prise en compte de facteurs humains relationnels** est également indispensable afin de permettre une modélisation efficace de ces situations de coactivité ou de collaboration.

Nous définissons la cohérence d'un humain virtuel comme l'adéquation entre sa situation, son état mental (buts, émotions, personnalité, relations sociales) et son comportement.

Adaptabilité de l'humain virtuel

Les simulations informatiques permettent de rendre compte de scénarios complexes dans lesquels évoluent de nombreux individus. Il est nécessaire de spécifier, pour chaque scénario, les interactions et les comportements des humains virtuels afin qu'ils reflètent ceux observés dans les situations réelles. Ceci nécessite un travail d'implémentation dantesque, ainsi que des compétences en programmation dont ne disposent pas forcément les utilisateurs. Il existe donc un réel besoin de proposer des humains virtuels dont le raisonnement est indépendant d'un cadre d'application particulier. Cela nécessite que l'humain virtuel dispose de **processus de raisonnement indépendants du domaine d'application** et soit capable d'interpréter des **connaissances spécifiques modélisées dans un formalisme particulier**.

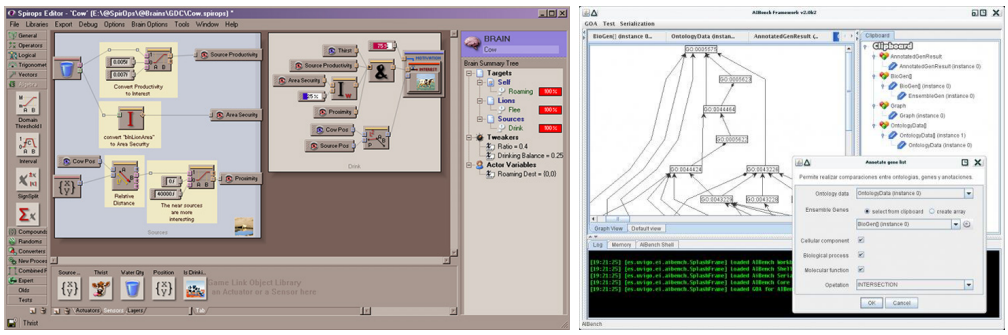

FIGURE 0.0.2: Exemples d'outils auteurs pour la spécification de comportements : *Spirops Editor* (à gauche), *AI Bench* (à droite)

Spécification du comportement

L'objectif d'un environnement virtuel pour l'apprentissage humain étant de proposer des situations d'apprentissages pertinentes, il est généralement nécessaire d'imaginer et de configurer plusieurs scénarios. Ces situations d'apprentissage doivent pouvoir être adaptées facilement. Or, les formateurs ont rarement les compétences nécessaires pour entrer dans le cœur de l'architecture et programmer de nouveaux scénarios. Il est donc nécessaire de proposer des formalismes et des outils auteurs afin de permettre aux formateurs d'ajouter de nouvelles situations d'apprentissage en utilisant les objets et entités déjà existantes. Cependant, un environnement informatique pour l'apprentissage humain nécessite une grande quantité de modules. Le module responsable des humains virtuels côtoie donc bien souvent un module de suivi de l'apprenant ainsi que des modules de scénarisation et une interface graphique. Afin de proposer un apprentissage efficace, il est nécessaire que ces modules communiquent et interagissent de manière cohérente. Le formalisme utilisé pour spécifier les situations d'apprentissage et les entités du monde doit donc être partagé par ces différents modules, qui doivent pouvoir l'interpréter sans traitement humain supplémentaire.

Afin de permettre au formateur de spécifier des scénarios d'apprentissage, il est nécessaire que les **modèles utilisés pour déterminer les composantes individuelles des humains virtuels soient compréhensibles et configurables par des non-informaticiens.** De plus, pour que l'apprenant comprenne l'impact de ses actions et ses erreurs éventuelles, il est nécessaire de pouvoir expliquer le comportement de l'humain virtuel, c'est-à-dire **recenser les raisons de ses actions** et donc de **pouvoir expliciter les différentes variables qui composent son état mental**.

Problématique

L'objectif de cette thèse est de proposer des humains virtuels dont le comportement est cohérent, adaptable et déterminé par des dimensions humaines.

La majorité des simulations informatiques utilise des humains virtuels dont les comportements sont scriptés, c'est-à-dire que la réponse comportementale est spécifiée pour chaque situation qu'il est possible de rencontrer. Les comportements obtenus sont donc cohérents et explicables car écrits en parfaite adéquation avec les besoins de la simulation. Ils ne sont cependant pas du tout adaptables.

Les **architectures cognitives** permettent de générer des **comportements rationnels** en séparant la connaissance du raisonnement [Newell 90, Laird 12]. Il est alors nécessaire de disposer d'**algorithmes de haut niveau** indépendants du domaine d'application qui permettent de guider la cognition, ainsi que de **modèles de connaissances** dépendants du domaine. Ces derniers doivent répondre à un

formalisme compris par les algorithmes de haut niveau afin de permettre à l'humain virtuel d'appliquer son raisonnement sur des objets et des actions concrètes de l'environnement. La principale difficulté réside dans l'intégration de dimensions humaines telles que la personnalité, les émotions ou les relations sociales. Cette intégration doit être réalisée de manière explicite afin de pouvoir expliciter l'état mental de l'humain virtuel et pouvoir expliquer son comportement.

EMA [Gratch 04] **modélise explicitement les émotions et les processus mis en œuvre pour leur élicitation (*appraisal*) et leur influence sur le comportement (*coping*)**. Cependant, les autres dimensions humaines (comme la personnalité et les relations sociales) sont agrégées et intégrées directement dans le comportement et donc non explicites. Les humains virtuels générés présentant un comportement cohérent et adaptable, mais, dans une certaine mesure, non explicable.

Le paradigme de la théorie de l'esprit stipule que le caractère social de l'humain réside dans sa capacité à déchiffrer l'état mental des individus qui l'entourent. Dans cette thèse, nous restreignons le cadre d'étude des interactions entre humains virtuels à la modélisation des interactions émotionnelles, à savoir l'empathie. Les théories en psychologie sociale définissent deux types d'empathie. Tout d'abord, l'**empathie cognitive** est un processus cognitif conscient qui consiste à se projeter mentalement à la place d'autrui pour deviner son état émotionnel. Ce processus nécessite que l'humain virtuel dispose d'une théorie de l'esprit, c'est-à-dire qu'il puisse inférer, à chaque instant, l'état mental des individus qui l'entourent (humains virtuels ou avatars qui représentent l'apprenant). Il doit donc disposer et maintenir en temps réel un modèle de chacun d'entre eux, sur lequel il applique ses propres processus cognitifs afin d'inférer son état mental, en particulier son état émotionnel. Ce processus est très couteux lorsque le nombre d'humains virtuels augmente ce qui pose un problème informatique de **passage à l'échelle**[1]. Le second type, l'**empathie affective**, représente les changements induits chez un individu par la perception de l'état mental d'autrui. Il s'agit d'un processus inconscient et automatique, sur lequel l'esprit n'a aucune prise, et on parle souvent de « contagion émotionnelle ». **Les modèles d'empathie affective existants ne prennent généralement pas en compte les caractéristiques individuelles des humains virtuels**. Le modèle ASCRIBE [Bosse 09] est la seule exception à cette règle et détermine deux composantes individuelles, le pouvoir de contagion qui représente la force avec laquelle un individu observé « émet » son état émotionnel et la susceptibilité à la contagion qui détermine la sensibilité d'un individu à être influencé par les émotions d'autres humains virtuels. Dans ce modèle, ces deux

1. De plus, nous ne pensons pas qu'un individu imagine consciemment l'état émotionnel de chaque personne lorsqu'il est entouré par un grand nombre d'individus, mais qu'il a conscience de cet état de manière naturelle.

caractéristiques sont déterminées de manière *ad hoc* ce qui nuit à la cohérence et à l'explicabilité du modèle de contagion.

Proposition

Nous proposons un modèle d'humain virtuel émotionnel et social dont le comportement est cohérent, adaptable et explicable.

En accord avec [Bouchet 11], nous choisissons d'intégrer les dimensions humaines dans une architecture d'agent rationnel. Nous utilisons une architecture cognitive qui permet de définir des algorithmes de haut niveau indépendants du domaine et proposons un langage de description des entités, des actions et des activités de l'humain virtuel qui permet de représenter les connaissances spécifiques du domaine d'application.

Afin de modéliser les objets du monde ainsi que l'activité des humains virtuels, nous proposons un formalisme inspiré des méthodologies issues de l'ergonomie cognitive, discipline qui vise à décrire l'activité réelle. Ce formalisme permet de décrire des comportements qui ne sont ni prescrits, ni erronés : l'humain peut s'écarter de l'activité prescrite pour atteindre différents compromis. Il est suffisamment simple pour être renseigné, à travers un outil auteur par des individus sans connaissances en programmation, et suffisamment expressif et robuste pour être interprété directement par les humains virtuels.

Nous avons choisi de baser nos travaux sur les études de l'humain réalisées en psychologie qui proposent une représentation explicite des composantes et de leur dynamique. D'une part, les formateurs sont plus familiers avec ces modèles qu'avec des modèles ad hoc et d'autre part ils permettent d'expliquer le comportement. Notre humain virtuel dispose ainsi d'un état mental composé d'une personnalité basée sur les modèles en traits [Cattell 46, Eysenck 76, Costa 92] [2] et d'émotions issues des théories de l'évaluation cognitive [Smith 91, Scherer 01]. Ils sont également liés les uns avec les autres par des relations sociales [Ochs 09, Svennevig 99]. Nous proposons un ensemble de processus qui permettent de gérer la dynamique de ces dimensions humaines (élicitation des émotions, mise à jour des relations sociales, impact de l'état mental sur le comportement). Afin de garantir l'adaptabilité de l'humain virtuel, ces processus sont indépendants du domaine.

La prise en compte des interactions affectives entre les humains virtuels est réalisée à travers un modèle d'empathie affective basé sur le modèle ASCRIBE. Nous déterminons une correspondance entre les dimensions humaines (personnalité, état émotionnel, relations sociales) et les composantes du modèle de contagion émotionnelle afin de garantir l'explicabilité de la contagion émotionnelle.

2. La personnalité est stable et n'a donc pas de dynamique [McCrae 03].

Organisation du mémoire

Cette thèse est découpée en trois parties.

La première s'intéresse à la modélisation de l'humain virtuel. Dans un premier chapitre, nous décrivons les travaux réalisés en psychologie, en particulier la personnalité, les émotions et les processus cognitifs qui régissent la prise de décision de l'humain. Nous étudions ensuite les implémentations informatiques d'humains virtuels émotionnels. Le second chapitre s'intéresse aux interactions qui lient les humains virtuels, et nous focalisons notre étude sur l'empathie qui est l'interaction émotionnelle entre les individus. Nous distinguons deux types d'empathie : l'empathie cognitive, un processus cognitif conscient qui consiste à se projeter mentalement à la place d'autrui pour deviner son état émotionnel et l'empathie affective, un processus inconscient et automatique, qui représente les effets de la perception de l'état mental d'autrui sur celui d'individu et qu'on appelle bien souvent contagion émotionnelle. Le troisième chapitre est consacré aux techniques utilisées pour permettre à un humain virtuel de s'adapter à son environnement. Nous distinguons deux modèles : le modèle du monde qui décrit les entités et les actions et le modèle d'activité permet d'intégrer les actions dans un contexte rationnel dirigé par des buts.

La seconde partie regroupe nos contributions. Le premier chapitre décrit les langages de description que nous proposons afin de représenter les connaissances spécifiques au domaine. À cette occasion, nous proposons DOMAIN-DL qui permet de décrire les entités du monde et les actions, ainsi qu'ACTIVITY-DL pour la description des activités. Le second chapitre décrit notre humain virtuel, en particulier sa personnalité et ses relations sociales, puis les différentes étapes qui régissent sa prise de décision. Nous détaillons en particulier les processus de haut niveau, et comment ceux-ci utilisent les modèles de connaissances spécifiques au domaine pour permettre à l'humain virtuel d'adapter son comportement à ce qui l'entoure. Le troisième chapitre concerne notre modèle de contagion émotionnelle ; nous expliquons le lien entre les dimensions humaines de chaque humain virtuel et son rapport à la contagion émotionnelle, puis comment ce phénomène s'insère dans son cycle décisionnel.

La troisième partie concerne l'implémentation et les résultats obtenus à l'issue de ces travaux. Le premier chapitre décrit REPLICANTS, le module responsable des humains virtuels intégré dans HUMANS, la plateforme logicielle de notre équipe dont l'objectif est de générer des environnements artificiels s'appuyant sur des modèles de l'activité humains à des fins d'apprentissage humain. Le second chapitre présente les résultats obtenus sur quelques scénarios.

Nous concluons en résumant nos contributions et présentons nos perspectives.

État de l'art : Modélisation de l'humain virtuel

> Finally, we'll get machines that think about themselves and make up theories, good or bad, of how they, themselves might work. Perhaps, when our machines get to that stage, we'll find it very easy to tell it has happened. For, at that point, they'll probably object to being called machines. To accept that will be difficult, but only by this sacrifice will machines free us from our false mottos.
>
> *(Marvin Minsky, "Why People Think Computers Can't." AI Magazine (1982))*

Cette partie propose un état de l'art relatif aux différentes thématiques rencontrées pour la modélisation d'humains virtuels émotionnels et sociaux.

Le premier chapitre concerne l'humain en tant qu'entité individuelle et décrit les modèles réalisés par les psychologues pour décrire les composantes qui régissent son comportement, en particulier la personnalité, les émotions et l'humeur. Nous nous attardons ensuite sur les architectures cognitives dont l'objectif est de représenter informatiquement les processus cognitifs de l'humain. Nous présentons enfin quelques implémentations d'humains virtuels utilisant ces modèles.

Le second chapitre s'intéresse aux humains en tant qu'entités sociales et aux interactions qui ont lieu entre eux. Nous focalisons notre étude sur les interactions émotionnelles et distinguons à cette occasion deux types d'empathie : l'empathie cognitive et l'empathie affective. Nous étudions comment ces modèles d'empathies sont associés aux modèles de l'humain virtuel présentés dans le premier chapitre.

Le troisième et dernier chapitre de cet état de l'art concerne la manière dont l'humain se représente le monde et les interactions que celui-ci lui offre. Nous nous intéressons en particulier à la manière dont le raisonnement et les connaissances spécifiques sont découplés afin de permettre une adaptation à différents domaines. Nous étudions deux niveaux de spécification des connaissances de domaine : les connaissances sur le monde et les connaissances sur l'activité.

Chapitre 1

L'humain virtuel

Les humains virtuels sont des entités informatiques dont le comportement, aux yeux d'un observateur externe, est similaire à celui d'un humain réel.

Nous avons défini la cohérence d'un humain comme l'adéquation entre son état mental, la situation dans laquelle il se trouve et son comportement. Il est donc nécessaire de connaitre l'impact de l'état mental sur les processus qui régissent la prise de décision d'un humain réel afin de pouvoir les implémenter informatiquement au sein d'un humain virtuel.

Malgré la multitude d'études réalisées par les philosophes, depuis l'antiquité, et par les psychologues, plus récemment, le cerveau humain reste une boite noire ; les entrées et les sorties sont connues, mais on ne peut que stipuler sur les processus internes qui permettent de passer des unes aux autres. Et si la présence de certains composants tels que la personnalité, les émotions et la conscience fait consensus, leur rôle au sein de la machinerie interne reste un mystère.

Le débat qui préoccupe les philosophes depuis Platon jusqu'à Descartes (au XVIIème siècle) s'intéresse de façon quasi exclusive à distinguer corps et esprit. Au cours de la seconde moitié du XIXème siècle, William James définit la conscience comme un processus, un flux constant de pensée et d'idées, et non plus comme une entité physique : « Consciousness does not appear to itself shopped in bits... It is nothing jointed ; it flows. » Cette nouvelle façon de voir les choses donne naissance à la psychologie qui s'intéresse dorénavant aux processus mentaux comme l'attention, la mémoire, le raisonnement, l'imagination et l'intention.

Dans les années 1940 nait la psychologie cognitive, inspirée par les avancées faites dans le domaine des télécommunications et des neurosciences. Elle étudie les processus mentaux et considère le cerveau humain comme une machine de traitement de l'information. S'inscrivant naturellement dans ce contexte, l'intelligence artificielle s'intéresse à modéliser des humains virtuels. La problématique générale,

qui est donc également la nôtre, réside dans l'adaptation de modèles issus de la psychologie et de la sociologie dans des architectures informatiques. Nous focalisons donc ici notre étude sur les modèles issus de la psychologie qui permettent d'expliciter l'état mental d'un humain et qui décrivent les processus mis en jeu pour assurer leur dynamique.

Nous décrirons tout d'abord les principaux modèles de personnalité, d'émotions et d'humeur existant dans la littérature cognitiviste. Ensuite, nous nous intéresserons aux architectures cognitives qui visent à modéliser et à intégrer les processus cognitifs des humains dans des architectures informatiques. Nous étudierons enfin en détail les deux principales architectures informatiques qui permettent de créer des humains virtuels dotés de comportements émotionnels basés sur les théories de l'évaluation cognitive et dont les processus décisionnels sont régis par une architecture cognitive.

1.1 Personnalité

Depuis la Grèce antique, de nombreuses études se sont intéressées à la personnalité. Par exemple, l'approche biologique estime que la personnalité est déterminée génétiquement, tandis que l'approche humaniste rend responsable l'environnement. On s'écartera ici de ce débat inné *versus* acquis dans la mesure où notre objectif n'est pas de justifier ce qui détermine une personnalité, mais au contraire, d'en trouver les effets sur le comportement. On s'intéressera donc aux approches qui permettent de définir les tendances à l'action d'individus en fonction de leur personnalité. Cette caractérisation en terme d'effets est en accord avec la vision populaire : lorsque l'on demande à une personne de décrire sa personnalité, elle répondra naturellement par une énumération d'adjectifs qui représentent ses comportements habituels (par exemple timide, amicale, indépendante ou sérieuse).

1.1.1 Approche topologique

L'approche topologique définit des catégories de personnalité dans laquelle les individus sont catalogués, une personne ne pouvant appartenir qu'à une seule catégorie à la fois. Cette approche s'inspire de la théorie de l'humorisme développée par Empedocles (495 - 435 avant J.-C.) selon laquelle les quatre éléments basiques (eau, air, feu, terre) constituent l'ensemble des substances connues. Hippocrates (460 - 370 avant J.-C.) applique cette théorie à la médecine et associe des éléments aux fluides corporels, appelés humeurs (du latin *umor*). 200 ans plus tard, Galen (129 - 201) étend ce modèle à la personnalité en reliant directement le niveau des humeurs corporelles aux tendances émotionnelles d'un individu. Il définit ainsi quatre classes de tempéraments : sanguin (trop de sang), mélancolique (trop de bile

noire), colérique (trop de bile) et flegmatique (trop de glaire). Il serait donc possible de corriger un comportement inadéquat en diminuant le niveau de l'humeur responsable. Cette conception a façonné la médecine jusqu'à la Renaissance.

L'approche topologique est aujourd'hui rejetée par les psychologues car jugée trop basique. Elle reste néanmoins profondément ancrée dans la culture populaire, comme le prouve par exemple l'engouement pour les horoscopes (signes du zodiaque, signes astrologiques chinois).

1.1.2 Approche par traits

Les psychologues ont proposé d'utiliser des dimensions continues afin de permettre d'expliquer une plus grande diversité de comportement. On parle ici de « traits de personnalité ». Un trait est une dimension de personnalité utilisée pour catégoriser une personne en fonction du degré de manifestation de ce trait. Ici, un individu n'appartiendra pas à une seule catégorie, mais à toutes à la fois, cette appartenance étant modulée. Par exemple, les valeurs du trait « timidité » peuvent aller de « pas du tout timide » à « extrêmement timide » et permettent de représenter la timidité d'un individu par rapport au reste de la population.

La distribution globale d'une population sur un trait suit une courbe normale, c'est-à-dire que la majorité des individus sera dans la moyenne, avec quelques extrêmes positifs et négatifs. La description de la personnalité d'une personne en terme de traits présume que celle-ci ne varie pas (ou peu) en fonction du temps ou des situations ; une personne sociale le sera de manière générale et quelles que soient les circonstances, même si des fois elle préfèrera s'isoler.

Les chercheurs ont tenté d'identifier les traits de personnalité basiques qui permettent de définir la personnalité d'un individu de manière complète et non redondante. Or, la littérature (anglaise) recense plus de 4500 adjectifs utilisés pour décrire la personnalité, et il est délicat de définir à priori lesquels utiliser, un modèle minimal nécessitant qu'ils soient indépendants. De nombreuses équipes se sont lancées à la recherche du modèle minimal en utilisant la technique statistique d'analyse factorielle qui permet de repérer, parmi le grand nombre de traits proposés, ceux qui sont indépendants et orthogonaux.

À l'issue de ses travaux, Eysenck propose un modèle en deux traits (*Extraversion* et *Neuroticisme*) [Eysenck 47], qu'il enrichit par la suite d'un troisième (*Psychotisme*) après avoir étudié le comportement de personnes souffrant de troubles mentaux [Eysenck 76]. Après plus de dix ans de recherche, Cattell identifie quant à lui 16 traits basiques [Cattell 46].

Par la suite, un consensus semble apparaître autour d'un modèle comportant 5 traits basiques appelé FFM (five-factors model), Big Five ou encore OCEAN (de la première lettre de chacune des dimensions) [Goldberg 92, Costa 92]. Il a été prouvé, aussi bien d'un point de vue théorique qu'expérimental, que ce modèle reprend de

manière exhaustive toutes les dimensions des autres modèles de personnalité. C'est donc celui que nous étudierons plus en détail.

Le modèle OCEAN

Les tests de personnalité permettent de cerner la personnalité, en termes de traits, d'un individu à partir d'un questionnaire comportant d'une vingtaine de questions à plusieurs centaines pour les tests les plus fiables. Ils sont largement utilisés dans les hôpitaux pour détecter les patients susceptibles d'avoir des troubles mentaux ainsi que par les recruteurs pour cerner la personnalité de potentiels employés.

La décomposition d'une personnalité en cinq traits se révèle trop imprécise. Par exemple, alors que les comportements induits sont très différents, les caractères « social » et « aventureux » sont tous définis par la valeur d'*extraversion*, tout comme « l'anxiété » et la « dépression » sont des sous-ensembles du *neuroticisme*. Il est donc nécessaire de raffiner les traits principaux. C'est pourquoi les chercheurs proposent des sous-questionnaires permettant de déterminer les valeurs de ces sous traits, appelés « facettes » [Costa 95] et de préciser les tendances émotionnelles ou comportementales des individus au sein d'un même trait.

Openness représente l'ouverture à l'expérience et concerne l'imagination, la curiosité intellectuelle, l'intérêt pour les idées nouvelles et divergentes. Les individus ayant une forte valeur d'*openness* sont plutôt non conventionnels, tandis que ceux ayant une faible valeur préfèrent rester dans les sentiers battus.

Le trait *openness* est décomposé en six facettes, regroupées dans le Tableau 1.1.

Facette	Définition
Fantasy	vivacité de l'imagination
Esthetic	sensibilité à l'art et à la beauté
Feelings	recherche de sensations fortes
Actions	curiosité à expérimenter
Ideas	curiosité dans les idées
Values	curiosité dans les valeurs sociales

Table 1.1: OCEAN - Facettes du trait *Openness*

Conscientiousness représente le contrôle de soi et la discipline d'un individu. Les individus ayant une forte valeur de *conscientiousness* sont organisés et déterminés, tandis que ceux ayant une faible valeur sont plus négligents et plus facilement distraits de leur objectif. Les facettes sont décrites par le Tableau 1.2.

Facette	Définition
Competence	rationalité
Order	organisation
Dutifulness	sens du devoir
Achievement striving	perfectionnisme
Self discipline	discipline
Deliberation	planification

Table 1.2: OCEAN - Facettes du trait *Conscientiousness*

Extravertion permet de représenter les extravertis (à l'extrême positif) et les introvertis (à l'extrême négatif). Les individus extravertis sont très sociables, énergiques, amicaux et autoritaires, alors que les individus introvertis sont plutôt réservés et indépendants. Les facettes qui composent l'*extravertion* sont définies dans le Tableau 1.3.

Facette	Définition
Warmth	tendance à être formel, impersonnel ou amical
Gregariousness	désir d'être entouré
Assertiveness	capacité à prendre des décisions, exprimer ses idées
Activity	tendance à rechercher une activité (ne pas rester inactif)
Excitement seeking	recherche de nouveauté, prise de risques
Positive emotions	tendance à ressentir des émotions positives

Table 1.3: OCEAN - Facettes du trait *Extravertion*

Agreeableness spécifie la relation d'un individu avec les autres. Une personne très agréable fera beaucoup de compromis et aidera les autres ; il aura des relations sociales plaisantes. Les individus ayant un score d'*agreeableness* faible seront au contraire centrés sur leurs désirs et intérêts propres. Le Tableau 1.4 définit les facettes qui composent ce trait.

Neuroticism caractérise la stabilité émotionnelle d'un individu. Une détresse émotionnelle régulière et de brusques changements d'humeur correspondent à une valeur élevée : les individus ayant une forte valeur de *neuroticism* s'énervent plus facilement et sont plus enclins à ressentir des émotions négatives (tristesse, colère, anxiété, culpabilité,...). À l'inverse, les individus qui présentent une valeur de *neuroticism* faible sont plus calmes et moins sujets à des réactions émotionnelles extrêmes. La majorité des facettes du *neuroticism* est

Facette	Définition
Trust	tendance à faire confiance aux autres
Straightforwardness	capacité à inspirer la confiance des autres
Altruism	prise en considération des autres, tendance à les aider
Compliance	tendance à faire des compromis plutôt que d'imposer ses choix
Modesty	modestie dans le comportement et les compétences
Tender-mindedness	tendance à être charitable et impacté par les « bonnes causes »

Table 1.4: OCEAN - Facettes du trait *Agreeableness*

donc centrée sur les tendances à ressentir des émotions négatives, tel que le montre le Tableau 1.5.

Facette	Définition
Anxiety	tendance à ressentir de la peur
Angry hostility	tendance à ressentir de la colère
Depression	tendance à ressentir de la tristesse, de la solitude
Self-consciousness	tendance à ressentir de la honte, de l'embarras
Impulsiveness	tendance à succomber à la tentation
Vulnerability	tendance à paniquer, à dépendre des autres

Table 1.5: OCEAN - Facettes du trait *Neuroticism*

Comportement et personnalité

Plusieurs études ont démontré qu'une fois stabilisée (vers vingt-cinq ans) la personnalité d'un adulte n'évolue plus au fil du temps [McCrae 03]. Cependant, au cours de son existence, un individu évolue, change et s'adapte. Si elle est statique, la personnalité ne peut donc pas être l'unique responsable du comportement.

À partir du modèle de traits OCEAN, McCrae et Costa développent une théorie de la personnalité qui définit les liens entre la personnalité statique d'un individu et d'autres composantes dynamiques. Cette théorie permet d'expliquer l'impact des influences extérieures telles que la culture et la biographie personnelle sur le comportement. Cette théorie est reprise par la Figure 1.1.1 et permet d'inscrire la personnalité dans une structure individuelle complète et dynamique. Ainsi, la personnalité, bien que statique, influence le comportement de manière indirecte : elle détermine les aspirations personnelles d'un individu et ce sont ces dernières qui modulent les réactions émotionnelles et donc son comportement.

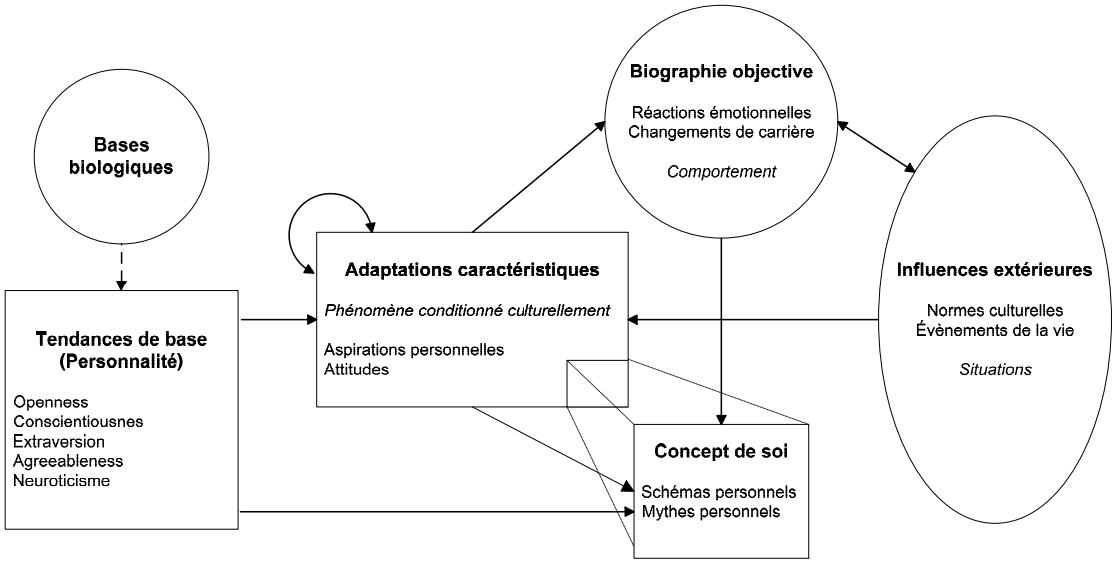

FIGURE 1.1.1: Théorie des Five Factors (d'après [McCrae 03])

1.2 Émotions et Humeur

Pour certains auteurs, il existe un ensemble d'émotions distinctes qui correspondent à des corrélations neurologiques [Ekman 71, LeDoux 96], tandis que pour d'autres, les émotions sont des épiphénomènes qui reflètent des processus internes. Parmi eux, les processus physiologiques génèrent des émotions afin de susciter une réaction cognitive [James 84], pour d'autres, cette causalité est inversée [Smith 91], ou une combinaison des deux [Damasio 06]. Il est donc clair qu'il n'existe pas de définition universelle des émotions.

> « The meanings of words like "feelings", "emotions" or "consciousness" seems so natural, clear, and direct to us that we cannot see how to start thinking about them. » [Minsky 06, p.2]
>
> « Everyone knows what emotion is, until asked to give a definition. Then, it seems, no one knows. » [Fehr 84]

Du point de vue de l'Intelligence Artificielle, l'intérêt n'est pas de déterminer exactement ce que sont les émotions, ni leur nombre ; mais plutôt quels sont les processus mis en cause et comment les implémenter. Deux principales thématiques sont donc généralement adressées.

La première est statique et concerne la définition, la représentation et la classification des émotions.

La seconde aborde les aspects dynamiques et tente de définir les processus qui permettent d'une part, de générer un état émotionnel (*appraisal*) et d'autre part,

de déterminer l'impact de cet état émotionnel sur le comportement (*coping*). On parle dans ce cas des théories de l'évaluation cognitive.

Définition cognitive des émotions

Certains théoriciens définissent l'émotion comme la plus évoluée des trois façons dont un individu dispose pour s'adapter à son environnement [Smith 91]. Ainsi, le premier niveau d'adaptation est le réflexe, une réponse directe et dirigée à un stimulus de l'environnement. Le réflexe étant étroitement lié au stimulus, la réponse est automatique et non adaptable : par exemple, les battements de cil pour hydrater l'œil, ou les frissons en cas de froid. Le second niveau d'adaptation concerne les besoins physiologiques tels que la faim ou la soif. Ces besoins sont associés à des stimuli internes cycliques et leur réponse adaptative est relativement flexible, car un individu est capable de déterminer plusieurs moyens de répondre à ces besoins. Les émotions sont le troisième niveau d'adaptation environnementale : elles ne sont pas des réponses directes à des stimuli de l'environnement, mais elles sont représentatives de la relation qui relie l'individu à son environnement. Les émotions permettent de signifier à l'individu que quelque chose dans l'environnement a changé, remettant en jeu son bien-être, mais sans forcément indiquer la source de ce changement. La réponse adaptative est donc très flexible et permet une grande variété de stratégies d'ajustements spécifiques qui composent un ensemble organisé et cohérent de réponses cognitives, comportementales, émotionnelles et somatiques.

Les théories de l'évaluation cognitive s'intéressent donc à définir, d'une part, les processus responsables de la génération d'un état émotionnel (phase d'*appraisal*) et, d'autre part, l'impact sur la prise de décision (phase de *coping*). Ces modèles s'articulent en plusieurs étapes successives que nous décrirons.

Nous présenterons ensuite les trois principales théories de l'évaluation cognitive : la théorie d'Ortony, Clore et Collins (OCC) [Ortony 88], celle de Smith & Lazarus [Smith 91] et celle de Scherer [Scherer 01]. Nous indiquerons, pour chacune d'entre elles, la structure utilisée pour représenter un état émotionnel.

1.2.1 Théories de l'évaluation cognitive

En psychologie émotionnelle, le terme *appraisal* réfère à l'évaluation cognitive qui précède un épisode émotionnel. L'idée centrale est que des individus différents (avec différentes motivations, buts, normes) vont évaluer la même situation de manière différente et donc présenter des réponses émotionnelles différentes. Les théories de l'évaluation cognitive se distinguent les unes des autres par les dimensions d'évaluation qu'elles utilisent, c'est-à-dire les aspects de la situation qui vont être

évalués afin de générer un état émotionnel. Cependant, le processus d'*appraisal* reste globalement le même et suit les étapes décrites par la Figure 1.2.1.

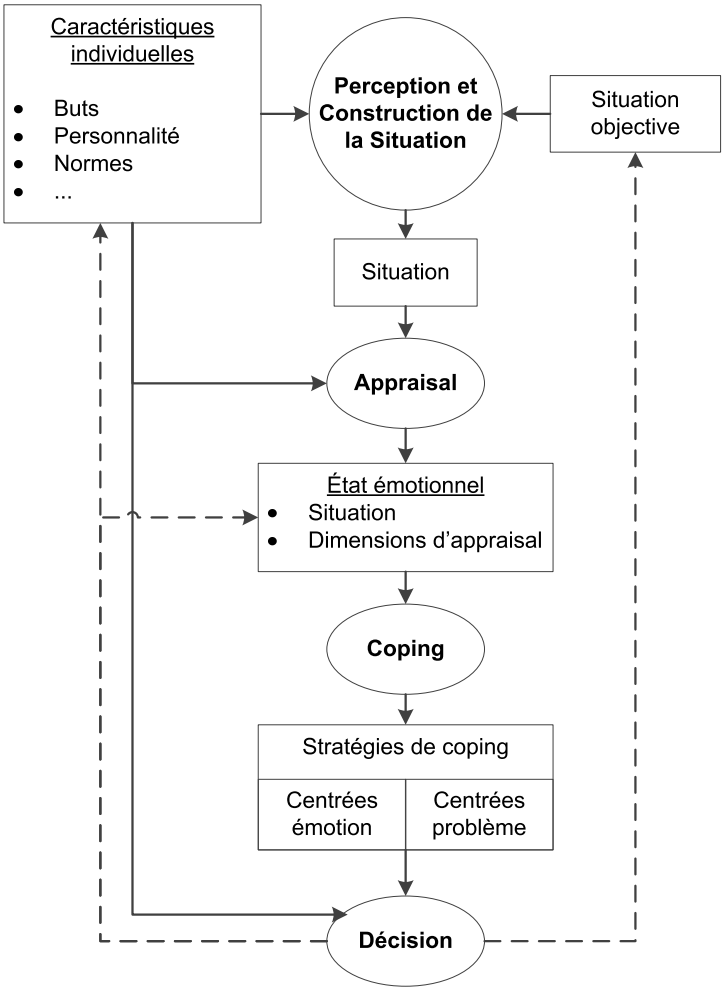

FIGURE 1.2.1: Théories de l'évaluation cognitive - Processus général

Perception et construction de la situation

Les émotions sont représentatives d'une réaction induite par un changement de la relation entre l'individu et son environnement, généralement appelée *situation*. La perception de la situation se fait de manière très dépendante de l'individu. Ce processus va au-delà du simple « encodage » des données visibles : certaines informations peuvent être ignorées, erronées ou amplifiées. Les informations manquantes sont déduites d'après des causes, intentions ou motivations réelles ou imaginées par l'individu.

Appraisal

La seconde étape est l'étape d'*appraisal*, c'est-à-dire la confrontation de la *situation* générée à l'étape précédente avec les caractéristiques individuelles de l'individu. Cette confrontation est réalisée suivant plusieurs axes afin de rendre compte de dimensions propres à chaque modèle. Ainsi, la théorie de Smith et Lazarus définit 6 dimensions d'*appraisal*, tandis que celle de Scherer en contient 16. La théorie OCC évalue différemment cette relation en fonction du fait qu'il s'agisse d'un évènement, d'une action réalisée par un agent, ou de la présence d'un objet dans l'environnement.

Le résultat de l'*appraisal* permet de déterminer de manière explicite l'émotion ressentie par l'individu. Les théories de Smith et Lazarus et de Scherer ont une approche dimensionnelle des émotions : celles-ci sont définies par les différentes valeurs d'*appraisal* et il est possible de les catégoriser par l'intermédiaire de fonctions d'étiquetage. Au contraire, la théorie OCC utilise directement une approche catégorielle qui définit 22 types d'émotions.

Coping

L'étape suivante est la phase de *coping* lors de laquelle l'individu propose une ou plusieurs stratégies d'adaptation. Les théories de l'évaluation cognitive définissent généralement deux types de stratégies : le *coping centré problème* qui consiste à agir dans l'environnement pour améliorer la situation et le *coping centré émotion* qui consiste à modifier l'état individuel, par exemple en modifiant la réponse émotionnelle sans en changer la signification (stratégie d'évitement), ou en en changeant la signification (stratégie de déni ou prise de distance).

Décision

Enfin, en fonction de ses caractéristiques individuelles, l'individu peut décider d'agir en accord avec son état émotionnel en suivant les stratégies d'adaptation suggérées, ou bien choisir de moduler son comportement. Les caractéristiques individuelles et plus particulièrement les normes sociales impactent fortement cette décision. L'action sélectionnée aura donc un impact sur l'environnement et/ou sur l'état interne de l'agent.

1.2.1.1 Ortony, Clore et Collins

La théorie émotionnelle d'Ortony, Clore et Collins [Ortony 88], habituellement appelée OCC, est centrée sur la phase d'*appraisal* et ne s'intéresse donc qu'à la phase de génération de l'état émotionnel à partir de la relation entre l'individu et son environnement.

La théorie OCC utilise une approche catégorielle, avec 22 types d'émotions organisés dans une hiérarchie que l'on peut voir sur la Figure 1.2.2.

La situation est évaluée différemment selon qu'elle soit causée par un évènement, une action ou un objet. Pour chacun de ces cas, la théorie utilise trois variables individuelles (les buts, les standards et les préférences) qui permettent de déterminer comment les évènements, les actions des autres agents et la perception des objets sont confrontés aux buts de l'agent, à ses critères d'acceptabilité morale (standards), et à l'attractivité des objets (préférences) afin de générer l'état émotionnel ressenti.

Par exemple, la perception d'un évènement permettant d'atteindre ou de s'approcher d'un but déclenche de la joie ; réaliser une action contraire à la morale génère de la honte ; la perception d'un objet donnera de l'envie ou du dégout.

À ces émotions sont associées des intensités qui reflètent l'impact de l'émotion sur l'état mental général de l'agent.

Bien que cette théorie ne décrive pas les processus entrant en jeu lors de l'étape de *coping*, son grand avantage est que ceux qui interviennent durant la phase d'*appraisal* sont clairement définis. De plus, le découpage de l'espace émotionnel en catégories distinctes la rend compatible avec une implémentation informatique, ce qui était d'ailleurs l'objectif principal des auteurs.

> "(...) we would like to lay the foundation for a computationally tractable model of emotion. In other words, we would like an account of emotion that could in principle be used in an Artificial Intelligence (AI) system that would, for example, be able to reason about emotion." [Ortony 88, p. 2]

1.2.1.2 Smith &Lazarus

Smith & Lazarus sont les premiers théoriciens de l'évaluation cognitive à séparer l'*appraisal* en deux parties, la première étant responsable de l'évaluation de la situation par rapport au bien-être de l'agent et la seconde étant chargée d'estimer la capacité de l'individu à faire face à la situation [Lazarus 66].

La théorie de l'évaluation cognitive de Smith & Lazarus est représentative des théories de l'évaluation cognitive qui utilisent une représentation dimensionnelle explicite des états émotionnels. Dans leurs premiers travaux, les auteurs définissent une théorie émotionnelle complète, à savoir la phase d'*appraisal,* différentes stratégies de *coping* qui permettent d'obtenir une réponse émotionnelle adaptée, ainsi qu'une description des émotions continue en fonction des valeurs des dimensions d'*appraisal* [Smith 91]. Les processus mis à contribution sont globalement identiques à ceux de la figure 1.2.1.

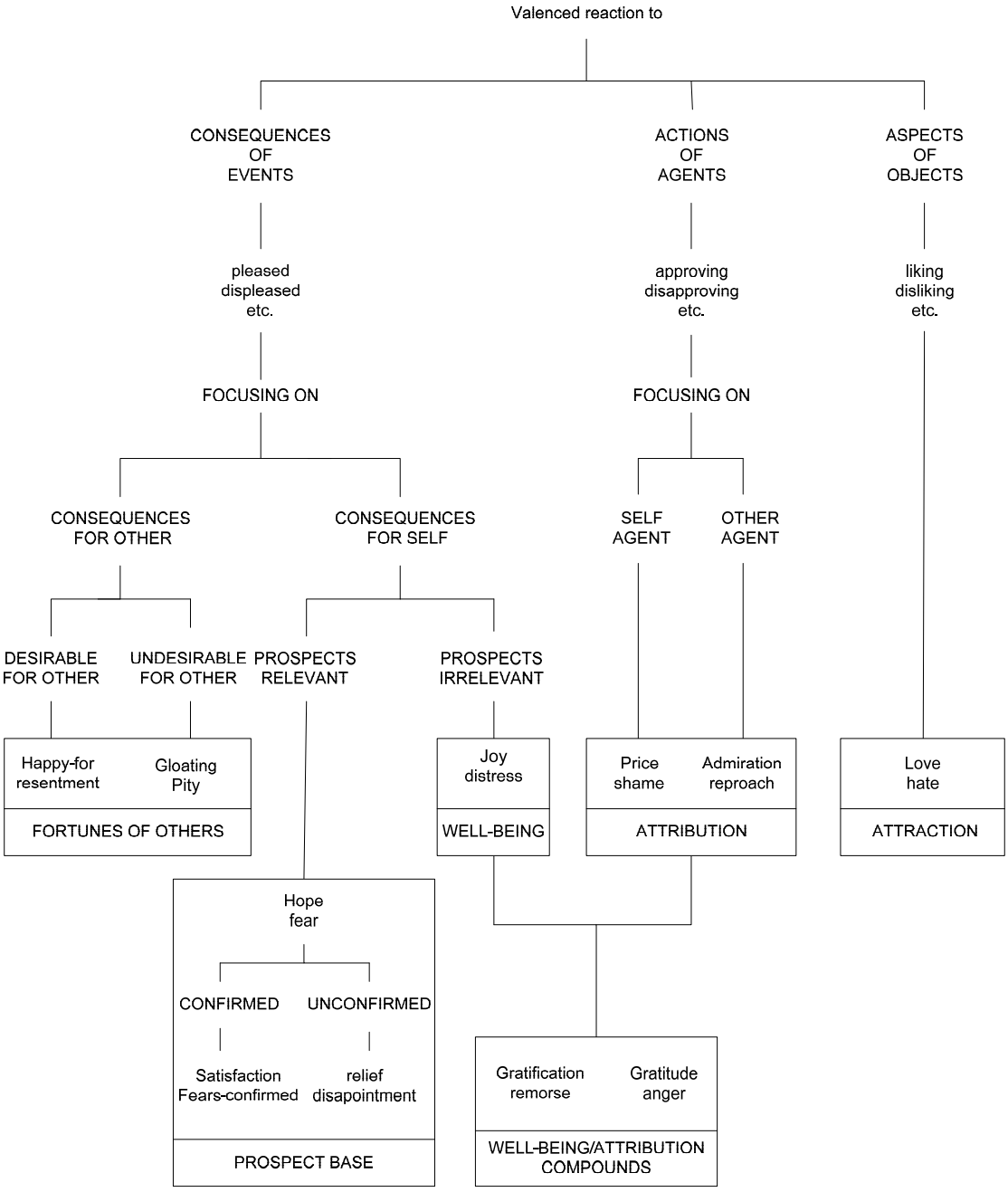

FIGURE 1.2.2: Types d'émotions de la théorie OCC (d'après [Ortony 88, p. 19])

Appraisal

L'étape d'*appraisal* est séparée en deux étapes. La première détermine l'impact de la situation sur le bien-être de l'agent à travers le calcul de deux valeurs :

- *la pertinence motivationnelle* : mesure l'impact du stimulus sur les buts et les préoccupations de l'individu ;
- *la congruence motivationnelle* : représente le degré avec lequel le stimulus facilite ou menace les buts et les préoccupations de l'individu.

Une situation pour laquelle ces valeurs sont nulles n'a pas d'impact sur l'individu et ne modifiera pas son comportement. Seules celles ayant des valeurs non nulles sont donc évaluées par la seconde étape qui renseigne la capacité de l'individu à gérer cette situation. Quatre évaluations sont réalisées lors de cette seconde étape :

- *la responsabilité* : détermine l'objet ou l'individu (parfois lui-même) responsable du changement de l'environnement ayant initié le stimulus émotionnel. Cette information fournit la cible de la réponse émotionnelle, par exemple qui doit recevoir du blâme ou du crédit. Contrairement à l'attribution de causalité qui est « froide », l'attribution de la responsabilité prend en compte les facteurs motivationnels ; ainsi, une personne ayant causé du tort pourra être jugée moins responsable si ses actions néfastes sont jugées non intentionnelles et/ou inévitables ;
- *le potentiel de coping centré problème* : représente la capacité et l'efficacité de l'individu à agir directement sur la situation, en modifiant l'environnement ;
- *le potentiel de coping centré émotion* : représente la capacité et l'efficacité de l'individu à modifier son évaluation de la situation, en modifiant son état personnel ;
- *l'estimation de changements futurs* : recense tous les changements possibles pouvant modifier l'impact émotionnel de la situation.

État émotionnel et Coping

Smith & Lazarus définissent l'étape de *coping* comme un processus biologique et automatique, qui, à partir de la relation environnement-individu enrichie des valeurs d'*appraisal* générées précédemment, détermine un état émotionnel ainsi qu'un ensemble de tendances à l'action et de réponses physiologiques.

Une émotion correspond à une structure composée d'une valeur pour chacune de ces dimensions d'*appraisal*. Par exemple, la peur est représentée par une évaluation dont la pertinence motivationnelle est haute (la situation impacte les buts de l'agent), l'incongruence motivationnelle est haute (la situation menace le bien-être de l'agent), et la responsabilité est attribuée à autrui ; la tristesse sera quant à elle représentée par une évaluation dont la pertinence et la congruence motivationnelles sont hautes (la situation impacte l'agent, et ce, de manière négative), un faible

potentiel de *coping* centré problème (l'agent ne peut rien changer pour améliorer la situation), ainsi qu'une estimation de changements futurs modérée (il y a peu de chances pour que la situation change).

Ce processus de génération de la réponse émotionnelle est biologique et automatique, et les caractéristiques individuelles de l'individu n'ont aucun impact sur cette étape.

> « If two individuals make the same appraisals they will experience the same emotion, regardless of the actual circumstances. » [Smith 91, p. 634]

Conclusion sur la théorie de Smith et Lazarus

Cette théorie très complète fait intervenir tellement de processus qu'elle en devient difficile à implémenter informatiquement. Elle a cependant le mérite d'être plus générique que la théorie OCC ; d'une part, les perceptions sont toutes évaluées suivant le même processus ; d'autre part, elle propose une continuité entre l'évaluation cognitive, l'impact de l'état émotionnel ressenti, et le comportement résultant (via la définition des stratégies de *coping*).

1.2.1.3 Scherer

La théorie des émotions de Scherer [Scherer 01] est calquée sur le modèle général des théories de l'évaluation cognitive de la figure 1.2.1, et plus particulièrement sur la théorie de Smith & Lazarus [Smith 91].

Cependant, quand la théorie de Smith & Lazarus utilise 6 dimensions d'*appraisal* évaluées au cours de 2 étapes, celle de Scherer en définit 13, appelées « *stimulus evaluation checks* » (SEC), regroupées en quatre groupes et évaluées lors de 4 étapes. Ces dimensions et sous-dimensions sont décrites dans la table 1.6.

Ainsi, on évaluera d'abord la pertinence de la situation (*relevance*), puis les implications de cette situation pour l'individu par rapport à ses besoins et à ses buts (*implication*). Ensuite, l'évaluation de la capacité de l'individu à faire face à la situation a lieu (*coping potential*), suivie de la confrontation de la situation avec les normes socioculturelles (*normative significance*).

La théorie de l'évaluation cognitive de Scherer propose un plus grand nombre de dimensions que celle de Smith & Lazarus, et leur évaluation est réalisée séquentiellement, ce qui permet d'écarter rapidement les stimuli non pertinents. Elle ne propose aucun apport en ce qui concerne l'influence de l'état émotionnel sur la phase de *coping* et sur la prise de décision par rapport aux travaux de Smith & Lazarus.

Dimension	**Sous-dimension**	**Description**
Relevance		pertinence de la situation
	Novelty	attention à apporter à la situation (en fonction de la soudaineté et de la familiarité)
	Intrinsic pleasantness	classification positive ou négative de la situation (décorrélée des buts)
	Goal relevance	impact de la situation sur les buts
Implication		degré d'implication dans la situation
	Causal attribution	individu(s) et raison (ou enchainement de raisons) responsables de la situation
	Outcome probability	probabilité d'occurrence de la situation (et de ses conséquences)
	Descrepancy from expectation	différence entre la situation réelle et la situation imaginée
	Goal/need conduciveness	apport de la situation (et de ses conséquences) sur la réalisation des buts
	Urgency	urgence de la situation
Coping potential		capacité de l'individu à gérer la situation
	Control	degré de possibilité de modification de la situation (et de ses conséquences) par l'individu ou autrui
	Power	capacité de l'individu à modifier la situation (éventuellement avec l'aide d'autrui) en fonction de son état et de ses ressources
	Adjustment	capacité de l'individu à s'adapter à la situation si elle n'est pas modifiable
Normative significance		acceptabilité sociale de la situation
	Internal standards compatibility	compatibilité avec les valeurs internes (normes et valeurs de l'individu)
	External standards compatibility	compatibilité avec les valeurs externes (normes socio-culturelles)

Table 1.6: Scherer - Dimensions d'*appraisal*

1.3 Architectures cognitives

1.3.1 Définition

Depuis les débuts de l'Intelligence Artificielle dans les années 1960, les travaux visant à doter les machines d'une intelligence « humaine » [1] ont fleuri.

Dans le cadre de la résolution de problèmes bien définis et appliqués à des domaines restreints, les Intelligences Artificielles obtiennent des résultats supérieurs aux humains grâce à leurs capacités calculatoires. Un bon exemple est la victoire en 1997 de Deep Blue, un programme joueur d'échecs, face au champion du monde Garry Kasparov. Cependant, les performances de ces Intelligences Artificielles sont médiocres une fois sorties de leur contexte.

En effet, l'intelligence ne réside non pas dans la puissance de calcul mais dans la diversité des raisonnements applicables et dans la capacité d'adapter ces raisonnements à la situation. Ainsi, un humain confronté à un nouveau problème peut raisonner par déduction, par analogie, tester différentes stratégies et apprendre de ses erreurs. Afin d'obtenir une Intelligence Artificielle capable de raisonner comme un humain, il est donc nécessaire de comprendre et modéliser ces stratégies avant de pouvoir les implémenter dans une architecture informatique.

En 1959, Ernst et Newell révolutionnent le domaine de l'Intelligence Artificielle avec le « General Problem Solver » (GPS) et la technique de résolution de problème par *means-end analysis* [Newell 59]. Cette technique de résolution de problème, désormais un mécanisme classique en Intelligence Artificielle, consiste à guider la recherche d'une solution dans un espace d'états en sélectionnant l'opérateur permettant de réduire la différence entre l'état courant et l'état solution (« si la différence entre l'état actuel et l'état souhaité est A, essayer de réduire cette différence en appliquant la méthode B »). Cette méthode sous-entend qu'on dispose d'un moyen d'évaluer l'impact des opérateurs sur les états. Dans GPS, ces connaissances sont encodées dans la *Table of connection*s. Dans d'autres systèmes plus avancés tels que STRIPS [Fikes 71] il est possible de calculer cet impact à partir des effets des actions associées aux opérateurs. Bien que GPS soit limité sur le type de connaissances qu'il peut utiliser, il s'agit du premier système d'Intelligence Artificielle capable de résoudre de manière non aléatoire et non linéaire des problèmes que les concepteurs n'ont pas directement pré-résolus.

Les travaux théoriques d'Allen Newell sur la cognition humaine continuent

1. Il serait présomptueux d'assumer que seuls les humains sont capables d'intelligence. Ainsi, Kohler (1887 - 1967), fondateur de la psychologie Gelstalt, remet en cause l'approche behaviouriste de Pavlov en montrant que les chimpanzés n'utilisent pas un apprentissage passif basé sur la récompense attendue, mais qu'au contraire ils mémorisent le problème initial et font appel à un processus cognitif d'essai-erreur afin d'adapter leur méthode de résolution de problème, faisant ainsi preuve d'intelligence (phénomène d'*insight*).

dans ce sens et les « théories unifiées de la cognition » constituent la première tentative de développement d'un cadre théorique de compréhension de la cognition [Newell 90]. Il y définit un modèle d'architecture proche des systèmes informatiques : celui d'une architecture cognitive. Une architecture cognitive est similaire à une architecture informatique dans le sens où toutes deux disposent de mémoires, de composants de contrôle et de traitement, de structures de données et de dispositifs d'entrée-sortie. Mais l'objectif de la seconde est de permettre des traitements classiques, alors que celui de la première est de permettre la représentation, l'acquisition et l'utilisation de la connaissance de manière indépendante du domaine afin d'atteindre un but dépendant du domaine. Ces travaux voient l'homme comme un « système symbolique de traitement de l'information » ; cette hypothèse est fondée sur la possibilité de séparer la cognition de la perception et de l'action et affirme que la cognition consiste en des opérations logiques sur des représentations symboliques implantées indifféremment dans un cerveau ou un ordinateur. Le défi majeur des architectures cognitives réside donc dans la coordination de toutes ces compétences (perception, raisonnement, planification, traitement du langage et apprentissage) afin de résoudre un ensemble de problèmes non déterminés à l'avance.

1.3.2 Principales architectures cognitives

Bien qu'une architecture cognitive unifiée telle que décrite par Newell, capable de tenir la comparaison avec l'intelligence humaine, semble chimérique, la preuve de concept apportée par GPS a permis de lancer les paris. Depuis, de nombreuses architectures cognitives ont vu le jour. Celles-ci peuvent être classées selon trois grands domaines d'application [Laird 12] :

Modélisation cognitive : ces architectures visent à obtenir des comportements spécifiques qui concordent avec ceux observés chez l'humain, au niveau de la rapidité de réponse, des taux d'erreur, voire même des images cérébrales obtenues par IRMf (Imagerie par Résonance Magnétique fonctionnelle).

Développement d'humains virtuels : ces architectures ont pour objectif de générer des personnages virtuels. Ces derniers ne disposent pas de toutes les compétences cognitives des humains, mais sont capables d'interagir avec leur environnement de manière dynamique en utilisant diverses techniques d'Intelligence Artificielle (réactivité, comportement dirigé par des buts, planification).

Modélisation de l'humain : l'objectif de ces architectures est de générer des humains virtuels capables de raisonner sur un vaste champ de problématiques, et d'obtenir les mêmes comportements, en terme d'effets, que ceux des humains.

Les deux principales architectures cognitives sont ACT et Soar. Toutes deux s'inspirent des théories unifiées de la cognition de Newell [Newell 90] et proposent des implémentations informatiques au fonctionnement très similaires. Le fonctionnement décrit ci-dessous est celui de Soar. ACT-R a un fonctionnement quasi-identique dont les subtilités sont expliquées dans [Johnson 97].

ACT, ACT* et ACT-R

ACT-R (Adaptive Control of Thought - Rational) est une théorie générale de la cognition développée par John Anderson et son équipe à l'université Carnegie Mellon.

En 1976, Anderson développe ACT [Anderson 76], une théorie construite sur HAM (Human Associative Memory), un modèle de mémoire sémantique proposé par Anderson et Bower en 1973.

En 1983, Anderson reprend sa théorie et en montre les limites : c'est la naissance de ACT*. Finalement, après vingt ans de travaux sur l'implémentation de la théorie dans des humains virtuels, ACT-R voit le jour en 1993 sous une forme plus générale que ses prédécesseurs [Anderson 93].

L'objectif de ACT-R est la modélisation cognitive des processus cognitifs de l'humain et leur implémentation au sein d'agents informatiques et s'attarde particulièrement sur les questions de mémoire et d'apprentissage. La principale particularité de l'architecture cognitive ACT-R est qu'elle permet une comparaison entre les comportements humains et les comportements informatiques modélisés en fournissant des données quantitatives sur le raisonnement de l'humain virtuel.

Soar

Soar (State, Operator And Result) est une architecture cognitive dont le développement a été initié par John Laird, Paul Rosenbloom et Allen Newell à l'université Carnegie Mellon [Laird 87]. Depuis sa création, l'architecture cognitive Soar a été utilisée dans chacun des trois grands axes. C'est donc sur celle-ci que nous attarderons notre étude.

Au début des années 1980, Newell et Laird reprennent la suite des travaux sur GPS. Leur objectif est d'utiliser plusieurs techniques classiques de résolution de problème à la fois (en particulier les techniques *means-end analysis*, *deep-first search*, *best-first search*, *min-max*, *A**, *generate-and-test*). Ils proposent de décomposer la connaissance utile pour la résolution de problème en éléments primitifs capables de s'assembler afin de générer dynamiquement un algorithme de résolution de problème en fonction de la tâche donnée. Leur approche combine l'utilisation d'espace-problèmes et de systèmes de production. C'est la première version de Soar.

1.3.2.1 Principe de fonctionnement

Un espace-problème est une représentation de la connaissance utilisée pour résoudre un type de problème. Il est composé d'un ensemble d'états, d'un ensemble d'opérateurs qui permettent de naviguer entre ces états, ainsi que de contraintes sur l'application de ces opérateurs (règles). Un espace-problème représente donc un algorithme de résolution de problème.

Le système de production représente la partie dynamique de l'architecture cognitive. Il permet de choisir quel opérateur appliquer afin de guider le cheminement dans l'espace-problème, à travers deux types de mémoires : la mémoire à long terme qui contient les règles de proposition et de choix des opérateurs (mémoire procédurale) et la mémoire à court terme (mémoire de travail) qui contient les connaissances spécifiques sur lesquelles l'humain virtuel raisonne.

Le contenu de la mémoire de travail est représenté sous forme d'un graphe sémantique (Figure 1.3.1) qui peut être mis à plat sous forme de triplets (Figure 1.3.2).

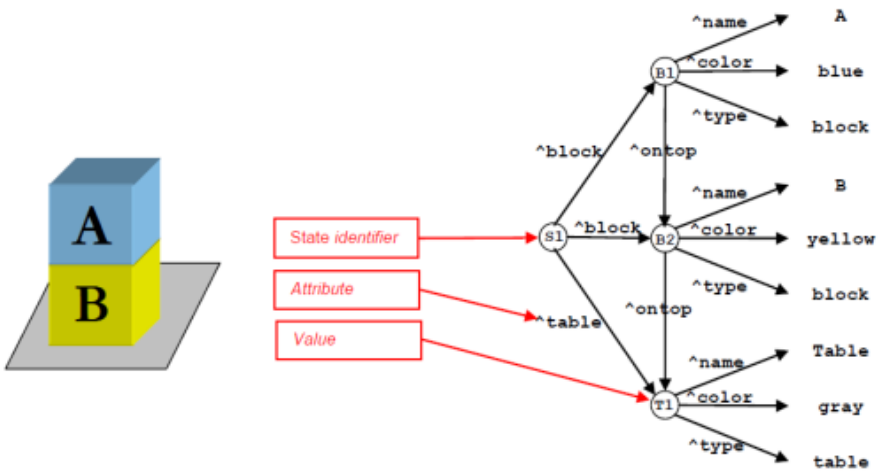

FIGURE 1.3.1: Représentation de la mémoire de travail Soar sous forme de graphe sémantique (d'après [Laird 98])

Les règles de la mémoire procédurales sont composées de deux parties : les conditions et les actions. Lorsque les conditions d'une règle sont compatibles avec les éléments de la mémoire de travail, les actions sont « appliquées ». Soar fait la distinction entre deux types de règles procédurales : les élaborations, qui permettent d'ajouter des connaissances à la mémoire de travail, et les règles relatives aux opérateurs qui permettent de changer d'état (règles de proposition) et ainsi de guider le parcours (règles de préférences).

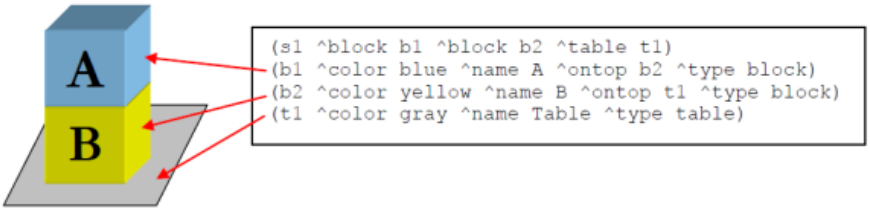

FIGURE 1.3.2: Représentation de la mémoire de travail Soar sous forme de triplets (d'après [Laird 98])

La particularité de Soar est de combiner plusieurs espace-problèmes ce qui permet une grande flexibilité. La solution, si elle existe, émerge au bout d'un certain nombre de cycles de décision. La Figure 1.3.3 représente les étapes de ce cycle. Tout d'abord, toutes les règles procédurales sont lancées en parallèle jusqu'à ce que plus aucune ne soit applicable. Si plusieurs opérateurs ont été proposés, les règles de préférences sont appliquées afin de sélectionner le plus pertinent. Celui-ci est ensuite « appliqué », ce qui correspond à un changement d'état. Le cycle recommence jusqu'à ce que le problème soit résolu, c'est-à-dire qu'une solution ait été trouvée, qu'aucune solution n'existe ou que le mécanisme soit bloqué. Il arrive en effet que le système dispose de trop (ou au contraire, de pas assez) de connaissance pour pouvoir sélectionner l'opérateur à appliquer. Pour résoudre ce blocage, Soar s'est enrichi d'un mécanisme de création automatique de sous-buts. Dans le cas où plusieurs opérateurs sont proposés mais il n'existe aucune règle de préférence qui permette de déterminer celui qui est le plus pertinent, une copie conforme de l'état courant est créée afin de pouvoir évaluer les effets de chaque opérateur et sélectionner le plus pertinent. Dans le cas où aucun opérateur n'est proposé, ce sous-but permet de choisir d'autres espace-problèmes qui seront peut-être moins spécifiques et moins directs, mais éventuellement susceptibles d'apporter une solution.

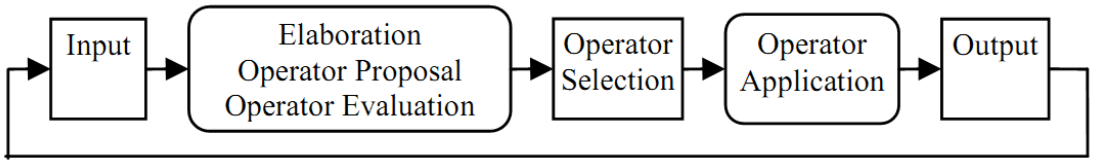

FIGURE 1.3.3: Cycle Soar (d'après [Laird 98])

Architecture

La Figure 1.3.4 représente l'architecture fonctionnelle de Soar. Les modules grisés sont ceux qui composent le socle initial, tandis que les autres sont des ap-

ports successifs. Deux autres types de mémoires à long terme ont ainsi vu le jour : la mémoire sémantique qui permet de modéliser les connaissances générales de l'humain virtuel et la mémoire épisodique qui représente sa mémoire temporelle, ainsi que pour chacune d'entre elles les processus d'apprentissage associés. Une description exhaustive des modules composant Soar 9 peut être trouvée dans [Laird 08, Laird 12].

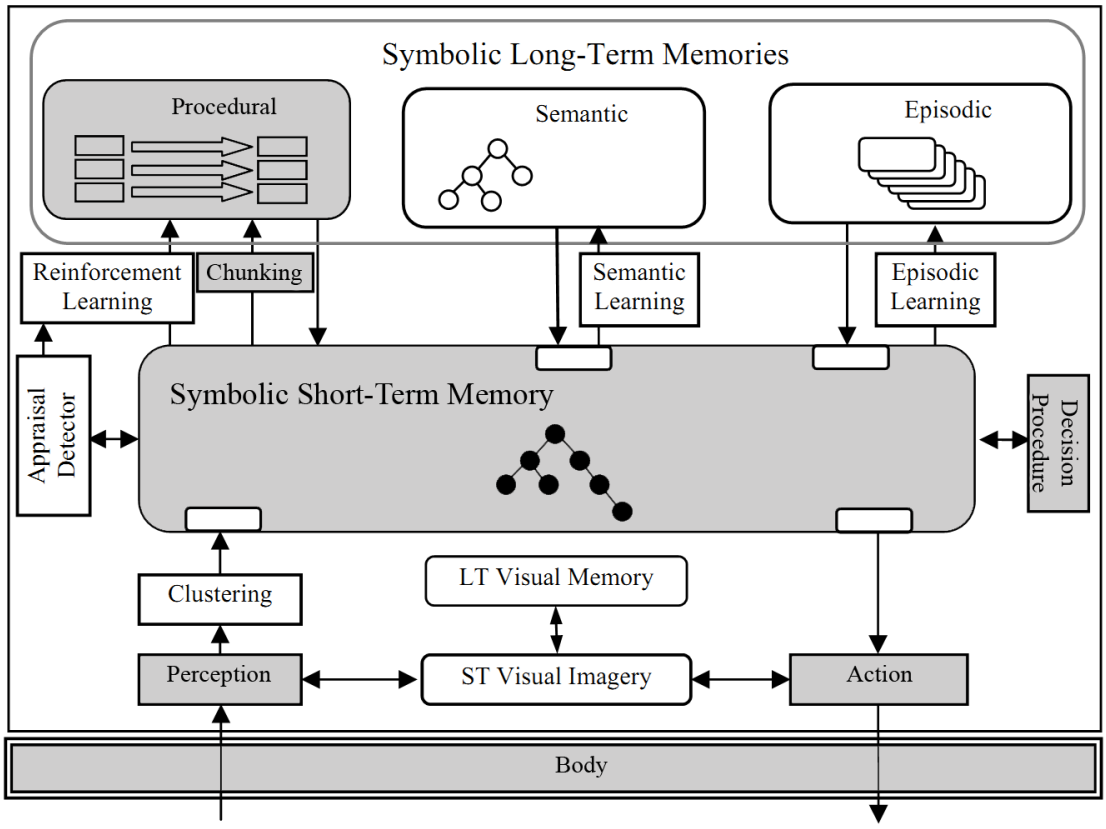

FIGURE 1.3.4: Architecture fonctionnelle de Soar 9

1.4 Implémentations informatiques

De nombreuses implémentations d'humains virtuels ont eu lieu. Nous étudierons ici en particulier les architectures qui permettent de générer des humains virtuels dotés de comportements émotionnels basés sur les théories de l'évaluation cognitive et dont les processus décisionnels sont régis par une architecture cognitive. Le lecteur intéressé pourra se référer à [Marsella 10] pour une description exhaustive des modèles computationnels des émotions.

Nous examinerons ici les deux architectures les plus complètes, celle de Marinier [Marinier 09] et celle de Gratch et Marsella (EMA) [Gratch 04].

Leurs principales caractéristiques sont résumées dans la table 1.7.

	Marinier	**EMA**
Théorie de l'évaluation cognitive originelle	Scherer [Scherer 01]	Inspiré de Smith & Lazarus [Smith 91]
Nombre de dimensions dans la théorie	13	12
Nombre de dimensions utilisées	11	12
Représentation des émotions	continue avec étiquetage catégoriel (OCC)	continue avec étiquetage catégoriel (OCC)
Modélisation du coping	non	oui

Table 1.7: Comparaison des différentes implémentations d'humains virtuels émotionnels

1.4.1 Le modèle de Marinier

Marinier propose un modèle d'architecture cognitive émotionnelle [Marinier 09] qui vise à combiner la théorie cognitive des émotions de Scherer [Scherer 01] (voir 1.2.1.3) avec les éléments des architectures cognitives définis par Newell dans ses théories unifiées de la cognition [Newell 90] et en particulier le modèle de cognition PEACTIDM.

Ce modèle est implémenté au sein de l'architecture cognitive Soar [Laird 08] mais pourrait utiliser n'importe quelle architecture compatible avec les théories unifiées de la cognition (comme par exemple ACT-R).

PEACTIDM

PEACTIDM vise à modéliser les processus de la cognition humaine à travers un ensemble d'opérateurs abstraits, représentés par les boites de la Figure 1.4.1. Il s'agit de *Perceive*, *Encode*, *Attend*, *Comprehend*, *Intend*, *Decode* et *Motor*. Chacun de ces opérateurs encapsule un ensemble de processus qui permettent de transformer des données (représentées en gras le long des flèches).

Perceive représente la perception de l'environnement par l'humain virtuel et la création des informations brutes.

Encode modélise le processus de construction de la relation humain virtuel-environnement sur laquelle sont basées les théories cognitivistes et génère un ensemble de stimuli.

Attend correspond à la phase consciente de l'humain virtuel qui sélectionne un stimulus parmi l'ensemble de ceux générés par *Encode*.

Comprehend représente l'évaluation de la situation et génère les tendances à l'action.

Intend permet de choisir une action à réaliser.

Decode transforme cette action en commandes motrices.

Motor applique ces commandes motrices et induit des transformations dans l'environnement.

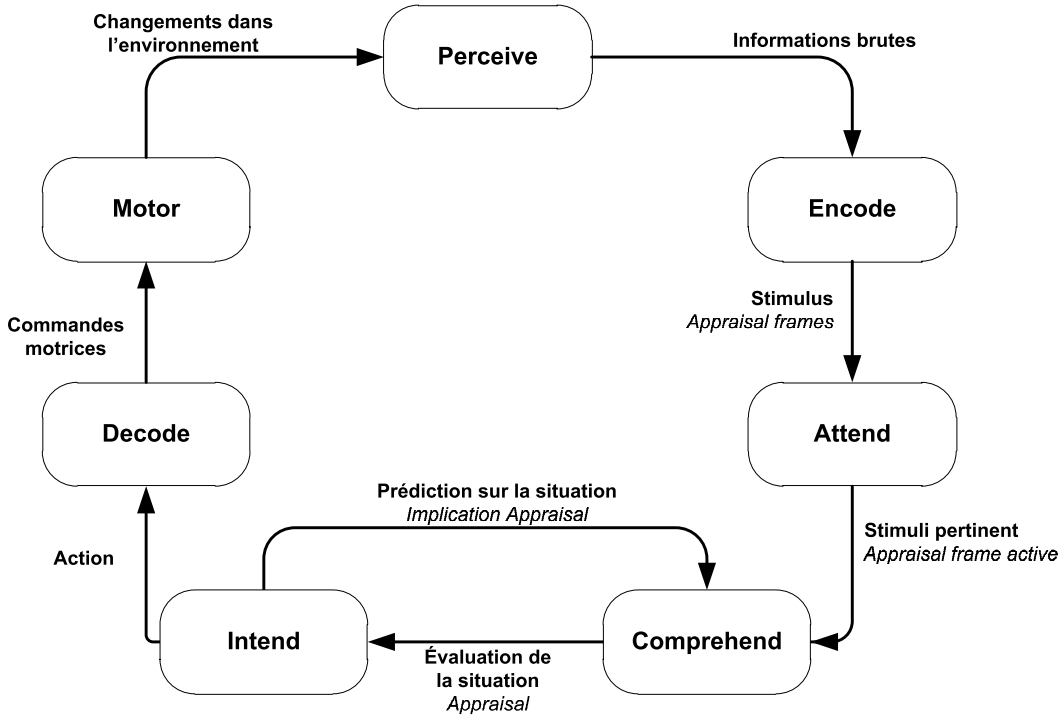

FIGURE 1.4.1: MARINIER - Cycle PEACTIDM et évaluation cognitive (d'après [Marinier 09])

PEACTIDM et appraisal

Le couplage du cycle PEACTIDM avec les étapes des théories d'évaluation cognitives des émotions est relativement direct. En effet, le principe sous-jacent est

que les processus PEACTIDM vont modéliser les étapes de la théorie émotionnelle et les émotions vont fournir les entrées et sorties des modules (représentées en italique le long des flèches sur la Figure 1.4.1.

Encode va générer des *appraisal frames* pour chaque stimuli. Les *appraisal frames* modélisent une certaine appréciation de la relation humain virtuel-environnement ; plusieurs frames sont donc générées pour chaque stimuli. Les dimensions d'appraisal *novelty*, *goal relevance* et *intrinsic pleasantness* sont évaluées et renseignées.

Attend consiste cette fois-ci à sélectionner l'*appraisal frame* active, à savoir celle qui semble la plus pertinente et a donc une *relevance* maximale.

Comprehend représente l'étape d'*appraisal* qui renseigne dans l'*appraisal frame* les dimensions *causal attribution*, *discrepancy from expectation*, *goal/need conduciveness*, *control* et *power*. On obtient donc une frame dont les dimensions émotionnelles sont toutes renseignées [2].

Intend représente l'étape de *coping* : l'humain virtuel génère des tendances à l'action, ainsi qu'un ensemble de prédictions sur l'évolution possible de la situation qui seront utilisées par la suite pour calculer la dimension émotionnelle *Outcome probability*. *Intend* permet de sélectionner parmi toutes ces actions celle à effectuer et la transmet à Decode.

Le comportement de *Decode*, *Motor* et *Perceive* reste inchangé par rapport au fonctionnement défini par PEACTIDM.

L'apport fondamental de ce modèle par rapport à la théorie de l'évaluation cognitive qu'il utilise réside dans la définition de fonctions de calcul d'intensité. Ainsi, l'humain virtuel est capable d'évaluer une intensité émotionnelle à partir des différentes valeurs d'*appraisal*. Il ajoute également une représentation de l'humeur et son évolution dans le temps.

Émotion, Humeur, Sentiment

Marinier fait la distinction suivante entre émotion, humeur et sentiment :

L'émotion correspond à l'*appraisal frame* active.

L'humeur est l'agrégation temporelle des états émotionnels ressentis par l'humain virtuel et représente son historique émotionnel. Elle est représentée dans une *frame* identique à celles utilisées pour représenter les émotions.

Le sentiment correspond à l'émotion ressentie par l'humain virtuel. C'est une combinaison de l'émotion active avec l'humeur, décrite par la figure 1.4.2. Cette

2. Certaines dimensions d'*appraisal* définies par Scherer sont ignorées (*urgency*, *adjustment* et les deux dimensions du groupe *normative significance : external standards compatibility* et *internal standard compatibility*).

combinaison est réalisée suivant une fonction de combinaison qui dépend des différentes valeurs de chacune des deux frames à combiner (humeur et émotion) et est détaillée dans [Marinier 09].

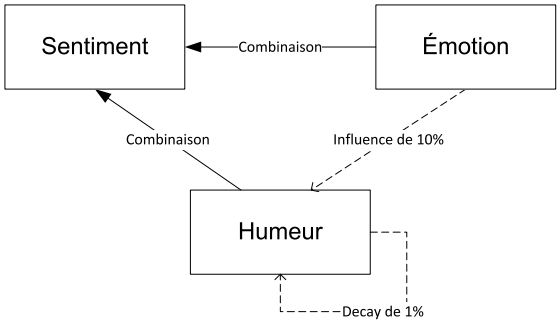

FIGURE 1.4.2: MARINIER - Relation entre émotion, humeur et sentiment

Prise de décision

Dans ce modèle, l'impact de l'état émotionnel sur la prise de décision est minimaliste. En effet, l'auteur n'aborde pas la phase de *coping*. L'apport des émotions sur la prise de décision permet à l'humain virtuel d'évaluer quand il ne fait plus aucun progrès vers l'obtention d'un but et il peut donc l'abandonner. Aucune des stratégies de *coping* centrées problème ou centrées émotion suggérées par les théories de l'évaluation cognitive ne sont implémentées dans ce modèle. L'humain virtuel sélectionne parmi les actions possibles celle qui lui permet d'optimiser son état émotionnel.

Conclusion

Le modèle de MARINIER a été appliqué dans un scénario simple : l'humain virtuel fait face à deux interrupteurs et à deux lampes. Une de ces lampes s'allume et l'objectif est d'appuyer sur l'interrupteur correspondant. Si l'humain virtuel appuie sur le bon interrupteur, la lumière s'éteint, la tâche est réussie et il ressent une émotion positive ; sinon, il ressent une émotion négative. L'intensité de l'émotion ressentie dépend de deux paramètres : ce qui est important pour lui et ce qu'il a imaginé. Ainsi, l'intensité émotionnelle ressentie va dépendre de l'importance qu'il apporte à sa tâche (dimensions du groupe *relevance*), ainsi que de la projection future (dimensions du groupe *implication* et en particulier *discrepancy from expectation* et *outcome probability*).

Dans la suite de ses travaux, MARINIER a utilisé son modèle dans d'autres cas d'application : un humain virtuel qui se déplace dans un labyrinthe afin d'atteindre un but et un robot nettoyeur qui parcourt un ensemble de pièces afin de

rassembler tous les objets qu'il rencontre. Dans ces contextes, Marinier utilise l'état émotionnel résultant d'une action comme mécanisme d'apprentissage par renforcement. Cette approche permet un apprentissage par renforcement interne à l'humain virtuel sans nécessité d'une fonction d'évaluation externe.

Marinier propose une implémentation des processus d'évaluation cognitive de la théorie de Scherer dans un modèle de la cognition humaine (PEACTIDM). Il se focalise sur le couplage des différents processus qui permettent de générer un état émotionnel explicite (*appraisal*) et propose une implémentation fonctionnelle dans l'architecture cognitive Soar ainsi qu'un modèle d'apprentissage par renforcement émotionnel. La phase de *coping* qui gère les stratégies d'adaptation suggérées par les théories de l'évaluation cognitive n'est pas abordée. La modélisation d'autres dimensions humaines telles que la personnalité ou la prise en compte de relations entre les humains virtuels n'est pas prise en compte.

1.4.2 EMA

EMA est un modèle informatique développé par Gratch & Marsella dont l'objectif est de rendre compte des processus cognitifs d'humains virtuels émotionnels [Gratch 04], aussi bien au niveau de l'*appraisal* que du *coping* et de la prise de décision finale.

Le modèle EMA s'inspire des théories de l'évaluation cognitive des émotions et en particulier du livre de Smith et Lazarus dans lequel ces derniers défendent leur théorie unifiée des processus d'*appraisal* et de *coping* [Smith 91] (voir 1.2.1.2).

EMA respecte les étapes classiques définies par les théories de l'évaluation cognitive (pour plus de détails, voir 1.2.1) et les intègre au sein de l'architecture cognitive Soar [Laird 87].

Perception et construction de la situation

Dans EMA, un humain virtuel modélise son rapport au monde par l'intermédiaire d'une interprétation causale qui correspond à une représentation enrichie de l'environnement sous forme de plans augmentée de processus issus de la théorie de la décision et d'informations intentionnelles. L'utilisation combinée de ces trois techniques permet d'associer des plans, des désirs, des intentions, des probabilités et des utilités à chaque situation. Le graphe causal représente les connaissances dépendantes du domaines et est généré à partir d'une ontologie. L'utilité est une valeur agrégée qui représente l'intérêt que l'humain virtuel associe à la situation. Elle est spécifiée « à la main » dans le graphe causal.

La perception de l'environnement permet de faire évoluer l'interprétation causale ce qui déclenche parfois des évènements, décrits par les auteurs comme des situations susceptibles de générer des émotions et devant donc être évaluées.

> « [an event is] any physical action represented in the causal interpretation that is believed to facilitate or inhibit some (past, present or furure) state with non-zero utility for the humain virtuel (or for entities that the humain virtuel is aware of). » [Gratch 04, p. 10]

Appraisal

Chaque évènement génère des structures de données particulières appelées *appraisal frame*. Un évènement pouvant être décrit suivant plusieurs points de vue (c'est-à-dire concerner plusieurs objets dans la représentation causale), plusieurs *appraisal frames* seront donc associées à un même évènement afin de représenter ces différentes situations.

Une *appraisal frame* contient la description de la situation ainsi que les valeurs d'*appraisal* qui lui sont associées. EMA utilise ses propres dimensions d'*appraisal*, reprises dans le tableau 1.8 (d'après [Gratch 04, p. 6]).

Dimension	**Sous-dimension**	**Définition**
Relevance		degré d'attention à apporter à la situation
Desirability		impact de la situation sur les buts de l'humain virtuel
Causal attribution	*Agency*	humain virtuel responsable de la situation
	Blame/credit	blâme ou crédit à apporter au responsable
Likelihood		probabilité d'occurrence de la situation (passée ou future)
Unexpectedness		prévision antérieure de l'occurrence de cette situation
Urgency		délai dans lequel traiter la situation
Ego involvement		impact sur l'image de soi (estime sociale, morale,...)
Coping potential	*Controllability*	degré d'influence sur la situation
	Changeability	possibilité que la situation change d'elle-même
	Power	pouvoir d'un humain virtuel de modifier la situation
	Adaptability	capacité à gérer les conséquences d'une situation

Table 1.8: EMA - Dimensions d'appraisal

L'interprétation causale permet d'évaluer chacune des dimensions des *appraisal*

frames. Une fois la *frame* complétée, un ensemble de règles permet de lui associer un type d'émotion et une intensité. Si plusieurs émotions sont actives en même temps, elles sont agrégées. EMA utilise six émotions issues de la classification catégorielle OCC (décrite en 1.2.1.1) : *Hope, Joy, Fear, Distress, Anger, Guilt.*

Coping

L'objectif du *coping* est de sélectionner une stratégie d'adaptation pertinente par rapport à la situation courante. Les auteurs pensent qu'il est possible de sélectionner cette stratégie à partir de l'interprétation causale. En effet, l'étape de *coping* peut être vue comme l'inverse de l'*appraisal* : il faut remonter les liens de causalité pour modifier les éléments qui ont causé l'émotion [Marsella 09]. Les auteurs proposent donc une correspondance entre des stratégies de *coping* classiques et des éléments clés de l'interprétation causale.

Chaque opération cognitive réalisée par l'humain virtuel va générer une opportunité de coping via une *coping elicitation frame* dont les composantes sont définies par la table 1.9.

Dimension	**Définition**
Focus-agency	humain virtuel ou objet qui a provoqué l'opération cognitive.
Interpretation-objects	éléments de l'interprétation causale référencés par l'opération cognitive. Il peut s'agir de tâches, d'objets, d'états.
Max-interpretation	objet pour lequel l'interprétation causale contient l'appraisal frame la plus forte.

Table 1.9: EMA - Dimensions de coping

Les auteurs ont implémenté un ensemble conséquent de stratégies de coping, recensées dans le tableau 1.10 [Marsella 09].

Cinq étapes sont nécessaires pour modéliser le processus de coping. Elles sont détaillées dans l'algorithme 1.1.

Conclusion

L'architecture EMA est utilisée dans la simulation *Mission Rehearsal Exercise* (MRE) dont l'objectif est de former des dirigeants militaires à la prise de décision dans des contextes dangereux. Dans le scénario décrit l'entrainé est en charge de maintenir la paix dans un village lorsqu'un des véhicules de son équipe renverse un enfant, le blessant gravement. Des humains virtuels intelligents dirigés par EMA jouent le rôle du sous-officier, du médecin et de la mère de l'enfant. L'entrainé

Type de *coping*	**Nom de la stratégie**	**Description**
Centré problème	*Planning*	trouver une action pour modifier la situation
Centré émotion	*Positive reinterpretation*	modifier l'importance des causes de la situation
	Acceptance	abandonner la recherche de solution pour améliorer la situation
	Denial	adapter les liens de causalité et les probabilités d'occurrence
	Mental disengagement	diminuer la désirabilité de la situation
	Shift blame	modifier les causes de la situation

Table 1.10: EMA - Stratégies de coping

Algorithme 1.1 EMA - Algorithme de *coping*

1. Identifier une opportunité de coping si la valeur *max-interpretation* d'une *coping elicitation frame* est supérieure à un certain seuil.
2. Élaborer la situation en appliquant différents raisonnements sur les *coping elicitation frames*.
3. Proposer des stratégies en sélectionnant parmi l'ensemble de stratégies celles dont les conditions d'application sont vérifiées.
4. Évaluer chaque stratégie proposée.
5. Choisir une stratégie et l'appliquer. Des heuristiques permettent de départager plusieurs stratégies concurrentes ou de les combiner si elles sont compatibles.

peut communiquer avec ces humains virtuels par l'intermédiaire d'un module de reconnaissance et de synthèse vocale. Les humain virtuels disposent également de comportements non verbaux afin de représenter graphiquement leur état émotionnel. La Figure 1.4.3 est une capture d'écran de la simulation MRE.

FIGURE 1.4.3: EMA - Capture d'écran de *Mission Rehearsal Exercise*

EMA est un modèle complet de la théorie de l'évaluation cognitive qui propose un ensemble d'algorithmes indépendants du domaine qui permettent d'éliciter des émotions (*appraisal)* et de modéliser leur influence sur le comportement (*coping*). La cohérence du comportement est donc assurée.

Dans une certaine mesure, les comportements sont adaptables. En effet, le graphe causal qui représente les connaissances spécifiques au domaine est généré à partir d'une ontologie. Cependant, il reste nécessaire de spécifier l'utilité de chaque situation. Il est ainsi possible d'associer deux utilités complètement opposées à deux situations pourtant très similaires.

L'explicabilité est également limitée. En effet, si le choix d'un comportement plutôt qu'un autre pourra se justifier en utilisant les *coping frames* sur lesquelles figurent l'état émotionnel, la justification de la valeur d'utilité responsable de cet état reste vague car elle n'utilise pas un modèle explicite, compréhensible par l'apprenant.

1.5 Conclusion

Nous avons décrit les principales théories en psychologie qui permettent de décrire les dimensions humaines de manière explicite et qui détaillent les processus gérant leur dynamique.

Nous avons tout d'abord vu que les modèles de personnalité en traits permettent de décrire la personnalité d'un individu de manière exhaustive et explicite. Le modèle OCEAN [Goldberg 92, Costa 92] semble faire l'unanimité. Nous avons également constaté que la personnalité est considérée comme statique.

Nous avons ensuite présenté les théories de l'évaluation cognitive, qui présentent l'émotion comme la plus évoluée des trois façons dont un individu dispose pour s'adapter à son environnement. L'émotion y est modélisée de manière explicite et un ensemble de processus cognitifs permet de générer un état émotionnel à partir d'une situation (*appraisal*) et de sélectionner une stratégie d'adaptation appropriée (*coping*). Selon les modèles, ces processus cognitifs sont plus ou moins détaillés. Par exemple, Ortony, Clore et Collins [Ortony 88] ne s'intéressent qu'à l'étape d'*appraisal* tandis que Smith & Lazarus [Smith 91] et Scherer [Scherer 01] définissent l'*appraisal* et le *coping*.

L'intelligence de l'humain réside dans sa capacité d'adaptation [Newell 90]. Ainsi, confronté à un nouveau problème il peut raisonner par déduction, par analogie, tester différentes stratégies et apprendre de ses erreurs. Les architectures cognitives visent à modéliser des entités capables d'associer différentes stratégies afin de s'adapter à de nouvelles situations. Toutes deux basées sur cet objectif, ACT-R [Anderson 76] et Soar [Laird 87], sont les deux principales architectures cognitives. ACT-R s'attarde particulièrement sur les questions de mémoire et sur la comparaison entre les comportements humains et les comportements informatiques modélisés en fournissant des données quantitatives sur le raisonnement de l'humain virtuel. Au contraire, Soar a une visée plus large qui comprend la reproductibilité de comportements humains et la modélisation d'humains virtuels. Nous avons présenté le fonctionnement de Soar, celui d'ACT-R étant sensiblement identique.

Enfin, nous avons vu comment ces différents éléments ont été combinés pour permettre l'implémentation informatique d'humains virtuels émotionnels et cognitifs, en particulier à travers l'utilisation d'architectures cognitives. Nous avons présenté à cette occasion l'architecture de Marinier [Marinier 09] et EMA [Gratch 04], qui visent toutes deux à intégrer les dimensions émotionnelles de la théorie de l'évaluation cognitive de Smith & Lazarus [Smith 91] au sein de l'architecture cognitive Soar. Ces architectures modélisent la dynamique des émotions et leur impact sur le comportement de manière explicite et permettent donc de répondre au besoin de cohérence de l'humain virtuel. Cependant, l'explicabilité de la prise de décision n'est pas garantie car la décision est basée sur une valeur d'utilité qui est spécifiée à la main au sein du graphe comportemental.

Chapitre 2

Interactions émotionnelles entre humains virtuels

La psychologie, fondée par les travaux de W. James à la fin du XIXème siècle s'intéresse au fonctionnement de l'esprit et à l'adaptation de l'humain à son environnement. Dans ces études, l'individu est toujours seul. Bien que John Dewey décrive l'humain comme un « animal social » en 1917, il faudra attendre les années 1930 pour voir naitre la psychologie sociale et les premières études sur les interactions entre les individus.

Théorie de l'esprit

En sciences cognitives, le terme « théorie de l'esprit » [1] désigne les processus cognitifs permettant à un individu d'expliquer et de prédire ses propres actions, ainsi que celles des individus qui l'entourent [Premack 78]. Cette théorie repose sur la capacité à attribuer à d'autres individus des états mentaux (composés de leurs croyances, perceptions, émotions et intentions).

Les travaux de S. Baron-Cohen visent à expliquer les difficultés sociales vécues par les autistes et les personnes atteintes du syndrome d'Asperger [Baron-Cohen 09]. D'après lui, ces troubles s'expliquent par des difficultés survenues au cours du développement de la théorie de l'esprit chez ces personnes qui deviennent incapables d'attribuer à d'autres individus des croyances différentes des leurs [2]. La théorie « empathisation-systémisation » classe les individus selon deux axes :

1. à ne pas confondre avec la « philosophie de l'esprit » qui étudie la séparation entre le corps et l'esprit.

2. Par exemple, la phrase « Marie croit que John a une liaison avec sa collègue » est vraie si Marie le croit, indépendamment du fait que John ait effectivement une liaison ou non. Il est donc nécessaire de pouvoir représenter ce que l'on croit vrai (ici, que John n'a pas de liaison), de ce que quelqu'un d'autre croit (ici, Marie croit qu'il a une liaison).

- Empathisation (E) : capacité à identifier l'état mental d'une autre personne et y répondre émotionnellement de manière adaptée
- Systémisation (S) : capacité à analyser toute chose et en concevoir un système logique

La Figure 2.0.1 représente les cinq classes d'individus que l'on obtient en faisant le ratio entre le score d'empathisation et le score de systémisation. Plus de 54% des hommes sont du type « S » et plus de 44% des femmes sont du type « E ». 65% des personnes autistes ou atteintes du syndrome d'Asperger sont du type « S extrême » et présentent donc un grand écart entre leur capacité d'empathisation et leur capacité de systémisation, soit un déficit empathique.

Type S extrême **S >> E**	**Type S** **S > E**	**Type B** **(Équilibré)** **S = E**	**Type E** **E > S**	**Type E extrême** **E >> S**

FIGURE 2.0.1: Classification empathisation-systémisation (d'après [Baron-Cohen 09])

Empathie cognitive, empathie affective

Appliquée aux émotions, la théorie de l'esprit forme ce que l'on appelle communément l'*empathie*.

> « Empathy seems like a simple concept - one feels what another feels - but the more one learns about it, the more it becomes, and the complexity is in the domain [...] of processus of interaction between affect and cognition. » [Hoffman 84, p. 103]
>
> « My thesis is that our fellow-feeling for the misery of others comes from our imaginatively changing places with the sufferer, thereby coming to conceive what he feels or even to feel what he feels. [...] Just as being in pain or distress of any kind arouses the most excessive sorrow, so conceiving or imagining being in pain or distress arouses some degree of the same emotion, the degree being large or small depending on how lively or dull the conception is. »[Smith 12]

D'une manière générale, les psychologues sociaux proposent de distinguer deux composantes à l'empathie [Smith 12, Spencer 70, Davis 94] :

l'empathie cognitive représente la capacité à identifier l'état émotionnel d'une autre personne (correspond à l'attribution d'état mentaux de la théorie de l'esprit)

l'empathie affective représente le phénomène qui fait ressentir à l'observateur les émotions d'un observé ; on parle souvent de **contagion émotionnelle**

La distinction principale entre ces deux concepts porte sur le fait que l'empathie cognitive est un processus conscient et cognitif, qui fait appel à la théorie de l'esprit, tandis que l'empathie affective (ou contagion émotionnelle) est un phénomène inconscient et automatique. Ces deux processus ne se déroulent donc pas au même niveau, bien que l'empathie cognitive précède très souvent l'empathie affective (l'attribution d'un état émotionnel à un individu observé implique une modification émotionnelle chez l'observateur).

Empathie et théories de l'évaluation cognitive

Notre objectif étant de modéliser des humains sociaux, il est indispensable que ceux-ci soient liés par des relations sociales et que celles-ci aient une influence sur leur comportement. Nous proposons donc de doter nos humains virtuels de relations sociales et de processus qui permettent de modéliser l'influence qu'ils ont les uns sur les autres, en particulier d'un point de vue émotionnel.

Les théories de l'évaluation cognitive proposent de modéliser de manière explicite les émotions. Celles-ci sont générées en réponse à une situation par des processus cognitifs (*appraisal*). Certaines de ces théories considèrent que le comportement est une réponse adaptative à cet état émotionnel et définissent un ensemble de stratégies régies par des processus cognitifs (*coping*).

La théorie des émotions OCC définit les émotions de manière catégorielle et propose une catégorie particulière d'émotions dites « empathiques » que nous présenterons.

Les architectures de Marinier et EMA qui modélisent l'*appraisal* ainsi que le *coping* de manière fine n'abordent pas les interactions entre les humains virtuels. Reprenant les fondements de EMA, Thespian vise à modéliser le phénomène d'empathie cognitive au sein d'une plateforme pour la scénarisation adaptative [Si 09]. Les humains virtuels sont dotés d'une théorie de l'esprit. Nous expliquerons pourquoi cette modélisation peut poser des problèmes de performance.

Nous décrirons ensuite les modèles d'empathie affective existants, ainsi que leurs implémentations.

2.1 Empathie cognitive et théories de l'évaluation cognitive

Les théories de l'évaluation cognitive permettent de générer un état émotionnel à partir d'une situation. La construction de cette dernière est influencée par

les caractéristiques individuelles de l'individu concerné (personnalité, buts, préférences,...). Ainsi, pour qu'un individu A ressente des émotions d'empathie pour un individu B [3], il doit appliquer les processus d'évaluation cognitive qui lui servent à générer son propre état émotionnel en utilisant les caractéristiques de B. Il est donc nécessaire que A dispose d'un modèle de B, potentiellement incomplet et erroné.

Cette conception est compatible avec la définition de l'empathie cognitive qui consiste à se projeter à la place d'autrui pour imaginer son état émotionnel et qui nécessite de disposer d'une théorie de l'esprit permettant d'évaluer les états mentaux de l'individu observé.

2.1.1 Empathie cognitive et OCC

La théorie des émotions OCC (voir 1.2.1.1) définit une classe d'émotions particulière pour représenter les émotions que l'on ressent par rapport à autrui, intitulée *Fortunes-of-others*.

Les émotions de cette classe sont déclenchées lorsque A perçoit qu'un évènement survient pour B. Ces émotions sont définies dans la table 2.1 et plusieurs paramètres permettent de définir leur intensité :

- la **désirabilité** de l'évènement pour A
- la **désirabilité** (présumée) de l'évènement pour B
- le degré selon lequel A juge que B mérite cet évènement (**mérite**)
- l'estime que A a pour B (relation d'**appréciation**)

Désirabilité A	**Désirabilité B**	**Appréciation**	**Mérite**	**Émotion**
+	+	+	+	*happy-for* (heureux pour)
-	-	+	-	*pity* (pitié)
+	-	-	+	*gloating* (jubilation)
-	+	-	-	*resentment* (ressentiment)

Table 2.1: OCC - Émotions de la classe *Fortunes-of-others*

Les deux premiers types d'émotion, *happy-for* et *pity* sont des émotions d'empathie dans le sens où l'individu qui les ressent est en empathie avec l'autre, et ce, au sens strict du terme. C'est-à-dire que d'une part, A et B sont impactés de la

3. Dans la suite, on étudiera toujours l'empathie du point de vue de l'individu A par rapport à l'individu B, ce processus étant évidemment symétrique.

même manière par l'évènement (jugé désirable ou indésirable) et que d'autre part A estime B (relation d'appréciation positive).

Les deux autres types d'émotion, *gloating* et *resentment* ne sont pas, au sens strict du terme, des relations d'empathie car l'état émotionnel des deux individus n'est pas le même. Elles permettent de représenter les émotions ressenties par un individu par rapport à l'occurrence d'un évènement arrivant à quelqu'un qu'il n'apprécie pas. Pour cette raison, elles sont appelées *Ill-will emotions* (émotions de mauvaise volonté).

Dynamique

Pour qu'un individu A ressente des émotions d'empathie pour un individu B, deux calculs de désirabilité sont nécessaires. Tout d'abord, A doit évaluer la désirabilité de l'évènement pour lui même en calculant l'impact des conséquences de l'évènement sur ses propres buts. Il doit ensuite évaluer la désirabilité de l'évènement pour B en confrontant les conséquences de l'évènement aux buts de B. Pour cela, A doit disposer d'une représentation (qui peut être incomplète ou erronée) des buts de B. A doit ensuite évaluer le mérite qu'il attribue à B, en accord avec ses standards. Ces deux valeurs de désirabilité permettent ensuite de déterminer quel type d'émotion sera déclenché parmi les quatre types du groupe *Fortunes-of-others*. L'intensité de l'émotion est enfin évaluée en fonction de l'estime que A a pour B (relation d'appréciation) ainsi que du degré que A attribue au fait que B mérite que cet évènement lui arrive.

Conclusion

La théorie OCC définit deux types d'émotions d'empathie liées à l'occurrence d'évènements. Les émotions empathiques liées aux perspectives futures (*hope, fear*), aux actions des humains virtuels (*admiration, reproche*) ou aux aspects des objets (*attirance, dégoût)* ne sont pas prises en compte, ce qui limite grandement ce modèle.

2.1.2 Empathie cognitive et théories de l'évaluation cognitive

Théories de l'évaluation cognitive

Les théories de l'évaluation cognitive de Smith et Lazarus (voir 1.2.1.2) et de Scherer (voir 1.2.1.3) n'abordent pas la thématique de l'empathie ni d'ailleurs celle plus générale des relations inter-individus.

La figure 2.1.1 reprend le processus général de ces théories (pour une explication détaillée, se référer à 1.2.1) et montre comment le processus serait adapté pour deux individus. Cet exemple est centré autour de Pablo ; un processus similaire a

évidemment lieu pour Anna. Conformément aux théories de l'évaluation cognitive, Pablo construit la situation en fonction de ses caractéristiques individuelles ; il l'évalue, puis détermine les stratégies de coping appropriées. Enfin, il détermine la stratégie la plus pertinente et agit éventuellement sur son environnement. Ces processus sont en blanc sur la figure 2.1.1

Dans le cas présent, Pablo va également construire une situation en fonction des caractéristiques individuelles d'Anna (ce modèle peut être incomplet ou erroné), puis évaluer cette situation en utilisant son propre processus d'appraisal combiné aux caractéristiques d'Anna (en gris). Ceci lui permet de déterminer l'état émotionnel dans lequel se trouve Anna par rapport à la situation perçue.

EMA

EMA n'intègre aucun mécanisme permettant de modéliser l'état émotionnel d'autrui. La principale raison réside dans le fait que les humains virtuels basés sur EMA ne disposent pas de théorie de l'esprit et ont donc une représentation commune des croyances. En effet, l'intégralité des processus d'EMA dépend d'un graphe causal qui permet d'associer des plans, des désirs, des intentions, des probabilités et des utilités à chaque situation. Chaque humain virtuel dispose de son propre graphe causal, mais cette représentation ne permet pas de représenter des croyances divergentes sur un même objet (une seule valeur de chaque type peut être associée à une situation).

Thespian

Thespian [Si 05] est un système multi-agent destiné à la scénarisation interactive : l'objectif est de permettre à l'utilisateur de participer au développement d'une histoire en jouant le rôle d'un des personnages et en interagissant avec des humains virtuels.

Thespian est construit sur PsychSim, un framework destiné à simuler des agents sociaux dont la décision est basée sur des chaines de Markov partiellement observables [Marsella 04]. L'utilisation des chaines de Markov partiellement observables permet de modéliser des croyances différentes, erronées et parfois contradictoires et de leur attribuer des probabilités d'occurrence. Ces croyances sont représentées sous forme de prédicats. La personnalité et la motivation des humains virtuels sont modélisées à travers un ensemble de buts et une valeur numérique représente l'importance accordée à chacun d'entre eux. Le processus de décision sélectionne, parmi les actions réalisables, celle dont l'utilité est maximisée. Cette utilité prend en compte l'importance du (des) but(s) concerné(s) ainsi que la probabilité de réussite de l'action.

Thespian introduit tout d'abord la notion d'émotion au sein de PsychSim de

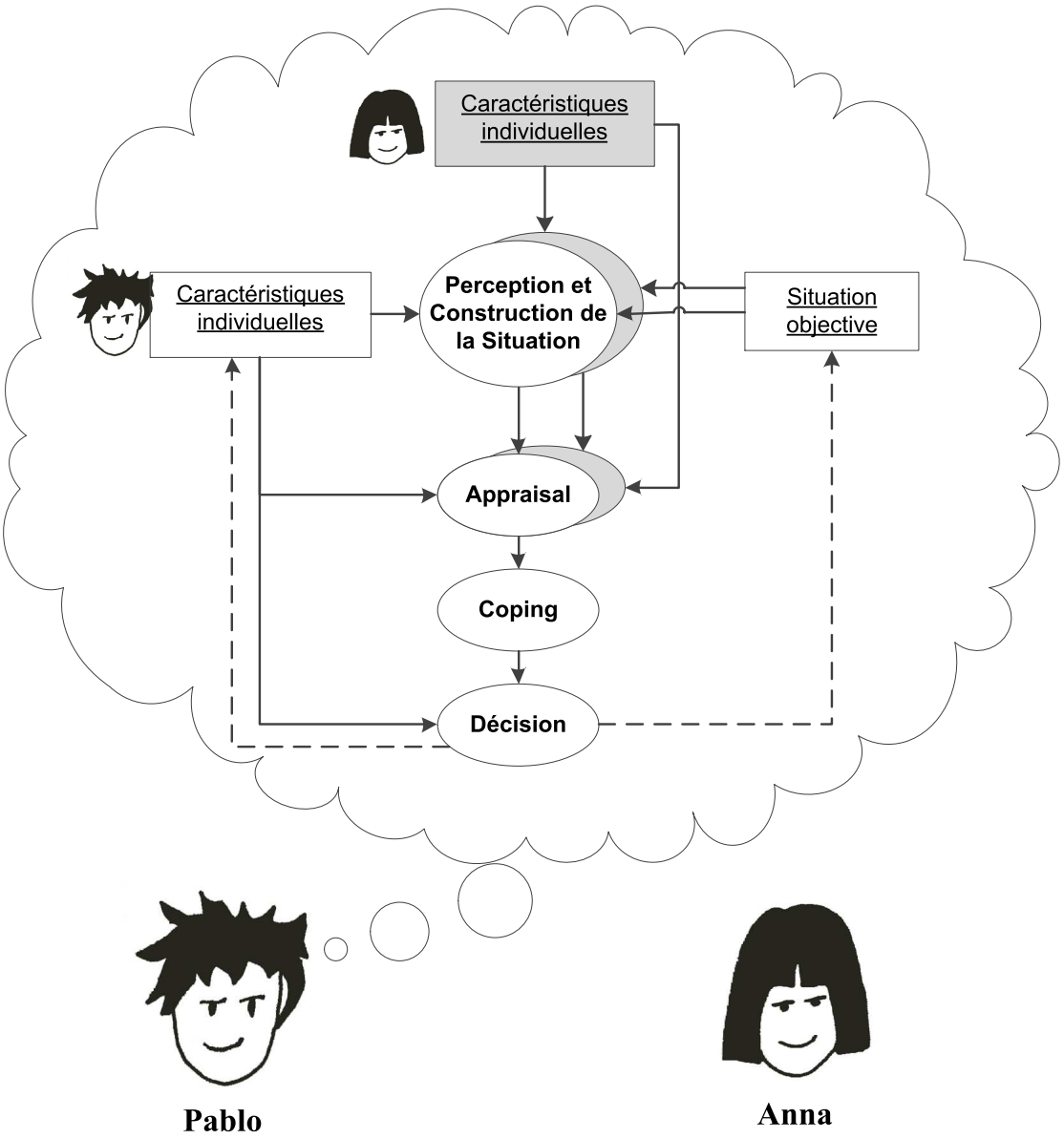

FIGURE 2.1.1: Modélisation de l'empathie cognitive avec deux individus.

manière implicite [Si 09]. Les *appraisal frames*, structures permettant de représenter l'impact émotionnel d'une situation dans EMA, disparaissent. Les dimensions de ces *appraisal frames* sont directement évaluées lors du calcul de l'utilité d'une action lors du processus de décision, qui prend la place du *coping*.

Thespian dote ensuite ses humains virtuels d'une théorie de l'esprit ; ils génèrent des croyances sur les croyances, sur les buts et sur les processus de décision des autres humains virtuels. Quand un humain virtuel choisit la prochaine action à réaliser, il se projette dans le futur pour évaluer les effets de chaque action sur l'état et les croyances de chacun des autres humains virtuels. Il n'évalue pas seulement l'effet immédiat, mais imagine également les réponses possibles de chacun d'entre eux, puis ses propres réactions à ces réponses, etc. La Figure 2.1.2 montre un exemple de cette projection.

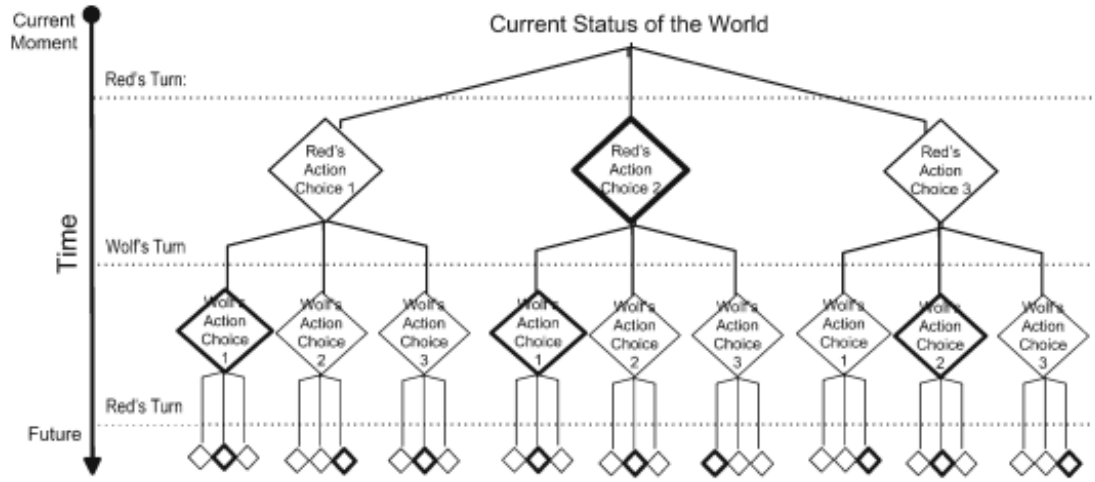

FIGURE 2.1.2: Thespian - Projection dans le futur lors de la prise de décision ([Si 09])

Le principal avantage de Thespian réside dans la représentation explicite des croyances que chacun des humains virtuels génère sur les autres. D'une part, la prise en compte des émotions d'autrui permet d'améliorer la réponse émotionnelle en la rendant sociale : par exemple, la nouveauté n'est plus évaluée uniquement comme l'occurrence d'un stimulus inattendu, mais également par rapport à ce que l'on attend des autres humains virtuels. Enfin, cela permet de modéliser des émotions centrées sur les relations sociales et donc de modéliser un humain virtuel dont le but est d'aider les autres.

Le problème principal de cette approche réside dans la nécessité de déterminer le niveau de projection utilisé lors de la prise de décision. En effet, il se pose rapidement un problème de performance et il est nécessaire de trouver des heu-

ristiques qui permettent de limiter l'explosion combinatoire d'une telle projection, par exemple lorsque l'utilité se stabilise pour chacun des humains virtuels considérés. Les auteurs ont cependant démontré que différents niveaux de projection mènent à des prédictions différentes sur les évènements futurs, en particulier sur la nouveauté d'un évènement, sur ses causes et sur les changements que l'on peut y apporter.

De plus, Thespian souffre des mêmes manques d'explicabilité que EMA.

2.1.3 Conclusion

Le modèle OCC utilise une approche catégorielle pour spécifier les émotions et définit un groupe particulier d'émotions pour modéliser l'empathie. Ces émotions particulières ne sont déclenchées que lors de l'occurrence d'un évènement et ne permettent pas de modéliser des émotions empathiques liées aux perspectives futures (*hope, fear*), aux actions des humains virtuels (*admiration, reproche*) ou aux aspects des objets (*attirance, dégoût)*, ce qui limite grandement ce modèle.

Les théories de l'évaluation cognitive n'envisagent pas le cas où plusieurs individus sont présents et encore moins les interactions émotionnelles qui peuvent survenir. Les intégrer nécessiterait que chaque humain virtuel soit doté d'une théorie de l'esprit afin d'avoir un modèle des autres individus et soit capable d'évaluer chaque situation de leur point de vue.

Thespian est une architecture basée sur des chaines de Markov partiellement observables qui permet de modéliser des humains virtuels cognitifs, émotionnels et sociaux. Les émotions sont modélisées de manière implicite et ont un impact direct sur le calcul d'utilité d'une action possible. Chaque humain virtuel est doté d'une théorie de l'esprit qui lui permet d'évaluer l'impact de ses actions sur les autres, d'imaginer leurs réactions, puis ses propres réactions à ces réactions, etc. Il est nécessaire de déterminer un nombre de pas de projection maximum afin d'éviter l'explosion combinatoire due au nombre d'états à évaluer. Cette limitation pose un problème de cohérence car différents niveaux de projection mènent à des états émotionnels différents. Comme pour EMA, le comportement des humains virtuels Thespian n'est pas explicable car les caractéristiques individuelles d'un individu (telles que sa personnalité) sont modélisées de manière *ad hoc* sous la forme d'un ensemble de buts et d'importances associées.

2.2 Empathie affective, contagion émotionnelle

L'empathie affective représente le phénomène qui fait ressentir à l'observateur les émotions qu'il attribue à l'individu qu'il observe. Dans le paradigme de la théorie de l'esprit, ces émotions font partie de l'état mental de l'individu observé,

celui-ci étant calculé au cours de l'empathie cognitive. Ce phénomène de contagion émotionnelle a cependant été observé bien avant l'éclosion de ce paradigme ; à la fin du XIXème siècle, l'accroissement des populations et la démocratisation de la politique, de l'économie, de la culture et la communication transforment l'époque en « ère des foules ». Les sociologues commencent à s'intéresser à la différence flagrante de comportement de l'individu lorsqu'il est seul et lorsqu'il est entouré de ses pairs, à fortiori dans le contexte de foules dont ils dressent un portrait péjoratif[4] [De Tocqueville 50, Tarde 90, Le Bon 95].

En particulier, Gustave Le Bon montre comment l'individu est altéré par la foule, tandis que la foule acquiert une « âme » commune dotée d'une unité mentale composée par imitation, contagion et suggestion. Selon lui, le groupe exerce sur l'individu une influence qui le dénature et le pousse à régresser vers un stage primaire de l'humanité.

> « Chez une foule, tout sentiment, tout acte est contagieux, et contagieux à ce point que l'individu sacrifie très facilement son intérêt personnel à l'intérêt collectif. [...] Il est facile de constater combien l'individu en foule diffère de l'individu isolé ; mais il est moins facile de découvrir les causes de cette différence. » [Le Bon 95].

Les avis sont plus nuancés chez ses contemporains. Bien que Freud encense les travaux de Le Bon dans son étude consacrée aux rapports psychoaffectifs entre l'individu et le groupe [Freud 21], il est moins catégorique quant à la supposée régression de l'individu au sein d'un groupe et explique cette influence sur le comportement par une mutation psychoaffective due à la réincarnation du mythe de la horde originaire dirigée par le père aimé et haï à la fois, le père étant dans ce contexte le leader de la foule.

Pour le sociologue scientifique Durkheim, « la conscience collective est la forme la plus haute de la vie psychique, puisque c'est une conscience de consciences » [Durkheim 68] ; elle apporte un soutien intellectuel à l'individu en lui permettant de s'inscrire dans un groupe et d'acquérir une rationalité et logique conceptuelle dont il ne disposerait pas autrement.

Tout comme Freud et Durkheim, nous ne pensons pas que l'individu raisonne de manière moins intelligente lorsqu'il est en groupe. Nous reprenons le postulat que l'influence du groupe sur l'individu est une mutation des rapports psychoaffectifs, et que celle-ci a lieu via des phénomènes de contagion et de suggestion. De nombreux psychologues et sociologues se sont intéressés à ces phénomènes.

En 1905, Vigouroux et Juquelier introduisent le terme de « contagion mentale ».

4. Le lexicographe Émile Littré définit à l'époque une foule comme « le vulgaire, le commun des hommes ».

> « Sous le titre de contagion mentale, nous nous proposons d'étudier la contagion des actes réflexes, des états émotifs, des sentiments, des perceptions, des mouvements volontaires, des idées, des croyances, c'est-à-dire, de toutes les manifestations de l'activité de l'axe cérébro-spinal. [...] La contagion mentale implique donc l'existence d'un sujet inerte, apte à subir une influence. Elle implique aussi, et par là même, un sujet actif qui exerce cette influence. Il y a toujours un *contagionneur* s'il y a un *contagionné*. Mais le premier est inconscient de l'influence qu'il exerce, le second est inconscient de l'influence qu'il subit. » [Vigouroux 05]

Hatfield, Cacioppo et Rapson reprennent également cette définition de la contagion.

> « The tendency to catch (experience/express) another person's emotions (his or her emotional appraisals, subjective feelings, expressions, patterned physiological processes, action tendencies, and instrumental behaviors). » [Hatfield 94]

Les auteurs précisent que cette contamination découle de notre capacité à percevoir, analyser, interpréter et mimer les émotions et les intentions d'autrui, de manière inconsciente et automatique. N'étant pas consciente cette contagion s'applique entre toutes les personnes du groupe, bien que son intensité dépende de nombreux facteurs. Cette théorie a été confirmée et étoffée par de nombreuses études et expériences. Par exemple, appliquée au contexte professionnel [Barsade 02] note que la contagion d'émotions positives dans un groupe améliore la coopération, permet d'éviter les conflits et augmente l'impression de performances ressentie par ses membres. Elle est également compatible avec la définition de l'empathie affective donnée par la théorie de l'esprit.

2.2.1 Modèles catégoriels de contagion émotionnelle

Il existe de nombreux travaux sur le phénomène de contagion dans les domaines de la biologie (modèles épidémiologiques) et des sciences sociales (propagation des rumeurs, des tendances d'achat, des avis politiques). La contagion y est définie comme résultant du contact entre un individu contaminé et un individu cible. Ces modèles permettent de déterminer l'état de l'individu cible à l'issue de chaque contact.

Ces modèles peuvent être regroupés au sein de deux catégories :

les modèles d'interaction indépendante sont dits « sans mémoire » car ils considèrent que chaque contact est indépendant. La majorité des modèles épidémiologiques sont classés dans cette catégorie. La Figure 2.2.1 représente la catégorie des modèles SIR (pour *Susceptible*, *Infectious*, *Recovered*). Son implé-

mentation la plus simple, réalisée par Kermack-McKendrick [Kermack 27] a permis d'expliquer la fluctuation du nombre d'individus infectés lors d'épidémies telles que la peste (Londres 1665-1666, Bombay 1906) et le choléra (Londres 1865). Ce modèle suppose que la taille de la population est fixée (pas de naissances ni de décès) et homogène et qu'il n'y a pas de période d'incubation. L'ajout de catégories et de transitions permet de rendre compte d'autres états dans lesquels peuvent se trouver les individus (décès, en incubation, immunisation permanente, immunisation temporaire) et d'autres types d'épidémies (tuberculose, fièvre typhoïde, ...).

les modèles à seuil prennent en compte les expositions précédentes. Ils définissent un certain nombre de conditions nécessaires afin que la contagion puisse avoir lieu, représentées par des seuils, et l'impact d'un contact dépend de la proximité du seuil concerné : si le seuil est dépassé, l'individu devient contaminé (Figure 2.2.2). Ces modèles permettent de modéliser les phénomènes de contagion sociale dans lesquels les individus décident du comportement à adopter en fonction des décisions prises par les autres. Par exemple, un joueur de hockey sur glace ne choisit de porter un casque que si au moins un certain nombre de joueurs en portent déjà un [Schelling 73, Granovetter 83]. Durupinar [Durupinar 10] propose un modèle de foule[5] dans laquelle la contagion des émotions utilise un modèle à seuil : chaque humain virtuel dispose d'une jauge de contamination et d'un seuil au-dessus duquel il est considéré comme contaminé. À chaque pas de temps, les valeurs de contamination des voisins contaminés s'ajoutent à la jauge. Si cette valeur de contamination est supérieure au seuil de l'humain virtuel cible, celui-ci devient contaminé.

Figure 2.2.1: Contagion - modèles SIR ([Kermack 27])

Dodds et Watts proposent un modèle global qui combine ces deux approches [Dodds 05]. Chaque individu est dans l'un des trois états suivants : contaminé, non contaminé ou immunisé. Dès qu'il entre en contact avec un individu contaminé, il reçoit une dose de contamination dont la valeur est tirée dans distribution aléatoire. Cette dose s'ajoute à son quota de contamination. S'il est non contaminé

5. La foule est composée d'humains virtuels dotés de personnalité (modèle OCEAN, voir 1.1.2) et d'émotions (modèle OCC, voir 1.2.1.1). Les actions qu'un humain virtuel peut réaliser sont définies par son rôle qui est statique (par exemple manifestant, leader, policier). La prise de décision est directement déterminée par des tendances à l'action reliées à la personnalité des humains virtuels.

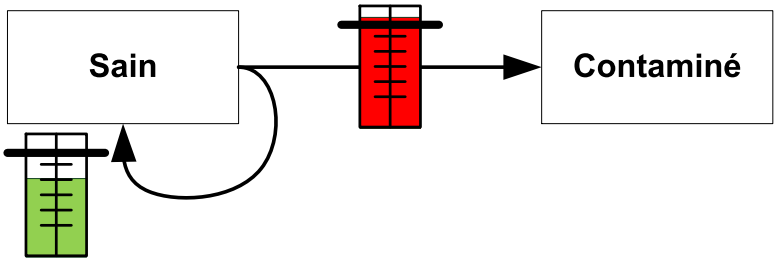

FIGURE 2.2.2: Contagion - modèles à seuil

et si le seuil de contamination, est dépassé, il devient alors contaminé. S'il est immunisé, il peut redevenir contaminé selon une probabilité fixée. Les différents paramètres de ce modèle permettent de rendre compte des deux approches dont il s'inspire en réglant la mémoire des individus relative aux expositions (une longue mémoire représente un modèle à seuil, l'absence de mémoire équivaut à un modèle d'interaction indépendante).

Conclusion

La principale critique apportée à ces modèles de contagion réside dans leur approche catégorielle : un individu ne peut appartenir qu'à une seule catégorie à la fois. Or, contrairement à une infection, un humain virtuel dispose toujours d'un état émotionnel et est dont continuellement « contaminé ». La contagion émotionnelle ayant toujours lieu, chaque humain virtuel est donc à la fois contaminé et contaminant. Ces différents modèles ne sont donc pas pertinents pour modéliser la contagion des émotions.

2.2.2 Modèle ASCRIBE

BOSSE et al. proposent ASCRIBE (Agent-based Social Contagion Regarding Intentions Beliefs and Emotions), un modèle computationnel de contagion émotionnelle basé sur les modèles de dissipation thermique utilisés en physique [Bosse 09].

Présentation

À chaque pas de temps, les individus transmettent leur état émotionnel et sont impactés par celui des autres. Afin de prendre en compte les spécificités individuelles, le modèle ASCRIBE définit un ensemble de paramètres qui déterminent la capacité d'absorption et de transmission des émotions de chaque humain virtuel et qui sont repris dans le tableau 2.2.

Paramètre	Description
q_E	Intensité de l'émotion ressentie par l'émetteur
q_R	Intensité de l'émotion ressentie par le receveur
η_E	Capacité de l'émetteur à contaminer les autres humains virtuels
δ_R	Capacité du receveur à être contaminé
α_{ER}	Force du canal d'échange de l'émetteur vers le receveur

TABLE 2.2: ASCRIBE - Paramètres régulant la contagion émotionnelle d'un émetteur E vers un receveur R

À chaque pas de temps, chaque humain virtuel calcule la valeur moyenne transmise par ceux qui l'entourent et met à jour son état émotionnel. En particulier, les auteurs démontrent que l'émotion du groupe converge vers la moyenne pondérée des états émotionnels des humains virtuels. L'évolution dans le temps de l'émotion ressentie par chaque humain virtuel R, situé au sein d'un groupe d'humains virtuels G, est définie par l'équation différentielle 2.2.1. γ_R représente la force globale avec laquelle R reçoit l'émotion des humains virtuels du groupe. Elle est définie par la formule 2.2.2 et nécessite de calculer γ_{RE}, la force avec laquelle R reçoit l'émotion de chaque humain virtuel E du groupe (2.2.2). Plus le canal de communication α_{ER} entre les deux est fort, plus les émotions sont transmises intensément. Ainsi, si $\gamma_{ER} = 0$, R n'est pas impacté par l'émotion de E. Si $\gamma_{ER} = 1$, la contagion de cette émotion est à son maximum.

$$\frac{dq_R}{dt} = \gamma_R(q_R^* - q_R) \tag{2.2.1}$$

$$\gamma_R = \sum_{E \in G} \gamma_{ER} \tag{2.2.2}$$

q_R^*, définie par la formule 2.2.3, représente la moyenne de l'émotion transférée à R, pondérée par la capacité de l'émetteur E et la force du canal d'échange entre E et R (2.2.4).

$$q_R^* = \sum_{E \in G} \omega_{ER} \times q_E \tag{2.2.3}$$

$$\omega_{ER} = \frac{\eta_E \times \alpha_{ER}}{\sum_{X \in G} \eta_X \times \alpha_{XR}} \tag{2.2.4}$$

BOSSE et al. proposent de lier la capacité à influencer émotionnellement les autres d'un humain virtuel à sa capacité à externaliser et afficher l'émotion ressentie [Bosse 09]. Ainsi, l'extraversion, l'expressivité ou l'énergie de a pourraient permettre de déterminer la valeur η_a (par exemple, l'émotion propagée par un hu-

main virtuel introverti sera moins intense que s'il était extraverti). Ils proposent de lier δ_a (la capacité d'un humain virtuel a à être influencé par les émotions des autres) à sa flexibilité émotionnelle. Enfin, α_{BA} (la force du canal d'échange de l'émetteur b vers le receveur a) serait corrélée au type et à l'intensité du contact entre les deux humains virtuels, par exemple la distance qui les sépare ou le degré d'attachement. Ils ne proposent cependant pas de modèle formel permettant de déterminer ces valeurs.

Validation et résultats

Afin de valider ce modèle, Bosse et al. tentent de reproduire, à partir d'une vidéo, une situation de panique de foule et mesurent les écarts de comportement entre la foule réelle et la foule modélisée [Bosse 11]. Ils s'inspirent d'un incident survenu le 4 mai 2010 à Amsterdam : lors de la minute de silence d'un rassemblement national, le hurlement d'un individu a provoqué un mouvement de panique générale.

Cette modélisation se déroule en plusieurs étapes. Tout d'abord, comme indiqué sur la figure 2.2.3, les auteurs extraient manuellement les comportements d'une dizaine d'individus présents sur la vidéo (en l'occurrence, il s'agit de reporter, frame par frame, leurs coordonnées géographiques). Ensuite, une foule est modélisée : les comportements possibles des humains virtuels sont définis (« rester au même endroit » ou « se déplacer dans une direction ») et leurs paramètres ASCRIBE sont initialisés par défaut. Les auteurs définissent ensuite des fonctions d'évaluation afin de pouvoir comparer les différences entre le comportement des individus de la vidéo et celui des humains virtuels. Dans le cas présent, il s'agit de calculer la distance euclidienne entre la position notée sur la vidéo et celle des humains virtuels dans la simulation. Les auteurs appliquent ensuite une méthode de calibrage pour déterminer les valeurs optimales des paramètres de chaque humain virtuel en accord avec les résultats voulus.

Dans [Tsai 11a], Tsai et al. proposent de comparer les modèles de contagion émotionnelle de Durupinar [Durupinar 10] et de Bosse et al. [Bosse 09]. Pour cela, ils implémentent ces modèles de contagion émotionnelle au sein de trois foules dans le moteur de simulation d'évacuation ESCAPES [Tsai 11b] et obtiennent une foule sans contagion émotionnelle (modèle ESCAPES brut), une foule avec le modèle de contagion de Durupinar et une foule avec le modèle de contagion ASCRIBE. Les auteurs utilisent le processus de validation par vidéo expliqué ci-dessus et notent que le modèle de contagion ASCRIBE produit un résultat 14% meilleur (en terme d'écarts observés) que celle avec le modèle de contagion de Durupinar, celui-ci étant équivalent au modèle ESCAPES brut, sans contagion émotionnelle.

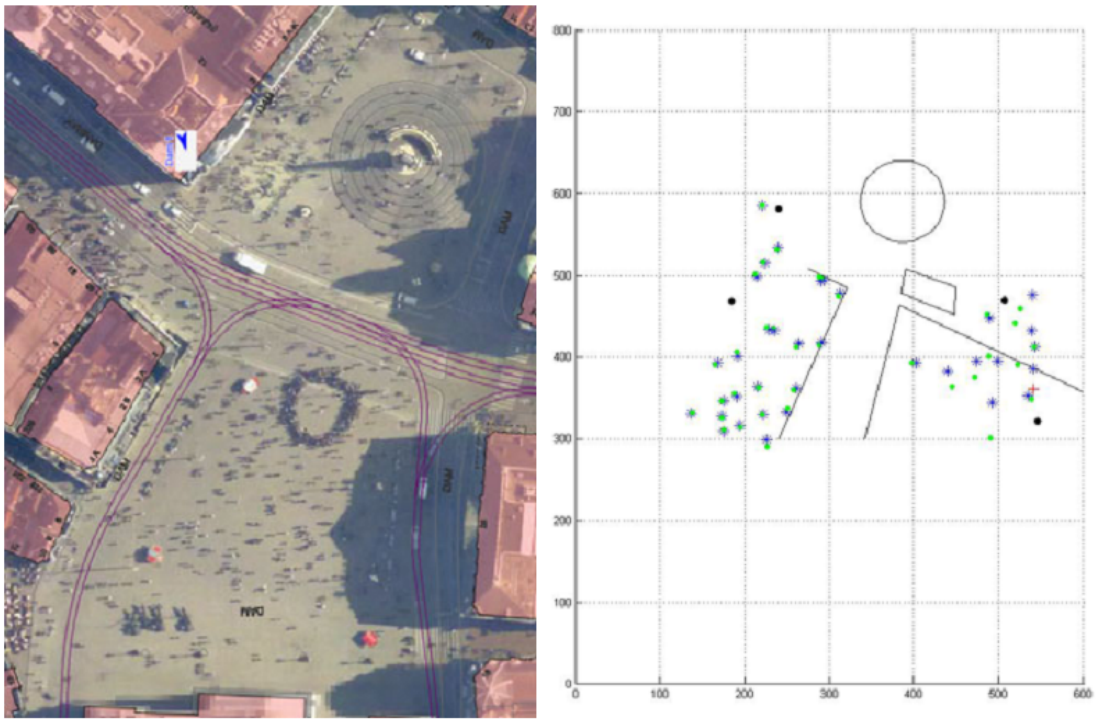

FIGURE 2.2.3: ASCRIBE - Vidéo source (à gauche) et capture d'écran de la simulation (à droite) [Bosse 11]

Critique

Bien qu'il soit compatible avec la définition de la contagion émotionnelle et semble offrir des résultats en accord avec des comportements de foule observés en situation réelle, deux principales critiques peuvent être apportées au modèle ASCRIBE.

Dans ce modèle, il est nécessaire de déterminer des distributions pour chacune des deux caractéristiques des humains virtuels (capacité à contaminer et capacité à être contaminé) ainsi que la force du canal d'échange de chaque couple. Or, ses auteurs les calibrent afin qu'ils soient en accord avec les résultats attendus. Il est donc délicat de conclure sur une supériorité du modèle ASCRIBE car son efficacité dans des cas d'utilisation autres que le rejeu d'un scénario existant et ayant servi à calibrer le modèle reste à prouver.

De plus, bien qu'ancré dans la théorie en psychologie, le modèle ASCRIBE reste flou sur la détermination des caractéristiques individuelles régissant la contagion émotionnelle, ce qui empêche d'implémenter des humains virtuels cognitifs capables de prendre des décisions complexes. En effet, il existe un lien entre les composantes individuelles (telles que la personnalité et les buts) et le comportement. Or la proposition faite par les auteurs d'ASCRIBE manque de formalisme ; comment déterminer l'extraversion, l'expressivité d'un humain virtuel ? Et quel est l'impact de ces attributs sur la prise de décision ? En toute logique, un individu extraverti aura une grande force de contagion et sera donc un leader naturel. Il imposera son état émotionnel à des humains virtuels moins extravertis qui auront tendance à adapter leur comportement pour améliorer leur propre état émotionnel, et donc par la même occasion, celui du leader. Dans le modèle existant, il n'existe pas de lien entre ces différents niveaux, ce qui limite les comportements à des quasi réflexes (l'humain virtuel a trop peur, il décide de fuir l'endroit où il se trouve) et ne permet donc pas de raisonnements cognitifs.

2.2.3 Conclusion sur les modèles de contagion émotionnelle

Les modèles de contagion émotionnelle présentés modélisent la contagion comme un processus inconscient qui permet à chaque individu d'être impacté par les émotions de ceux qui l'entourent. Ces modèles permettent de générer les « vagues » émotionnelles telles qu'elles sont observées et décrites par les psychologues et les sociologues. On notera que les résultats obtenus par le modèle ASCRIBE, dans des conditions d'exécution particulières, ont été jugés conformes à la réalité. Cependant, le modèle de l'individu qu'il utilise est ad hoc, aussi bien d'un point de vue statique (personnalité, structures émotionnelles et sociales) que dynamique (évaluation cognitive, prise de décision).

2.3 Conclusion

Dans ce chapitre, nous avons présenté les deux principales façons de modéliser les interactions émotionnelles entre humains virtuels.

Dans une première section, nous nous sommes intéressés à l'empathie cognitive, le processus cognitif qui vise à se projeter à la place d'autrui pour imaginer et ressentir ses émotions. L'empathie cognitive permet de modéliser de manière très fine les interactions émotionnelles entre les individus, mais elle nécessite qu'ils disposent d'une théorie de l'esprit leur permettant d'évaluer l'état émotionnel de chacun des autres humains virtuels. Nous avons vu avec Thespian [Si 09], une architecture qui reprend les processus d'évaluation cognitive de EMA, comment cette empathie est intégrée aux théories de l'évaluation cognitive. Cette approche nécessite de déterminer de manière explicite le niveau de projection réalisé lors de la prise de décision et peut poser des problèmes de passage à l'échelle.

La seconde partie s'intéresse à l'empathie affective (contagion émotionnelle), le processus inconscient qui fait que l'état émotionnel d'un observateur est automatiquement impacté par celui des individus qui l'entourent. Nous avons présenté les modèles de contagion émotionnelle existant dans la littérature. La majorité des modèles utilisés en épidémiologie nécessite que l'individu soit soit contaminé, soit contaminant. Un individu ressentant constamment une émotion, il est en ce sens contaminant et contaminé. Ces modèles sont donc incompatibles avec la contagion émotionnelle. Le modèle ASCRIBE [Bosse 09] est un modèle de contagion émotionnelle qui permet cette dualité. Il s'inspire des observations faites par les psychologues et sociologues en définissant pour chaque individu une susceptibilité à la contagion et un pouvoir de contagion, ainsi qu'un canal de contagion entre chaque couple d'individus. Les résultats obtenus sont cohérents au niveau macroscopique (c'est-à-dire au niveau de la foule ou du groupe). Cependant, les dimensions humaines des individus sont déterminées de manière ad hoc, et ce, aussi bien d'un point de vue statique (personnalité, structures émotionnelles et sociales) que dynamique (évaluation cognitive, prise de décision). Ce modèle n'est donc pas explicable.

Chapitre 3

Modèles d'action et d'activité

En psychologie cognitive et en sciences cognitives, un système est dit « intelligent » si ses **actions sont basées sur la manipulation de symboles** qui représentent les connaissances sur le monde et sur lui-même, et si son comportement est **motivé par des buts**. Ces connaissances lui permettent de planifier et de modifier ses actions [Hollnagel 99].

Pour pouvoir s'adapter à différents domaines, l'humain virtuel doit découpler son raisonnement des connaissances spécifiques au domaine. Il est donc nécessaire qu'il dispose d'un ensemble de processus décisionnels indépendants du domaine capables de raisonner sur des connaissances spécifiques. Ces connaissances doivent être représentées selon un formalisme qui assure leur bonne compréhension par les processus de haut niveau.

Dans ce chapitre, nous décrivons tout d'abord les formalismes généralement utilisés par les architectures cognitives pour représenter les connaissances spécifiques à un domaine d'application. L'exemple de la résolution du jeu des tours de Hanoï nous permet en particulier de montrer le couplage entre les règles d'action et les instances du monde.

Nous présentons ensuite différentes théories selon lesquelles ces règles d'action s'inscrivent ensuite dans des structures plus globales qui représentent l'activité d'un humain virtuel afin d'atteindre des buts spécifiques. Nous décrivons en particulier HAWAI-DL, un formalisme inspiré des approches socio-cognitivistes qui permet de décrire l'activité humaine pour guider le comportement d'humains virtuels.

3.1 Modèle d'actions

L'intelligence humaine se trouve dans la diversité des raisonnements applicables et dans la capacité d'adapter ces raisonnements à la situation courante. Ainsi, un humain confronté à un nouveau problème peut raisonner par déduction, par

analogie, tester différentes stratégies et apprendre de ses erreurs. Pour faire cela, l'humain fait appel à ses capacités d'abstraction, c'est-à-dire qu'il est capable de prendre du recul par rapport à la situation concrète.

> « If the organism carries a « small-scale » model of external reality and of its possible actions within its head, it is able to try out various alternatives, conclude which is the best of them, react to future situations before they arise, utilize the knowledge of past events in dealing with the present and the future, and in every way to react in a much fuller, safer, and more competent manner to the emergencies which face it. » [Craik 67, p. 61]

Un système intelligent est donc doté d'une représentation interne de l'environnement qu'il utilise pour planifier et prendre ses décisions. Par exemple, les architectures cognitives présentées précédemment (voir 1.3) utilisent une mémoire de travail pour représenter les instances de l'environnement. Les actions sont définies dans la mémoire procédurale sous la forme de règles d'action qui permettent de spécifier comment l'action affecte la mémoire de travail de l'agent.

Ces règles sont généralement présentées sous la forme *SI [conditions] ALORS [effets]*. Les *conditions* sont un ensemble de triplets qui doivent être vrais pour que l'action puisse avoir lieu et les *effets* représentent les modifications induites par l'application de l'action. Plus précisément, si les triplets des *conditions* sont présents dans la mémoire de travail, les triplets des *effets* y sont ajoutés, supprimés ou modifiés (la modification d'un triplet équivalent à la suppression de l'ancien et l'ajout d'un nouveau).

Parmi les formalismes les plus courants, nous citerons STRIPS [Fikes 72] et PDDL [Ghallab 98]. Le formalisme utilisé pour décrire les règles procédurales Soar est inspiré de STRIPS, et il existe un moteur de génération de règles procédurales Soar à partir d'un modèle PDDL [1].

Nous utiliserons l'exemple des tours de Hanoï pour décrire les différents niveaux d'abstraction proposés par ces règles d'action.

Exemple des tours de Hanoï

Le problème des tours de Hanoï est utilisé par les spécialistes de la cognition. Ce jeu a été inventé par le mathématicien français Édouard Lucas (1842-1891) et se présente sous la forme de trois piquets sur lesquels on peut enfiler des disques de diamètres différents. La photo 3.1.1 montre la configuration de départ, avec les disques empilés du plus grand en bas au plus petit en haut. Le but du jeu consiste à transférer les disques du piquet de gauche au piquet de droite, en ne déplaçant à

1. Disponible sur le site du groupe Soar de Universiy of Michigan *http ://code.google.com/p/soar/wiki/DomainGenPDDL*

chaque mouvement qu'un seul disque et en ne pouvant le poser que sur un piquet vide ou sur un disque plus grand. L'exemple de la photo comporte huit disques.

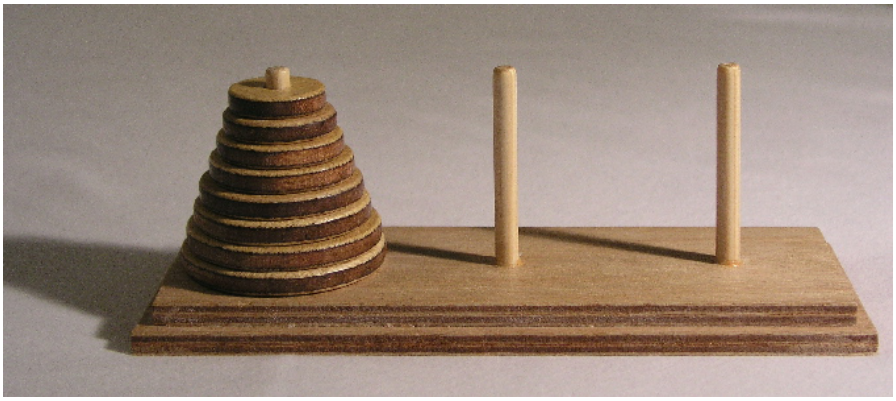

FIGURE 3.1.1: Photographie du jeu des tours de Hanoï avec 8 disques
(Ævar Arnfjörð Bjarmason, license creative commons)

Un humain sachant résoudre le problème de la tour de Hanoï avec quatre disques sera capable de le résoudre s'il en comporte cinq, voire plus. Cette résolution est cependant loin d'être triviale pour un humain virtuel. Nous laisserons de côté ici l'algorithme de résolution du problème pour nous consacrer à la représentation des connaissances, à savoir les actions et les objets de l'environnement.

3.1.1 Sans abstraction, un modèle ad hoc

Si l'humain virtuel n'est doté d'aucune capacité d'abstraction, chaque objet, chaque action, est considéré comme une entité distincte. Quelles que soient les similitudes présentées, l'humain virtuel est incapable de les associer.

Si l'on applique ceci à la résolution du jeu des tours de Hanoï, l'humain virtuel devra disposer, pour chaque disque, de 6 règles d'action. Le tableau 3.1 liste les 12 actions de mouvement nécessaires pour modéliser les connaissances nécessaires à la résolution d'un jeu comportant trois disques, chaque action étant codée de manière explicite comme le montre l'algorithme 3.1. Dans cet algorithme, nous utilisons deux éléments de notation particuliers qui sont le « + » et le « - » avant un effet de l'action. Ils signifient que l'on ajoute (respectivement que l'on enlève) un élément de connaissance.

3.1.2 L'utilisation de variables, verbosité contre complexité

Les architectures cognitives proposent une première réponse à ce problème de combinatoire en permettant de séparer le raisonnement des objets concrets sur

disque A	disque B	disque C
piquet 1 → piquet 2	piquet 1 → piquet 2	piquet 1 → piquet 2
piquet 2 → piquet 3	piquet 2 → piquet 3	piquet 2 → piquet 3
piquet 2 → piquet 1	piquet 2 → piquet 1	piquet 2 → piquet 1
piquet 2 → piquet 3	piquet 2 → piquet 3	piquet 2 → piquet 3
piquet 3 → piquet 1	piquet 3 → piquet 1	piquet 3 → piquet 1
piquet 3 → piquet 2	piquet 3 → piquet 2	piquet 3 → piquet 2

TABLE 3.1: Actions de mouvement nécessaires pour la résolution du jeu des tours de Hanoï avec 3 disques

Algorithme 3.1 Pseudo-code pour une des actions de mouvement du jeu de Hanoï sans abstraction des connaissances

```
Action-deplacer-disqueA-piquet1-piquet2
conditions:
    (disqueA, position, piquet1)
effets:
    -(disqueA, position, piquet1)
    +(disqueA, position, piquet2)
```

lesquels il s'applique.

L'abstraction consiste à utiliser dans l'écriture des actions des variables qui vont s'instancier de manière dynamique en fonction de l'état du monde sur l'ensemble de la règle. Les objets ne sont plus nommés « en dur » dans le code, mais généricisés et classés en fonction des attributs qui les définissent.

La Figure 3.1.2 représente cette classification des objets du monde sous forme de diagramme de classes [2].

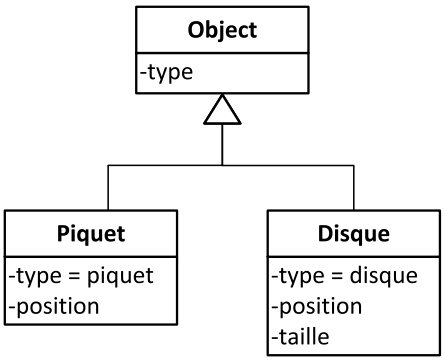

FIGURE 3.1.2: Modélisation des objets du monde pour le jeu de Hanoï

Le pseudo-code 3.2 définit une règle d'action générique qui permet de modéliser l'ensemble des actions de mouvement des tours de Hanoï, quel que soit le nombre de disques (et éventuellement le nombre de piquets, bien que le jeu ne prévoie pas un tel changement). Les variables sont notées ici entre chevrons $< >$. Ainsi, on peut déplacer n'importe quel objet ayant un attribut de nom « type » et de valeur « disque ». Cet objet devra également avoir un attribut « position » ainsi qu'un attribut « taille », avec pour chacun une valeur associée.

L'écriture de règles d'actions génériques avec variables est cependant complexe.

En effet, la généricité nécessite d'expliciter certaines contraintes qui étaient implicites dans les versions sans abstraction ; ici, on vérifie le type des objets, le fait que la destination est différente de la position initiale, et enfin le fait que s'il y a un disque à l'emplacement cible, la taille de ce dernier doit être supérieure à celle du disque que l'on déplace. En l'occurrence, on vérifie qu'il n'y a pas à l'emplacement cible de disque dont la taille est inférieure à celle de celui que l'on déplace. En effet, le formalisme utilisé par les architectures cognitives (Soar, en particulier) ne permet d'utiliser que des conjonctions dans l'écriture des règles. Cette subtilité

2. On notera ici que le nom de la classe n'a aucune sémantique, c'est la valeur de l'attribut « type » qui permet de distinguer les catégories d'objets.

Algorithme 3.2 Pseudo-code pour une action générique de mouvement du jeu de Hanoï

```
Action-déplacer-disque
conditions:
    (<disque>, type, disque)
    (<p-init>, type, piquet)
    (<p-init>, position, <p-init-pos>)
    (<p-dest>, type, piquet)
    (<p-dest>, position, <p-dest-pos>)
    (<disque>, position, <p-init-pos>)
    (<p-dest-pos>, !=, <p-init-pos>)
    NOT (
        (<disque2>, position, <p-dest-pos>)
        (<disque2>, taille, <disque2-taille>)
        (<disque>, taille, <disque-taille>)
        (<disque2-taille>, <, <disque-taille>)
    )
effets:
    -(<disque>, position, <p-init-pos>)
    +(<disque>, position, <p-dest-pos>)
```

permet ici de prendre en compte dans la même règle, le cas où il n'y a pas de disque à l'emplacement cible et le cas où il y en a un. Lorsqu'il est impossible d'obtenir une seule conjonction, il suffit d'écrire l'ensemble des conditions sous la forme normale disjonctive et de créer ensuite une règle par bloc disjonctif. On notera toutefois que lorsqu'il y a plusieurs blocs disjonctifs de conditions et d'actions, générer et écrire à la main l'ensemble des règles associées peut poser des problèmes de combinatoire.

La majorité des architectures visant à modéliser des humains virtuels autonomes utilise ce niveau d'abstraction. Cependant, la complexité d'écriture des règles d'action est généralement insurmontable pour un non-informaticien, voire même pour un informaticien non familier avec le fonctionnement des langages d'action.

3.2 Modèle d'activité

Un modèle d'activité est une représentation du comportement de l'humain qui reflète l'organisation temporelle et/ou structurelle des actions afin de réaliser un objectif de haut niveau.

On parle généralement de procédure.

> « Put most simply, a procedure is a prescription of how to do something, a specification of the active means whereby an objective may be achieved. Formally speaking, a procedure describes an ordered set of actions which enables a user to achieve a specified objective, provided that the actions are performed correctly and that the conditions otherwise are as expected. Procedures are used for all complex work situations and can be regarded as an explicit representation of the knowledge of an experienced operator. Procedures can either be mandatory such as pre-flight checklists in aviation, be prescribed guidelines for actions, or simply be a set of recommendations. » [Niwa 02]

La procédure correspond donc à une description parfaite de l'activité (enchainement d'actions) à accomplir pour atteindre un certain objectif.

Nous survolons ici les différentes approches en psychologie qui s'intéressent à la modélisation de l'activité humaine, et confronterons en particulier deux grands paradigmes, celui de la psychologie cognitive et celui du sociocognitivisme, en particulier la théorie de l'action située et la théorie de l'activité.

3.2.1 Approche cognitiviste

La théorie cognitiviste est basée sur les travaux de Ernst et Newell et en particulier la technique de résolution de problème par *means-end analysis* [Newell 59],

décrite en 1.3.1. Cette approche repose sur la notion centrale de représentation en buts et en moyens pour atteindre ces buts, ce qui permet d'obtenir un arbre qui décrit la tâche en buts et sous-buts, et associe à chaque élément de la décomposition hiérarchique les moyens disponibles pour les réaliser. L'objectif est de déterminer une structure de l'action abstraite, représentable sous forme de plans rationnels et décontextualisés, et qui permettent de prédire le comportement de l'humain, celui-ci étant vu comme un « système symbolique de traitement de l'information ».

Le modèle belief-desires-intentions

Dans le livre fondateur de la théorie des désirs-croyances-intentions (beliefs-desires-intentions ou BDI), Bratman s'intéresse à la rationalité du comportement humain [Bratman 87]. Cette théorie stipule que le comportement humain est dirigé par trois concepts principaux (*mental states*) qui sont les désirs, les croyances et les intentions. Ces concepts permettent d'expliquer la rationalité des actions. Par exemple, l'action d'allumer l'air conditionné est expliquée (partiellement) par le désir qu'on a de se rafraichir et la croyance que la climatisation est efficace pour réduire la température. Ce désir et cette croyance permettent de générer l'intention d'agir.

L'humain doit répondre à deux besoins principaux qui sont la *coordination interpersonnelle* (agencer les activités présentes et futures pour pouvoir atteindre des buts complexes) et la *coordination intrapersonnelle* (agencer ses propres activités avec celles des autres individus afin de répondre aux interactions sociales). Il dispose cependant de ressources limitées, c'est-à-dire qu'il n'est pas capable, à chaque instant, de considérer toutes les options qui se présentent avant de faire un choix. Au contraire, les décisions sont guidées par les actions passées et les intentions futures. L'humain dispose de plans larges qui lui permettent d'envisager le futur et de coordonner ses propres activités et celles des autres. *Ces plans sont partiels, assemblés et complétés à la volée.*

3.2.2 Approche sociocognitiviste

Les deux grands courants de l'approche sociocognitiviste qui s'intéressent à la représentation de l'activité sont la théorie de l'action située et la théorie de l'activité. Ils dénoncent les limites du rôle fonctionnel des plans rationnels de l'approche cognitiviste en surlignant l'importance des facteurs contextuels dans la réalisation d'une activité. En effet le plan rationnel n'est qu'une des multiples ressources en rapport avec la tâche que l'humain utilise. La situation évoluant à tout moment, l'humain ajuste donc constamment, et de manière ad hoc, son comportement. Il est donc capable de produire des plans en cours d'action, ceux-ci prenant en compte le contexte de la situation.

Cette conception de la cognition a donné naissance au courant de l'apprentissage situé [Resnick 89] qui s'intéresse à l'importance du contexte social au cours de l'apprentissage. L'acquisition d'une compétence s'effectue dans la situation où celle-ci sera utilisée. En d'autres termes, il faut que l'apprentissage s'inscrive dans un contexte pour que l'apprenant puisse lui donner un sens.

Théorie de l'action située

La théorie de l'action située émerge des travaux de Winograd et Flores, célèbres dans le domaine de la conception de systèmes sensés reproduire certaines capacités de l'intelligence humaine [3] [Winograd 87], ainsi que de ceux de Suchman [Suchman 87]. Ces auteurs s'intéressent à l'apport des sciences sociales et la prise en compte du contexte dans l'analyse de l'activité humaine, deux points ignorés par l'approche cognitiviste qu'ils remettent en question.

Ainsi, la théorie de l'action située est un courant des sciences sociales qui voit l'activité de l'humain comme un ensemble d'interactions. Elle sous-tend que toute action dépend étroitement des circonstances matérielles et sociales dans lesquelles elle a lieu et limite le rôle fonctionnel apporté par les plans rationnels et génériques.

> « J'introduis l'expression « action située » pour souligner que tout cours d'action dépend de façon essentielle de ses circonstances matérielles et sociales. Plutôt que d'essayer d'abstraire l'action de ses circonstances et de la représenter comme un plan rationnel, mieux vaut étudier comment les gens utilisent les circonstances pour effectuer une action intelligente » [Suchman 87, p. 50]
>
> « Les actions sont toujours socialement et physiquement situées, et la situation est essentielle à l'interprétation de l'action. Par situation on doit entendre un complexe de ressources et de contraintes, qui peuvent toutes le cas échéant jouer un rôle significatif sans pour autant que ce rôle soit nécessairement réductible à un jeu de représentations mentales préalablement objectivées dans les appareils cognitifs. »[Visetti 89]

Théorie de l'activité

La théorie de l'activité découle quant à elle d'études réalisées dans les années 1920 par des psychologues soviétiques révolutionnaires, Vygotsky et Leontjev [Leontjev 59]. La théorie de l'activité situe la conscience dans la pratique quotidienne et affirme que les actions sont toujours insérées dans une matrice sociale composée d'individus et d'artefacts. Leontjev [Leontjev 59] distingue trois

3. Winograd est l'un des fondateurs de l'informatique linguistique

concepts principaux : l'activité, associée à un but de haut niveau, l'action, dépendante d'une activité et d'un de ses sous-buts, et l'opération, instanciation d'une action qui prend en compte le contexte dans lequel elle se déroule.

3.2.3 HAWAI-DL : Modélisation de l'activité pour les humains virtuels

Un modèle de l'activité humaine consiste généralement en un ensemble de tâches, liées entre elles de manière temporelle et/ou hiérarchique, qui permet de déterminer une séquence d'actions à effectuer pour atteindre un but.

La plupart des langages de description de l'activité se concentrent sur les tâches prescrites, c'est-à-dire sur la séquence d'actions à réaliser pour atteindre un but de manière optimale. Ces modèles sont renseignés par des informaticiens à partir d'analyses rédigées par des experts du domaine. Si ce type d'approche permet de générer des situations d'apprentissage qui peuvent être pertinentes, elle peine à représenter la réalité du terrain. En effet, en situation réelle, il est fréquent que l'humain s'écarte volontairement de l'activité prescrite et commette une entorse à la règle, faisant ainsi des compromis entre production, délais et sécurité. Des écarts existent souvent entre les analyses des experts et les comportements réels observés sur le terrain.

HAWAI-DL (Human Activity and Work Analysis for sImulation-Description Language) est un langage de description de l'activité humaine qui utilise un formalisme inspiré des méthodologies issues de l'ergonomie cognitive, discipline qui vise à décrire l'activité réelle. HAWAI-DL offre la possibilité aux ergonomes de modéliser directement l'activité humaine en prenant en compte les observations faites sur le terrain.

Ces modèles d'activité permettent ensuite d'une part de suivre l'apprenant et de détecter ses erreurs (travaux de thèse de K. Amokrane [Amokrane 10]) et d'autre part de générer le comportement d'humains virtuels (travaux de thèse de L. Edward [Edward 11]). Nous ne donnerons ici qu'une brève description d'HAWAI-DL, une version plus complète pouvant être trouvée dans les thèses citées précédemment.

Principaux concepts

Les principaux concepts de ce langage s'inspirent des méthodologies utilisées en ergonomie cognitive pour modéliser l'activité humaine. La Figure 3.2.1 reprend les principaux concepts utilisés par ce langage.

Au centre du modèle d'activité figure la notion de *Tâche* dont l'objectif est de répondre à un *But* représenté par l'état du monde désiré après la réalisation de la tâche.

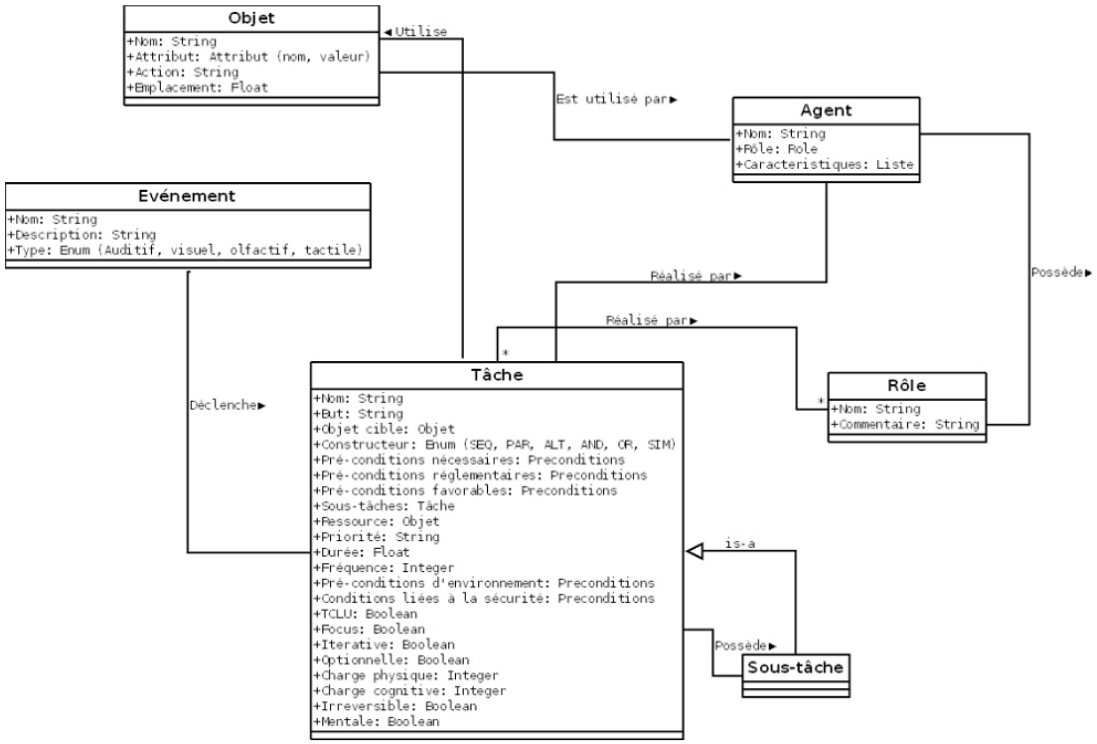

FIGURE 3.2.1: HAWAI-DL - Principaux concepts, tiré de [Edward 11]

Une tâche peut posséder une ou plusieurs sous-tâches ordonnées avec l'aide d'un constructeur qui, selon les cas, apportent des informations sur l'ordonnancement temporel des sous-tâches (*SEQ* pour séquentiel, *SIM* pour simultané, *PAR* pour parallèle) ou sur les conditions de réussite de la tâche principale (*AND* signifie que toutes les sous-tâches doivent être réalisées, *ALT* et *OR* qu'au moins l'une des sous-tâches doit être réalisée).

Un certain nombre de conditions sont attachées à la tâche. Les *pré-conditions nécessaires*, les *pré-conditions règlementaires*, les *pré-conditions favorables,* les *pré-conditions liées à l'environnement*, les *pré-conditions liées à la sécurité.* Ces *pré-conditions* déterminent l'état du monde et/ou de l'agent nécessaires pour que la réalisation ait lieu.

La tâche est également renseignée par un ensemble de caractéristiques liées à l'état de l'humain virtuel. La *charge physique* impacte le niveau de fatigue de l'humain virtuel et la *charge cognitive* son niveau d'attention.

Arbre d'activité

Les arbres d'activité peuvent être spécifiés par des ergonomes avec Visual Hawai, un outil auteur. La Figure 3.2.2 montre un arbre d'activité formalisé avec Visual Hawai.

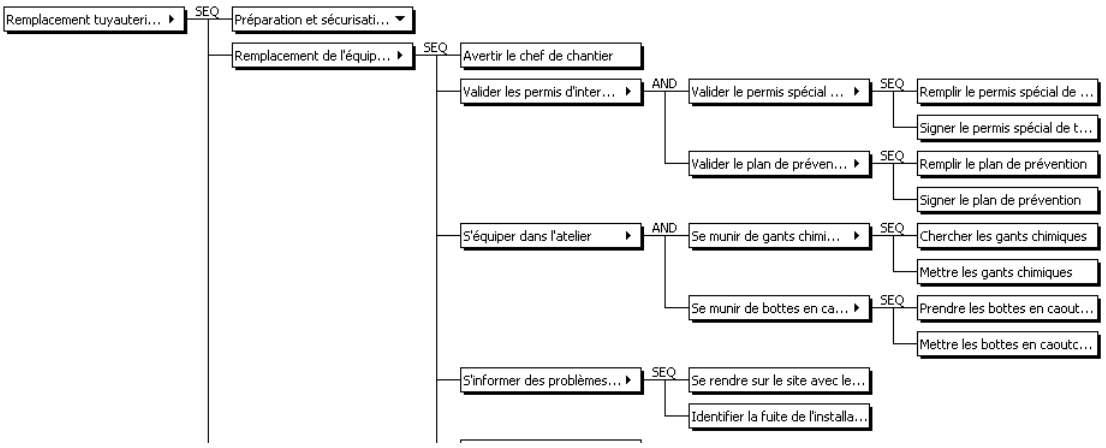

FIGURE 3.2.2: HAWAI-DL - Extrait d'un modèle d'activité

Cet exemple comporte une tâche principale, *Remplacement tuyauterie* qui est composée de deux tâches filles, *Préparation et sécurisation* et *Remplacement de l'équipement.* Ces deux tâches sont liées par un constructeur *SEQ* et elles doivent donc être réalisées séquentiellement. De la même manière, *Remplacement de l'équipement* est constitué d'un ensemble de sous-tâches réalisées séquentiellement. *S'équiper dans l'atelier* est l'une d'entre elles et consiste en deux sous-tâches, *Se munir*

de gants chimiques et *Se munir de bottes en caoutchouc*. Le constructeur associé à *S'équiper dans l'atelier* est un constructeur *AND* qui signifie que la réalisation de cette tâche dépend de celle de ses sous-tâches.

Conclusion

HAWAI-DL est un formalisme qui s'inspire des méthodologies en ergonomie cognitive et reprend un ensemble de concepts utilisés dans cette discipline. Il permet de modéliser des arbres d'activité rendant compte de procédures extrêmement complexes observées *in vivo*, et ces arbres ont été utilisés avec succès dans plusieurs applications pour la formation. La Figure 3.2.3 montre une capture d'écran du projet V3S. Ce formalisme permet, à partir d'un même modèle, de suivre l'apprenant et générer des comportements d'humain virtuel.

FIGURE 3.2.3: Capture d'écran de V3S

Plusieurs critiques peuvent cependant être apportées.

Tout d'abord, l'absence de distinction entre action et tâche pose un problème d'adaptabilité. Dans ce formalisme, l'action est modélisée de manière implicite au sein de la tâche. Elle n'existe donc que dans un contexte particulier, pour répondre à un but précis. Cette conception pose problème dès lors que l'arbre d'activité n'est pas complet, ce qui est probablement le cas étant donné qu'il est renseigné par des non-informaticiens. Par exemple, si le créateur de l'arbre modélise une tâche d'ouverture de porte mais que la porte est verrouillée et que l'arbre ne contient aucune tâche indiquant qu'il faut d'abord déverrouiller la porte, l'humain virtuel

se trouve dans une impasse : il ne dispose pas de règles d'action décontextualisées qui lui permettraient de lever cette impasse (en l'occurrence, il pourrait réaliser déverrouiller la porte avant de l'ouvrir).

HAWAI-DL offre la possibilité de spécifier des types d'objets plutôt que des instances comme cible des tâches. Cependant, les différents types de pré-conditions utilisent quant-à-elles directement des instances du monde. Le potentiel d'abstraction est donc présent mais inutilisé.

Les constructeurs qui lient les sous-tâches d'une tâche sont *ad hoc*. Ils peuvent porter sur l'ordonnancement temporel des actions ou bien sur les conditions de réussite de la tâche principale en fonction de celles des sous-tâches. Et dans ce dernier cas, le lien avec le but de la tâche principale n'est pas expliqué.

3.3 Conclusion

La représentation des connaissances a un impact majeur sur la capacité d'adaptation d'un humain virtuel. Nous avons vu que pour pouvoir s'adapter à son environnement, l'humain virtuel doit disposer de connaissances spécifiques. Nous avons en particulier distingué deux niveaux de modèles :

Le modèle d'actions décrit les actions du monde, en particulier les conditions nécessaires pour leur application ainsi que leurs effets. Nous avons vu que l'absence d'abstraction mène à un humain virtuel peu adaptable, ainsi qu'à un vaste travail d'implémentation. Les formalismes utilisés par les moteurs de planification, dont font partie les architectures cognitives, proposent d'utiliser des variables dans l'écriture de ces règles afin de permettre un premier niveau d'abstraction entre les règles et les entités de l'environnement. Une même règle peut donc s'appliquer à toutes les instances d'un type d'objet. Par exemple, une règle d'ouverture de porte pourra être utilisée quelle que soit la porte considérée. Pour pouvoir s'adapter à différents domaines, l'humain virtuel doit découpler son raisonnement des connaissances spécifiques au domaine. Il est donc nécessaire qu'il dispose d'un ensemble de processus décisionnels indépendants du domaine capables de raisonner sur des connaissances spécifiques. Ces connaissances doivent être représentées selon un formalisme qui assure leur bonne compréhension par les processus de haut niveau. Cependant, l'utilisation de variables nécessite d'expliciter certaines connaissances ce qui complexifie grandement l'écriture des règles d'action.

Le modèle d'activité permet d'inscrire les actions de l'humain virtuel dans des activités dirigées par des buts. La modélisation de l'activité peut se réaliser selon deux grands paradigmes : l'approche cognitiviste vise à abstraire l'action de son contexte afin de proposer des mécanismes génériques pouvant prédire le comportement. Cette approche ne permet cependant de modéliser que l'activité prescrite, sans prendre en compte les facteurs humains ni la réalité du terrain. L'approche

sociocognitiviste, formée par la théorie de l'action située et par la théorie de l'activité, stipule que toute action dépend étroitement des circonstances matérielles et sociales. L'action étant par définition contextualisée, il est donc impossible de l'abstraire. À l'inverse de l'approche cognitiviste qui vise à prédire le comportement, l'approche sociocognitiviste vise à décrire l'activité humaine telle qu'elle est observée.

Nous avons décrit HAWAI-DL, un langage de description de l'activité qui vise à combiner ces deux approches et permet de générer des arbres d'activité qui guident le comportement d'humains virtuels ainsi que des mécanismes de suivi de l'apprenant en utilisant directement des modèles d'activités spécifiés par des ergonomes. HAWAI-DL souffre cependant d'un certain nombre de critiques : l'absence de distinction entre tâche et action, l'absence d'abstraction dans l'écriture des préconditions ainsi que l'utilisation de constructeurs *ad hoc* pour relier les sous-tâches remettent en cause la cohérence et l'adaptabilité du comportement des humains virtuels qui utilisent ce modèle.

Proposition

You're a robot ?
I should have known...
No human being is that humane.

("Alien : Resurrection" de Jean-Pierre Jeunet, Twentieth Century Fox Film Corporation, 1997.)

Dans notre système, nous voulons modéliser des humains virtuels dont le comportement est cohérent, adaptable et explicable.

EMA est une plateforme qui vise à intégrer les émotions dans l'architecture cognitive Soar [Gratch 04]. Elle est basée sur la théorie de l'évaluation cognitive et représente les émotions de manière explicite et intègre les processus d'*appraisal* et de *coping* de manière indépendante du domaine. Les connaissances du domaine sont représentées par un graphe causal généré à partir d'une ontologie. Le comportement de l'humain virtuel EMA est cohérent et adaptable. Ils n'est cependant pas explicable car les processus d'*appraisal* et de *coping* utilisent des valeurs d'utilité qui représentent la personnalité et les buts de l'agent de manière agrégée et sont spécifiées « à la main » dans le graphe causal.

Nous proposons de reprendre les bases d'EMA et d'expliciter l'impact des buts et de la personnalité dans les connaissances spécifiques au domaine. Pour cela, nous modélisons les buts de l'humain virtuel avec des modèles d'activité. Le langage de description HAWAI-DL [Amokrane 10, Edward 11] permet de spécifier des modèles d'activité complexes et rendant compte de comportements observés, mais l'absence de distinction entre action et tâche nuit à l'adaptabilité du comportement. Nous proposons donc de séparer les connaissances spécifiques du domaine en deux modèles : le modèle d'actions et le modèle d'activité. Le premier chapitre de cette partie décrit DOMAIN-DL, notre modèle d'action, puis ACTIVITY-DL, notre modèle d'activité. Il est possible de spécifier la personnalité à travers les buts [Rizzo 97, Rousseau 96]. Nous proposons d'associer un système de marqueurs aux éléments d'ACTIVITY-DL afin de représenter cette influence.

Le second chapitre concerne le modèle de notre agent. Nous décrivons les dimensions humaines de notre agent, en particulier la personnalité OCEAN [Costa 92], ainsi qu'un ensemble de relations sociales. La personnalité pouvant être considérée comme statique, il ne se pose pas la question de sa mise à jour. Les relations sociales utilisent le modèle de Ochs & al. [Ochs 09] qui les représente de manière explicite et définit l'impact des émotions sur les relations sociales.

Nous détaillerons ensuite comment les modèles de connaissances sont intégrés dans le processus décisionnel de notre humain virtuel, en particulier leur couplage avec les processus hérités de l'évaluation cognitive (*appraisal* et *coping).*

Enfin, EMA ne prend pas en compte les interactions qui relient les humains, à fortiori les interactions émotionnelles.

Thespian tente de remédier à ce dernier point en modélisant l'empathie cognitive entre les humains virtuels [Si 09]. Ils sont dotés d'une théorie de l'esprit, c'est-à-dire la capacité à disposer d'un modèle interne des autres humains et de pouvoir évaluer leurs réactions. Cette approche nécessite cependant de déterminer de manière explicite le niveau de projection réalisé lors de la prise de décision et pose des problèmes de passage à l'échelle.

Les interactions émotionnelles peuvent également être modélisées par l'intermédiaire d'un modèle d'empathie affective, c'est-à-dire sous la forme d'un processus inconscient et automatique semblable à un phénomène de contagion. Le modèle ASCRIBE [Bosse 09] modélise la contagion émotionnelle entre des humains virtuels. Chaque individu dispose d'une valeur de susceptibilité à la contagion et d'une valeur de pouvoir de contagion, toutes deux déterminées de manière *ad hoc*.

Nous proposons d'ignorer l'empathie cognitive, qui pose des problèmes de performance et mettons en place un modèle d'empathie affective qui prend en compte les composantes individuelles de chaque individu et ce, afin de gérer les interactions émotionnelles entre les individus de manière fine.

Chapitre 4

Connaissances

Les connaissances sont une composante majeure de l'intelligence d'un humain virtuel. D'une part, elles lui permettent de s'adapter à son environnement en prenant du recul par rapport à la situation concrète en raisonnant sur des concepts. D'autre part, elles offrent un contexte rationnel aux actions, en les inscrivant dans des activités dirigées par des buts.

L'état de l'art a mis en exergue deux problèmes principaux.

Tout d'abord, nous avons vu que l'absence d'abstraction dans l'écriture des règles d'action mène à un humain virtuel peu adaptable, ainsi qu'à un vaste travail d'implémentation. Au contraire, l'utilisation de variables nécessite d'expliciter certaines connaissances ce qui complexifie grandement le couplage de l'humain virtuel avec son environnement.

Ensuite, nous avons vu que la modélisation de l'activité humaine est un domaine complexe. L'approche cognitive permet d'abstraire le raisonnement des instances concrètes, mais la dimension contextuelle et humaine est dès lors complètement ignorée. L'approche sociocognitiviste privilégie quant à elle le réalisme de la situation en mettant l'accent sur les situations observées et leurs circonstances matérielles et sociales, ce qui empêche toute abstraction permettant un raisonnement rationnel.

HAWAI-DL est un langage de description qui vise à coupler ces deux approches afin de générer des règles de comportement pour des humains virtuels ainsi que des mécanismes de suivi de l'apprenant, en utilisant directement des modèles d'activités spécifiés par des ergonomes. HAWAI-DL souffre cependant de quelques manques, que nous proposons de combler.

La principale critique concerne l'**absence de distinction entre modèle d'action et modèle d'activité.** En effet, les actions sont modélisées implicitement au sein des tâches, ce qui pose un soucis d'adaptabilité lorsque le modèle d'activité est incomplet. Une autre critique concerne l'**absence de logique dans la descrip-**

tion des constructeurs qui lient les sous-tâches d'une tâche principale. Certains d'entre eux apportent des informations temporelles alors que d'autres représentent les conditions de réussite de la tâche principale.

Pour répondre à ces problématiques, nous [1] proposons de **séparer modèle d'action et modèle d'activité**. En utilisant les concepts définis dans DOMAIN-DL pour modéliser l'activité de l'humain virtuel, on s'assure que l'activité décrite utilise des objets définis dans l'environnement et ainsi que les humains virtuels disposent des connaissances (c'est-à-dire des règles d'action) nécessaires pour suivre leur modèle d'activité. Cette distinction permet également à l'humain virtuel de disposer de connaissances en cas d'impasse. Nous proposons de **standardiser les constructeurs en utilisant une logique temporelle**, la logique temporelle d'Allen. Afin d'expliciter l'impact de la personnalité sur le comportement, nous proposons un système de marqueurs qui permettent de spécifier l'impact de la personnalité sur le comportement humain.

Nous proposons un nouveau formalisme, HUMANS-DL qui regroupe plusieurs langages de description :

- DOMAIN-DL, pour décrire les objets du monde et les actions qui s'y déroulent
- ACTIVITY-DL, pour modéliser les arbres d'activité
- CAUSALITY-DL, pour spécifier les liens de causalité entre les évènements et les probabilités associées (risques, prédiction des erreurs,...) [2]

Ces langages sont dits « de description » car ils permettent de décrire de manière générique les types d'objets présents dans l'environnement et donc de s'abstraire des instances ; en effet, on se situe ici au niveau des concepts, un niveau au dessus des instances qui peuplent effectivement le monde.

Il serait d'ailleurs plus judicieux de parler de *sous-langages*, car certains concepts sont partagés par les différents langages et ce afin d'apporter une continuité logique. Par exemple, les entités et actions de DOMAIN-DL sont utilisées pour renseigner le modèle d'activité.

Dans ce chapitre, nous décrivons tout d'abord DOMAIN-DL, le formalisme utilisé pour décrire le modèle d'actions, qui rassemble les entités et les actions spécifiques au domaine. Nous nous intéressons ensuite à ACTIVITY-DL, le langage de description utilisé pour décrire des arbres d'activité. Les connaissances issues de ces formalismes seront ensuite utilisées par les processus cognitifs des humains virtuels afin de leur permettre de s'adapter rationnellement à leur environnement (voir chapitre 5.2).

1. Correspond ici à l'équipe HUMANS, composée de D. Lourdeaux, K. Amokrane, C. Barot, K. Carpentier, V. Lanquepin et dont je fais partie.

2. CAUSALITY-DL n'est pas utilisé par les humains virtuels et est donc ignoré dans cette thèse.

4.1 Intégration des connaissances dans l'humain virtuel

Dans cette thèse, nous nous intéressons uniquement aux humains virtuels et focalisons notre étude à la génération de comportements à partir des modèles de domaine et de modèles d'activité. La figure 4.1.1 représente le couplage entre les connaissances spécifiées par les ergonomes et notre humain virtuel.

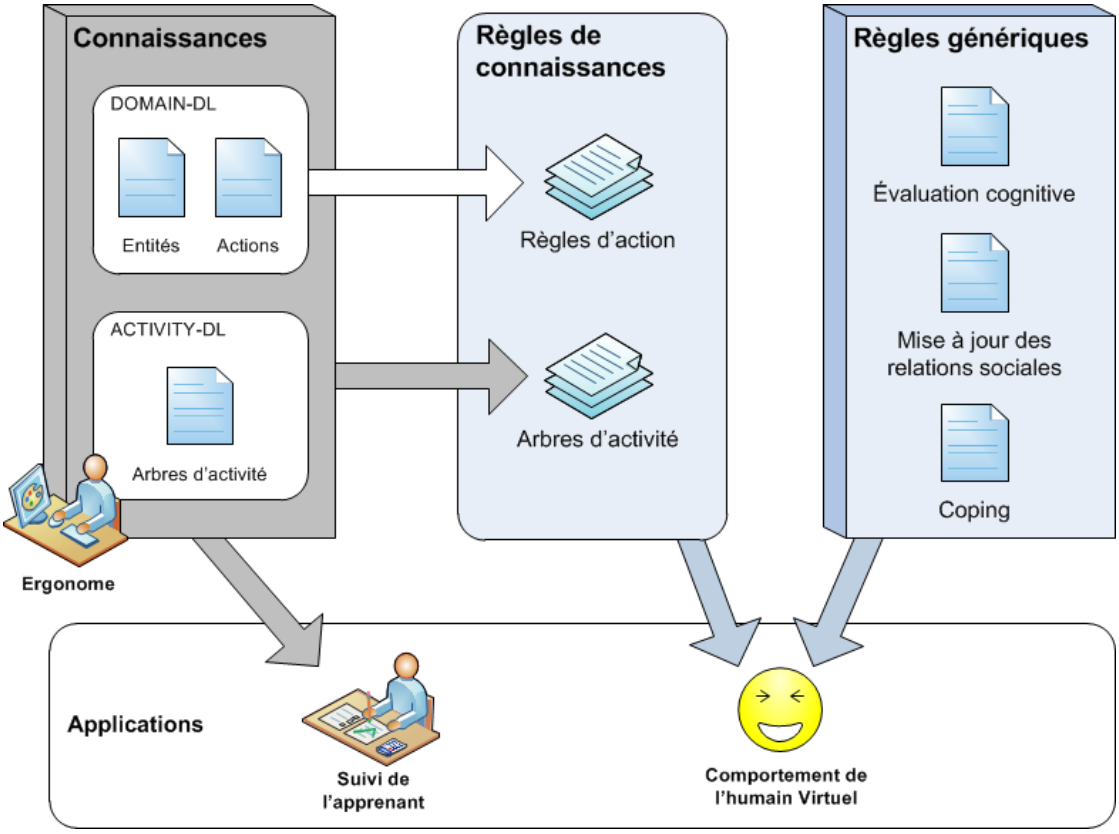

FIGURE 4.1.1: Modèles de connaissances

Le bloc de gauche, *Connaissances*, correspond aux connaissances décrites par les ergonomes en utilisant les deux langages de description, DOMAIN-DL et ACTIVITY-DL.

Le comportement de l'humain virtuel utilise deux groupes de règles :

- Les *Règles génériques* régulent les étapes de son cycle décisionnel. Ce cycle et les règles génériques sont définis dans le prochain chapitre (voir 5.2).
- Les *Règles de connaissances* sont générées de manière automatique à partir des connaissances spécifiées par l'ergonome. Ce groupe est composé des règles

d'action qui déterminent les conditions et les effets des actions, ainsi que de règles qui permettent à l'humain virtuel de construire un (ou plusieurs) arbre(s) d'activité. Les règles d'action sont générées à partir du modèle du monde utilisant le formalisme DOMAIN-DL, tandis que les règles d'activité utilisent à la fois le modèle du monde et l'arbre d'activité afin d'assurer une continuité entre les deux modèles.

4.2 DOMAIN-DL, le modèle d'action

DOMAIN-DL est le langage de description qui vise à décrire les objets du monde ainsi que les actions qui peuvent s'y dérouler. Pour plus de clarté, La version présentée ici est restreinte à l'utilisation qu'en fait un humain virtuel, la version complète permettant également le suivi de l'apprenant ainsi que la scénarisation pédagogique.

4.2.1 Principaux concepts

Ce formalisme est composé de trois concepts principaux comme le montre la moitié supérieure de la figure 4.2.1 :

- *Object :* représente un type d'entité du monde. Il peut s'agit d'objets physiques (une porte, un humain virtuel) ou de représentations plus ou moins abstraites (un feu, un bruit).
- *State :* décrit les attributs des entités (ouvert, fermé, verrouillé, etc.). Un état est associé à un objet par l'intermédiaire d'un *Statement*.
- *Action :* permet de spécifier les conditions et les effets des actions réalisables dans l'environnement. Chaque action est déterminée par une *Condition* et un *Effe*ct, chacun d'entre eux pointant sur un groupe de *Statements*. Les *Statements* de la *Condition* représentent les couples d'objet/état nécessaires pour pouvoir réaliser l'action, et ceux associés à *Effect* sont les conséquences de l'application de l'action.

Sur cette figure, les flèches pleines indiquent un héritage, tandis que les flèches en pointillés représentent signalent une relation d'association.

4.2.2 Exemple

4.2.2.1 Modèle du monde

L'exemple utilisé ici concerne la modélisation d'une action *Open*. Une telle action peut s'appliquer sur n'importe quel type d'objet, pourvu qu'il ne soit pas préalablement ouvert ; son effet est d'ouvrir l'objet en question. La partie inférieure de

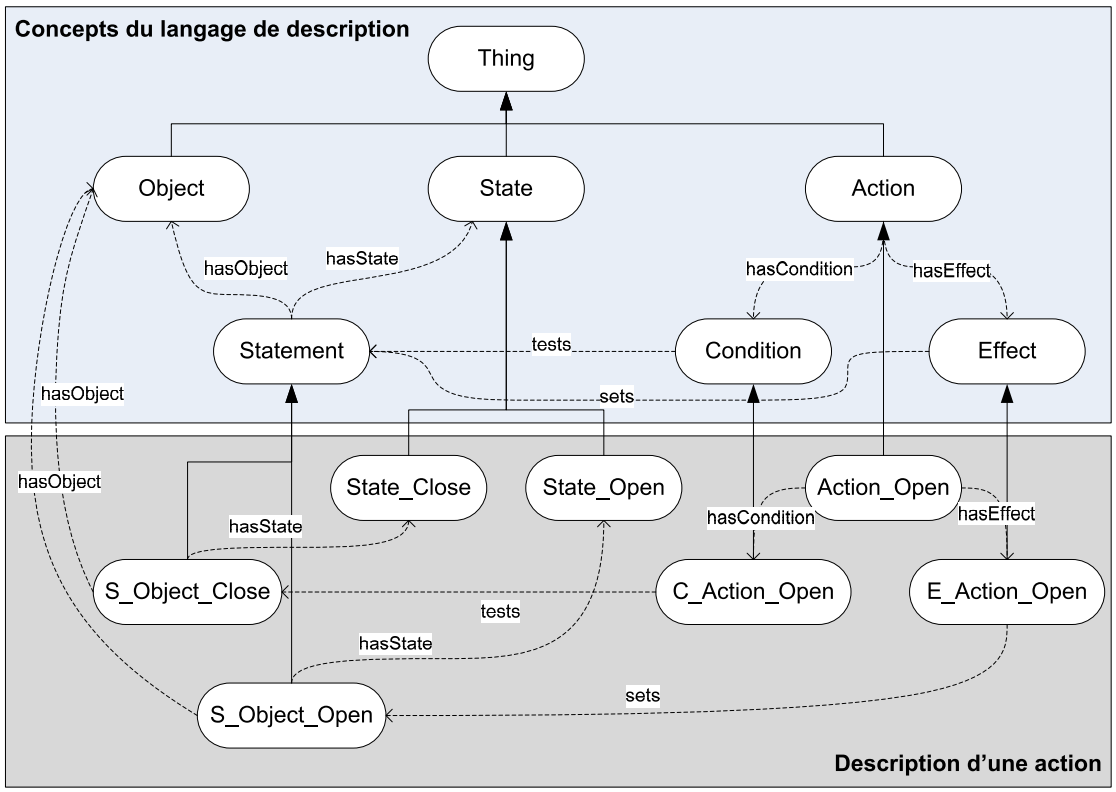

FIGURE 4.2.1: Concepts principaux de DOMAIN-DL (en haut) et extrait d'un modèle du monde (en bas)

la figure 4.2.1 représente comment cette modélisation est réalisée en utilisant le langage de description DOMAIN-DL.

La condition de cette action teste l'existence d'un objet dont l'état est fermé (*statement S_Object_Close*).

Son unique effet est de modifier l'état de l'objet concerné (*statement S_Object_Open*).

Cette règle d'action est générique car elle ne teste pas le type de l'objet. N'importe quel objet étant dans un état *Close* pourra être ouvert.

4.2.2.2 Règle d'action

La règle d'action correspondante est indiquée par l'algorithme 4.1. Les *Statements* sont transformés en prédicats qui forment les conditions et les effets de la règle d'action.

Algorithme 4.1 Règle d'action générique générée à partir de DOMAIN-DL

```
Action-Open
conditions:
    (<obj>, type, Object)
    (<obj>, hasState, State_Close)
effets:
    -(<obj>, hasState, State_Close)
    +(<obj>, hasState, State_Open)
```

Cette action, définie sous forme de concept dans DOMAIN-DL, est proposée de manière dynamique dans l'environnement si les conditions sont valides. C'est le processus décisionnel qui est chargé de sélectionner, parmi toutes les actions proposées, celle qu'il est pertinent de réaliser.

4.2.2.3 Application

La figure 4.2.2 illustre l'application de la règle d'action sur une instance *#Door_1* qui hérite du concept *Door*. A l'instant t_0, #Door1 est dans l'état fermé et ceci est indiqué par l'existence d'une relation *hasState* qui lie l'instance *#Door1* à une instance *#State_Close* du concept *State_Close*. Les conditions de la règle d'action définie par l'algorithme 4.1 sont donc valides. La règle d'action *Open* peut donc s'appliquer. La seconde moitié de la figure 4.2.2 représente l'état du monde après l'application de la règle et les changements sont indiqués en rouge. On en note deux : le premier est lié à la première ligne des effets de l'action et correspond à la suppression du premier prédicat. Le second ajoute un nouveau prédicat et lie *#Door_1* à l'état *#State_Open* avec une relation *hasState*.

On remarque que d'un point de vue dynamique, les *Statements* sont transparents. Ils permettent de modéliser des relations ternaires *Sujet, Relation*, *Valeur* dans l'ontologie, modélisation qui est rendue implicite avec l'utilisation de prédicats.

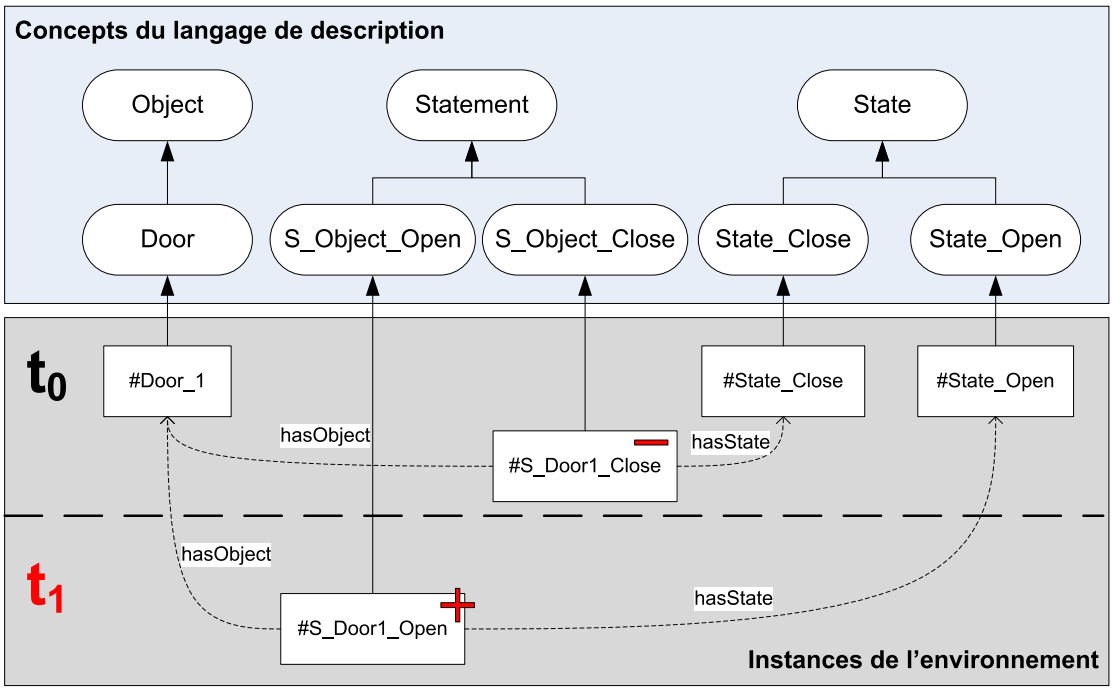

FIGURE 4.2.2: Exemple d'application de la règle d'action décrite par l'algorithme 4.1 sur une instance de l'environnement

4.3 ACTIVITY-DL, le modèle d'activité

ACTIVITY-DL est le langage de description de l'activité qui reprend et le positionnement de HAWAI-DL (présenté en 3.2.3). Comme pour la description de DOMAIN-DL, la version présentée ici est une restreinte et centrée sur l'utilisation qu'en font les humains virtuels, la version complète permettant également le suivi de l'apprenant ainsi que la scénarisation pédagogique.

4.3.1 Principaux concepts

Nous visons à décrire l'activité telle qu'elle est observée en situation réelle et nous nous inspirons des méthodologie en ergonomie cognitive. Notre formalisme utilise

donc un certain nombre de composantes particulières, essentiellement pour représenter les conditions qui poussent l'humain à effectuer une action plutôt qu'une autre. La figure 4.3.1 présente les principaux concepts proposés par ACTIVITY-DL pour décrire l'activité humaine.

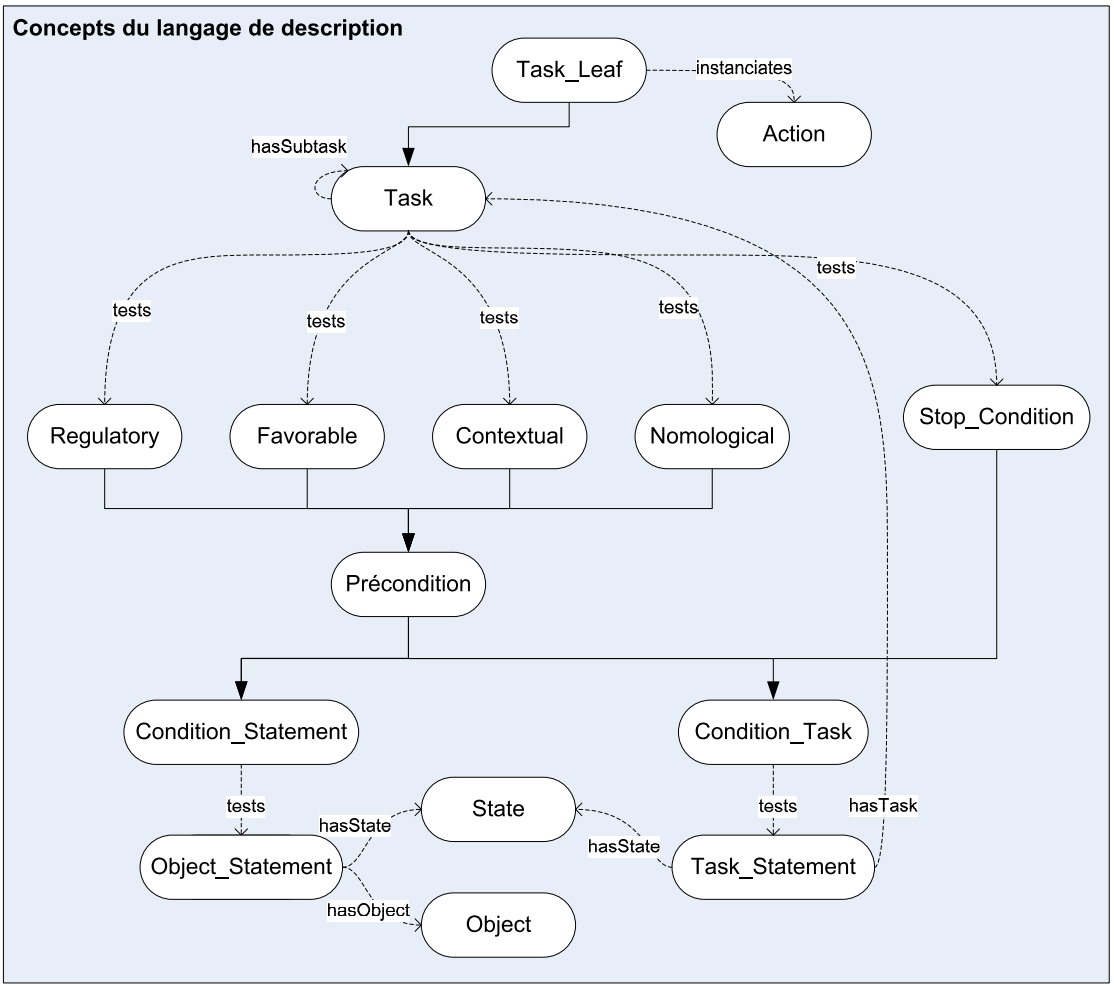

FIGURE 4.3.1: ACTIVITY-DL - Concepts principaux

4.3.1.1 Tâches

Un modèle d'activité se présente sous forme d'arbre hiérarchique, composé de tâches (*Task*s).

L'organisation de ces tâches se fait à trois niveaux :

- hiérarchique : une tâche peut être composée de plusieurs sous-tâches. On parle alors de tâche mère et de ses tâches filles,

- temporelle : un constructeur est associé à chaque tâche mère pour spécifier l'ordonnancement temporel de ses tâches filles,
- logique : la condition d'arrêt d'une tâche mère peut être décrite en fonction de l'état de ses taches filles associés à des opérateurs logiques (AND, OR, XOR). Il est ainsi possible de spécifier que les taches filles sont différentes alternatives pour réaliser la tache mère, que la réussite de chacune d'entre-elles est nécessaire, ou d'autres compositions plus complexes.

Une tâche feuille (*Task_Leaf)* est située au niveau le plus bas de l'arbre et représente une action que l'humain virtuel entreprend dans l'environnement.

Constructeurs

Le constructeur associé à une tâche mère permet de définir la relation temporelle qui relie ses taches filles. Ces relations temporelles sont représentées à l'aide de relations d'Allen, rappelées dans le tableau 4.1.

Afin de faciliter la saisie et d'améliorer la lisibilité du modèle d'activité, ACTIVITY-DL propose un ensemble de constructeurs prédéfinis correspondant à des agrégats de relations d'Allen couramment utilisées. Ces constructeurs sont référencés dans le tableau 4.2. Il reste toutefois possible d'exprimer l'ordonnancement temporel en utilisant directement les opérateurs de relation temporelle d'Allen.

Relations	**Illustration**	**Description**
$X < Y,\ Y > X$	X Y	X a lieu avant Y
$X\, m\, Y,\ Y\, mi\, X$	X Y	Y commence quand X finit
$X\, o\, Y,\ Y\, oi\, X$	X Y	X et Y se superposent
$X\, s\, Y,\ Y\, si\, X$	X Y	X a lieu au début de Y
$X\, d\, Y,\ Y\, di\, X$	X Y	X a lieu pendant Y
$X\, f\, Y,\ Y\, fi\, X$	X Y	X a lieu à la fin de Y
$X = Y$	X Y	X et Y ont lieu en même temps

Table 4.1: Relations temporelles d'Allen

Constructeur	Relations	Description
IND	$X\,\{<, >, m, mi, o, oi, s, si, d, di, f, fi, =\}\,Y$	Tâches indépendantes n'ayant aucune contrainte
SEQ	$X\,\{<, >, m, mi\,\}\,Y$	Tâches n'ayant pas d'intervalle de temps commun, s'exécutant les unes à la suite des autres, sans ordre spécifique.
SEQ-ORD	$X\,\{<, m\,\}\,Y$	Tâches n'ayant pas d'intervalle de temps commun, s'exécutant les unes à la suite des autres, dans un ordre déterminé.
PAR	$X\,\{o, oi, s, si, d, di, f, fi, =\}\,Y$	Tâches partageant au moins un intervalle de temps.
PAR-SIM	$X\,\{=\}\,Y$	Tâches simultanées.
PAR-START	$X\,\{s, si, =\}\,Y$	Tâches débutant en même temps.
PAR-END	$X\,\{f, fi, =\}\,Y$	Tâches se terminant en même temps.

Table 4.2: Constructeurs temporels d'ACTIVITY-DL

4.3.1.2 Conditions

Les conditions sont constituées d'un ensemble d'assertions représentées sous forme de triplets, liées par des conjonctions ou des disjonctions. Ces assertions (*statements*) peuvent porter sur deux types d'information : les *Object Statements* sont relatifs à l'état d'objets du monde, tandis que les *Task Statements* concernent l'état des tâches de l'arbre d'activité.

Plusieurs types de conditions peuvent être associées à une tâche, chacun ayant une sémantique particulière qui fait sens pour les ergonomes.

Conditions nomologiques

Par exemple : *il est nécessaire qu'une porte ne soit pas déjà ouverte pour qu'on puisse l'ouvrir.*

La nomologie correspond à l'étude des phénomènes physiques rationnels. « Nomological constraints are constraints dictated by natural law »[Goel 92]

Par extension, nous définissons les conditions nomologiques comme étant les configurations du monde nécessaires pour qu'une tâche soit physiquement réalisable.

Conditions contextuelles

Par exemple : *si le tuyau est sale, il est utile de nettoyer avant de le démonter.*
Les conditions contextuelles permettent de représenter les configurations qui font sens pour l'exécution d'une tâche. Dans cet exemple, si le tuyau est sale, il sera nettoyé (création d'une tâche implicite) avant d'être démonté ; sinon, la tâche peut être ignorée.

Conditions favorables

Par exemple : *s'il fait beau, il est préférable d'aller à la piscine.*
Les conditions favorables permettent d'expliciter les paramètres qui permettent de choisir une tâche plutôt qu'une autre lorsqu'elles sont en conflit.

Conditions règlementaires

Par exemple : *il faut avoir les mains propres pour pouvoir cuisiner.*
Les conditions réglementaires sont des conditions contextuelles liées à la procédure.

Conditions d'arrêt

Les conditions d'arrêt permettent de déterminer lorsqu'une tâche doit être arrêtée. Dans ce contexte, les statements relatifs aux états d'objets permettent de déterminer les états du monde à atteindre afin de déterminer que la tâche est terminée. Les statements liés aux états des tâches évaluent l'état de la tâche en cours (durée écoulée depuis le début de la tâche, nombre d'itération effectuées) ou celui de ses sous-tâches.

4.3.2 Marqueurs personnalisés

Les travaux de Rousseau et Hayes-Roth [Rousseau 96], ainsi que ceux de Rizzo et al. [Rizzo 97] stipulent que le traits de personnalité d'un individu s'expriment sous la forme d'influence sur les actions et les plans, et que la personnalité d'un individu peut être décrite en terme de ses buts et plans. L'interprétation de ce postulat donne généralement lieu à une intégration implicite de l'influence de la personnalité sous forme de valeur d'utilité, comme c'est le cas dans EMA.

Afin d'intégrer cette influence de manière explicite, nous proposons un ensemble de marqueurs de personnalités qui, associés aux tâches et aux conditions, permettent à l'ergonome de spécifier les points d'intérêt d'un individu.

Ces marqueurs personnalisés peuvent être définis selon les besoins de l'application. Ce moyen permet de modéliser l'impact de la personnalité sur la prise de décision.

Une discussion sur ce mécanisme pourra être trouvée en conclusion.

4.4 Couplage entre DOMAIN-DL et ACTIVITY-DL

Afin d'assurer une continuité logique entre les entités et les actions de l'environnement et les modèles d'activité des humains virtuels, nous proposons de coupler ACTIVITY-DL à DOMAIN-DL. La figure 4.4.1 montre comment ce lien est réalisé : les concepts en blancs sont issus d'ACTIVITY-DL, ceux en noir de DOMAIN-DL et les concepts grisés sont communs aux deux langages.

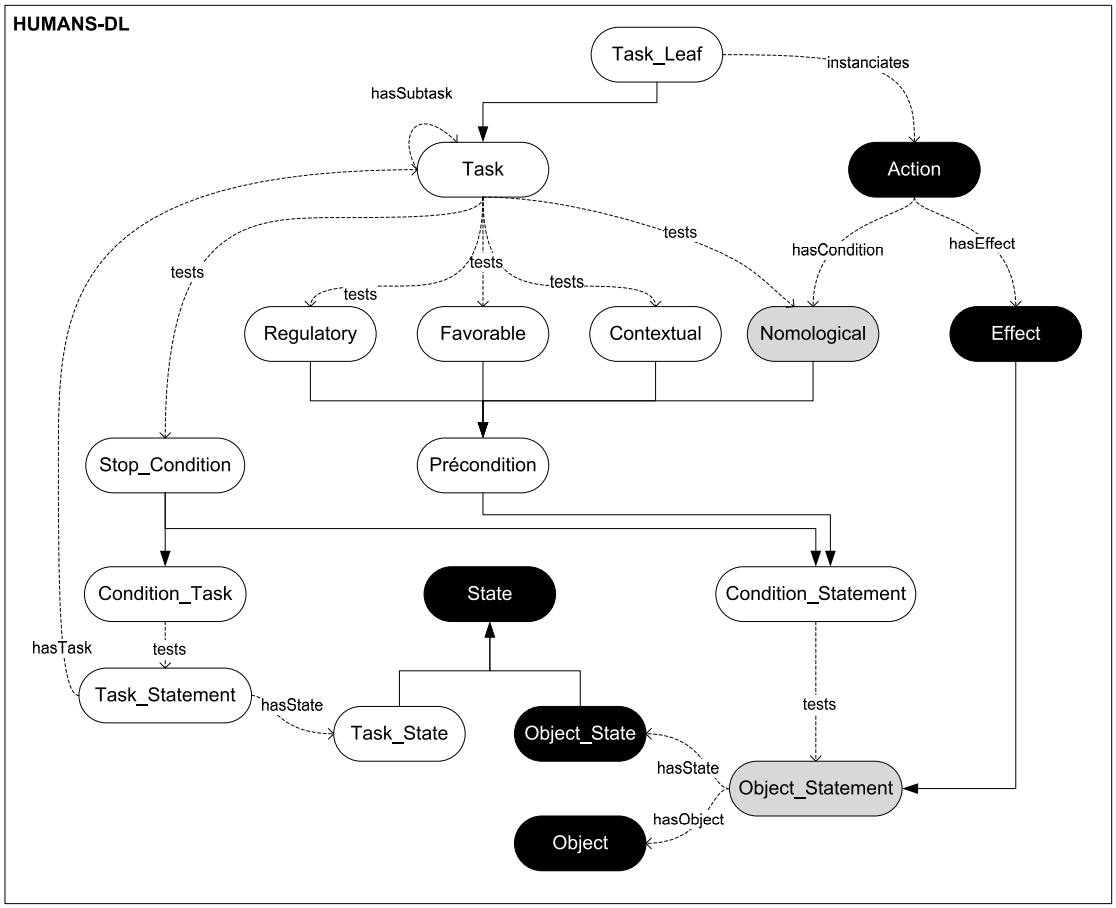

FIGURE 4.4.1: HUMANS-DL - Concepts communs à DOMAIN-DL et ACTIVITY-DL

Les conditions nomologiques d'une activité sont constituées des conditions de réalisation de l'action associée. Il s'agit en effet des conditions sur le monde nécessaires à la réalisation de la tâche. Les concepts communs peuvent être spécifiés dans DOMAIN-DL, puis surchargés dans ACTIVITY-DL. Par exemple, le modèle du monde permet de définir les conditions d'exécution d'une action ; une tâche instanciant cette action héritera automatiquement de ces conditions (en tant que conditions nomologiques), et il sera éventuellement possible pour l'ergonome d'ajouter de nouvelles conditions d'exécution à cette tâche sans en modifier l'action.

4.5 Arbre d'activité

Afin d'assurer l'adaptation de l'humain virtuel à son environnement et d'inscrire son comportement dans un contexte dirigé par des buts, il dispose d'un arbre d'activité généré par le modèle d'activité (voir 4.3). Cet arbre est modélisé dans sa mémoire de travail sous la forme de triplets. La Figure 4.5.2 représente la structure de l'arbre ACTIVITY-DL tel qu'il est traduit [3], cette modélisation étant ici volontairement simplifiée afin de pouvoir être représentée sous la forme d'un graphe sémantique.

La Figure 4.5.2 détaille la structure commune à tous les types de conditions. L'attribut *met* permet d'indiquer si la condition est vérifiée (valeur **yes**) ou non (valeur **no**). Un ensemble de paramètres permet de spécifier les conditions de validation de la condition. Chaque paramètre est composé de trois valeurs représentant un triplet (*Object*, *predicate*, *state*).

Un *Object_Statement* de DOMAIN-DL définit ces trois valeurs et sa correspondance avec cette structure est directe, tandis que les *Task_Statement* ne précisent que deux valeurs. Pour ce dernier cas, le *predicate* a automatiquement pour valeur *state.*

L'attribut *utility* associé à chaque tâche et condition est un attribut dynamique qui est mis à jour à chaque changement de situation. Cette valeur permet de renseigner l'apport de la réalisation de la tâche ou de la validation de la condition par la réalisation de la tâche.

4.6 Conclusion

Les langages de description DOMAIN-DL et ACTIVITY-DL, ainsi que leur couplage, nous permettent de répondre aux critiques apportées à HAWAI-DL et assurent donc une plus grande liberté et réutilisabilité. Ces formalismes sont également suffisamment expressifs pour pouvoir générer automatiquement des connais-

3. Le processus de traduction est présenté en 7.3.

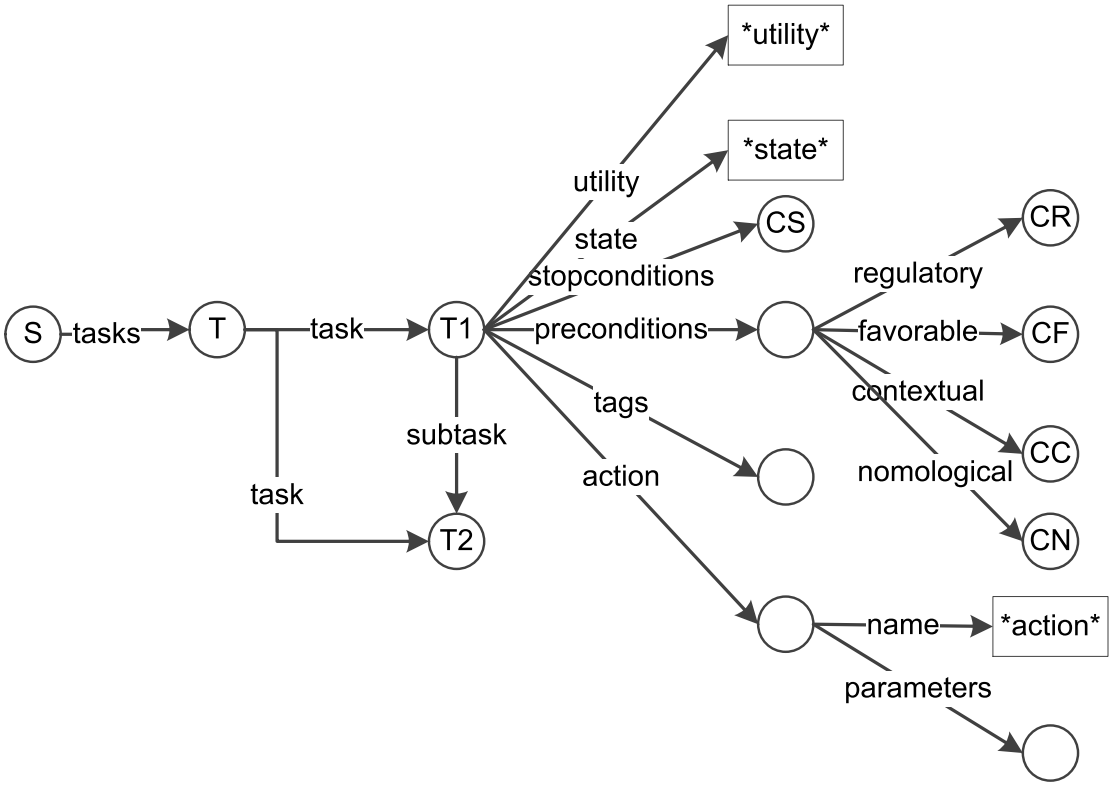

FIGURE 4.5.1: ACTIVITY-DL - Modèle d'arbre d'activité

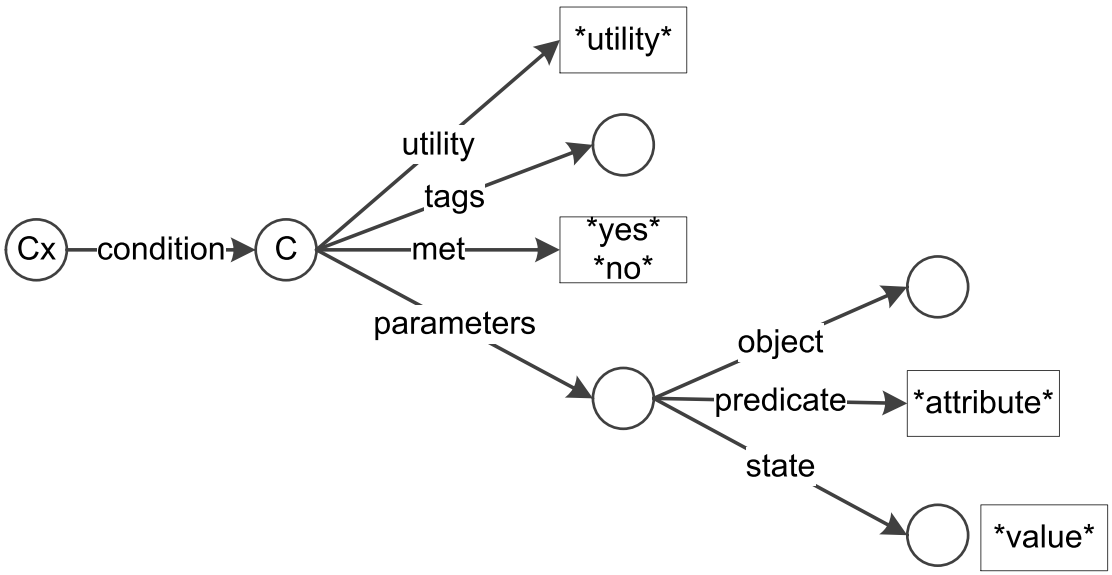

FIGURE 4.5.2: ACTIVITY-DL - Modèle des conditions de l'arbre d'activité

sances directement utilisables par les humains virtuels, à travers les règles d'action et les arbres d'activité, garantissant leur adaptabilité à l'environnement spécifié via ces formalismes.

DOMAIN-DL permet ainsi de générer un ensemble de règles d'action sous la forme de règles procédurales qui permettront de proposer des actions dans l'environnement. Le modèle d'activité formalisé avec ACTIVITY-DL permet de générer des connaissances dans la mémoire de travail sous la forme d'un arbre d'activité qui évolue de manière dynamique et guide le comportement de l'agent. Cette dynamique est expliquée au chapitre suivant.

Chapitre 5

Humain virtuel

Dans la lignée des chercheurs en Intelligence Artificielle, nous souhaitons modéliser des humains virtuels dont le comportement est cognitif, émotionnel et social.

Nous avons vu à travers l'état de l'art présenté dans la partie précédente qu'il est possible de créer des humains virtuels cognitifs et émotionnels (EMA). Cependant, la spécification des différences entre les individus (personnalité, buts) est représentée de manière agrégée et implicite par l'association d'une valeur d'utilité à chacune des situations spécifique du domaine. De plus, l'objectif étant de modéliser des humains sociaux, il est indispensable que ceux-ci soient liés par des relations sociales.

Nous proposons d'expliciter l'impact de la personnalité, des relations sociales et des buts sur la prise de décision de l'humain virtuel. Pour cela, nous reprenons le socle fourni par les théories de l'évaluation cognitive, en particulier l'interprétation qu'en fait EMA, et dotons notre humain virtuel d'une personnalité, de relations sociales et d'un arbre d'activité. Les modèles utilisés seront présentés dans la première partie de ce chapitre.

Nous détaillerons ensuite les processus dynamiques qui entrent en jeu afin d'intégrer ces dimensions humaines, en particulier leur couplage avec les processus d'évaluation cognitive qui permet de générer un état émotionnel et le *coping* qui permet d'évaluer différentes stratégies avant de prendre une décision.

5.1 Dimensions humaines

5.1.1 Personnalité

Le modèle OCEAN, décrit en détail en 1.1.2, permet de décrire la personnalité d'un individu par l'intermédiaire d'un vecteur de 5 valeurs [1] [Costa 92]. Nous avons choisi d'utiliser ce modèle pour décrire la personnalité de notre humain virtuel car il est couramment utilisé par les psychologue.

La personnalité P_i d'un humain i est définie par le vecteur 5.1.1. La valeur de chaque trait est comprise dans l'intervalle$[-1; 1]$,1 représentant une grande valeur de ce trait, -1 une faible valeur, et 0 une valeur neutre. Par exemple, un humain virtuel ayant une valeur d'extraversion entre -1 et 0 sera considéré comme introverti, alors qu'une valeur entre 0 et 1 signifie qu'il sera plutôt extraverti.

$$P_i = \begin{vmatrix} P_O \\ P_C \\ P_E \\ P_A \\ P_N \end{vmatrix} \in [-1, 1]^5 \tag{5.1.1}$$

La théorie des traits stipule que les différents traits sont indépendants entre eux, il est donc possible de générer les différentes personnalités d'une population en utilisant une distribution gaussienne [McCrae 03].

5.1.2 Relations sociales

Nous souhaitons modéliser des humains virtuels qui rendent compte de processus cognitifs émotionnels et sociaux. Après avoir analysé la littérature, nous avons retenu le modèle de Ochs [Ochs 09] qui reprend et affine le modèle dimensionnel des relations interpersonnelles de [Svennevig 99]. Ce modèle utilise les dimensions suivantes pour représenter la relation sociale d'un humain virtuel a envers un humain virtuel b.

- l'*appréciation* que a a de b ;
- la *dominance*, ou pouvoir que a peut exercer sur b ;

1. Il aurait également été possible de modéliser les facettes de chacun des traits de personnalité. On aurait alors obtenu un vecteur de 30 valeurs. Ne souhaitant pas entrer à un niveau de détail aussi fin, nous modéliserons uniquement les cinq traits principaux. L'ajustement du modèle pour prendre en compte ces facettes serait cependant direct.

On pourrait également utiliser tout type de modèle de personnalité, du moment qu'il s'agit d'un modèle en traits. Par exemple, le modèle d'Eysenck en 3 traits [Eysenck 76] est compatible avec notre approche.

- la *solidarité*, c'est-à-dire le degré de similarité entre a et b, en termes de buts, croyances et valeurs ;
- la *familiarité* qui caractérise le type et la quantité d'informations échangées entre a et b.

Nous utilisons ce modèle pour représenter les relations sociales entre nos humains virtuels.

Du point de vue d'un individu a, la relation sociale entre deux individus a (lui-même) et b est définie par le vecteur 5.1.2. Cette relation n'est pas symétrique, c'est-à-dire qu'il est possible que $R_{a/b} \neq R_{b/a}$, autrement dit que la relation qui lie a à b ne soit pas identique à celle qui lie b à a.

$$R_{a/b} = \begin{bmatrix} Appreciation_{a/b} \\ Dominance_{a/b} \\ Solidarite_{a/b} \\ Familiarite_{a/b} \end{bmatrix} \in \left[-1; 1\right] \tag{5.1.2}$$

5.2 Cycle de l'humain virtuel

Le cycle de nos humains virtuels est basé sur le cycle classique des agents cognitifs : dans un premier temps, ils perçoivent leur environnement et ce qu'il s'y déroule, ils prennent une décision en fonction de leurs caractéristiques internes et enfin réalisent une action dans l'environnement (Figure 5.2.1 à gauche).

Afin de doter nos humains virtuels d'émotions, nous avons choisi d'intégrer les différentes étapes définies par les théories de l'évaluation cognitive, et nous reprenons en particulier les travaux de Smith et Lazarus [Smith 91] (Figure 5.2.1 à droite).

Le cycle de notre humain virtuel, représenté par la figure 5.2.2 est donc une combinaison entre le cycle classique des agents cognitifs et celui défini par les théories de l'évaluation cognitive.

L'intégralité des étapes qui interviennent dans le cycle de l'humain virtuel correspond à des processus cognitifs, qui sont donc encapsulés dans le « cerveau » de l'humain virtuel. Nous utilisons l'architecture cognitive Soar [Laird 87] (décrite en détail en 1.3.2) pour modéliser les processus cognitifs de chaque humain virtuel.

5.2.1 Perception et construction de la situation

D'après la description qu'en donnent les théories de l'évaluation cognitive, la perception et la construction de la situation se déroulent de manière simultanée au cours d'un processus subjectif qui dépend de l'état mental dans lequel se trouve l'humain virtuel. De nombreux travaux mettent l'accent sur la modélisation de

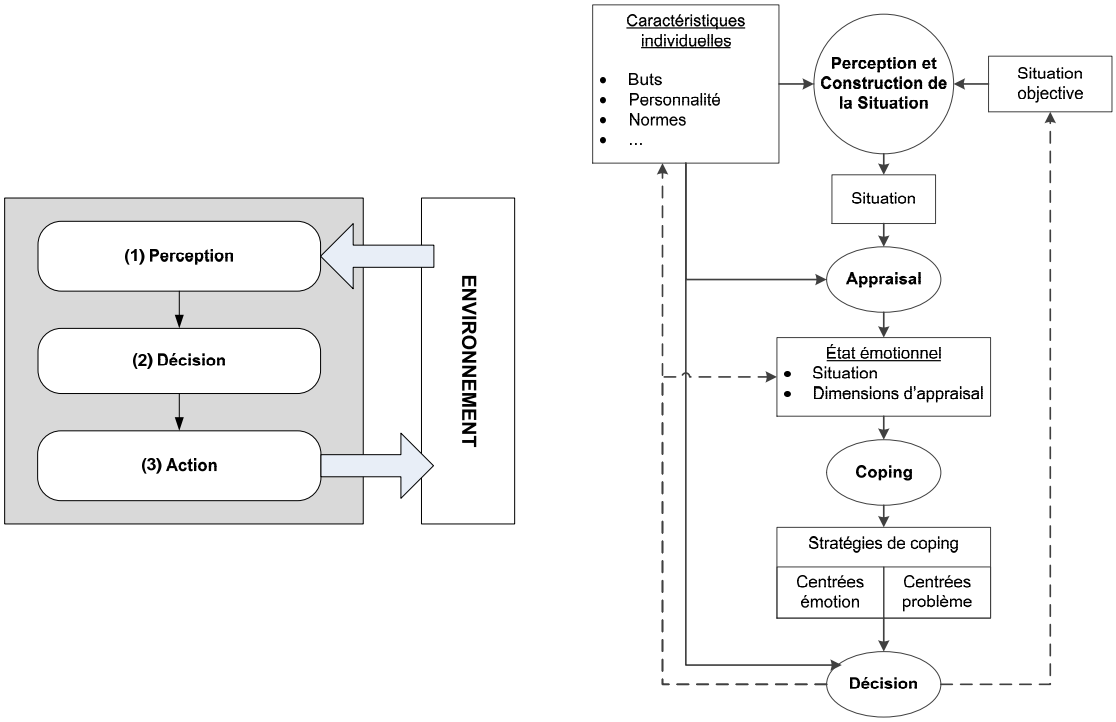

FIGURE 5.2.1: Cycle classique des agents cognitifs (à gauche), cycle défini par les théories de l'évaluation cognitive (à droite)

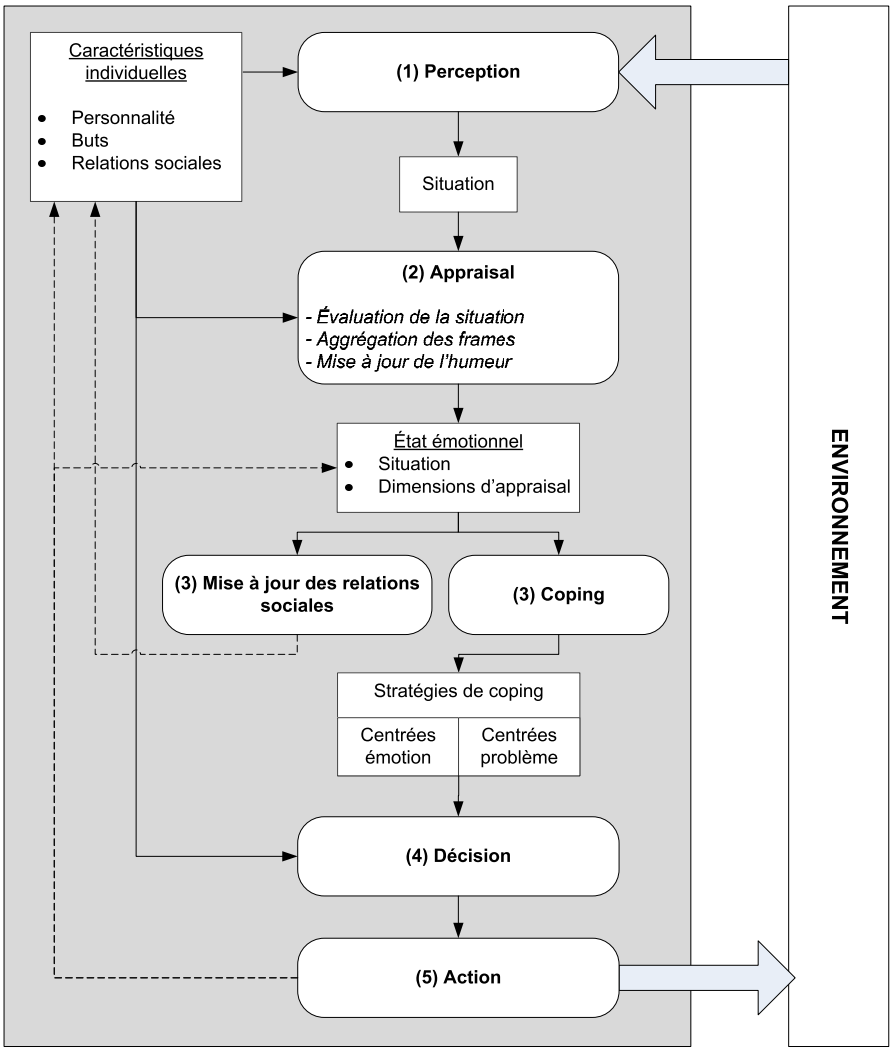

FIGURE 5.2.2: Cycle décisionnel d'un humain virtuel

l'attention visuelle, mais c'est un sujet que nous ignorons totalement ici. La perception se fait donc de manière brute, c'est-à-dire que l'état mental de humain virtuel n'a aucun impact sur sa perception.

Perception

Les entités perceptibles sont les éléments de la simulation qui sont visibles. Plus exactement, ce sont les états visibles qui déterminent si les entités et ses états associés sont visibles. On note quatre types d'entités : les objets, les humains virtuels, les évènements et l'humain virtuel lui-même.

L'ensemble des entités perçues par l'humain virtuel est écrit dans la mémoire de travail Soar sous forme de triplets. La mémoire de travail est un graphe sémantique dont le nœud racine est l'état courant. Le seul point d'entrée du graphe est un nœud particulier appelé *input-link.*

Chaque objet perçu est ajouté comme un sous-noeud de l'input-link auquel il est lié par un arc *object*. La figure 5.2.3 montre un exemple d'écriture de triplets représentant trois entités perçues sur l'*input-link* (un bloc bleu, une porte et l'humain virtuel lui-même).

L'humain virtuel perçoit également les autres humains virtuels ainsi que des évènements, que ceux-ci soient exogènes ou réalisés par un des humains virtuels. Comme pour les objets, ceux-ci sont ajoutés sur des sous-nœuds du nœud *input-link* et sont liés à ce dernier par un arc *agent* ou *event*.

Conformément avec la philosophie de l'esprit, l'humain virtuel perçoit également son état physique. Ces perceptions sont attachées au nœud *input-link* par un arc *self*.

L'écriture des triplets sur l'*input-link* est réalisée par l'interface entre l'environnement et Soar (voir 7.2).

Construction de la situation

La construction de la situation consiste à recopier les perceptions depuis l'*input-link* vers un sous-nœud de la situation courante appelé *perceptions*. Un certain nombre d'élaborations s'appliquent ensuite sur les sous-nœuds attachés à *perceptions* afin d'enrichir et d'abstraire les perceptions.

Ces élaborations sont en particulier chargées d'effectuer tous les calculs et comparaisons mathématiques, Soar n'offrant pas la possibilité de réaliser des calculs dans les conditions des règles. Il est donc nécessaire d'ajouter une étape supplémentaire qui attache à l'objet un triplet avec le résultat du calcul afin de pouvoir en tester la valeur dans les conditions des règles.

Certaines élaborations permettent de mettre à jour le statut des conditions présentes dans l'arbre d'activité. Si le ou les prédicats spécifiés dans la condition sont

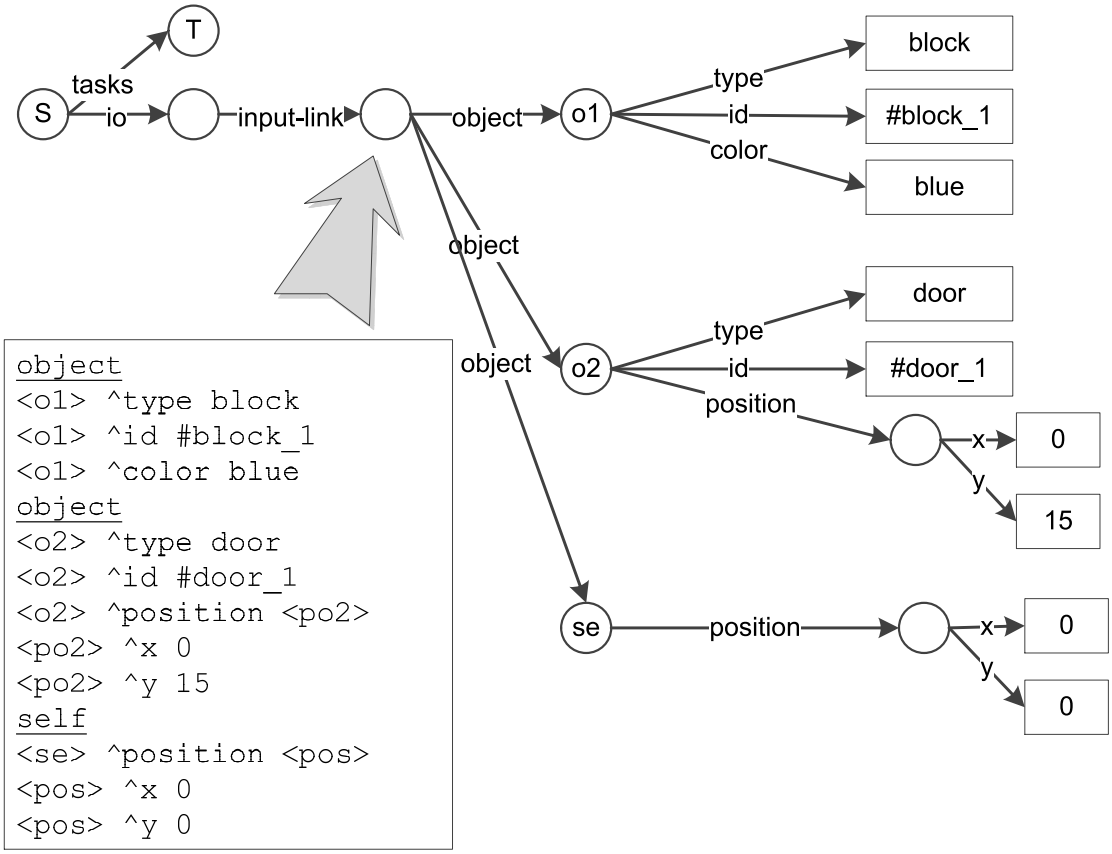

FIGURE 5.2.3: Exemple d'écriture des entités perçues dans la mémoire de travail

présent dans la mémoire de travail, la valeur de l'attribut *met* de la condition passe de **no** à **yes**, indiquant que la condition est vérifiée. Un mécanisme inverse permet de faire basculer cette valeur de **yes** à **no** lorsqu'un des prédicats n'est plus vrai.

5.2.2 Évaluation Cognitive

Création et évaluation des *appraisal frames*

Nous évaluons les évènements et actions qui surviennent dans l'environnement. Pour chaque évènement/action présent sur le nœud *perceptions*, nous créons une *appraisal frame* qui contient les dimensions d'*appraisal* définies dans le tableau 5.1.

Dimension	**Sous-dimension**	**Définition**
Relevance		degré d'attention à apporter à la situation
Desirability	**Activité**	**impact de la situation sur les buts de l'humain virtuel**
	Personnalité	**impact de la situation par rapport à la personnalité de l'humain virtuel**
Causal attribution	*Agency*	humain virtuel responsable de la situation
	Blame/credit	blâme ou crédit à apporter au responsable
Likelihood		probabilité d'occurrence de la situation (passée ou future)
Unexpectedness		prévision antérieure de l'occurrence de cette situation
Urgency		délai dans lequel traiter la situation
Ego involvement		impact sur l'image de soi (estime sociale, morale,...)
Coping potential	*Controllability*	degré d'influence sur la situation
	Changeability	possibilité que la situation change d'elle-même
	Power	pouvoir d'un humain virtuel de modifier la situation
	Adaptability	capacité à gérer les conséquences d'une situation

Table 5.1: Dimensions d'*appraisal*

La plupart des dimensions sont tirées de EMA et nous ne détaillerons pas les processus qui permettent de les évaluer. Nous insisterons par contre sur les différences de notre modèle (en gras), en particulier l'évaluation de la dimension *desirability* en deux évaluations distinctes, la première liée à l'avancement dans l'arbre d'activité (et donc relative aux buts) et la seconde à la personnalité de l'humain virtuel, évaluation qui dans EMA, prenait en compte la valeur d'utilité spécifiée à la main dans le graphe causal.

Évaluation de la désirabilité liée à l'avancement dans l'arbre d'activité

L'avancement dans l'arbre d'activité est associé à deux objets : les tâches et les conditions par le biais de l'attribut *utility*. L'évaluation de la dimension *desirability-activity* suit les étapes de l'algorithme 5.1.

Algorithme 5.1 Évaluation de l'avancement dans l'arbre d'activité

1. Initialiser la dimension *desirability-activity* de l'*appraisal-frame* avec la valeur 0
2. Si l'évènement a permis de terminer une tâche t (t, *state*, *terminated*)
   - ajouter l'utilité de la tâche (*desirability-activity <- desirability-activity + t.utility*)
3. Si l'évènement permet de valider une condition c (c, *met*, **yes**)
   - ajouter l'utilité de la condition (*desirability-activity <- desirability-activity + c.utility*)
4. Si l'action rend impossible la réussite d'une tâche (t, *state, failure*)
   - enlever l'utilité de la tâche (*desirability-activity <- desirability-activity - t.utility*)
5. Si l'action rend invalide une condition (c, *met*, **no**)
   - enlever l'utilité de la condition (*desirability-activity <- desirability-activity - c.utility*)

Évaluation de la désirabilité liée à la personnalité

L'évaluation de la désirabilité par rapport à la personnalité d'un individu porte sur la présence des marqueurs personnalisés. Ils peuvent être associés à deux objets : les tâches et les conditions par le biais de l'attribut *tags*. L'évaluation de la dimension *desirability-personality* suit les étapes de l'algorithme 5.2.

Algorithme 5.2 Évaluation de la personnalité dans l'arbre d'activité

1. Initialiser la dimension *desirability-personality* de l'*appraisal-frame* avec la valeur 0
2. Si l'évènement a permis de terminer une tâche t qui est marquée par le tag (t, *state*, *terminated*) (t, *tag, tag*)
   - ajouter l'importance apportée au marqueur (*desirability-personality* <- *desirability-personality* + tag.*value*)
3. Si l'évènement permet de valider une condition c (c, *met*, **yes**) (c, *tag, tag*)
   - ajouter l'utilité de la condition (*desirability-personality*<- *desirability-personality* + tag.*value*)
4. Si l'action rend impossible la réussite d'une tâche (t, *state, failure*) (t, *tag, tag*)
   - enlever l'utilité de la tâche (*desirability-personality* <- *desirability-personality* - tag.*value*)
5. Si l'action rend invalide une condition (c, *met*, **no**) (c, *tag, tag*)
   - enlever l'utilité de la condition (*desirability-personality* <- *desirability-personality* - tag.*value*)

Correspondance entre les catégories émotionnelles OCC et les *appraisal frames*

Pour plus de clarté, nous proposons d'associer aux *appraisal frames* une catégorie émotionnelle définie par la théorie OCC (voir 1.2.1.1). La théorie OCC catégorise les émotions tandis que le second les place dans un espace continu. Trois dimensions sont utilisées pour déterminer la catégorie associée à une *appraisal frame* : la désirabilité (*desirability*) peut être positive (+) ou négative (-) [2] ; la probabilité d'occurrence de l'évènement (*likelihood*) peut être sûre s'il s'est déjà produit (1) ou incertaine (< 1) ; la responsabilité causale des actions peut être l'humain virtuel lui-même (*self*) ou un autre humain virtuel (*other*). Cette correspondance se fait de la manière indiquée par le tableau 5.2.

Algorithme

Le processus d'évaluation cognitive est résumé par les étapes de l'algorithme 5.3. Les règles Soar génériques associées à ces étapes sont présentées en annexe A.1.

2. Pour évaluer la désirabilité globale d'une appraisal frame, on réalise la moyenne entre la désirabilité relative à l'avancement dans l'arbre d'activité et celle relative à la personnalité.

Dimension de l'*appraisal frame*			**Catégorie émotionnelle OCC**
desirability	*likelihood*	*agency*	
+	1		joie
-	1		détresse
+	1	other	admiration
-	1	other	colère
+	1	self	fierté
-	1	self	honte
+	< 1		espoir
-	< 1		peur

TABLE 5.2: Correspondance entre les catégories OCC et les dimensions des *appraisal frames*

Algorithme 5.3 Algorithme de mise à jour des relations sociales

1. Pour chaque *event* attaché au nœud *perceptions* :
   a) appliquer les règles d'élaboration pour créer l'*appraisal frame*
   b) appliquer les règles d'élaboration pour attribuer une catégorie émotionnelle OCC (tableau 5.2)

5.2.3 Mise à jour des relations sociales

Les théories de l'évaluation cognitive ne prennent pas en compte les relations sociales. Or, nous souhaitons modéliser des humains virtuels émotionnels et sociaux. Il est donc nécessaire d'étudier l'impact des émotions sur les relations sociales.

Cette étape doit être réalisée après l'étape d'*appraisal* car elle utilise les *appraisal frames*, et avant de prendre une décision car les relations sociales peuvent y jouer un rôle.

Ochs et al. définissent un modèle de la dynamique des relations sociales fondé sur les types d'émotions ressenties par les humains virtuels [Ochs 09]. Ce modèle est basé sur une version simplifiée de la théorie émotionnelle OCC (voir 1.2.1.1) et n'est donc pas directement compatible avec notre modèle qui utilise sur la théorie de l'évaluation cognitive de Smith et Lazarus (voir 1.2.1.2). Nous utilisons les catégories émotionnelles associées aux *appraisal frames* lors de l'étape précédente pour adapter ce modèle au nôtre.

Évolution des relations sociales

Le modèle de Ochs et al. représente la dynamique émotionnelle des relations sociales par l'intermédiaire de schémas ; la figure 5.2.4 reprend l'un d'entre eux. Il définit l'impact des émotions sur la relation d'appréciation entre deux humains virtuels, i et j. Les émotions sont ressenties par i et causées, d'après lui, par j.

Ces schémas sont résumés par les tableaux 5.3 et 5.4.

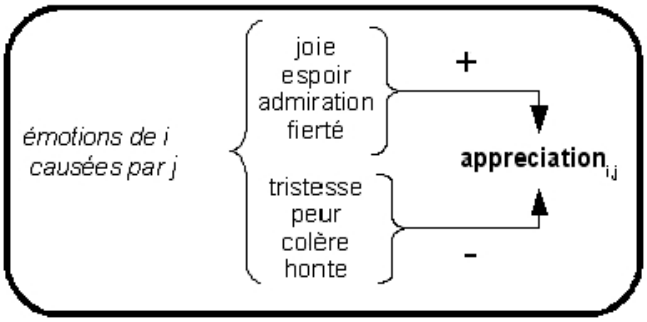

Figure 5.2.4: Dynamique des émotions sur la relation d'appréciation (extrait de [Ochs 09])

Algorithme

La mise à jour des relations sociales suit les étapes de l'algorithme 5.4. Les règles Soar associées sont définies en annexe A.2.

Catégorie émotionnelle OCC	**Impact sur la relation de i vers j**	
	Impact	**Dimension**
admiration	+	appréciation
	-	dominance
colère	+	dominance
	-	appréciation
	-	solidarité

TABLE 5.3: Impact des émotions ressenties sur la relation sociale pour une émotion causée par j (d'après [Ochs 09])

Catégorie émotionnelle OCC		**Impact sur la relation de i vers j**	
ressentie par i	**affichée par j**	**Impact**	**Dimension**
joie	joie	+	solidarité
	tristesse	-	
tristesse	tristesse	+	
	joie	-	
espoir	espoir	+	
	peur	-	
peur	peur	+	
	espoir	-	
peur		+	dominance
tristesse		+	

TABLE 5.4: Impact des émotions affichées de j et des émotions ressenties de i (d'après [Ochs 09])

Algorithme 5.4 Algorithme de mise à jour des relations sociales

1. Pour chaque *appraisal-frame* :
   a) proposer et appliquer l'opérateur de modification des relations inter-individuelles correspondant à la catégorie émotionnelle OCC (tableau 5.3)
2. Pour chaque humain virtuel attaché au nœud *perceptions* :
   a) proposer et appliquer l'opérateur de modification des relations inter-individuelles correspondant à la catégorie émotionnelle OCC (tableau 5.4)

5.2.4 *Coping* centré problème

Parmi toutes les stratégies de *coping* proposées par [Smith 91] et implémentées dans EMA, nous ne modélisons que le *coping* centré problème, c'est-à-dire celui qui vise à agir dans l'environnement.

Afin d'inscrire ses actions dans un cadre rationnel dirigé par des buts, l'humain virtuel est doté d'arbre(s) d'activité. À chaque instant, il tente d'avancer dans cet arbre en réalisant les actions associées aux tâches feuilles. Dans le cas où l'arbre d'activité est complet et où les conditions de sélection des tâches sont explicites et n'entrent pas en conflit, ce parcours est relativement simple.

Or, comme les arbres d'activité visent à décrire l'activité réelle et sont renseignés par des ergonomes, ils sont rarement complets. L'humain virtuel doit donc être capable de raisonner sur son arbre d'activité et de planifier des actions afin de l'adapter et le compléter. Dans ce cadre, il évalue toutes les options qui se présentent à lui en fonction de plusieurs paramètres.

L'algorithme 5.5 est appliqué, les règles génériques correspondantes sont définies en Annexe (voir A.3).

Algorithme 5.5 Algorithme de coping et de prise de décision

1. Pour chaque action possible :
   a) Créer une *coping-frame*
   b) Évaluer l'état émotionnel résultant
   c) Évaluer la désirabilité par rapport aux marqueurs personnalisés
   d) Évaluer l'avancement dans l'arbre d'activité
   e) Combiner ces évaluations en fonction des préférences de l'humain virtuel
2. Sélectionner l'action ayant l'évaluation globale la plus haute

Création d'une *coping-frame*

À chaque action possible est associée une *coping-frame*, structure de donnée similaire à l'*appraisal frame* et composée des deux dimensions définies dans le tableau 5.5. D'une manière générale, les évaluations du coping suivent le même schéma que lors de l'évaluation d'un évènement lors de la phase d'*appraisal*.

La règle générique de création d'une *coping-frame* est définie en annexe (voir A.3.1).

Dimension	Description
Personnalité	évaluer le respect des marqueurs personnalisés
Activité	évaluer l'avancement dans l'arbre activité

TABLE 5.5: Dimensions d'une *coping-frame*

Algorithme 5.6 Évaluation du respect des marqueurs personnalisés

1. Initialiser la dimension *tags* de la *coping-frame* avec la valeur 0
2. Si l'action considérée est associée à une tâche dans le modèle d'activité
   - Pour chaque marqueur associé à la tâche dans le modèle d'activité, augmenter la valeur de *tags* du poids de ce marqueur
   - Pour chaque marqueur associé aux conditions de la tâche dans le modèle d'activité, augmenter la valeur de *tags* du poids de ce marqueur
3. Si l'action permet de terminer une tâche
   - ajouter les marqueurs de la tâche
4. Si l'action permet de valider une condition
   - ajouter les marqueurs de la condition
5. Si l'action rend impossible la réussite d'une tâche
   - enlever les marqueurs de la tâche
6. Si l'action rend invalide une condition
   - enlever les marqueurs de la condition

Évaluation relative aux marqueurs personnalisés

ACTIVITY-DL, le formalisme utilisé pour spécifier les arbres d'activité permet de définir un certain nombre de marqueurs personnalisés (voir 4.3.2). Ces marqueurs permettent de taguer une tâche ou une condition comme étant relative à un domaine particulier (hygiène, sécurité, production, ...).

Nous proposons de réaliser une évaluation pour chaque type de marqueur présent dans l'arbre d'activité et de doter chaque humain virtuel de préférences relatives à ces marqueurs. Ainsi, confronté à une alternative, un humain virtuel qui n'apportera de l'importance qu'à l'hygiène évaluera positivement une tâche présentant ce marqueur et affectera une valeur nulle aux autres.

L'algorithme 5.6 réalise cette évaluation. Les règles Soar génériques associées sont définies en Annexe (voir A.3.2.1).

Évaluation relative au suivi de l'arbre

Chaque action est également évaluée relativement à l'avancement qu'elle apporte dans l'arbre d'activité. Par exemple, une action ne permettant pas d'avancer aura une évaluation nulle, tandis qu'une action permettant de résoudre une tâche mère aura une évaluation très positive. Au contraire, une action qui rend invalides certaines conditions nécessaires à l'avancement sera évaluée négativement. L'algorithme 5.7 permet de réaliser cette évaluation, et les règles Soar génériques sont présentées en Annexe (voir A.3.2.2).

Algorithme 5.7 Évaluation de l'avancement dans l'arbre d'activité

1. Initialiser la dimension *activity* de la *coping-frame* avec la valeur 0
2. Si l'action considérée est associée à une tâche dans le modèle d'activité
   - ajouter l'utilité de la tâche associée
3. Si l'action permet de terminer une tâche
   - ajouter l'utilité de la tâche
4. Si l'action permet de valider une condition
   - ajouter l'utilité de la condition
5. Si l'action rend impossible la réussite d'une tâche
   - enlever l'utilité de la tâche
6. Si l'action rend invalide une condition
   - enlever l'utilité de la condition

Agrégation des dimensions de la *coping-frame*

La combinaison de ces différentes évaluations est réalisée en fonction des préférences de l'humain virtuel par une règle générique définie en Annexe (voir A.3.3).

5.3 Conclusion

Dans ce chapitre, nous avons présenté notre modèle d'humain virtuel.

Nous avons tout d'abord décrit ses caractéristiques individuelles : sa personnalité, décrite avec le modèle OCEAN, et les relations sociales qu'il entretient avec les autres humains virtuels, avec le modèle proposé par Ochs [Ochs 09].

Notre humain virtuel suit le cycle classique des agents cognitifs. Il perçoit son environnement, raisonne, puis agit. En accord avec les sciences cognitives, tous les processus de raisonnement sont réalisés par une architecture cognitive. Nous avons choisi d'utiliser Soar, présenté longuement en 1.3.2.

La théorie de l'évaluation cognitive permet de doter l'humain virtuel d'émotions en affinant les processus de raisonnement. Les travaux de Smith et Lazarus [Smith 91] nous permettent d'ajouter deux processus cognitifs : l'*appraisal*, qui consiste à évaluer l'impact émotionnel des situations, et le *coping*, qui permet de choisir une stratégie adaptée. Nous ajoutons également une étape responsable de la mise à jour des relations sociales avec autrui, basée sur les travaux de Ochs et al. [Ochs 09]. Deux configurations sont envisagées : le cas où autrui est responsable de l'état émotionnel de l'humain virtuel ; et le cas où les états émotionnels des deux individus sont en congruence (ou incongruence).

Nous avons ensuite présenté l'algorithme de prise de décision de l'humain virtuel, ainsi que l'intégration des modèles d'activité dans ce processus, en particulier lors des processus *d'appraisal* et de *coping* centré problème.

Pour chacune de ces étapes, nous avons défini un ensemble de règles génériques Soar indépendantes du domaine. Celles-ci sont présentées en Annexe (A).

Chapitre 6

Contagion émotionnelle

Dans notre modèle, les relations entre les humains virtuels sont modélisées par des interactions émotionnelles.

L'état de l'art a présenté les deux façons de modéliser les interactions émotionnelles pour des humains virtuels.

L'empathie cognitive permet de modéliser de manière très fine les interactions émotionnelles entre les individus, mais elle nécessite que les individus disposent d'une théorie de l'esprit leur permettant d'évaluer l'état émotionnel de chacun des autres humains virtuels. Cette approche pose des problèmes de passage à l'échelle lorsqu'on souhaite implémenter plusieurs humains virtuels.

L'empathie affective, ou contagion émotionnelle, est un processus inconscient qui fait que l'état émotionnel d'un observateur est automatiquement impacté par celui des individus qui l'entourent. Comme ce processus est inconscient, il se situe à un niveau « inférieur » par rapport à celui de l'évaluation cognitive. Nous avons présenté les modèles de contagion émotionnelle existant dans la littérature, et en particulier le modèle ASCRIBE. Ces modèles s'inspirent des observations faites par les psychologues et sociologues au niveau macroscopique (c'est-à-dire au niveau de la foule ou du groupe) mais les modèles des individus sont ad hoc, aussi bien d'un point de vue statique (personnalité, structures émotionnelles et sociales) que dynamique (évaluation cognitive, prise de décision).

Nous proposons d'ignorer les processus d'empathie cognitive, qui posent des problèmes de performance, et mettons en place un modèle d'empathie affective qui prend en compte les composantes individuelles de chaque individu afin de gérer les interactions émotionnelles entre les individus de manière fine.

La première section de ce chapitre explique comment les composantes individuelles de chaque individu sont utilisées pour déterminer comment ce phénomène l'impacte, tandis que la seconde section décrit comment cette contagion émotionnelle s'intègre dans le cycle des humains virtuels.

6.1 Caractéristiques individuelles

ASCRIBE [Bosse 09] est un modèle computationnel de contagion émotionnelle que nous avons présenté en 2.2.2.

Pour chaque individu, le modèle ASCRIBE utilise deux caractéristiques qui sont le pouvoir de contagion (η) et sa susceptibilité à la contagion (δ). L'impact d'un individu sur un autre est impacté, d'une part, par les caractéristiques de chacun d'entre eux, et d'autre part, de la force du canal de communication qui les lie. Bien que l'approche générale soit cohérente avec les études réalisées par les psychologues, le modèle d'humain virtuel qu'il met en jeu n'est pas issu de la psychologie et de la sociologie et ne permet pas de modéliser des prises de décision complexes. En effet, il est nécessaire de relier les composantes individuelles telles que la personnalité ou les buts des humains virtuels aux processus régissant la contagion émotionnelle et la prise de décision afin d'obtenir des comportements rationnels.

Nous proposons d'enrichir le modèle ASCRIBE en associant les paramètres individuels qui régissent la contagion émotionnelle aux caractéristiques individuelles des humains virtuels.

Capacité d'infection et Susceptibilité à la contagion

S'il n'existe pas de modèle formel reliant les caractéristiques individuelles d'un individu à sa participation au phénomène de contagion émotionnelle, de nombreuses expériences en psychologie se sont intéressées à ce sujet et ont permis de faire ressortir quels facteurs pourraient influencer la contagion émotionnelle.

Selon Hatfield et al. [Hatfield 92], la contagion émotionnelle intervient entre chaque individu (celui-ci étant à la fois contaminant et contaminé) et est modulée par deux facteurs : la capacité d'infection des individus contaminants, et la sensibilité à la contagion des contaminés. Ces deux facteurs correspondent aux paramètres η et δ du modèle ASCRIBE.

Capacité d'infection

Deux facteurs modulent la capacité d'infection d'un individu :

1. il doit ressentir, ou donner l'impression de ressentir des émotions fortes ;
2. il doit être capable d'exprimer ces émotions fortes (expressions faciales, vocales et/ou posturales) ;

Susceptibilité à la contagion

Au contraire, les traits suivants caractérisent un individu susceptible à la contagion :

1. il est attentif aux autres ;
2. il se considère plutôt comme dépendant ;
3. il peut décoder les émotions faciales, vocales et/ou posturales des autres individus ;
4. il tend à imiter les émotions faciales, vocales et/ou posturales des autres individus ;
5. il est conscient de sa réponse émotionnelle (expressions faciales, vocales et/ou posturales) ;
6. il est émotionnellement réactif.

Relations sociales

Certaines configurations de relations sociales peuvent influencer le processus de contagion émotionnelle :

1. les relations d'appréciation (amour) ou relations proches rendent les individus concernés plus susceptibles
2. les relations de pouvoir rendent l'individu dominant plus résistant aux émotions de l'autre, tandis que ce dernier sera plus influençable.

Adaptation à notre modèle d'humain virtuel

Le pouvoir de contagion d'un humain virtuel ne dépend que de sa personnalité et est donc statique (formule 6.1.1).

Π est une fonction permettant de combiner les traits de personnalité et est dépendante de l'implémentation.

$$\eta = \Pi(-P_E, P_A, P_E, -P_C) \tag{6.1.1}$$

La susceptibilité à la contagion (t) d'un humain virtuel dépend de son humeur et évolue au cours du temps. Elle est définie par la formule 6.1.2. ϕest une fonction permettant de combiner les traits de personnalité, et Π'est la fonction qui combine l'importance de la personnalité à l'humeur.

$$\delta(t) = \phi(Humeur(t), \Pi'(P_A, -P_O, P_A, P_O, -P_E)) \tag{6.1.2}$$

La force du canal de communication entre deux humains virtuels a et b dépend de la relation sociale de a vers b et est définie par la formule 6.1.3. Δest une fonction qui calcule la force de la relation sociale en transformant le vecteur 3D de relation sociale en valeur sur l'intervalle $[1; 1]$.

$$\alpha_{a/b}(t) = \Delta(R_{/a}) \tag{6.1.3}$$

$\eta/\delta/\alpha$	**Description**	**Interprétation**
Pouvoir de contagion		
η	Ressent des émotions	E-
	Exprime des émotions	A+
		E+
		C-
Susceptibilité à la contagion		
δ	Attentif aux autres	A+
		Humeur > 0
	Dépendant des autres	O-
		A+
	Imite les émotions des autres	A+
	Conscient de ses émotions	O+
	Émotionnellement réactif	E-
Relations sociales		
α	Amour, admiration	$Affection_{a.b} > 0$
	Respect, pouvoir	$Dominance_{a/b} < 0$

TABLE 6.1: Influence des différences individuelles sur la contagion émotionnelle

6.2 Proposition d'implémentation

Le calcul des valeurs η, δ et α est réalisé par des fonction d'aggrégation Π, Φ et Δ.

Nous en proposons une première interprétation ci-dessous, l'idée étant d'offrir la possibilité de les modifier via une interface graphique. Ceci permettra de régler la rapidité et l'importance de la contagion selon les scénarios.

Nous avons implémenté de la manière suivante les fonctions de calcul de pouvoir de contagion (formule 6.2.1) et de susceptibilité à la contagion (formule 6.2.2) d'un humain virtuel.

$$\eta = \frac{f\left(-P_E\right)}{2} + \frac{f\left(P_A\right)}{4} + \frac{f\left(P_E\right) + f\left(-P_C\right)}{8} \tag{6.2.1}$$

$$\delta(t) = \frac{f\left(P_A\right) + f\left(-P_O\right)}{10} + \frac{f\left(P_A\right) + f\left(P_O\right) + f\left(-P_E\right)}{5} + \frac{1 + Humeur_t}{5} \tag{6.2.2}$$

avec $f\left(x\right) = \begin{cases} x, & si\, x \geqq 0 \\ 0, & sinon \end{cases}$.

La force du canal de communication entre deux humains virtuels a et b dépend de la relation sociale de a vers b et est définie par la formule 6.2.3. La force du canal de communication est définie par la relation sociale (formule 6.2.3).

$$\alpha_{a/b}(t) = \frac{App_{b/a} + Domin_{b/a} + Solid_{b/a} + Fami_{b/a}}{4} \tag{6.2.3}$$

6.3 Dynamique

La figure 6.3.1 montre comment la contagion émotionnelle s'intègre dans le cycle des humains virtuels.

Cette étape a lieu après l'évaluation cognitive, une fois que tous les humains ont évalué l'impact des évènements et actions de l'environnement sur leur état émotionnel ainsi que sur leurs relations sociales. Comme il s'agit d'un phénomène inconscient et automatique, il est situé en dehors de la partie grisée qui représente les processus cognitifs de l'humain virtuel.

Le module de contagion émotionnelle récupère le pouvoir de contagion, la susceptibilité à la contagion ainsi que l'état émotionnel et les relations sociales de chaque humain virtuel. La position des agents est également prise en compte par ce module afin de pouvoir représenter la distance de perception [1].

Le module de contagion émotionnelle applique ensuite le processus de contagion émotionnelle du modèle ASCRIBE (décrit en 2.2.2) en utilisant les valeurs de η et δ définies ci-dessus et les relations sociales α comme canal de communication afin de mettre à jour l'état émotionnel de l'humain virtuel en fonction de celui de ses voisins.

1. On considère que les voisins d'un humain virtuel a sont situés à une distance inférieure à d.

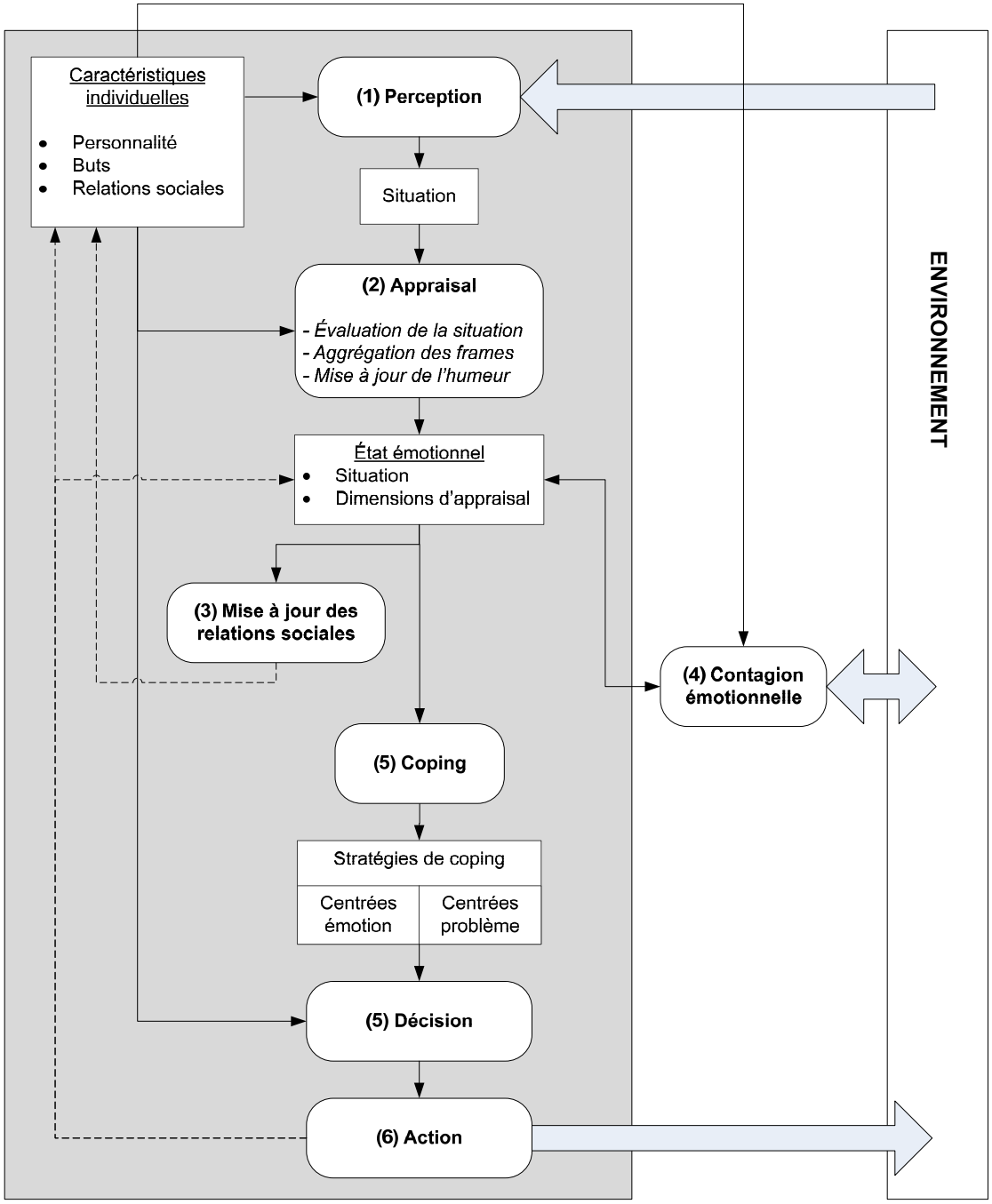

FIGURE 6.3.1: Cycle d'un humain virtuel - prise en compte de la contagion émotionnelle

Implémentation et résultats

I've seen things you people wouldn't believe.
Attack ships on fire off the shoulder of Orion.
I watched C-beams glitter in the dark near the Tannhauser gate.
All those moments will be lost in time.
Like tears in rain.
Time to die.

("Blade Runner" de Ridley Scott, Warner Brothers, 1982.)

Chapitre 7

La plateforme HUMANS

Les travaux abordés durant cette thèse s'insèrent dans un large contexte de recherche sur les environnements informatiques pour l'apprentissage humain (EIAH). Les projets de l'équipe consistent à modéliser l'activité humaine et sociale dans des environnements virtuels afin de permettre et/ou améliorer l'apprentissage.

Par exemple, le projet V3S (Virtual Reality for Safe Seveso Subcontractors) a pour objectif de simuler des situations de travail dangereuses touchant aux activités de maintenance réalisées par des entreprises extérieures sur des sites Seveso. Le projet SIMADVF présente différents scénarios sous forme de serious game dans le but de former des assistantes de vie.

Deux principales thématiques de recherche sont abordées :

- la scénarisation adaptative et le suivi de l'apprenant, pour proposer des situations d'apprentissage permettant l'acquisition de connaissances ;
- le comportement d'humains virtuels autonomes, afin de permettre des apprentissages dans des situations collectives complexes.

C'est dans le second axe que s'inscrivent ces travaux de thèse.

La première problématique adressée dans cette thèse concerne la problématique de passage à l'échelle induite par la complexité des raisonnements sur autrui lorsqu'un grand nombre d'humains virtuels est modélisé. Pour y répondre, nous avons proposé un modèle d'humain virtuel émotionnel, cognitif et social ainsi qu'un modèle de contagion émotionnelle qui régit les interactions émotionnelles entre ces humains.

Une seconde problématique abordée par cette thèse concerne la représentation des connaissances, et d'un point de vue plus général, l'intégration de cet humain virtuel dans son environnement. Un des objectifs de notre équipe de recherche est de proposer des outils auteur afin de permettre à des non-informaticiens de modéliser des situations d'apprentissage. Ces descriptions sont réalisées avec l'aide de formalismes, DOMAIN-DL et ACTIVITY-DL, afin de permettre aux modules

de scénarisation, de suivi de l'apprenant et aux humains virtuels de partager les mêmes connaissances. Il est nécessaire que les humains virtuels puissent interpréter directement les modèles spécifiés avec ces langages afin de se comporter de manière cohérente avec l'environnement associé.

Dans ce chapitre, nous présentons tout d'abord HUMANS, la suite logicielle pour la génération d'EIAH de notre équipe. Nous détaillons ensuite nos deux contributions principales, le module REPLICANTS qui gère les humains virtuels, puis l'outil ACTIVITY-GEN qui opérationnalise les modèles spécifiés avec les formalismes HUMANS-DL en connaissances interprétables par les humains virtuels.

7.1 Présentation de la plateforme

7.1.1 Architecture globale

HUMANS est l'acronyme de *HUman Models based Artificial eNvironments Software platform*, soit « plateforme logicielle pour environnements artificiels s'appuyant sur des modèles de l'humain ».

La suite logicielle HUMANS regroupe l'ensemble des outils et modules de l'équipe HUMANS du laboratoire Heudiasyc, et vise à permettre le développement d'environnements virtuels rendant compte de comportements humains, ceux-ci étant basés sur des analyses de l'activité humaine.

Elle est composée de modules dont les fonctionnalités sont bien définies. Ces modules s'articulent autour de modèles qui leur permettent de disposer de représentations communes. On parle de HUMANS comme d'une « suite logicielle », d'une « plateforme regroupant plusieurs logiciels ».

La Figure 7.1.1 regroupe ces différents modules. Notons qu'il existe deux types de modules externes à la plateforme ; le module GUI, qui gère l'affichage et les interactions avec l'apprenant ; et tous les autres modules, regroupés sous l'appellation E.T.

7.1.1.1 HUMANS>>COMMUNICATOR, l'interface de communication

COMMUNICATOR est le module qui gère la communication entre les modules de la plateforme et ainsi qu'avec les modules externes.

Il a été développé sous ma supervision par des étudiants ingénieurs de l'Université de Technologie de Compiègne.

La plateforme HUMANS est un système multi-agent, actuellement implémenté en JADE[1]. Chaque module est un agent qui évolue sur la plateforme.

1. http ://jade.tilab.com/

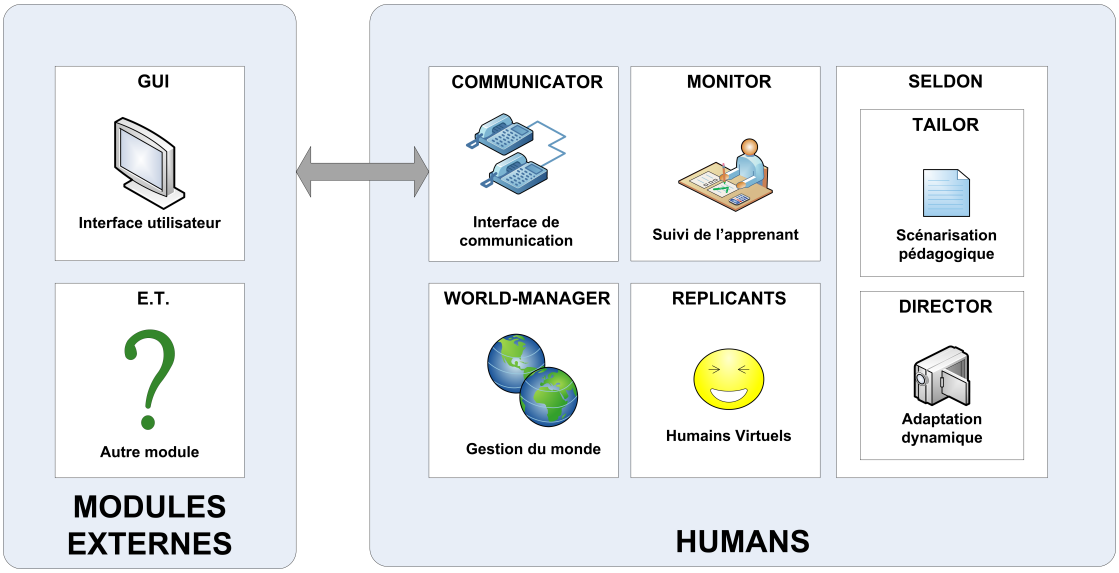

FIGURE 7.1.1: HUMANS - Principaux modules

COMMUNICATOR n'est donc pas un module séparé et autonome, mais le socle qui permet aux autres modules d'interagir ensemble.

La figure 7.1.2 représente le diagramme UML des différentes classes impliquées dans la communication entre les différents modules.

La communication entre les modules utilise un certain nombre de classes :

- L'agent central *Communicator* qui transfère les messages aux destinataires tout en vérifiant les droits ;
- *ComLog*, un agent Jade qui enregistre tous les messages échangés et les affiche dans la console ou les écrit dans un fichier texte ;
- *DFCom,* une surcharge de l'agent Jade *DF*, le service de pages jaunes, afin de gérer les droits de communication de la plate-forme ;
- La classe abstraite d'interfaçage *ComModuleInterface* qui fait le lien entre le module extérieur et l'agent *Communicator ;*
- L'agent *ComModuleAgent* qui transmet les messages de l'interface vers l'agent central *Communicator.*

Couplage technique

Pour s'intégrer à la plateforme, chaque module doit s'interfacer avec un agent *ComModuleAgent* via une *ComModuleInterface* afin de recevoir et d'envoyer des messages. La figure 7.1.3 montre les classes fournies pour faciliter ce couplage.

Dans le cas où le module est développé en Java, cet interfaçage est relativement direct via l'utilisation de *JavaComInterface,* car il permet de réaliser des appels

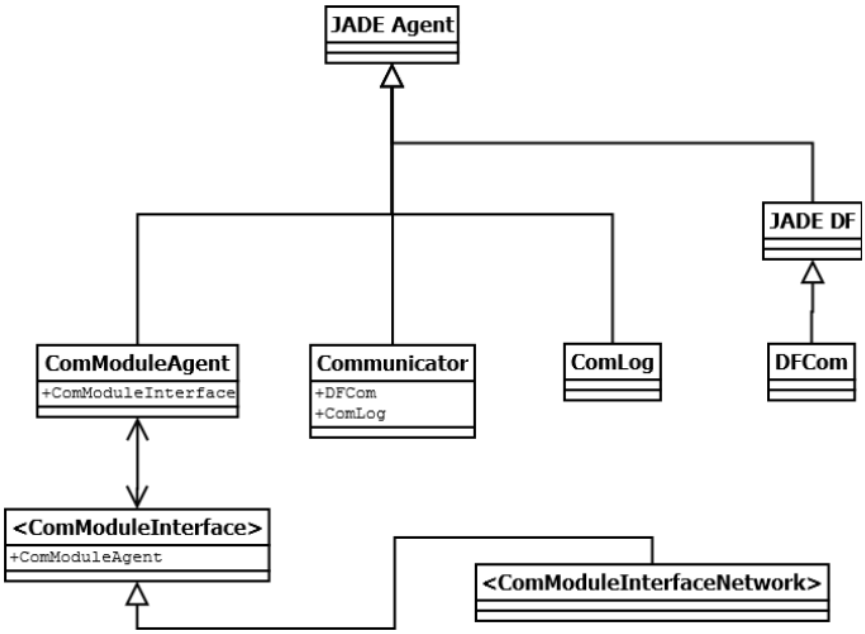

FIGURE 7.1.2: COMMUNICATOR - Classes principales

de code natifs. Cependant, il est fréquent que les modules utilisent des technologies différentes et ce couplage diffère donc radicalement selon le langage utilisé. La classe *ComModuleInterfaceNetwork* propose d'utiliser un serveur TCP/IP ; cette utilisation est complètement masquée pour l'utilisateur qui n'a qu'à surcharger un certain nombre de méthodes pour finaliser le couplage du module avec la plateforme.

Un langage de communication est également proposé. Celui-ci est basé sur les différents concepts des langages de description de HUMANS-DL afin d'assurer une continuité sémantique entre les connaissances utilisées par les modules, et celles qu'ils échangent.

7.1.1.2 HUMANS>>WORLD-MANAGER

WORLD-MANAGER est le module qui est responsable de maintenir l'état du monde. Il dispose d'une base de connaissance qui centralise l'intégralité des connaissances relatives à l'état du monde et qui utilise le formalisme défini par DOMAIN-DL. Il est également doté d'un moteur de raisonnement capable d'interpréter les requêtes et d'y répondre. Il est en particulier responsable des requêtes suivantes :

– perception, envoyées par les humains virtuels (« Que vois-je »), à laquelle il

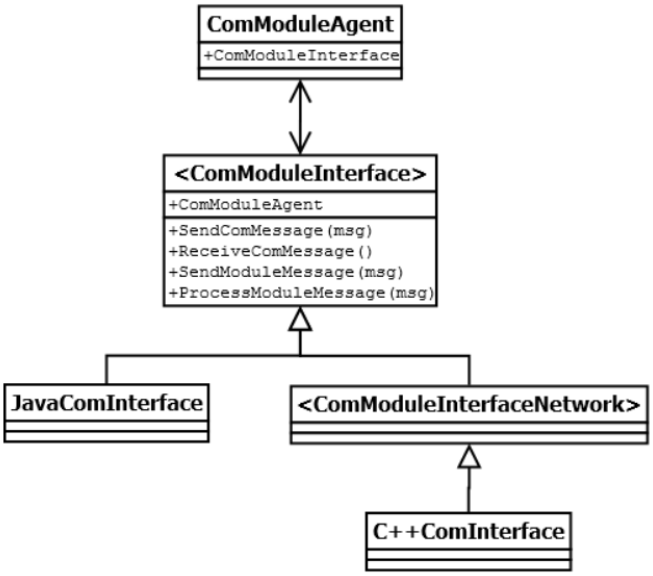

FIGURE 7.1.3: COMMUNICATOR - Interfaçage avec les modules

renvoie la liste des entités perçues ainsi que leurs états visibles ;
- demande d'action des humains virtuels (« Ouvrir la porte »), dont il vérifie la validité avant de renvoyer les changements d'états résultants.

WORLD-MANAGER dispose d'un ensemble de règles logiques, générées à partir des langages de description de HUMANS-DL, qui lui permettent de garantir l'intégrité de l'état du monde. Il est actuellement implémenté en Java et utilise la librairie Jena [2].

7.1.1.3 HUMANS>>MONITOR

MONITOR, anciennement HERA, est le module qui assure le suivi de l'apprenant. Il interprète les actions que l'apprenant réalise dans l'environnement et détecte ses erreurs par rapport à un arbre d'activité ACTIVITY-DL. Il envoie ensuite, en temps réel, les erreurs et performances de l'apprenant aux modules intéressés. MONITOR est le résultat de la thèse de K. AMOKRANE [Amokrane 10].

7.1.1.4 HUMANS>>SELDON

SELDON est le module de scénarisation pédagogique de la plateforme HUMANS. Il est composé de TAILOR et DIRECTOR.

2. http ://jena.apache.org/

À partir du profil de l'apprenant, de son activité et des objectifs pédagogiques, ADAPTOR détermine un ensemble de situations cibles et de situations à éviter (situation constraints).

Ces situations sont ensuite transmises à DIRECTOR qui se charge d'adapter dynamiquement l'environnement afin de parvenir ou d'éviter ces situations.

L'originalité de SELDON est de permettre un contrôle indirect des humains virtuels sans nuire à la cohérence du scénario.

TAILOR et DIRECTOR sont en cours de développement et respectivement issus des travaux de K. Carpentier et C. Barot.

7.2 HUMANS>>REPLICANTS - Les humains virtuels de HUMANS

7.2.1 Architecture générale de REPLICANTS

HUMANS>>REPLICANTS est le module de la plateforme HUMANS que j'ai développé au cours de ma thèse. Il est responsable du comportement des humains virtuels, et ce, uniquement au niveau décisionnel, c'est-à-dire que les processus relatifs au monde physique tels que la perception, les déplacements, les mouvements ne sont pas présents.

La figure 7.2.1 présente son architecture fonctionnelle. REPLICANTS dispose d'une interface de communication (*Replicants-COM*) qui lui permet d'échanger des connaissances avec l'extérieur. Les connaissances de REPLICANTS sont centralisées dans un module d'environnement (*Environnement*), qui se charge de fournir les informations nécessaires aux humains virtuels. Le module d'environnement communique avec le *WORLD-MANAGER* via le *COMMUNICATOR* afin d'obtenir les états du monde et d'envoyer les requêtes d'action formulées par les humains virtuels.

Les humains virtuels sont des entités représentées par deux parties, le replicant représente le « corps », et le SoarAgent « l'esprit ».

Un peu de technique ?

REPLICANTS est développé en C++ et utilise les librairies boost [3] et Qt [4]. REPLICANTS embarque l'architecture cognitive Soar [5] ; il intègre un noyau Soar (*SoarKernel*) sur lequel évoluent les SoarAgents.

3. http ://www.boost.org/
4. http ://qt.nokia.com/
5. La version de Soar utilisée est la Soar Suite 9.3.2

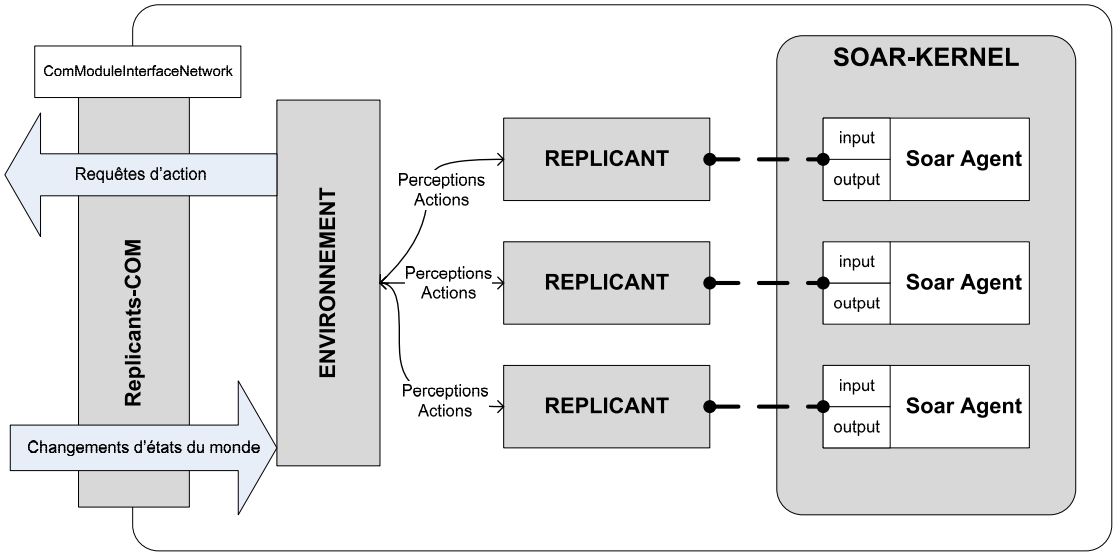

FIGURE 7.2.1: Architecture fonctionnelle de REPLICANTS

L'interface entre le « corps » et le « cerveau » des humains virtuels est réalisé via la lecture et l'écriture de connaissances dans la mémoire de travail du SoarAgent. Cette lecture est réalisée sur le noeud *output-link*, et l'écriture sur le noeud *input-link* en utilisant une interface particulière, appelée SoarSML (Soar Markup Language).

Visualisation et débugger

La figure 7.2.2 présente une capture d'écran de l'interface graphique de REPLICANTS. Cette interface simpliste permet de créer des humains virtuels et de modifier leurs caractéristiques individuelles, telles que leur personnalité, relations sociales ou humeur. La scène centrale affiche les replicants existants sur la plateforme sous forme de points dont la couleur, un dégradé du rouge vers le vert, représente l'humeur [6]. Conformément aux idées couramment admises, le rouge représente une mauvaise humeur, et le vert une bonne humeur. Le large cercle gris autour du replicant selectionné représente les entités qu'il perçoit.

Les processus cognitifs (évaluation cognitive et décision) peuvent être suivis via les outils fournis dans la Suite Soar. On dispose ainsi d'un débugger, SoarJavaDebugger, qui permet de visualiser en temps réel l'état des processus cognitifs des replicants évoluant sur le noyau Soar associé à REPLICANTS.

6. Dans le cas d'une impression en noir et blanc, une couleur claire correspond à une bonne humeur et une couleur foncée une mauvaise humeur.

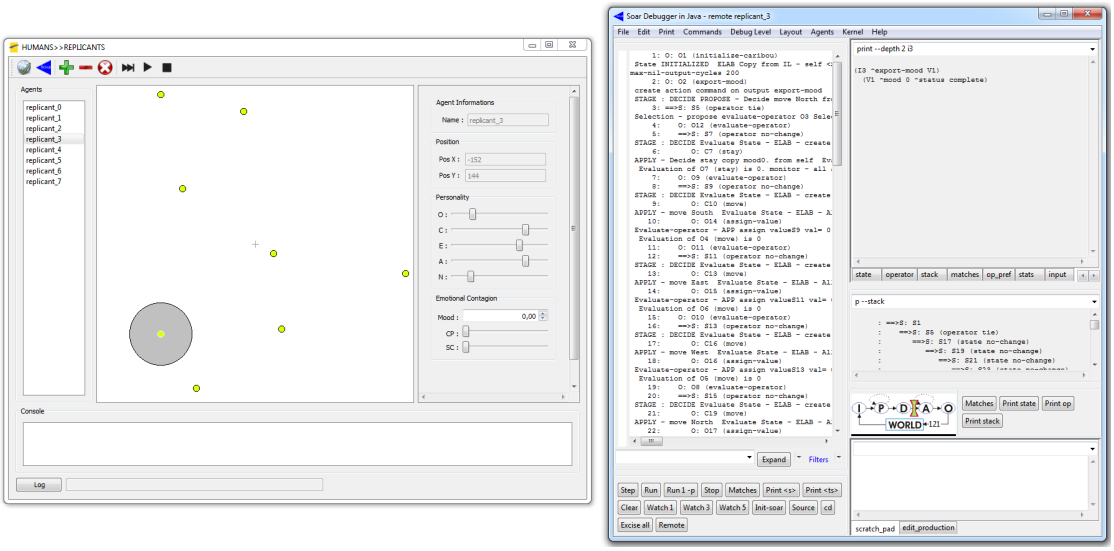

FIGURE 7.2.2: Capture d'écran de REPLICANTS (à gauche) et du Debugger Soar connecté (à droite)

7.2.2 Cycle des replicants

Le cycle de nos humains virtuels contient un certain nombre d'étapes, définies en 5.2 :

1. Perception et construction de la situation
2. Évaluation cognitive
3. Contagion émotionnelle
4. Coping, décision et action

Nous proposons de regrouper ces étapes dans des machines à états représentées par la figure 7.2.3. Le niveau global (en gris) comporte trois grandes étapes dont la première regroupe la perception et l'évaluation cognitive. L'entrée dans cet état entraine pour chaque replicant (en bleu) , le début des étapes de perception puis d'évaluation cognitive et de mise à jour des relations sociales (en blanc). Une fois ces étapes terminées pour chacun des replicants, la machine globale entre dans le processus de contagion émotionnelle ; cette étape est non cognitive, c'est à dire qu'elle est réalisée au niveau du « corps » du replicant, et donc en dehors de Soar. Enfin vient la phase de *coping* et d'action lors de laquelle chaque replicant prend une décision, puis envoie une action à l'environnement.

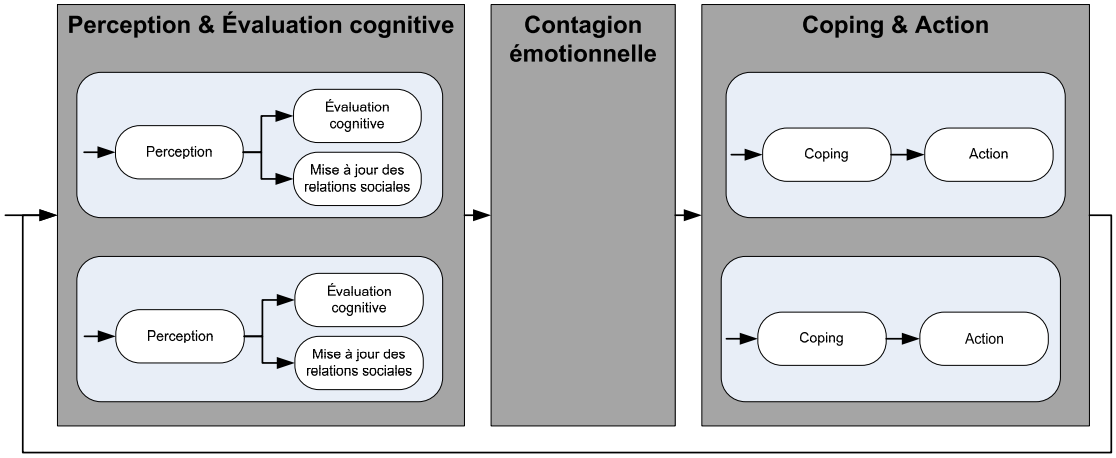

FIGURE 7.2.3: Cycle des replicants

7.2.2.1 Perception

L'étape de perception est une étape non cognitive qui se déroule en trois étapes, reprises dans l'algorithme 7.1. Nous rappelons ici que la perception est « brute », c'est-à-dire que l'humain virtuel perçoit l'ensemble des entités situées dans un certain rayon. L'état émotionnel, la personnalité n'influencent pas ses perceptions.

Algorithme 7.1 Algorithme de perception

1. Envoi d'une requête de perception à l'environnement
2. Réception de la réponse de l'environnement (liste de triplets)
3. Écriture des triplets sur l'*input-link* du SoarAgent

7.2.2.2 Évaluation cognitive

L'étape d'évaluation cognitive est intégralement réalisée par le SoarAgent et suit les points de l'algorithme 7.2. Elle consiste à générer un état émotionnel relatif aux perceptions, et à mettre à jour ses relations sociales. Les règles génériques utilisées sont détaillées dans la partie 5.2.2, l'intégralité d'entre elles se trouvant en annexe (voir A.1).

Une fois l'état émotionnel évalué par le SoarAgent et les relations sociales mises à jour, ces informations sont écrites par le SoarAgent sur le noeud *output-link*.

Algorithme 7.2 Algorithme d'évaluation cognitive

1. Pour chaque *event* attaché au nœud *perceptions* :
   a) appliquer les règles d'élaboration pour créer l'*appraisal frame*
   b) appliquer les règles d'élaboration pour attribuer une catégorie émotionnelle OCC
2. Pour chaque *appraisal-frame* :
   a) proposer et appliquer l'opérateur de modification des relations inter-individuelles correspondant à la catégorie émotionnelle OCC
3. Pour chaque humain virtuel attaché au nœud *perceptions* :
   a) proposer et appliquer l'opérateur de modification des relations inter-individuelles correspondant à la catégorie émotionnelle OCC
4. Écrire sur le noeud *output-link* l'état émotionnel ainsi que les relations inter-individuelles mises à jour

7.2.2.3 Contagion émotionnelle

La contagion émotionnelle est réalisée par l'environnement en suivant les étapes de l'algorithme 7.3.

Algorithme 7.3 Algorithme de contagion émotionnelle

1. Demander à chaque replicant de lui transmettre son état émotionnel et relations inter-individuelles depuis le noeud *output-link* de son SoarAgent associé
2. Calculer pour chaque replicant et chaque couple de replicants les caractéristiques individuelles à la contagion émotionnelles
3. Envoyer à chaque replicant son état émotionnel modifié

7.2.2.4 *Coping* et action

L'étape de *coping* consiste à évaluer les actions disponibles et à sélectionner la plus adaptée. C'est un processus cognitif (régulé par le SoarAgent de l'humain virtuel) qui suit les étapes de l'algorithme 7.4.

Nous utilisons le mécanisme de sous-buts de Soar, qui consiste à recopier l'état courant dans un sous-état et de lui appliquer les effets de l'action à évaluer. Ces effets sont considérés suivant plusieurs points de vue correspondant aux différentes évaluations à réaliser, qui peuvent être réalisées en parallèle.

Algorithme 7.4 Algorithme de coping et de décision

1. Pour chaque action possible, créer une *coping-frame*
   a) Évaluer l'état émotionnel résultant
   b) Évaluer la désirabilité par rapport aux marqueurs personnalisés
   c) Évaluer l'avancement dans l'arbre d'activité
   d) Combiner ces évaluations en fonction des préférences de l'humain virtuel
2. Sélectionner l'action ayant l'évaluation globale la plus haute
3. Écrire les paramètres de cette action sur le noeud *output-link*

- L'application des règles d'*appraisal,* réalisée de la même manière que pour évaluer l'état émotionnel réel de l'humain virtuel, permet d'obtenir l'état émotionnel dans lequel il serait si l'action venait à être réalisée.
- La désirabilité par rapport aux marqueurs personnalités consiste à calculer le nombre de tâches marquées que cette action permet de terminer, ou le nombre de conditions marquées qu'elle permet de valider.
- L'avancement dans l'arbre d'activité est évalué en comptant le nombre de contraintes relatives à l'arbre d'activité (conditions, tâches finalisées) qui sont levées (ou ajoutées) par cette action.

Ces évaluations ainsi que la fonction d'aggrégation des différentes évaluations permettant d'obtenir une évaluation globale sont définis par des règles génériques, définies en Annexe (voir A.3.2).

7.2.2.5 Action

Lors de la phase d'action, le replicant lit sur le noeud *output-link* l'action décidée par le SoarAgent et la transmet à l'environnement (algorithme 7.5. Les effets de l'action (réussite ou échec) ne sont pas communiqués directement à l'humain virtuel. Cette information est inférée lors de la phase de perception et d'élaboration de la situation suivante, au cours de laquelle il vérifie que l'état du monde correspond (ou non) aux effets attendus.

Algorithme 7.5 Algorithme d'action

1. Lire l'action demandée sur le noeud *output-link*
2. Envoyer une demande d'action à l'environnement

7.3 HUMANS-GEN

HUMANS-GEN est l'outil qui opérationnalise les connaissances décrites avec les formalismes HUMANS-DL en connaissances utilisables par les humains virtuels de REPLICANTS (figure 7.3.1). J'ai développé HUMANS-GEN en collaboration avec V. Lanquepin, doctorant à l'Université de Technologie de Compiègne depuis septembre 2012.

Deux types de connaissances sont utilisés par les humains virtuels : celles relatives au modèle du monde, c'est-à-dire les entités et les actions (DOMAIN-DL, décrit en 4.2), et les modèles d'activité (ACTIVITY-DL, voir 4.3). Les premières sont transformées en règles d'action et les secondes en arbres d'activités dans un formalisme interprétable par les replicants (règles Soar). Elles s'associent ensuite aux règles génériques afin de permettre de spécialiser les raisonnements et d'assurer l'adaptation du replicant dans son environnement.

Après avoir décrit le processus général et les modules impliqués, nous décrivons la librairie NGS (Soar New Goal System) puis les extensions que nous avons apportées afin de permettre cette génération.

7.3.1 Processus général

La transformation des connaissances sur le monde et sur les modèles d'activité suit le processus décrit par la figure 7.3.2. L'objectif est de transformer des fichiers XML de description d'entités, d'actions et d'activités (à gauche) en un ensemble de règles Soar (à droite).

Tout d'abord, un **parseur** interprète les fichiers contenant ces descriptions afin de construire un modèle de données.

Ce modèle est ensuite utilisé par un **générateur** qui produit deux ensembles de fichiers. Le premier regroupe l'ensemble des règles d'action, au format Soar. Le second rassemble des fichiers qui contiennent les arbres d'activités, au format ACTIVITY-NGS.

Le **moteur NGS** transforme ensuite ces règles depuis le formalisme ACTIVITY-NGS vers le formalisme NGS, puis génère des fichiers Soar classiques contenant ces informations.

7.3.2 NGS

NGS (pour *New Goal System*) est une librairie Soar développée par SoarTech [7] pour faciliter le développement d'agents Soar, en encapsulant un certain nombre d'éléments « bas niveau ».

7. http ://www.soartech.com/

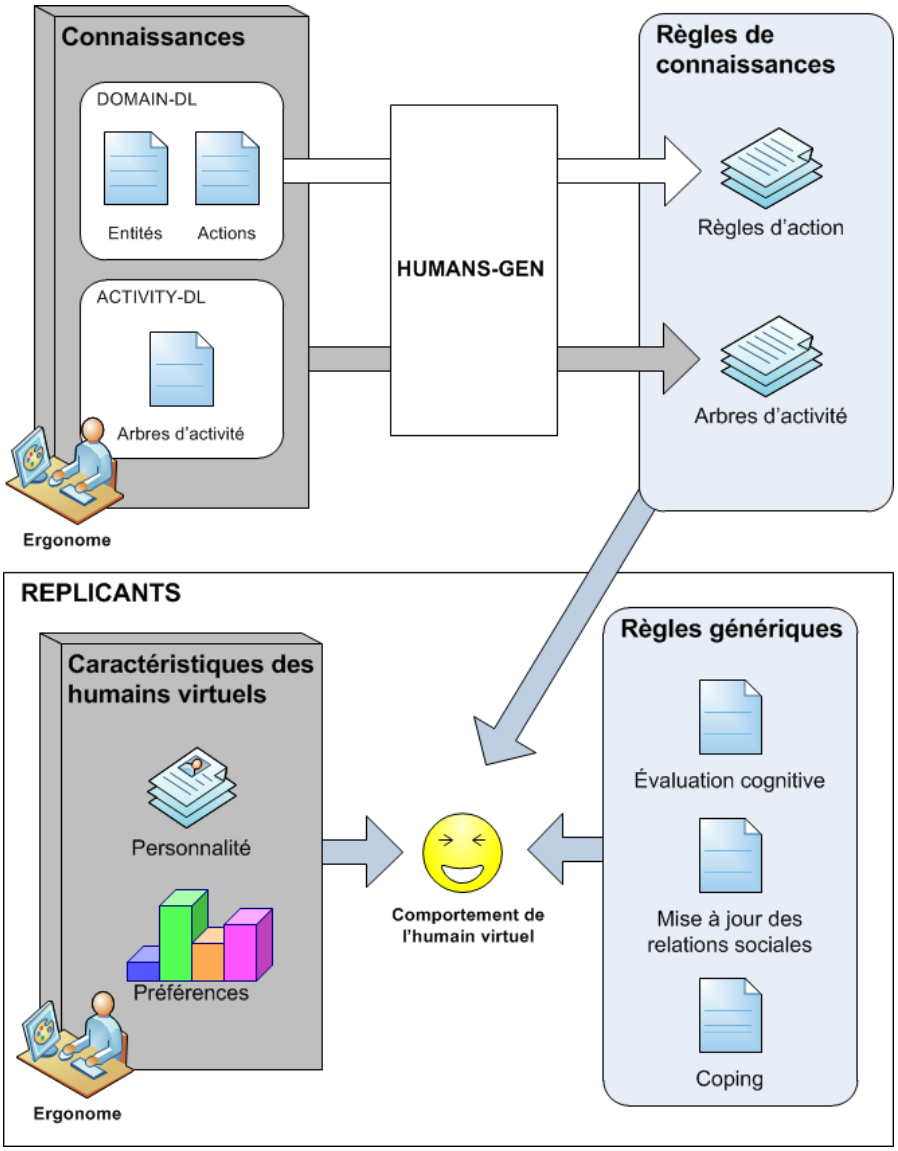

FIGURE 7.3.1: HUMANS-GEN, le module d'opérationnalisation des modèles de connaissances

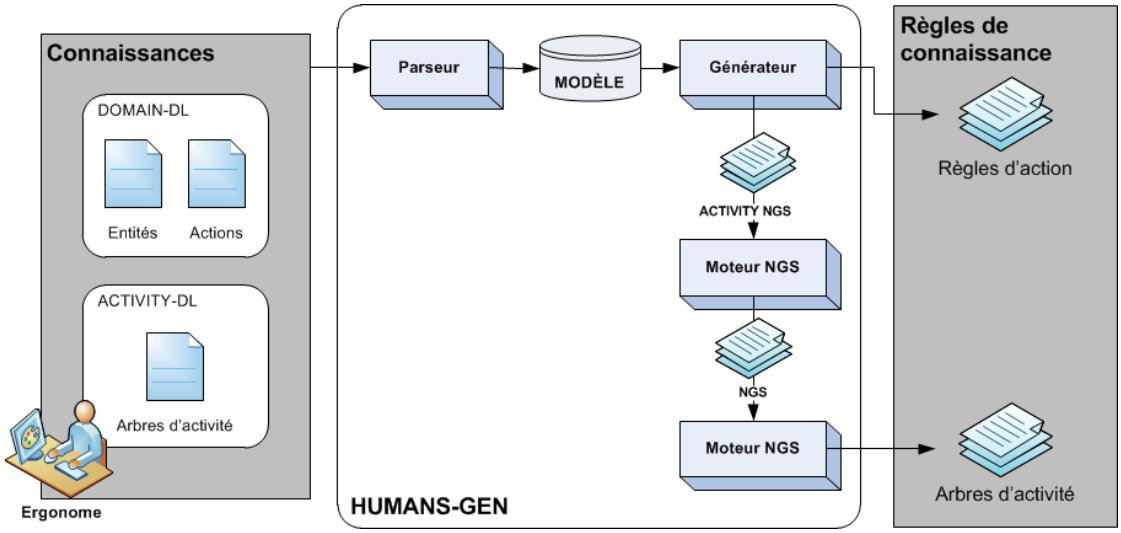

FIGURE 7.3.2: HUMANS-GEN - Processus d'opérationnalisation

Le formalisme NGS consiste en un ensemble de macros qui sont des templates d'écriture de règles. L'écriture de règles dans ce formalisme est donc un mélange de règles Soar classiques, et de macros NGS. Le moteur NGS permet de remplacer ces macros par le code correspondant et ainsi de générer des fichiers de règles en Soar « pur ».

NGS propose d'abstraire les connaissances à travers une hiérarchie de concepts, définis sur la figure 7.3.3.

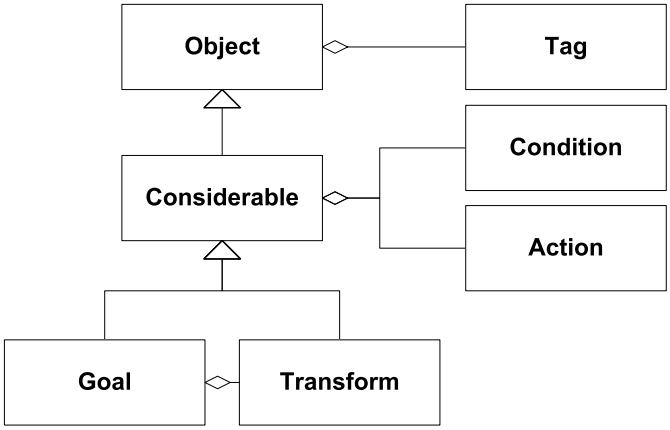

FIGURE 7.3.3: NGS - Concepts

7.3.2.1 Object

Un *Object* est l'élément de base de la hiérarchie de concepts définis par la librairie NGS. Un *Object* n'a aucune fonctionnalité particulière, mais dispose d'un certain nombre de caractéristiques qui permettent aux macros NGS de fonctionner.

En particulier, un *Object* dispose d'un type (ou d'une liste des types dont il hérite), et éventuellement d'un ou plusieurs *Tags.*

Un *Tag* est un attribut qui permet de regrouper un certain nombre d'informations utilisateurs sans nuire à la lisibilité de l'*Object. Tag* et attribut sont identiques, et l'unique différence réside dans leur position dans l'*Object* : un attribut est attaché à la racine de l'*Object,* tandis qu'un Tag est rattaché à l'attribut *Tag (*lui même attaché à *l'*Object).

NGS propose un certain nombre de fonctions permettant de rechercher des *Objets* d'un certain type ou présentant certain *Tags.*

7.3.2.2 Considerable

Un *Considerable* est un objet sur lequel l'humain virtuel peut raisonner explicitement. Il se trouve dans l'un des états définis par la figure 7.3.4.

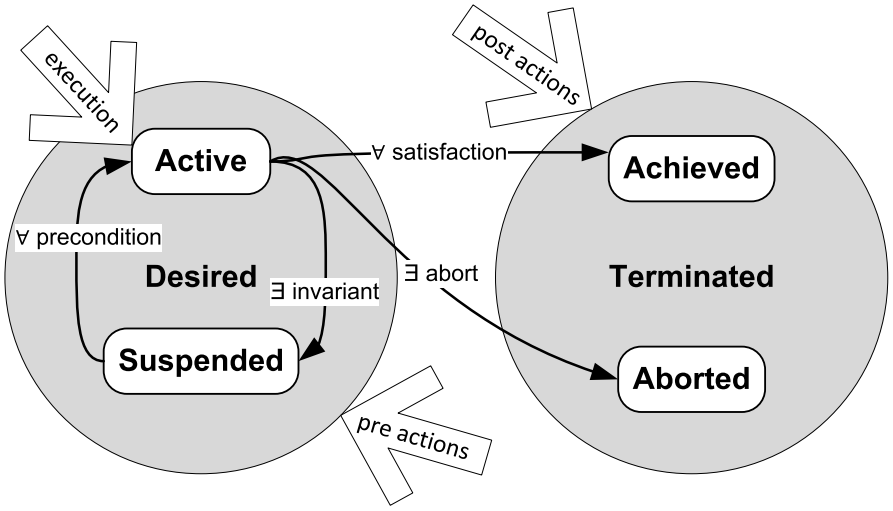

FIGURE 7.3.4: NGS - États d'un *Considerable*, *Conditions* et *Actions*

Lorsqu'un *Considerable* est dans l'état *Desired*, l'humain virtuel connait l'existence du considerable mais il ne lui a pas nécessairement affecté de ressources. L'état *Desired* comporte deux sous-états, *Active*, lorsque des ressources sont affectées, et *Suspended* lorsque les ressources sont temporairement désallouées. Un désirable qui se trouve dans l'état *Terminated* est considéré comme terminé, et

deux sous-états existent : *Achieved* lorsque le *Considerable* est réussi, et *Aborted* quand il a été définitivement arrêté. Un *Considerable* peut passer de l'un à l'autre des états *Active* à *Suspended*, par contre l'atteinte de l'état *Terminated* et de l'un de ses sous-états est définitif.

À un *Considerable* peuvent être attachés deux types d'entités : les *Actions* et les *Conditions.*

Une *Action* est une structure déclarative attachée à un état. Lors de l'entrée dans cet état, l'*action* est déclenchée, ce qui permet d'effectuer des changements dans la mémoire de travail ou d'envoyer un message. Nous n'utilisons pas les *actions* dans ACTIVITY-GEN et ne détaillons donc pas leur fonctionnement.

7.3.2.3 Conditions

Une *Condition* est une structure NGS qui permet de contrôler les changements d'état du *Considerable* auquel elle est attachée.

Une condition est déterminée par son rôle qui définit l'impact qu'a sa vérification sur l'état du *Considerable.* Il sont représentés sur la figure 7.3.4 et repris ici :

- *precondition :* toutes les conditions ayant ce rôle doivent être vérifiées pour que le *Considerable* puisse passer de l'état *Suspended* à l'état *Active ;*
- *invariant :* si une seule de ces conditions n'est plus respectée lorsque le *Considerable* est actif, il repasse dans l'état *Suspended* ;
- *satisfaction* : si toutes les conditions de *satisfaction* sont validées, un *Considerable* actif passe dans l'état *Achieved ;*
- *aborted :* si une seule de ces conditions n'est pas respectée, le *Considerable* passe de l'état *Active* à l'état *Aborted.*

Afin d'assurer une grande expressivité, NGS définit plusieurs types de *Condition*, regroupés dans le tableau 7.1. Il est donc possible de tester l'état et éventuellement la valeur d'un *Considerable*, l'existence d'un attribut ou *Tag* particulier.

7.3.2.4 Goals

Le *Goal* Est l'élément de base utilisé pour développer des agents Soar avec NGS et permettent de modéliser l'objectif d'atteindre un état du monde particulier.

Un *Goal* peut être défini comme le sous-but d'un autre *Goal.* Dans ce cas, tout changement d'état concernant l'activation du *Goal* principal implique le changement correspondant pour ses sus-buts. Ainsi, si un but passe de l'état *Active* à l'état *Suspended*, tous ses sous-buts sont immédiatement suspendus. Un *Goal* peut être le sous-but de plusieurs *Goals.*

Plusieurs types de Goals sont prédéfinis :

- *Achievement* : ce *Goal* est considéré comme réalisé une fois l'état du monde désiré atteint.

Type	Paramètres	Conditions de validation
active	*<cons>*	le considerable passé en paramètre est dans l'état *Active*
achieved	*<cons>*	le considerable passé en paramètre est dans l'état *Achieved*
unachievable	*<cons>*	le considerable passé en paramètre est dans l'état *Aborted*
has-tag	<obj> <tag>	l'objet passé en paramètre a un tag du nom passé en paramètre
has-tag-value	<obj> <tag> <vak>	l'objet passé en paramètre a un tag du nom et de la valeur passés en paramètre
has-attribute	<obj> <attr>	l'objet passé en paramètre a attribut du nom passé en paramètre
has-attribute-value	<obj> <attr> <val>	l'objet passé en paramètre a un attribut du nom et de la valeur passés en paramètre

TABLE 7.1: NGS - Types de *conditions*

– *Maintenance :* ce *Goal* vise à maintenir en continu l'état du monde désiré.

À un *Goal* peut être associé un ou plusieurs *Transforms* (figure *7.3.5*). Un *Transform* est une représentation interne des actions à réaliser. Plus techniquement, une fois le *Goal* dans l'état *Active*, le *Transform* est déclenché ce qui créé un operateur Soar. Ce dernier permet de réaliser une demande d'action en écrivant les informations nécessaires sur le noeud *output-link* afin de signifier qu'une décision a été prise.

7.3.3 ACTIVITY-NGS

ACTIVITY-NGS est une surcouche de NGS pour la modélisation d'arbres d'activité. Nous avons défini un ensemble de macros afin de pouvoir modéliser les spécificités du langage ACTIVITY-DL dans NGS. Ces macros sont utilisées par le générateur pour traduire le modèle en fichier ACTIVITY-NGS et sont listées en annexe (voir B). L'algorithme utilisé est défini par l'algorithme 7.6.

Une fois cette opération effectuée, le moteur NGS transforme les macros ACTIVITY-NGS en leur équivalent NGS, puis finalement en règles Soar.

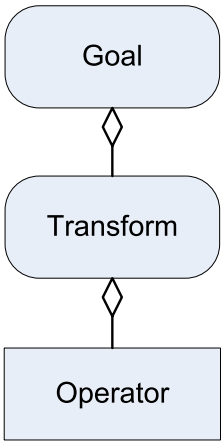

FIGURE 7.3.5: NGS - Lien entre les *Goals* et les *Operators Soar*

Algorithme 7.6 Algorithme de génération des fichiers Activity-NGS

1. Créer les Tâches mères : *NGS_ACT_goal_create-goal-XXX* (avec XXX le constructeur d'Allen liant les tâches filles)
2. Créer les Tâches feuilles : *NGS_ACT_goal_create-goal-perform*
3. Créer les conditions et les associer aux goals : *NGS_ACT_goal_create-condition-XXX* (avec XXX le type de condition)

7.3.4 Conclusion

Nous avons présenté HUMANS-GEN, l'outil qui permet de transformer les connaissances spécifiées dans les formalismes de HUMANS-DL, en particulier DOMAIN-DL et ACTIVITY-DL, en fichiers de règles Soar interprétables par les humains virtuels. Nous avons vu précédemment la difficulté rencontrée au cours de l'écriture de règles Soar. La librairie NGS vise à faciliter ce travail en proposant une abstraction des concepts à travers l'utilisation de macros. L'objectif de HUMANS-GEN est de simplifier encore plus ce processus en l'automatisant. L'écriture des règles se fait de manière implicite via la spécification des entités, des actions et des activités dans les formalismes de HUMANS-DL et un outil auteur, EDITOR.

L'utilisation de la librairie NGS et l'extension que nous lui avons apportée, ACTIVITY-NGS permet une très grande expressivité que nous pourrions facilement étendre ou adapter si des modifications des langages de description de HUMANS-DL devaient survenir.

HUMANS-GEN est quant à lui le premier outil à générer automatiquement des règles Soar à partir d'une ontologie.

7.4 Conclusion

Nous avons vu que ces humains virtuels sont dotés de règles génériques qui leur permettent d'effectuer des raisonnements de haut niveau. Ils disposent de processus leur permettant d'évaluer leur état émotionnel aux vues de leurs perceptions, ainsi que d'évaluer de différentes manières les différentes manières d'y réponde, en particulier en inscrivant leurs actions dans un contexte d'activité.

Nous avons présenté la plateforme HUMANS qui forme une suite de génération d'environnements virtuels pour la formation complète, avec des modules de suivi de l'apprenant, de scénarisation et d'humains virtuels. Ces modules partagent une même base de connaissance, décrite avec les formalismes de HUMANS-DL.

Nous avons ensuite décrit REPLICANTS, le module de génération d'humains virtuels qui implémente le modèle d'humain virtuel présenté précédemment.

Enfin, nous avons expliqué le fonctionnement de HUMANS-GEN, un outil qui permet aux humains virtuels de REPLICANTS d'interpréter les connaissances définies avec le formalisme HUMANS-DL.

Chapitre 8

Cas d'usage et résultats

Ce chapitre propose trois cas d'usage afin de montrer l'application de nos modèles sur des exemples concrets. Le premier cas s'intéresse uniquement à la contagion émotionnelle, le second prend également en compte l'évaluation cognitive et un processus de *coping* simple. Le troisième exemple montre comment l'humain virtuel combine ses règles génériques et les connaissances spécifiques afin de se comporter de manière cohérente en inscrivant ses actions dans un cadre rationnel dirigé par des buts.

8.1 Contagion émotionnelle

Dans ce premier exemple, nous proposons une implémentation simple de notre modèle de contagion émotionnelle.

Trois humains virtuels sont mis en situation, Bob, Anna et Pablo. Aucune action ne se produisant dans l'environnement, l'humeur des humains virtuels évolue donc uniquement à cause de la contagion émotionnelle.

8.1.1 Personnalité et relations interindividuelles

Les personnalités OCEAN de Bob, Anna et Pablo sont indiquées dans les matrices 8.1.1, et leurs relations dans le tableau 8.1. Pour rappel, les dimensions d'une matrice de personnalité sont définies par l'équation 5.1.1 et celles de la matrice de relations sociales par l'équation 5.1.2.

$$P_{Bob} = \begin{bmatrix} 0.78 \\ 0.6 \\ -0.4 \\ -0.4 \\ 0.6 \end{bmatrix} \quad P_{Anna} = \begin{bmatrix} -0.6 \\ 0.16 \\ -0.56 \\ -0.36 \\ 0.94 \end{bmatrix} \quad P_{Pablo} = \begin{bmatrix} -0.06 \\ -0.18 \\ 0.06 \\ 0.38 \\ 0.92 \end{bmatrix} \tag{8.1.1}$$

	Bob	**Anna**	**Pablo**
Bob		$R_{Bob/Anna} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$R_{Bob/Pablo} = \begin{bmatrix} 0.8 \\ 0 \\ 1 \\ 0.5 \end{bmatrix}$
Anna	$R_{Anna/Bob} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$		$R_{Anna/Pablo} = \begin{bmatrix} 0.6 \\ 0.5 \\ 0.3 \\ 0 \end{bmatrix}$
Pablo	$R_{Pablo/Bob} = \begin{bmatrix} 0.5 \\ 1 \\ 0.2 \\ 0.9 \end{bmatrix}$	$R_{Pablo/Anna} = \begin{bmatrix} 0.5 \\ 1 \\ 0.2 \\ 0.9 \end{bmatrix}$	

TABLE 8.1: Relations inter-personnelles

Bob et Anna ont des relations très mauvaises (valeurs nulles pour chaque composante des relations sociales) ; par contre, tous deux apprécient Pablo. De son côté, Pablo a des relations identiques avec Bob et Anna. Ainsi, le canal de communication entre Bob et Anna est nul dans les deux sens, alors qu'il est de force moyenne entre Pablo et Bob et Pablo et Anna.

8.1.2 Contagion émotionnelle et caractéristiques individuelles

L'application des règles définies en 7.2.2.3 permettent d'obtenir les valeurs de pouvoir de contagion (8.1.2) et de susceptibilité à la contagion [1] (8.1.2) de chaque humain virtuel. Le tableau 8.2 indique les résultats du calcul de la force du canal de communication.

1. Les valeurs indiquées ici correspondent à la susceptibilité à la contagion pour une humeur neutre.

$$\eta_{Bob} = 0.2 \quad \eta_{Anna} = 0.28 \quad \eta_{Pablo} = 0.125 \tag{8.1.2}$$
$$\delta_{Bob} = 0.436 \quad \delta_{Anna} = 0.172 \quad \delta_{Pablo} = 0.120 \tag{8.1.3}$$

	Bob	Anna	Pablo
Bob		$\alpha_{Anna/Bob} = 0$	$\alpha_{Pablo/Bob} = 0.575$
Anna	$\alpha_{Bob/Anna} = 0$		$\alpha_{Pablo/Anna} = 0.35$
Pablo	$\alpha_{Bob/Pablo} = 0.65$	$\alpha_{Anna/Pablo} = 0.65$	

TABLE 8.2: Force du canal de communication

8.1.3 Résultats

La figure 8.1.1 montre l'influence de la contagion émotionnelle sur l'évolution de l'humeur dans un cadre où les humains virtuels sont initialisés avec des humeurs arbitraires et où aucun évènement ne se produit dans l'environnement.

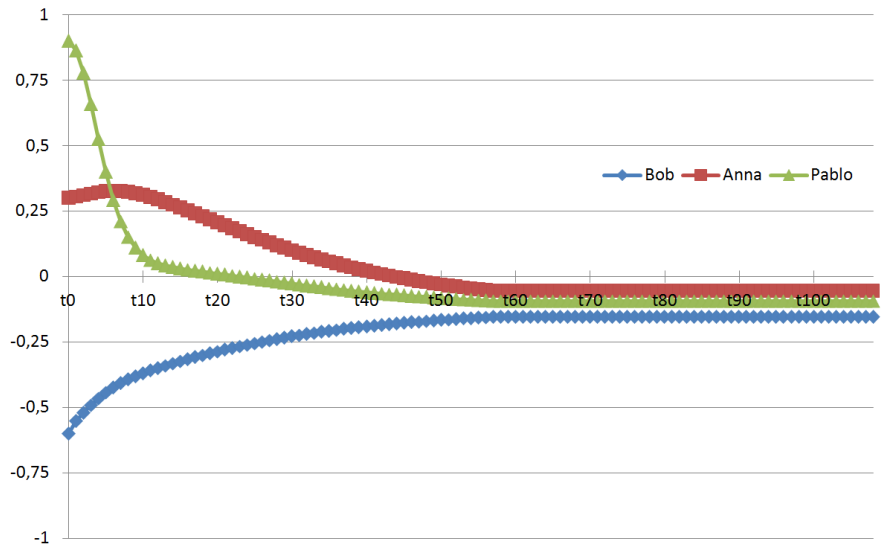

FIGURE 8.1.1: Évolution de l'humeur dans le temps

On peut remarquer que les états émotionnels des humains virtuels convergent assez rapidement. On peut également remarquer l'importance des relations sociales : entre t_0 et t_{10}, l'humeur d'Anna tend vers celle de Pablo, qu'elle apprécie, et ignore Bob, qu'elle n'apprécie pas. De plus, entre t_{10} et t_{50}, l'humeur de Pablo,

bien qu'influencée par celle de Bob, est tout de même plus proche de celle d'Anna. Ce phénomène s'explique par le fait qu'Anna a une plus forte capacité de contagion ($\eta_{Anna} = 0.28$) que Bob ($\eta_{Pablo} = 0.125$), et n'est pas lié à la force du canal de communication car Pablo a les mêmes relations sociales avec Anna qu'avec Bob.

8.2 Contagion émotionnelle, évaluation cognitive et *coping* simple.

Dans un second exemple, nous proposons d'intégrer l'évaluation cognitive ainsi que la prise de décision dans un scénario simple de panique au cours duquel des feux se déclenchent dans l'environnement. Les humains virtuels peuvent choisir entre cinq actions : ne rien faire ou se déplacer dans l'une des quatre directions cardinales ("stay", "moveNorth", "moveSouth", "moveEast", "moveWest"). La position des humains virtuels ainsi que leur personnalité et les relations inter-individuelles qui les lient sont générées aléatoirement à l'initialisation de la simulation.

Pour des raisons de simplification, cet exemple ne comporte que des évènements. Les relations sociales ne sont donc pas modifiées au cours du déroulement du scénario. De plus, nous ignorons les dimensions temporelles. D'après la classification OCC, les types d'émotions intégrées dans cet exemple sont donc la *Joy* et la *Distress*.

Ce scénario n'utilise pas les arbres d'activité, et considère une seule stratégie de *coping* qui est l'évaluation cognitive des actions réalisables.

8.2.1 Évaluation cognitive

8.2.1.1 Création de l'*appraisal-frame*

La structure de *l'appraisal-frame* que nous utilisons ici ne contient qu'une dimension qui correspond à la désirabilité agrégée de l'évènement, cet exemple ne prenant pas en compte les arbres d'activité. Pour chaque évènement, une *appraisal-frame* est créée par l'application de la règle *elaborations*elaborate*appraisal-frame*create* issue des règles génériques de l'humain virtuel

```
sp {elaborations*elaborate*appraisal-frame*create
   (state <s> ^perceptions <p>
              ^appraisal <app>)
   (<p> ^event <e>)
-->
   (<app> ^frame <f>)
   (<f> ^event <e>)
}
```

8.2.1.2 Évaluation de la désirabilité

L'évaluation de la désirabilité d'un évènement est réalisée par l'application d'un ensemble de règles Soar. Dans notre exemple, les humains virtuels n'ont qu'un seul but, celui de rester en vie, et les feux qui se déclarent dans l'environnement sont toujours jugés indésirables.

La règle Soar *evaluate*desirability*event*fire* permet d'associer, à un évènement perçu une désirabilité calculée en fonction de sa distance : plus l'évènement est près, plus il est indésirable.

```
sp {evaluate*desirability*event*fire
  (<s> ^perceptions.event <event>)
  (<event> ^type <fire>
           ^distance <d>)
-->
  (<event> ^desirability (- 100 <d>))
}
```

La règle générique *elaborations*elaborate*appraisal-frame*dimension*desirability* recopie cette valeur de désirabilité dans l'*appraisal-frame* correspondante :

```
sp {elaborations*elaborate*appraisal-frame*dimension*desirability
   (state <s> ^appraisal.frame <f>)
   (<f> ^event <e>)
   (<e> ^desirability <d>)
-->
   (<f> ^desirability <d>)
}
```

Une fois l'intégralité des *appraisal-frames* évaluées, elles sont aggrégées dans une *appraisal-frame* qui représente l'état émotionnel de l'humain virtuel. Cet état émotionnel est transmis à l'environnement via une action *export-mood*.

8.2.2 Contagion émotionnelle

L'environnement réalise la contagion émotionnelle (décrite en 7.2.2.3) entre les humains virtuels en considérant leur position et leurs caractéristiques individuelles. Un humain virtuel ne peut être contaminé et ne peut contaminer qu'un individu qu'il perçoit.

8.2.3 Prise de décision et coping

Ce processus est réalisé en plusieurs phases. Tout d'abord, les actions envisageables sont proposées. Par exemple, la règle *propose*action*moveNorth* correspond à l'action de déplacement vers le nord.

```
sp {propose*action*moveNorth
  (<s> ^agent-position <my-position>)
  (<my-position> ^at-north.free true)
-->
  (<s> ^operator <op> +)
  (<op> ^name moveNorth)
}
```

Dans le cas où plusieurs actions sont possibles, l'humain virtuel doit choisir celle qu'il préfère. Dans le formalisme Soar, cela s'appelle une *Impasse* : l'humain virtuel ne dispose pas de suffisamment ou au contraire, de trop, d'information. L'humain virtuel doit alors appliquer les stratégies de *coping* dont il dispose afin de faire un choix. L'unique stratégie de coping utilisée dans ce scénario consiste à évaluer l'état émotionnel dans lequel l'humain virtuel se trouverait s'il effectuait l'action. Pour réaliser cela, l'agent Soar recopie la situation courante et lui applique mentalement l'action. La règle *apply*action*moveNorth* correspond aux changements induits par un déplacement vers le nord.

```
sp {apply*action*moveNorth
  (<s> ^operator.name moveNorth
       ^agent-position <my-position>)
  (<my-position ^at-north <north-cell>)
-->
  (<s> ^agent-position <my-position>
      -^agent-position <north-cell>)
}
```

De cette opération résulte un nouvel état dans lequel sont modélisés les changements dus à l'action. Ici, c'est la position de l'humain virtuel qui a été modifiée. Ce changement de position mène à une modification de la distance des feux, et donc à une nouvelle évaluation de le désirabilité de chaque *event* de type feu perçu. Les règles d'évaluation cognitive s'appliquent de nouveau ce qui permet d'associer un état émotionnel à l'état résultant de cette action.

Une fois que l'humain virtuel a évalué chaque action possible, il sélectionne celle dont l'impact émotionnel est maximal. Les états imaginés sont supprimés et l'humain virtuel peut transmettre l'action choisie à l'environnement.

8.2.4 Résultats

La figure 8.2.1 regroupe deux captures d'écran de REPLICANTS prises lors de l'exécution du scénario décrit ci-dessus.

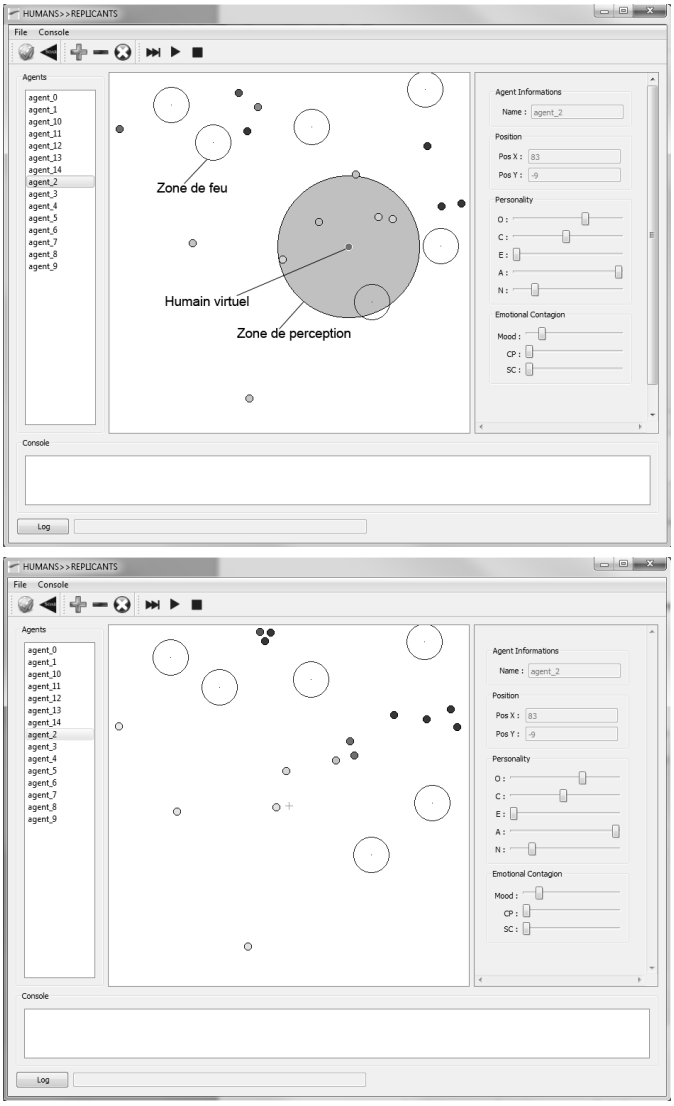

FIGURE 8.2.1: Captures d'écran de REPLICANTS montrant l'évolution de la situation

La capture d'écran du haut montre la situation initiale et celle du bas la situation après un certain temps. Chaque petit disque représente un replicant, dont le niveau de gris indique l'humeur. Plus le disque est clair, meilleure est l'humeur. Dans la situation initiale, le large disque grisé autour du replicant sélectionné représente

sa zone de perception. Les cercles de taille moyenne représentent les feux présents dans l'environnement. Il est possible de les déclencher via un clic droit dans la fenêtre.

On remarque tout d'abord que les replicants ont essayé de rallier les positions non exposées aux feux. Pour certains la situation est sans issue. Le groupe de trois replicants au nord est bloqué par des feux : chacun d'entre eux se trouve dans un état émotionnel désastreux, entretenu par la proximité des feux et par la contagion émotionnelle. Ceux situés au centre de l'environnement se sont regroupés dans une zone non exposée, et leur humeur s'est améliorée.

8.3 Utilisation des arbres d'activité lors de la phase de *coping*.

Notre troisième exemple vise à montrer le fonctionnement de notre modèle de *coping*. L'humain virtuel mis en situation tente de partir en vacances. Il dispose d'un modèle d'activité qui décrit les étapes généralement réalisées pour arriver à ce but.

8.3.1 Arbre d'activité

La figure 8.3.1 représente les principaux éléments de ce modèle d'activité, défini dans le formalisme ACTIVITY-DL. Le code complet de cet exemple, au format XML se trouve en Annexe C.1.

Ce modèle est constitué d'un seul arbre dont la tâche principale est *Go To Holidays.* Cette tâche est composée de deux tâches filles, *Get Ready* et *Leave*, liées par un opérateur *SEQ-ORD* : *Get Ready* doit être réalisée avant *Leave.* La condition de réussite de la tâche mère est définie par la réussite des deux tâches filles (*AND*).

Get Ready est également composée de deux sous tâches, *Load Car* et *Close House*, indépendantes l'une de l'autre. Ici aussi, toutes deux doivent être réussies pour que la tâche mère le soit (opérateur *AND*).

Load Car est une tâche feuille, c'est-à-dire qu'elle déclenche une action dans l'environnement. Les conditions nomologiques et de réussite de cette tâche sont les conditions d'exécution et les effets de l'action associée.

Close House est une tâche mère, composée de deux tâches filles concurrentes. La condition de réussite de la tâche mère est un opérateur *XOR* sur la réussite des sous-tâches, la tâche mère ne peut donc être réussie que si une seule des deux sous-tâches est réussie. Les tâches *Close Door* et *Lock Door* sont donc deux manières distinctes de résoudre la tache *Close House.* La première consiste à fermer la porte

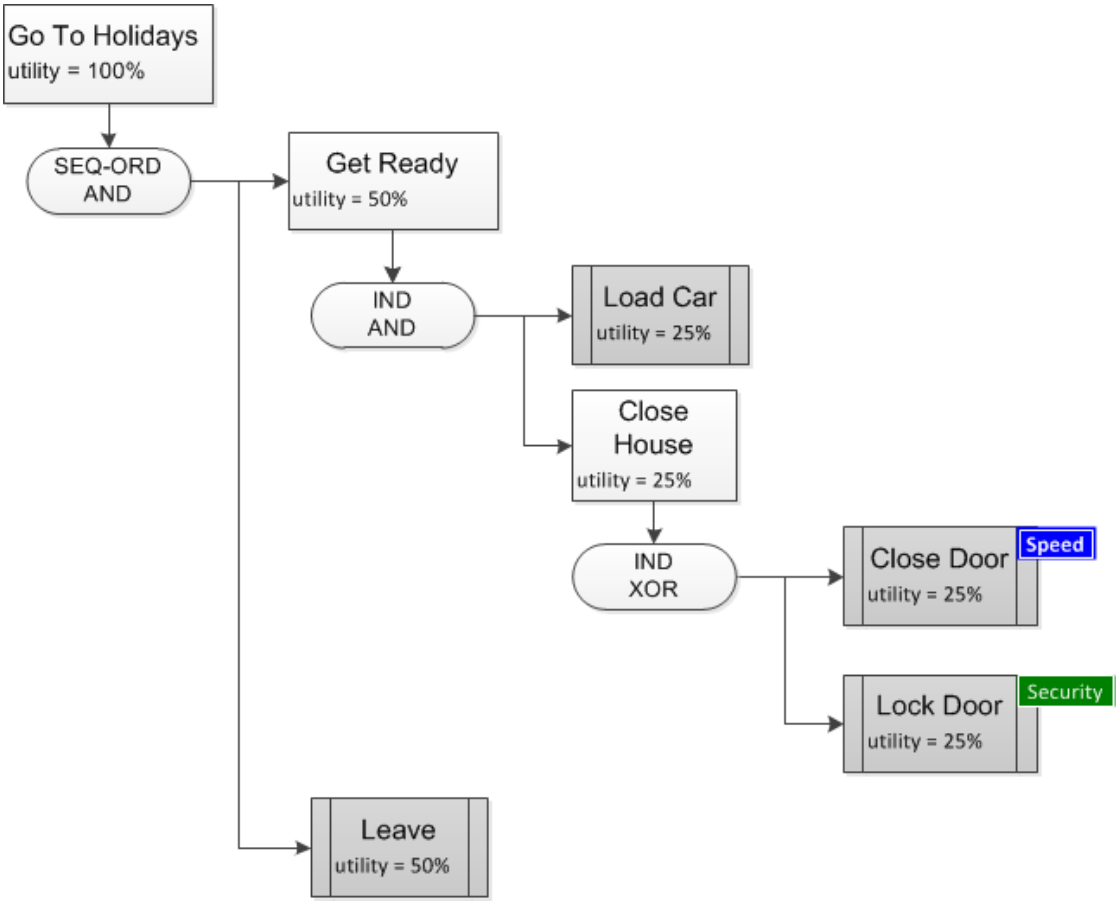

FIGURE 8.3.1: Arbre d'activité et utilités initiales

et est marquée comme rapide (marqueur *speed*), tandis que la seconde vise à la verrouiller et apporte un gain de sécurité (marqueur *security*).

Enfin, une fois cette première partie de l'activité réalisée, l'humain virtuel peut considérer la dernière tâche feuille, *Leave*, qui déclenche une action dans l'environnement.

La valeur d'utilité associée à chaque tâche correspond à l'avancement dans l'arbre qu'elle procure. Ces valeurs sont générées automatiquement par ACTIVITY-GEN en fonction de l'organisation de l'arbre, et mises à jour dynamiquement. Ici, les tâches filles nécessaires à la réussite d'une tâche mère se distribuent équitablement l'utilité de cette dernière.

On notera que dans cet exemple, les marqueurs personnalisés sont associés de manière binaire aux tâches. ACTIVITY-DL permet de spécifier la valeur de l'apport de la tâche (ou de la condition) relatif au marqueur (une tâche peut être marquée comme 50% *speed* et 50% *security*).

8.3.2 Marqueurs et préférences

Ce modèle d'activité utilise deux marqueurs, *security* et *speed.* L'humain virtuel de ce scénario accorde 80% d'importance à la sécurité, et 20% à la vitesse de réalisation de son activité.

```
#Définition de l'importance des marqueurs personnalisés
sp {propose*initialize-agent*tags
   (state <s> ^self.preferences <p>)
-->
   (<s> ^operator <op>+)
   (<op> ^name init-tags
         ^tags <t>)
   (<t> ^security 80
        ^speed 20)}
sp {apply*initialize-agent*tags
   (state <s> ^self.preferences <p>
              ^operator <op>)
    (<op> ^name init-tags
          ^tags <t>)
-->
   (<p> ^tags <t>)}
```

On considère ici que l'humain virtuel n'a pas de préférences quant à l'importance des dimensions de la *coping-frame.* Les deux dimensions ont donc le même poids dans le calcul de l'évaluation globale de la *coping-frame.*

```
sp {propose*initialize-agent*coping-preferences
```

```
   (state <s> ^self.preferences <p>)
-->
   (<s> ^operator <op>+)
   (<op> ^name init-coping-preferences
         ^preferences <cp>)
   (<cp> ^tags 1
         ^activity 1)}
sp {apply*initialize-agent*tags
   (state <s> self.preferences <prefs>)
              ^operator <op>)
    (<op> ^name init-coping-preferences
          ^tags <cp>)
-->
   (<p> ^coping-preferences <cp>)}
```

8.3.3 Actions possibles

À chaque pas de temps, l'humain virtuel doit sélectionner une des actions parmi celles qui sont possibles. Ce scénario recense quatre actions dont les noms sont, pour des raisons de compréhension, identiques à ceux des tâches feuilles qu'elles permettent de réaliser, et une cinquième action (*Take Key*) définie dans le modèle d'action. Le code de ces règles d'action est en Annexe C.2.

- *Load Car*
  - Précondition : l'humain virtuel n'est pas parti.
- *Close Door*
  - Préconditions : la porte n'est pas fermée et l'humain virtuel n'est pas parti.
  - Effet : la porte est fermée.
- *Lock Door*
  - Préconditions : la porte n'est pas fermée, l'humain virtuel a la clé et l'humain virtuel n'est pas parti.
  - Effet : la porte est fermée et la porte est verrouillée.
- *Leave*
  - Précondition : l'humain virtuel n'est pas parti
  - Effet : l'humain virtuel est parti
- *Take Key*
  - Préconditions : l'humain virtuel n'est pas parti et l'humain virtuel n'a pas la clé.
  - Effet : l'humain virtuel a la clé.

8.3.4 Exécution et résultats

La figure 8.3.2 décrit le parcours de l'arbre réalisé par un humain virtuel et montre les valeurs des *coping-frames* à chaque pas de temps (t_0, t_1, $t_2$et t_3). Dans ce scénario, nous considérons uniquement le suivi de l'arbre d'activité. La dimension émotionnelle est ignorée. Une *coping-frame* est donc composée des dimensions relatives à l'avancement dans l'arbre et au respect des marqueurs, indiquées respectivement par « a » et « t ». Le sens interdit signifie que les préconditions de l'action ne sont pas respectées et qu'elle n'est donc pas réalisable.

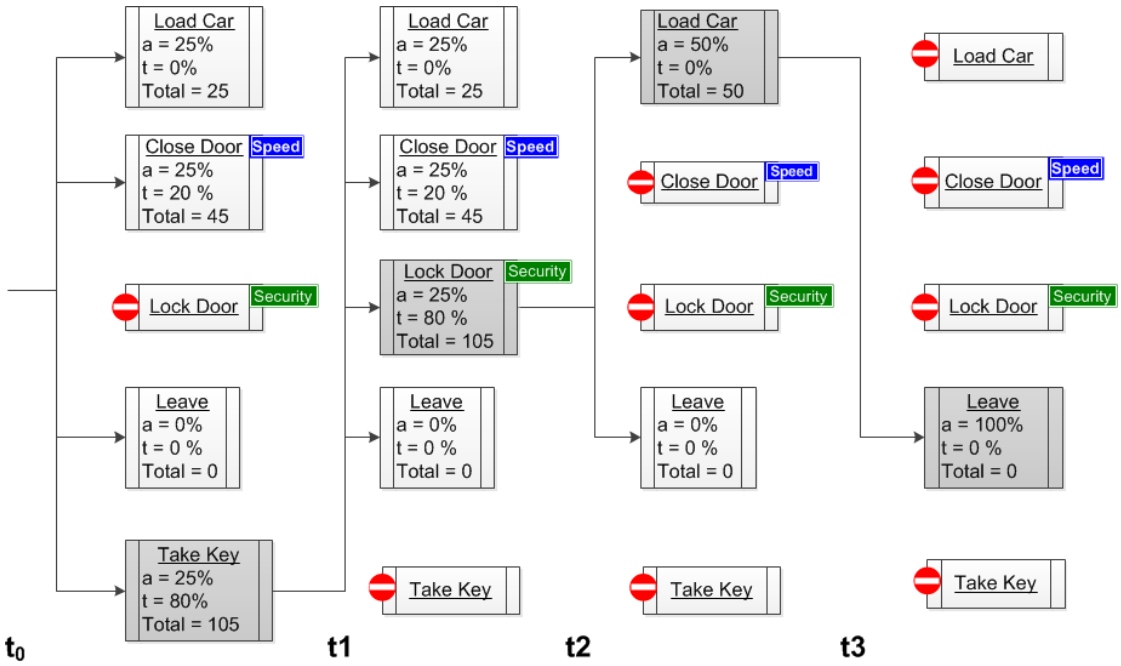

FIGURE 8.3.2: *Coping-frames* et comportement de l'agent

8.3.4.1 Instant t_0

À l'initialisation du scénario, quatre actions sont réalisables dans l'environnement. L'humain virtuel génère pour chacune d'entre elles une *coping-frame :*

Load Car

- L'avancement dans l'arbre de l'action *Load Car* est de 25% : cette action permet de résoudre la tâche associée dont l'utilité est de 25%.
- La valeur de respect des marqueurs est égale à 0% car aucun marqueur n'est associé à la tâche.

– La valeur globale de la *coping-frame* est de 25.

Close Door

– L'avancement dans l'arbre de l'action *Close Door* est de 25% : cette action permet de résoudre la tâche associée dont l'utilité est de 25%.
– La valeur de respect des marqueurs est égale à 20% car le marqueur *Speed* est associé à la tâche et l'humain virtuel lui accorde 20% d'importance.
– La valeur globale de la *coping-frame* est de 45.

Leave

– L'avancement dans l'arbre de l'action Leave est de 0% :
  – cette action permet de résoudre la tâche associée dont l'utilité est de 50%.
  – la réalisation de cette action empêche la réalisation de la tâche *Get Ready* dont l'utilité est de 50% (après l'application de l'action *Leave*, l'humain virtuel est parti ce qui rend irréalisables la tâche *Get Ready*).
– La valeur de respect des marqueurs est égale à 0% car aucun marqueur n'est associé à la tâche.
– La valeur globale de la *coping-frame* est de 0.

Take Key

– L'avancement dans l'arbre de l'action *Take Key* est de 25% : cette action rend possible la tâche *Lock Door* dont l'utilité est de 25%.
– La valeur de respect des marqueurs est égale à 80% : cette action rend possible la tâche *Lock Door* dont la valeur de respect des marqueurs est de 80% (le marqueur *Security* est associé à la tâche et l'humain virtuel lui accorde 80% d'importance.)
– La valeur globale de la *coping-frame* est de 105.

Sélection de l'action

L'humain virtuel sélectionne l'action dont la valeur globale de *coping-frame* est la plus élevée, soit *Take Key.*

8.3.4.2 Instant t_1

À l'instant t_1, les quatre actions de l'arbre d'activité sont maintenant réalisables dans l'environnement.

Load Car

- L'avancement dans l'arbre de l'action *Load Car* est de 25% : cette action permet de résoudre la tâche associée dont l'utilité est de 25%.
- La valeur de respect des marqueurs est égale à 0% car aucun marqueur n'est associé à la tâche.
- La valeur globale de la *coping-frame* est de 25.

Close Door

- L'avancement dans l'arbre de l'action *Close Door* est de 25% : cette action permet de résoudre la tâche associée dont l'utilité est de 25%.
- La valeur de respect des marqueurs est égale à 20% car le marqueur *Speed* est associé à la tâche et l'humain virtuel lui accorde 20% d'importance.
- La valeur globale de la *coping-frame* est de 45.

Lock Door

- L'avancement dans l'arbre de l'action *Lock Door* est de 25% : cette action permet de résoudre la tâche associée dont l'utilité est de 25%.
- La valeur de respect des marqueurs est égale à 80% car le marqueur *Security* est associé à la tâche et l'humain virtuel lui accorde 80% d'importance.
- La valeur globale de la *coping-frame* est de 105.

Leave

- L'avancement dans l'arbre de l'action Leave est de 0% :
  - cette action permet de résoudre la tâche associée dont l'utilité est de 50%.
  - la réalisation de cette action empêche la réalisation de la tâche *Get Ready* dont l'utilité est de 50% (après l'application de l'action *Leave*, l'humain virtuel est parti ce qui rend irréalisables la tâche *Get Ready*).
- La valeur de respect des marqueurs est égale à 0% car aucun marqueur n'est associé à la tâche.
- La valeur globale de la *coping-frame* est de 0.

Sélection de l'action

L'humain virtuel sélectionne l'action dont la valeur globale de *coping-frame* est la plus élevée, soit *Lock Door.*

8.3.4.3 Instant t_2

À l'instant t_1, seules deux actions sont réalisables dans l'environnement. En effet, la réalisation de l'action *Lock Door* a rendu invalides les conditions nomologiques associées aux actions *Close Door* et *Lock Door* (une porte ne doit pas déjà être fermée ou verrouillée pour pouvoir être fermée ou verrouillée).

L'utilité de chaque tâche est mise à jour avant le début du processus de coping.

Load Car

- L'avancement dans l'arbre de l'action *Load Car* est de 50% : cette action permet de résoudre la tâche associée dont l'utilité est de 50%.
- La valeur de respect des marqueurs est égale à 0% car aucun marqueur n'est associé à la tâche.
- La valeur globale de la *coping-frame* est de 50.

Leave

- L'avancement dans l'arbre de l'action Leave est de 0% :
  - cette action permet de résoudre la tâche associée dont l'utilité est de 50%.
  - la réalisation de cette action empêche la réalisation de la tâche *Get Ready* dont l'utilité est de 50%.
- La valeur de respect des marqueurs est égale à 0% car aucun marqueur n'est associé à la tâche.
- La valeur globale de la *coping-frame* est de 0.

Sélection de l'action

L'humain virtuel sélectionne l'action dont la valeur globale de *coping-frame* est la plus élevée, soit *Load Car.*

8.3.4.4 Instant t_3

À l'instant t_3, seule l'action *Leave* est réalisable dans l'environnement. L'utilité de chaque tâche est mise à jour avant le début du processus de coping.

Leave

- L'avancement dans l'arbre de l'action Leave est de 100% : cette action permet de résoudre la tâche associée dont l'utilité est de 100%.
- La valeur de respect des marqueurs est égale à 0% car aucun marqueur n'est associé à la tâche.

– La valeur globale de la *coping-frame* est de 100.

Sélection de l'action

L'humain virtuel sélectionne l'action dont la valeur globale de *coping-frame* est la plus élevée, soit *Leave.*

8.3.5 Conclusion

Cet exemple montre l'intégration des arbres d'activité dans le processus de *coping* d'un replicant. L'arbre d'activité, défini de manière statique est interprété, adapté et complété de manière dynamique en fonction des préférences de l'humain virtuel, et les compromis sont réalisés en fonction de ses préférences. Parce qu'il accorde plus d'importance à la sécurité qu'à la rapidité d'exécution de son activité, cet humain virtuel a adapté son plan global en y ajoutant une étape afin de pouvoir verrouiller sa porte. Un agent dont la priorité serait la sécurité aurait simplement fermé la porte.

Bien que simple, cet exemple montre la puissance du couplage entre les règles Soar génériques utilisées pour le *coping* et les arbres d'activité et les règles d'action générés par HUMANS-GEN.

Cet exemple ne prend pas en compte l'évaluation émotionnelle, cependant on peut facilement imaginer le résultat de ce couplage. Par exemple, si un feu survient dans la maison pendant que l'humain virtuel prépare son départ, les actions qui se déroulent dans la maison seront jugées indésirables, tandis que celle qui lui permet de s'éloigner (*Leave*) sera considérée comme très désirable. Il est alors probable que l'humain virtuel interrompe son activité pour prendre la fuite.

Conclusion et perspectives

Retour sur les objectifs

Cette thèse s'insère dans un large contexte de recherche sur les environnements informatiques pour l'apprentissage humain (EIAH). L'objectif de cette thèse est de permettre de modéliser des humains virtuels émotionnel et sociaux dont le comportement est cohérent, adaptable à tout domaine d'application et explicable.

La cohérence, l'adaptabilité et l'explicabilité du comportement de l'humain virtuel nécessitent de définir un ensemble de processus de haut niveau indépendants du domaine d'application ainsi qu'un formalisme de description du domaine compréhensible par des non-informaticiens qui soit directement interprétable par les humains virtuels.

Le couplage de modèles de l'humain issus de la psychologie et de la sociologie à des architectures informatiques permettent de rendre compte de processus décisionnels et de générer des comportements cohérents et explicables. En particulier, la théorie de l'évaluation cognitive rend explicite les émotions et la dynamique liée. Nous nous sommes particulièrement attardés sur EMA [Gratch 04], une architecture qui vise à implémenter les processus cognitifs de l'évaluation cognitive au sein d'un humain virtuel. Le comportement de ce dernier est cohérent et adaptable car les processus cognitifs sont indépendants du domaine. Cependant, le comportement est non explicable car des valeurs représentant les dimensions humaines (buts et personnalité) sont spécifiées à la main de manière *ad hoc* et agrégée dans les connaissances du domaine.

Nous nous sommes également intéressés aux interactions émotionnelles et avons illustré, à travers la présentation de Thespian [Si 05], que les processus cognitifs mis en jeu pour permettre à des humains émotionnels d'interagir nécessitent beaucoup de ressources. Les modèles d'empathie cognitive existants requièrent d'envisager la situation depuis le point de vue de chaque humain, ce qui génère un problème de passage à l'échelle. La modélisation de l'empathie affective est généralement peu cohérente, car les modèles de contagion issus de l'épidémiologie sont catégoriels. Le modèle ASCRIBE fait exception en déterminant pour chaque individu deux valeurs, sa susceptibilité à être impacté par les émotions des autres et son pouvoir

de contagion. L'attribution de ces deux valeurs est cependant non explicable car déterminée de manière *ad hoc.*

La difficulté réside dans l'intégration de dimensions humaines au sein de processus indépendants du domaine.

Bilan des travaux effectués

Dans cette thèse, nous avons présenté un modèle d'humain virtuel cognitif, émotionnel et social dont les processus cognitifs sont assurés par une architecture cognitive. Les modèles que nous avons utilisés sont issus de travaux en psychologie et en sociologie et permettent d'assurer d'une part, la cohérence entre les caractéristiques individuelles de chaque humain virtuel et le comportement résultant de ses processus décisionnels (cohérence) et d'autre part, la compréhension du comportement par l'apprenant et le formateur (explicabilité).

Nous avons proposé un modèle de contagion émotionnelle qui permet de résoudre la problématique de passage à l'échelle induite par les relations émotionnelles entre les humains virtuels. Ce modèle combine les composantes individuelles qui régissent les phénomènes d'empathie affective, en prenant en compte la personnalité et les relations sociales, et les transfère dans un modèle d'empathie affective afin de simuler l'impact de l'état émotionnel des humains virtuels les uns sur les autres de la même manière que s'ils disposaient d'une théorie de l'esprit (impact sur l'état émotionnel, impact sur la relation sociale).

Le modèle d'humain virtuel que nous proposons présente plusieurs niveaux d'abstraction des connaissances. Il dispose ainsi d'un ensemble de règles génériques qui lui permettent de gérer les processus cognitifs d'évaluation cognitive (*appraisal*) et de stratégies adaptatives (*coping*). En fonction des scénarios, les humains virtuels sont dotés de connaissances spécifiques qui leur permettent de s'adapter à leur environnement. Ces connaissances définissent les entités du monde et les actions qui s'y déroulent, ainsi que des arbres d'activités qui représentent l'activité d'humains observée *in vivo.* L'influence de la personnalité est modélisée par l'intermédiaire de marqueurs. L'humain virtuel combine les règles génériques et les connaissances spécifiques afin de se comporter de manière cohérente, en inscrivant ses actions dans un cadre rationnel dirigé par des buts, les émotions et la personnalité. Nous avons en particulier enrichi le mécanisme d'*appraisal* et de *coping* de EMA en ajoutant la prise en compte explicite des buts et de la personnalité.

Ces connaissances sont modélisées dans des langages de description et écrites via un outil auteur, EDITOR. DOMAIN-DL permet de spécifier les entités et les actions, tandis qu'ACTIVITY-DL s'intéresse à l'activité humaine. Nous proposons un outil, HUMANS-GEN, qui permet aux humains virtuels d'interpréter directement ces connaissances.

Notre modèle d'humain virtuel est implémenté au sein d'un module de gestion de

personnages virtuels autonomes, REPLICANTS. Les processus cognitifs de chaque humain virtuel sont gérés par l'architecture cognitive Soar. REPLICANTS est intégré sur la plateforme HUMANS.

Limites, propositions et discussions

De manière générale, il serait intéressant d'appliquer ces travaux à un scénario de grande envergure, avec un grand nombre d'humains virtuels disposant de plusieurs arbres d'activité complexes.

L'implémentation des processus d'évaluation cognitive est actuellement rudimentaire. En effet, notre contribution visant à intégrer la prise en compte des arbres d'activité et de la personnalité au sein de l'*appraisal* et du *coping*, les autres dimensions des *appraisal-frames ne* sont pas implémentées. Si le squelette est présent, il est nécessaire de fournir les fonctions qui permettent d'évaluer les dimensions de chaque *appraisal-frame.* L'intégration de l'architecture EMA permettrait de combler ce manque.

Le système de marqueurs que nous avons présenté n'est actuellement pas relié à la personnalité OCEAN. Il est cependant nécessaire, pour assurer la cohérence, que les traits de personnalité associés à un humain virtuel déterminent les marqueurs qui l'intéressent. Il serait possible de proposer à l'ergonome de travailler directement avec les vecteurs de personnalité OCEAN. Une autre des solutions possibles serait d'utiliser les travaux de Bouchet et Sansonnet [Bouchet 11] qui définissent des tendances comportementales (opérateurs d'influence) à partir des facettes OCEAN. Ces opérateurs d'influence sont des adjectifs issus de WordNet qui pourraient être utilisés comme marqueurs.

Dans ces travaux, et à l'image d'autres [Bouchet 11, Marsella 10], nous définissons la cohérence d'un humain virtuel comme l'adéquation entre sa rationalité, son état mental (buts, émotions, personnalité, situation, relations sociales) et son comportement. Les débats sur la rationalité et l'action occupent une place centrale en sociologie, dans laquelle « l'émotionnel s'oppose à l'une ou l'autre des caractéristiques générales couplage du rationnel, c'est-à-dire ce qui est réflexif, calculé, contrôlé, règlementé, prévisible »[Cuin 01]. Il serait donc intéressant de discuter de la compatibilité entre la rationalité et les dimensions humaines, en particulier les émotions, et de l'impact de ce couplage.

Les théories de l'évaluation cognitive définissent deux étapes au cycle de l'humain virtuel : l'*appraisal* qui consiste à confronter les buts et la personnalité de l'humain virtuel à la situation et le *coping* qui permet de proposer des réponses adaptatives pour améliorer ou maintenir l'état émotionnel. Cette vision n'est cependant pas compatible avec toutes les théories de l'émotion, en particulier avec la théorie de la dissonance cognitive [Festinger 57]. Cette théorie stipule qu'en présence de connaissances incompatibles, il est nécessaire d'abandonner l'une ou

l'autre afin de rétablir une certaine logique mentale. Cet abandon génère également des émotions [José F Fontanari 12]. Or, les stratégies de *coping centré émotion* comme la pensée positive (*Wishful Thinking*), la résignation et la prise de distance visent à abandonner des buts ou en modifier l'importance. D'après la théorie de la dissonance cognitive, des émotions devraient donc être générées lors du *coping,* ce qui risque de générer une oscillation entre *appraisal* et *coping* et remet en cause le cycle en deux temps de la théorie de l'évaluation cognitive.

Perspectives

Dans cette thèse, nous avons focalisé notre étude sur les relations émotionnelles interindividuelles et avons proposé une opérationnalisation de modèles issus de la psychologie permettant de répondre à des problématiques informatiques. Il existe cependant beaucoup d'autres types de relations interindividuelles qui influencent le comportement émotionnel des humains, comme par exemple le **lien de l'individu au groupe** et **l'importance des normes socioculturelles.**

> « Intergroup emotions theory was developed in an attempt to better understand the nature of emotions that arise from group identification and membership, with a particular focus on the emotional antecedents of prejudice, discrimination, and other intergroup behaviors. IET borrows broadly from appraisal theories of emotion, as well as from social identity theory and self-categorization theory, to extend theorizing about emotion from the individual to the group level. The basic premise of IET is that when an individual identifies with a group, that ingroup becomes part of the self, thus acquiring social and emotional significance. When an ingroup acquires such significance, events or objects that impinge on the ingroup are appraised for their emotional relevance, just like events that occur in an individual's personal life. » [Smith 07]

Imaginons un instant le scénario suivant : une manifestation dégénère. Des individus se retrouvent à proximité d'un magasin dont la vitrine, brisée, regorge de gadgets électroniques. S'ils étaient seuls, aucun d'entre eux n'oserait voler. Mais si l'un d'entre eux montre l'exemple, la plupart suivront.

Notre modèle ne permet pas pour l'instant de modéliser cette influence du « groupe » sur l'individu.

Notre approche considère que les humains sont autonomes et une modélisation macroscopique irait à l'encontre de nos principes. Comment prendre alors en compte cet impact du groupe sur la prise de décision ?

Nous pensons que le fait d'appartenir à un groupe apporte un biais dans l'évaluation des évènements, et que ce biais est du au fait que l'individu adapte ses

propres critères d'acceptabilité morale en fonction des normes socioculturelles du groupe. De la même manière que la théorie de l'esprit offre un cadre cognitif pour évaluer l'état émotionnel d'autrui, il serait possible d'évaluer l'acceptabilité morale d'une action en se projetant à la place de chaque individu qui nous entoure. Et comme pour l'empathie affective, cela poserait un problème de passage à l'échelle. L'approche que nous avons utilisée pour construire notre modèle de contagion émotionnelle permettrait de résoudre ce problème.

Un modèle de contagion des normes socioculturelles permettrait à chaque individu d'être impacté et d'impacter temporairement les valeurs de ceux qui l'entourent. Après avoir évalué son état émotionnel et mis à jour ses relations sociales, l'humain virtuel serait contaminé émotionnellement et moralement. Une dimension de la *coping-frame* évaluerait l'acceptabilité morale de chaque action par rapport à ces normes. Tout comme pour la contagion émotionnelle, il est nécessaire que les caractéristiques individuelles de l'humain virtuel entrent en compte dans le calcul de sa susceptibilité et pouvoir de contagion morale.

Un avantage supplémentaire de cette approche est qu'elle propose un moyen de modéliser l'impact du groupe sur l'individu sans niveau intermédiaire. Ici, les humains virtuels restent autonomes et indépendants les uns des autres, sans appartenance fixe. Le groupe n'est pas vu comme une entité distincte mais comme une vague fluctuante d'émotions et de normes socioculturelles.

À plus grande échelle, la réflexion menée pendant ces travaux de thèse pose les fondements d'une recherche qui vise à opérationnaliser les modèles issus de la psychologie et de la sociologie afin de modéliser des humains virtuels uniques. Si les processus réactifs sont trop simplistes pour pouvoir représenter les différences individuelles, les processus cognitifs sont trop complexes, et ce, aussi bien pour une machine que pour un humain. À nous de prouver qu'un « entre-deux » est adapté.

Annexes

Annexe A

Règles Soar génériques

A.1 Évaluation Cognitive

A.1.1 Création des *appraisal frames*

```
#Création de la structure d'appraisal-frame
sp {elaborations*elaborate*appraisal-frame*create
   (state <s> ^perceptions <p>
              ^appraisal <app>)
   (<p> ^event <e>)
-->
   (<app> ^frame <f>)
   (<f> ^event <e>)
}
```

A.1.2 Élaboration des dimensions d'une *appraisal frame*

A.1.2.1 Calcul de la désirabilité d'un évènement liée à un arbre d'activité

```
#Ajout de la désirabilité-activity associée à l'event
sp {elaborations*elaborate*appraisal-frame*dimension*desirability-activity
   (state <s> ^appraisal.frame <f>)
   (<f> ^event <e>)
   (<e> ^desirability-activity <d>)
-->
   (<f> ^desirability-activity<d>)
}
```

```
#Évaluation de la désirabilité de l'évèment si il a permis
#de terminer une tâche
sp {elaborate-state*elaborate*appraisal*des-activity*task-done
   (state <s> ^appraisal.frame <f>)
   (<f> ^event <e>)
   (<e> ^name <name>)
   (<act> ^search <task>)
   (<task> ^pool terminated
           ^utility <util>
           ^action.name <name>
   )
   (<f> ^desirability-activity <val>)
-->
   (<f> ^desirability-activity (+ <util> <val>)
                    <val> -) }


#Évaluation de désirabilité de l'évènement si il a permis
#de lever une condition sur la tâche mère
sp {elaborate-state*elaborate*appraisal*des-activity*condition-met
   (state <s> ^appraisal.frame <f>)
   (<act> ^search <task>)
   (<task> ^condition.<< regulatory contextual
                favorable nomologique >> <cond>)
   (<cond> ^met *yes*
           ^utility <util>)
   (<f> ^desirability-activity <val>)
-->
   (<f> ^desirability-activity (+ <util> <val>)
                    <val> -) }

#Évaluation de désirabilité d'un évènement s'il rend
#impossible l'exécution future d'une tâche
sp {elaborate-state*elaborate*appraisal*des-activity*task-failure
  (state <s> ^appraisal.frame <f>)
   (<f> ^event <e>)
   (<act> ^search <task>)
   (<task> ^pool failure>
           ^utility <util>)
```

```
(<f> ^desirability-activity <val>)
-->
   (<f> ^desirability-activity (- <util> <val>)
                     <val> -) }

#Évaluation de la désirabilité d'un évènement s'il rend
#invalide une condition sur une tâche
sp {elaborate-state*elaborate*appraisal*des-activity*condition-unmet
   (state <s> ^appraisal.frame <f>)
   (<act> ^search <task>)
   (<task> ^condition.<< regulatory contextual
               favorable nomologique >> <cond>)
   (<cond> ^met *no*
           ^utility <util>)
   (<f> ^desirability-activity <val>)
-->
   (<f> ^desirability-activity (- <util> <val>)
                     <val> -) }
```

A.1.2.2 Transfert des dimensions d'appraisal

```
#Ajout de la désirabilité-personality associée à l'event
sp {elaborations*elaborate*appraisal-frame*dimension*desirability-personality
   (state <s> ^appraisal.frame <f>)
   (<f> ^event <e>)
   (<e> ^desirability-personality <d>)
-->
   (<f> ^desirability-personality <d>)
}

#Ajout de la relevance associée à l'event
sp {elaborations*elaborate*appraisal-frame*dimension*relevance
   (state <s> ^appraisal.frame <f>)
   (<f> ^event <e>)
   (<e> ^relevance <r>)
-->
   (<f> ^relevance <r>)
}
```

```
#Ajout de l'urgency associée à l'event
sp {elaborations*elaborate*appraisal-frame*dimension*urgency
   (state <s> ^appraisal.frame <f>)
   (<f> ^event <e>)
   (<e> ^urgency <a>)
-->
   (<f> ^urgency <a>)
}


#Attribuer le responsable de l'action à la dimension agency de la frame
sp {elaborations*elaborate*appraisal-frame*dimension*agency-other
   (state <s> ^appraisal.frame <f>)
   (<f> ^event <e>)
   (<e> ^agent <a>)
-->
   (<f> ^agency <a>)
}


#Attribuer le responsable de l'action à la dimension agency de la frame
sp {elaborations*elaborate*appraisal-frame*dimension*agency-self
   (state <s> ^appraisal.frame <f>
              ^self <self>)
   (<f> ^event <e>)
   (<e> ^agent <self>)
-->
   (<f> ^agency self)
}

#Attribuer du blâme au responsable de l'action
sp {elaborations*elaborate*appraisal-frame*dimension*blame
   (state <s> ^appraisal.frame <f>)
   (<f> ^desirability < 0)
   (<f> ^agency <a>)
-->
   (<f> ^credit-blame blame)
}
```

```
#Attribuer du crédit au responsable de l'action
sp {elaborations*elaborate*appraisal-frame*dimension*credit
   (state <s> ^appraisal.frame <f>)
   (<f> ^desirability > 0)
   (<f> ^agency <a>)
-->
   (<f> ^credit-blame credit)
}
```

A.1.3 Attribution des catégories émotionnelles OCC

```
#Calcul de la moyenne des deux désirabilités
sp {elaborations*elaborate*appraisal-frame*dimension*desirability-activity
   (state <s> ^appraisal.frame <f>)
   (<f> ^event <e>)
   (<e> ^desirability-activity <d>)
-->
   (<f> ^desirability-activity <d>)
}

#Ajout de la catégorie OCC joy
sp {elaborations*elaborate*appraisal-frame*tag*joy
   (state <s> ^appraisal <app>)
   (<app> ^frame <f>)
   (<f> ^desirability > 0
        ^likelihood
       -^agency )
-->
   (<f> ^tag joy)
}

#Ajout de la catégorie OCC distress
sp {elaborations*elaborate*appraisal-frame*tag*distress
   (state <s> ^appraisal <app>)
   (<app> ^frame <f>)
   (<f> ^desirability < 0
        ^likelihood 1
       -^agency )
-->
   (<f> ^tag distress)
}
```

```
#Ajout de la catégorie OCC anger
sp {elaborations*elaborate*appraisal-frame*tag*anger
   (state <s> ^appraisal <app>)
   (<app> ^frame <f>)
   (<f> ^credit-blame blame
        ^likelihood 1
        ^agency <a>  <> self)
-->
   (<f> ^tag anger)
}

#Ajout de la catégorie OCC admiration
sp {elaborations*elaborate*appraisal-frame*tag*admiration
   (state <s> ^appraisal <app>)
   (<app> ^frame <f>)
   (<f> ^credit-blame credit
        ^likelihood 1
        ^agency <a> <> self)
-->
   (<f> ^tag admiration)
}

#Ajout de la catégorie OCC pride
sp {elaborations*elaborate*appraisal-frame*tag*pride
   (state <s> ^appraisal <app>)
   (<f> ^credit-blame credit
        ^agency self)
-->
   (<f> ^tag pride)
}

#Ajout de la catégorie OCC shame
sp {elaborations*elaborate*appraisal-frame*tag*shame
   (state <s> ^appraisal <app>)
   (<f> ^credit-blame blame
        ^likelihood 1
        ^agency self)
-->
   (<f> ^tag shame)
}
```

```
#Ajout de la catégorie OCC hope
sp {elaborations*elaborate*appraisal-frame*tag*hope
   (state <s> ^appraisal <app>)
   (<app> ^frame <f>)
   (<f> ^desirability > 0
        ^likelihood < 1
       -^agency)
-->
   (<f> ^tag hope)
}

#Ajout de la catégorie OCC fear
sp {elaborations*elaborate*appraisal-frame*tag*fear
   (state <s> ^appraisal <app>)
   (<app> ^frame <f>)
   (<f> ^desirability < 0
        ^likelihood < 1
       -^agency )
-->
   (<f> ^tag fear)
}
```

A.2 Mise à jour des relations sociales

A.2.1 Modifications des relations sociales dues à une émotion causée par autrui

```
#Augmentation de l'appreciation
sp {relations*propose*change-relationship*appreciation*positive
  (state <s> ^appraisal.frame <f>)
  (<f> ^tag admiration
       ^agency <a>
       ^desirability <d>)
-->
  (<s> ^operator <op> + =)
  (<op> ^name change-relationship
        ^dimension appreciation
        ^agent <a>
        ^value <d>
        ^sign |+|)
```

```
}

#Diminution de l'appreciation
sp {relations*propose*change-relationship*appreciation*negative
  (state <s> ^appraisal.frame <f>)
  (<f> ^tag anger
       ^agency <a>
       ^desirability <d>)
-->
  (<s> ^operator <op> + =)
  (<op> ^name change-relationship
        ^dimension appreciation
        ^agent <a>
        ^value <d>
        ^sign |-|)
}

#Augmentation de la dominance
sp {relations*propose*change-relationship*dominance*positive
  (state <s> ^appraisal.frame <f>)
  (<f> ^tag anger
       ^agency <a>
       ^desirability <d>)
-->
  (<s> ^operator <op> + =)
  (<op> ^name change-relationship
        ^dimension dominance
        ^agent <a>
        ^value <d>
        ^sign |+|)
}

#Diminution de la dominance
sp {relations*propose*change-relationship*dominance*negative
  (state <s> ^appraisal.frame <f>)
  (<f> ^tag << distress admiration >>
       ^agency <a>
       ^desirability <d>)
-->
  (<s> ^operator <op> + =)
```

```
  (<op> ^name change-relationship
        ^dimension dominance
        ^agent <a>
        ^value <d>
        ^sign |-|)
}

#Diminution de la solidarité
sp {relations*propose*change-relationship*solidarity*negative
   (state <s> ^appraisal.frame <frame>)
   (<frame> ^tag anger
            ^agency <a>
            ^desirability <d>)
-->
   (<s> ^operator <op> + =)
   (<op> ^name change-relationship
         ^dimension solidarity
         ^agent <a>
         ^value <d>
         ^sign |-|) }
```

A.2.2 Modifications des relations sociales dues à la perception des émotions d'autrui

```
#Augmentation de la solidarité
sp {relations*propose*change-relationship*solidarity*positive
  (state <s> ^appraisal.frame <f>
             ^perceptions.agent <a>)
  (<f> ^tag <tag>
       ^desirability <d>))
  (<a> ^mood.tag <t>)
  (<t> ^value <v>)
-->
  (<s> ^operator <op> + =)
  (<op> ^name change-relationship
        ^dimension solidarity
        ^agent <a>
        ^value (/ (+ <d> <v>) 2)
        ^sign |+|)
}
```

```
#Diminution de la solidarité
sp {relations*propose*change-relationship*solidarity*negative
  (state <s> ^appraisal.frame <f>
             ^perceptions.agent <a>)
  (<f> ^tag <tag>
       ^desirability <d>)
  (<a> ^mood <m>)
  (<mood> ^tag <> <tag>
          ^value <v>)
-->
  (<s> ^operator <op> + =)
  (<op> ^name change-relationship
        ^dimension solidarity
        ^agent <a>
        ^value (/ (+ <d> <v>) 2) ^sign |-|)
}

#Augmentation de la dominance
sp {relations*propose*change-relationship*perception*dominance*positive
   (state <s> ^perceptions.agent <a>)
   (<a> ^mood <m>)
   (<mood> ^tag << distress fear >>
           ^value <v>)
-->
   (<s> ^operator <op> + =)
   (<op> ^name change-relationship
         ^dimension solidarity
         ^agent <a>
         ^value <v>
         ^sign |+|) }
```

A.3 Coping

A.3.1 Création d'une *coping-frame*

```
#Création de la structure de coping-frame
sp {evaluate-state*elaborate*coping-frame*create
   (state <s> ^tried-tied_operator <a>
              ^appraisal <app>)
-->
```

```
   (<app> ^frame <f>)
   (<f> ^event <a>
        ^tags 0
        ^activity 0)}
```

A.3.2 Évaluation des dimensions d'une *coping-frame*

A.3.2.1 Évaluation des marqueurs personnalisés (impact de la personnalité)

```
#Incrémentation des marqueurs personnalisés associés à la tâche
sp {evaluate-state*elaborate*coping-frame*tags*task
   (state <s> ^tried-tied-operator <action>
              ^appraisal.tags <app-tags>
              ^activity <activity>
              ^self.preferences.tags.<tag> <pref>)
   (<activity> ^search <action>)
   (<action> ^tag <tag>)
   (<app-tags> ^value <val>)
-->
   (<app-tags> ^<tag> (+ <pref> <val>)
                      <val> -)}


#Incrémentation des marqueurs personnalisés associés à une condition
sp {evaluate-state*elaborate*coping-frame*tags*condition
   (state <s> ^tried-tied-operator <action>
              ^appraisal.tags <app-tags>
              ^activity <act
              ^self.preferences.tags.<tag> <pref>)
   (<act> ^search <action>)
   (<action> ^condition.<< regulatory contextual
                           favorable nomologique >>.tag <tag>)
   (<app-tags> ^<tag> <val>)
-->
   (<app-tags> ^<tag> (+ <pref> <val>)
                      <val> -)}
```

A.3.2.2 Évaluation de l'avancement dans l'arbre d'activité

```
#Évaluation de l'utilité intrinsèque de la tâche
```

```
sp {elaborate-state*elaborate*activity*self-task-utility-add
   (state <s> ^tried-tied-operator <action>
              ^appraisal.activity <app-act>
              ^activity <act>)
   (<act> ^search <task>)
   (<task> ^action <action>
           ^utility <util>)
   (<app-act> ^value <val>)
-->
   (<app-act> ^value (+ <util> <val>)
                     <val> -) }

#Évaluation de l'utilité de l'action si elle permet
#de terminer une tâche
sp {elaborate-state*elaborate*activity*task-utility-add
   (state <s> ^tried-tied-operator <action>
              ^appraisal.activity <app-act>
              ^activity <act>)
   (<act> ^search <task>)
   (<task> ^pool terminated
           ^utility <util>)
   (<app-act> ^value <val>)
-->
   (<app-act> ^value (+ <util> <val>)
                     <val> -) }

#Évaluation de l'utilité de l'action si elle permet
#de lever une condition sur la tâche mère
sp {elaborate-state*elaborate*activity*condition-utility-add
   (state <s> ^tried-tied-operator <action>
              ^appraisal.activity <app-act>
              ^activity <act>)
   (<act> ^search <task>)
   (<task> ^condition.<< regulatory contextual
                favorable nomologique >> <cond>)
   (<cond> ^met *yes*
           ^utility <util>)
   (<app-act> ^value <val>)
-->
   (<app-act> ^value (+  <util> <val>)
                     <val> -) }
```

```
#Évaluation de l'utilité d'une action si elle rend
#impossible l'exécution future d'une tâche
sp {elaborate-state*elaborate*activity*task-failure-utility-remove
   (state <s> ^tried-tied-operator <action>
              ^appraisal.activity <app-act>
              ^activity <act>)
   (<act> ^search <task>)
   (<task> ^pool failure>
           ^utility <util>)
   (<app-act> ^value <val>)
-->
   (<app-act> ^value (-  <util> <val>)
                     <val> -) }

#Évaluation de l'utilité de l'action si elle rend
#invalide une condition sur une tâche
sp {elaborate-state*elaborate*activity*condition-failure-utility-remove
   (state <s> ^tried-tied-operator <action>
              ^appraisal.activity <app-act>
              ^activity <act>)
   (<act> ^search <task>)
   (<task> ^condition.<< regulatory contextual
                favorable nomologique >> <cond>)
   (<cond> ^met *no*
           ^utility <util>)
   (<app-act> ^value <val>)
-->
   (<app-act> ^value (-  <util> <val>)
                     <val> -) }
```

A.3.3 Aggrégation des dimensions d'appraisal

```
sp {evaluate-operator*elaborate*sum*appraisals
   (state <s> ^appraisal <app>
              ^tried-tied-operator
              ^self.preferences.coping-preferences <prefs>)
   (<app> ^all-appraised true
          ^activity <act>
          ^tags <tags>)
   (<prefs> ^activity <p-act>
            ^tags <p-tags>)
```

```
-->
   (<app> ^value (+ (* <act> <p-act>)
                    (* <tags> <p-tags>) )) }
```

Annexe B

Macros Humans-Gen

B.1 Création des tâches mères (*Task*)

NGS_ACT_goal_create-goal-IND créer une tâche mère avec un constructeur IND

NGS_ACT_goal_create-goal-SEQ créer une tâche mère avec un constructeur SEQ

NGS_ACT_goal_create-goal-SEQ-ORD créer une tâche mère avec un constructeur SEQ-ORD

NGS_ACT_goal_create-goal-PAR créer une tâche mère avec un constructeur PAR

NGS_ACT_goal_create-goal-PAR-SIM créer une tâche mère avec un constructeur PAR-SIM

NGS_ACT_goal_create-goal-PAR-START créer une tâche mère avec un constructeur PAR-START

NGS_ACT_goal_create-goal-PAR-END créer une tâche mère avec un constructeur PAR-END

Si aucun constructeur n'est spécifié dans le modèle d'activité, la macro *NGS_ACT_goal_create-goal-IND* est utilisée par défaut.

B.2 Création des tâches feuilles (*Task_Leaf*)

NGS_ACT_goal_create-goal-perform créer un but dont l'objectif est d'effectuer une action. Une fois activé, ce but créé un *Transform* qui lui même génère un *operateur soar.*

B.3 Création des conditions

Ces macros utilisent les types définis par NGS et détaillés dans le tableau7.1.

NGS_ACT_goal_create-condition-nomological créer une condition nomologique

NGS_ACT_goal_create-condition-contextual créer une condition contextuelle

NGS_ACT_goal_create-condition-favorable créer une condition favorable

NGS_ACT_goal_create-condition-regulatory créer une condition regulatoire

NGS_ACT_goal_create-condition-stop créer une condition d'arrêt

Annexe C

Implémentation du 3ème cas d'usage

C.1 Arbre d'activité en ACTIVITY-DL

```
<?xml version="1.0" encoding="UTF-8"?>
<model>
  <tasks>

    <!-- Tâche principale Go To Holidays --!>
    <task_m>
      <task id="t_go_to_holidays" name="Go To Holidays">
        <!-- Tâches filles --!>
        <subtasks>
          <subtask id="t_get_ready"/>
          <subtask id="t_leave"/>
        </subtasks>
        <!-- Operateur d'allen --!>
        <constructor>
          <relation>
            <task id="t_get_ready"/>
            <operator id="<"/>
            <task id="t_leave"/>
          </relation>
        </constructor>
        <conditions>
          <stop-conditions>
            <stop-condition-task>
```

```
            <!-- Condition de réussite: les deux tâches
              filles sont satisfaites --!>
            <stop-condition consequence="satisfaction">
              <AND>
                <statement task="t_get_ready" value="*yes*"/>
                <statement task="t_leave" value="*yes*"/>
              </AND>
            </stop-condition>
          </stop-condition-task>
        </stop-conditions>
      </conditions>
    </task>
  </task_m>

  <!-- Tâche fille Get Ready --!>
  <task_m>
    <task id="t_get_ready" name="Get Ready">
      <!-- Tâches filles --!>
      <subtasks>
        <subtask id="t_load_car"/>
        <subtask id="t_close_house"/>
      </subtasks>
      <conditions>
        <stop-conditions>
          <stop-condition-task>
            <!-- Condition de réussite: les deux tâches
              filles sont satisfaites --!>
            <stop-condition consequence="satisfaction">
              <AND>
                <statement-task task="t_load_car" value="*yes*"/>
                <statement-task task="t_close_house" value="*yes*"/>
              </AND>
            </stop-condition>
          </stop-condition-task>
        </stop-conditions>
      </conditions>
    </task>
  </task_m>

  <!-- Tâche feuille Load Car --!>
```

```
<task_f>
  <task id="t_load_car" name="Load Car">
    <!-- Action associée --!>
    <operation>
      <action id="act_load_car"/>
      <!-- Ressources utilisées --!>
      <resources>
        <concept>
          <!-- Création/Utilisation d'une référence
            <one_car> sur un objet de type car --!>
          <ref-concept id="obj_car" ref="one_car"/>
        </concept>
      </resources>
    </operation>
  </task>
</task_f>

<!-- Tâche fille Close House --!>
<task_m>
  <task id="t_close_house" name="Close House"/>
    <!-- Tâches filles --!>
    <subtasks>
      <subtask id="t_close_door"/>
      <subtask id="t_lock_door"/>
    </subtasks>
    <conditions>
      <stop-conditions>
        <stop-condition-task>
          <!-- Condition de réussite: une seule des tâches
            filles est satisfaite --!>
          <stop-condition consequence="satisfaction">
            <XOR>
              <statement-task task="t_close_door" value="*yes*"/>
              <statement-task task="t_lock_door" value="*yes*"/>
            </XOR>
          </stop-condition>
        </stop-condition-task>
      </stop-conditions>
    </conditions>
  </task>
</task_m>
```

```
<!-- Tâche feuille Close the door --!>
<task_f>
  <task id="t_close_door" name="Close the door"/>
    <!-- Tag productivité associé à la tâche --!>
    <tags>
      <tag id="tag_productivity"/>
    </tags>
    <!-- Action associée --!>
    <operation>
      <action id="act_close"/>
    </operation>
    <conditions>
      <pre-conditions>
        <!-- Condition nomologique héritée de
          l'action associée --!>
        <pre-condition type="nomological">
          <statement-property concept="obj_door"
            ref-concept="the_door" state="close"
            value="*true*"/>
        </pre-condition>
      </pre-conditions>
      <stop-condition>
        <!-- Condition de réussite héritée
          de l'action associée --!>
        <stop-condition consequence="satisfaction">
          <statement-property concept="obj_door"
            ref-concept="the_door" state="close"
            ref-value="true"/>
        </stop-condition>
      </stop-conditions>
    </conditions>
  </task>
</task_f>

<!-- Tâche feuille Lock the door --!>
<task_f>
  <task id="t_lock_door" name="Lock the door">
    <!-- Tag sécurité associé à la tâche --!>
    <tags>
      <tag id="tag_security"/>
    </tags>
```

```
      <!-- Action associée --!>
      <operation>
        <action id="act_lock"/>
      </operation>
      <conditions>
        <pre-conditions>
          <!-- Condition nomologique héritée de
            l'action associée --!>
          <pre-condition type="nomological">
            <statement-property concept="obj_door"
              ref-concept="the_door" state="close"
              value="*true*"/>
          </pre-condition>
          <pre-condition type="nomological">
            <statement-property concept="obj_key"
              ref-concept="the_key" state="position"
              value="*inventory*"/>
          </pre-condition>
        </pre-conditions>
        <stop-condition>
          <!-- Condition de réussite héritée de
             l'action associée --!>
          <stop-condition consequence="satisfaction">
            <AND>
               <statement-property concept="obj_door"
                 ref-concept="the_door" state="locked"
                 ref-value="true"/>
               <statement-property concept="obj_door"
                 ref-concept="the_door" state="close"
                 ref-value="true"/>
            </AND>
          </stop-condition>
        </stop-condition>
      </conditions>
    </task>
  </task_f>

  <!-- Tâche feuille Leave --!>
  <task_f>
    <task id="t_leave" name="Leave"/>
```

```
        <!-- Action associée --!>
        <operation>
          <action id="act_leave"/>
          <!-- Ressources utilisées --!>
          <resources>
            <concept>
              <!-- Création/Utilisation d'une référence
                <one_car> sur un objet de type car --!>
              <ref-concept id="obj_car" ref="one_car"/>
            </concept>
          </resources>
        </operation>
        <conditions>
          <pre-conditions>
            <!-- Condition nomologique héritée de
               l'action associée --!>
            <pre-condition type="nomological">
              <statement-property concept="obj_car"
                ref-concept="one_car" state="ready"
                ref-value="true"/>
            </pre-condition>
          </pre-conditions>
        </conditions>
      </task>
    </task_f>
  </tasks>
</model>
```

C.2 Règles d'action

```
#Règles de l'action Load Car
sp {propose*action*act_load_car
  (<s> ^agent-position *here*)
-->
  (<s> ^operator <op> +)
  (<op> ^name act_load_car)
}
sp {apply*action*act_load_car
  (<s> ^operator.name act_load_car)
-->
```

```
}

#Règles de l'action Close Door
sp {propose*action*act_close
  (<s> ^agent-position *here*
       ^door <d>)
  (<s> ^closed *false*)
-->
  (<s> ^operator <op> +)
  (<op> ^name act_close
        ^param <d>)
}
sp {apply*action*act_close
  (<s> ^operator <op>)
   (<op> ^name act_close
         ^param <d>)
-->
  (<d> ^closed *true*
               <false>-))
}

#Règles de l'action Lock Door
sp {propose*action*act_lock
  (<s> ^agent-position *here*
       ^door <d>
       ^key <k>)
  (<s> ^closed *false*)
  (<k> ^position *inventory*)
-->
  (<s> ^operator <op> +)
  (<op> ^name act_close
        ^param <d>
        ^resource <k>)
}
sp {apply*action*act_lock
  (<s> ^operator <op>)
    (<op> ^name act_lock
          ^param <d>
          ^resource <k>)
    (<d> ^closed <false>)
-->
  (<d> ^closed *true*
```

```
                <false>-)
}

#Règles de l'action Leave
sp {propose*action*act_leave
  (<s> ^agent-position *here*)
-->
  (<s> ^operator <op> +)
  (<op> ^name act_leave)
}
sp {apply*action*act_leave
  (<s> ^operator <op>
       ^agent-position <pos>)
    (<op> ^name act_leave)
-->
  (<s> ^agent-position *away*
                       <pos>-)
}
```

Glossaire

contagion émotionnelle représente le phénomène qui fait ressentir à l'observateur les émotions (inférées par l'empathie cognitive) de l'observé ; on parle aussi d'empathie affective

elaborations processus cognitif et affectif, conscient, qui permet de se projeter à la place d'autrui afin d'imaginer son état émotionnel et éventuellement de générer des émotions empathiques et des réactions associées.

empathie affective représente le phénomène qui fait ressentir à l'observateur les émotions (inférées par l'empathie cognitive) de l'observé ; on parle souvent de contagion émotionnelle

empathie cognitive représente la capacité à identifier l'état émotionnel d'une autre personne (équivalent à la théorie de l'esprit)

espace-problème représentation de la connaissance utilisée pour résoudre un type de problème. Il est composé d'un ensemble d'états, d'un ensemble d'opérateurs qui permettent de naviguer entre ces états, ainsi que de contraintes sur l'application de ces opérateurs (règles). Un espace-problème représente donc un algorithme de résolution de problème.

Humain virtuel entités informatiques dont le comportement, aux yeux d'un observateur externe, est similaire à celui d'un humain réel.

règles de préférences processus cognitif et affectif, conscient, qui permet de se projeter à la place d'autrui afin d'imaginer son état émotionnel et éventuellement de générer des émotions empathiques et des réactions associées.

système de production représente la partie dynamique d'une architecture cognitive. Il permet de choisir quel opérateur appliquer afin de guider le cheminement dans l'espace-problème, à travers deux types de mémoires : la mémoire à long-terme qui contient les règles de proposition et de choix des opérateurs (mémoire procédurale), et la mémoire à court-terme (mémoire de travail) qui contient les connaissances spécifiques sur lesquelles l'agent raisonne.

théorie de l'action située courant des sciences sociales qui définit l'activité de l'humain comme un ensemble d'interactions. Elle sous-tend que toute action dépend étroitement des circonstances matérielles et sociales dans lesquelles elle a lieu et s'oppose donc aux approches cognitives qui visent à construire un modèle décontextualisé, applicable en toutes circonstances, du comportement de l'humain.

théorie de l'esprit désigne les processus cognitifs permettant à un individu d'expliquer et de prédire ses propres actions, celles des individus qui l'entourent (grâce à l'attribution d'états mentaux).

trait de personnalité Dimension de personnalité utilisée pour catégoriser une personne en fonction du degré de manifestation de ce trait.

état mental représente les croyances, perceptions, émotions et intentions des individus, dans le paradigme de la théorie de l'esprit

Bibliographie

[Amokrane 10] K. Amokrane. *Suivi de l'apprenant en environnement virtuel pour la formation à la prévention des risques sur des sites SEVESO.* PhD thesis, Université de Technologie de Compiègne, Compiègne, 2010.

[Anderson 76] J. R Anderson. Language, memory, and thought. Lawrence Erlbaum, 1976.

[Anderson 93] J. R Anderson. Rules of the mind. Lawrence Erlbaum, 1993.

[Baron-Cohen 09] S. Baron-Cohen. *Autism : The Empathizing–Systemizing (E-S) Theory.* Annals of the New York Academy of Sciences, vol. 1156, no. 1, page 68–80, 2009.

[Barsade 02] S. G Barsade. *The Ripple Effect : Emotional Contagion and Its Influence on Group Behavior.* Administrative Science Quarterly, vol. 47, no. 4, page 644–677, 2002.

[Bates 94] J. Bates*et al. The role of emotion in believable agents.* Communications of the ACM, vol. 37, no. 7, page 122–125, 1994.

[Bosse 09] T. Bosse, R. Duell, Z.A. Memon, J. Treur & C.N. Van Der Wal. *A Multi-agent Model for Mutual Absorption of Emotions.* In Proceedings of the 23rd European Conference on Modelling and Simulation, ECMS, page 212–218, 2009.

[Bosse 11] Tibor Bosse, Mark Hoogendoorn, Michel C. A. Klein, Jan Treur & C. Natalie Wal. *Agent-Based Analysis of Patterns in Crowd Behaviour Involving Contagion of Mental States.* In Kishan G. Mehrotra, Chilukuri K. Mohan, Jae C. Oh, Pramod K. Varshney & Moonis Ali, editeurs, Modern Approaches in Applied Intelligence, volume 6704, pages 566–577. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[Bouchet 11] Francois Bouchet & Jean-Paul Sansonnet. *Influence of Personality Traits on the Rational Process of Cognitive Agents.* pages 81–88. IEEE, August 2011.

[Bratman 87] M. Bratman. Intention, plans, and practical reason, volume 250. Harvard University Press Cambridge, MA, 1987.

[Cattell 46] R. B Cattell. *Description and measurement of personality.* 1946.

[Costa 92] Paul T. Costa & Robert R. McCrae. Revised NEO personality inventory (NEO PI-R) and neo five-factor inventory (NEO-FFI). Psychological Assessment Resources Odessa, FL, 1992.

[Costa 95] Paul T. Costa & Robert R. McCrae. *Domains and facets : Hierarchical personality assessment using the Revised NEO Personality Inventory.* Journal of personality assessment, vol. 64, no. 1, page 21–50, 1995.

[Craik 67] K. Craik. The nature of explanation. 1943. Cambridge University Press, Cambridge, 1967.

[Cuin 01] Charles-Henry Cuin. *Émotions et rationalité dans la sociologie classique : les cas de Weber et Durkheim.* Revue européenne des sciences sociales, vol. XXXIX-120, pages 77–100, 2001.

[Damasio 06] A. R Damasio & M. Blanc. L'erreur de descartes : la raison des émotions. Odile Jacob, 2006.

[Davis 94] M. H. Davis. Empathy : A social psychological approach. Westview Press, 1994.

[De Tocqueville 50] A. De Tocqueville. De la démocratie en amérique, volume 1. Pagnerre, 1850.

[Dodds 05] P. S Dodds & D. J Watts. *A generalized model of social and biological contagion.* Journal of Theoretical Biology, vol. 232, no. 4, page 587–604, 2005.

[Durkheim 68] E. Durkheim. Les formes élémentaires de la vie religieuse. Presses Univ. de France, 1968.

[Durupinar 10] Funda Durupinar. *From Audiences to Mobs : Crowd Simulation with Psychological Factors.* PhD thesis, Bilkent University, July 2010.

[Edward 11] L. Edward. *Modélisation décisionnelle de personnages virtuels pour la scénarisation de situations industrielles dangereuses.* PhD thesis, Université de Technologie de Compiègne, Compiègne, 2011.

[Ekman 71] P. Ekman*et al.* Universals and cultural differences in facial expressions of emotion. University of Nebraska Press Lincoln, 1971.

[Eysenck 47] H.J. Eysenck. Dimensions of personality. Transaction Publishers, 1947.

[Eysenck 76] H.J. Eysenck & S.B.G. Eysenck. Psychoticism as a dimension of personality. Hodder and Stoughton, London, 1976.

[Fehr 84] B. Fehr & J.A. Russell. *Concept of emotion viewed from a prototype perspective.* Journal of experimental psychology : General, vol. 113, no. 3, page 464, 1984.

[Festinger 57] L. Festinger. A theory of cognitive dissonance, volume 2. Stanford University Press, 1957.

[Fikes 71] R. E Fikes & N. J Nilsson. *STRIPS : A new approach to the application of theorem proving to problem solving.* Artificial intelligence, vol. 2, no. 3-4, page 189–208, 1971.

[Fikes 72] R. E. Fikes & N. J. Nilsson. *STRIPS : A new approach to the application of theorem proving to problem solving.* Artificial intelligence, vol. 2, no. 3, page 189–208, 1972.

[Freud 21] Sigmund Freud. *Psychologie des foules et analyse du moi.* Essais de psychanalyse, 1921.

[Ghallab 98] M. Ghallab, C. Aeronautiques, C. K. Isi, D. Wilkins*et al.* *PDDL-the planning domain definition language.* 1998.

[Goel 92] V. Goel. *Comparison of well-structured & ill-structured task environments and problem spaces.* In Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society, page 844–849, 1992.

[Goldberg 92] L. R Goldberg. *The development of markers for the Big-Five factor structure.* Psychological assessment, vol. 4, no. 1, page 26, 1992.

[Granovetter 83] M. Granovetter & R. Soong. *Threshold models of diffusion and collective behavior.* Journal of Mathematical sociology, vol. 9, no. 3, page 165–179, 1983.

[Gratch 04] J. Gratch & S. Marsella. *A domain-independent framework for modeling emotion.* Cognitive Systems Research, vol. 5, no. 4, page 269–306, 2004.

[Hatfield 92] E. Hatfield, J. T Cacioppo & R. L Rapson. *Primitive emotional contagion.* Emotion and Social Behavior, vol. 14, pages 151–177, 1992.

[Hatfield 94] E. Hatfield, J. T Cacioppo & R. L Rapson. Emotional contagion. Cambridge Univ Pr, 1994.

[Hoffman 84] M. L. Hoffman. *Interaction of affect and cognition in empathy.* Emotions, cognition, and behavior, page 103–131, 1984.

[Hollnagel 99] Erik Hollnagel & David Woods. *Cognitive Systems Engineering : New wine in new bottles.* International Journal of Human-Computer Studies, vol. 51, no. 2, pages 339–356, 1999.

[James 84] W. James. *What is an Emotion ?* Mind, vol. 9, no. 34, page 188–205, 1884.

[Johnson 97] T. R. Johnson. *Control in ACT-R and Soar.* In Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society, page 343–348, 1997.

[José F Fontanari 12] Marie-Claude Bonniot-Cabanac José F Fontanari. *A structural model of emotions of cognitive dissonances.* Neural networks : the official journal of the International Neural Network Society, vol. 32, pages 57–64, 2012.

[Kermack 27] W. O. Kermack & A. G. McKendrick. *A Contribution to the Mathematical Theory of Epidemics.* Proceedings of the Royal Society of London. Series A, vol. 115, no. 772, pages 700–721, August 1927.

[Laird 87] J.E. Laird, A. Newell & P.S. Rosenbloom. *SOAR : An architecture for general intelligence.* Artificial intelligence, vol. 33, no. 1, page 1–64, 1987.

[Laird 98] J. E. Laird & C. B. Congdon. *The Soar User's Manual Version 9.3.* 1998.

[Laird 08] J. E Laird. *Extending the Soar cognitive architecture.* In Artificial General Intelligence 2008 : Proceedings of the First AGI Conference, page 224, 2008.

[Laird 12] J. E Laird. The soar cognitive architecture. 2012.

[Lazarus 66] R. S Lazarus. *Psychological stress and the coping process.* 1966.

[Le Bon 95] Gustave Le Bon. Psychologie des foules. Presses Universitaires de France, Paris, 1995 edition, 1895.

[LeDoux 96] J. E LeDoux. *The emotional brain : The mysterious underpinnings of emotional life.* 1996.

[Leontjev 59] A. N. Leontjev. *Problems of the development of the mind.* 1959.

[Marinier 09] R. P Marinier, J. E Laird & R. L Lewis. *A computational unification of cognitive behavior and emotion.* Cognitive Systems Research, vol. 10, no. 1, page 48–69, 2009.

[Marsella 04] S. C. Marsella, D. V. Pynadath & S. J. Read. *PsychSim : Agent-based modeling of social interactions and influence.* In Proceedings of the international conference on cognitive modeling, page 243–248, 2004.

[Marsella 09] S. Marsella, J. Gratch, N. Wang & B. Stankovic. *Assessing the validity of a computational model of emotional coping.* In Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on, page 1–8, 2009.

[Marsella 10] Stacy Marsella, Jonathan Gratch & Paolo Petta. *Computational Models of Emotion.* In A blueprint for a affective computing : A sourcebook and manual. Oxford University Press, USA, Oxford, scherer, K.R., bänziger, t., & roesch, e. edition, 2010.

[McCrae 03] Robert R. McCrae & Paul T. Costa. Personality in adulthood : A five-factor theory perspective. The Guilford Press, 2003.

[Minsky 06] Marvin Minsky. The emotion machine. Simon & Schuster New York, 2006.

[Newell 59] A. Newell, J. C Shaw & H. A Simon. *Report on a general problem-solving program.* In Proceedings of the international conference on information processing, volume 1, page 256–264, 1959.

[Newell 90] Allen Newell. Unified theories of cognition, volume 187. Harvard University Press Cambridge, MA, 1990.

[Niewiadomski 10] R. Niewiadomski, V. Demeure & C. Pelachaud. *Warmth, competence, believability and virtual agents.* In Intelligent Virtual Agents, page 272–285, 2010.

[Niwa 02] Yuji Niwa & Erik Hollnagel. *Integrated computerisation of operating procedures.* Nuclear Engineering and Design, vol. 213, no. 2–3, pages 289–301, 2002.

[Ochs 09] M. Ochs, N. Sabouret & V. Corruble. *Simulation de la dynamique des emotions et des relations sociales de personnages*

virtuels. Revue d'intelligence artificielle, vol. 23, no. 2-3, page 327–357, 2009.

[Ortony 88] Andrew Ortony, Gerald L. Clore & Allan Collins. The cognitive structure of emotions. Cambridge University Press, 1988.

[Ortony 03] A. Ortony. *On making believable emotional agents believable*. Emotions in humans and artifacts, page 189–212, 2003.

[Premack 78] David Premack & Guy Woodruff. *Does the chimpanzee have a theory of mind ?* Behavioral and Brain Sciences, vol. 1, no. 04, pages 515–526, 1978.

[Resnick 89] L. B. Resnick. Knowing, learning, and instruction : Essays in honor of robert glaser. Lawrence Erlbaum, 1989.

[Rizzo 97] P. Rizzo, M. Veloso, M. Miceli & A. Cesta. *Personality-driven social behaviors in believable agents*. In Proceedings of the AAAI Fall Symposium on Socially Intelligent Agents, page 109–114, 1997.

[Rousseau 96] D. Rousseau & B. Hayes-Roth. *Personality in synthetic agents*. Knowledge Systems Laboratory Report No. KSL 96, vol. 21, 1996.

[Schelling 73] T. C. Schelling. *Hockey helmets, concealed weapons, and daylight saving : A study of binary choices with externalities*. Journal of Conflict Resolution, page 381–428, 1973.

[Scherer 01] K. R Scherer. *Appraisal considered as a process of multilevel sequential checking*. Appraisal processes in emotion : Theory, methods, research, vol. 92, page 120, 2001.

[Si 05] M. Si, S. C. Marsella & D. V. Pynadath. *Thespian : An architecture for interactive pedagogical drama*. In Proceeding of the 2005 conference on Artificial Intelligence in Education : Supporting Learning through Intelligent and Socially Informed Technology, page 595–602, 2005.

[Si 09] Mei Si, Stacy C. Marsella & David V. Pynadath. *Modeling appraisal in theory of mind reasoning*. Autonomous Agents and Multi-Agent Systems, vol. 20, no. 1, pages 14–31, May 2009.

[Smith 12] Adam Smith. The theory of moral sentiments. 1812.

[Smith 91] C.A. Smith & R.S. Lazarus. Emotion and adaptation. Oxford University Press, USA, 1991.

[Smith 07] E. R. Smith, C. R. Seger & D. M. Mackie. *Can emotions be truly group level ? Evidence regarding four conceptual criteria.* Journal of Personality and Social Psychology, vol. 93, no. 3, page 431, 2007.

[Spencer 70] H. Spencer. The principles of psychology, volume 2. Williams and Norgate, London, 1870.

[Suchman 87] L. A. Suchman. Plans and situated actions : the problem of human-machine communication. Cambridge Univ Pr, 1987.

[Svennevig 99] J. Svennevig. Getting acquainted in conversation : A study of initial interactions, volume 64. John Benjamins Publishing Company, 1999.

[Tarde 90] G. Tarde. Les lois de l'imitation : étude sociologique. Félix Alcan, 1890.

[Tsai 11a] J. Tsai, E. Bowring, S. Marsella & M. Tambe. *Empirical Evaluation of Computational Emotional Contagion Models.* In Intelligent Virtual Agents, page 384–397, 2011.

[Tsai 11b] J. Tsai, N. Fridman, E. Bowring, M. Brown, S. Epstein, G. Kaminka, S. Marsella, A. Ogden, I. Rika, A. Sheel, M.E. Taylor, Xuezhi Wang, A Zilka & M. Tambe. *ESCAPES-Evacuation Simulation with Children, Authorities, Parents, Emotions, and Social comparison.* In Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pages 457–464, 2011.

[Vigouroux 05] Auguste Vigouroux & Paul Juquelier. La contagion mentale. Octave Doin, Paris, 1905.

[Visetti 89] Y. M. Visetti. *Compte rendu de Plans and situated actions de L. Suchman.* Intellectica, vol. 1, no. 7, page 67–96, 1989.

[Winograd 87] Terry Winograd & Fernando Flores. Understanding computers and cognition : A new foundation for design. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1987.