CONCEPTION DE L'ARCHITECTURE D'UN HABITAT INTELLIGENT DEDIE A L'ASSISTANCE ET A LA TELESURVEILLANCE DE PERSONNES PRESENTANT DES TROUBLES COGNITIFS

par

Stéphanie Codron

mémoire présenté au Département d'Informatique
en vue de l'obtention du grade de maître ès sciences (M. Sc.)

FACULTÉ DES SCIENCES
UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, Mai 2004

National Library of Canada
Bibliothèque nationale du Canada

Acquisitions and Bibliographic Services
Acquisisitons et services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-94819-6
Our file Notre référence
ISBN: 0-612-94819-6

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Le 2 juin 2004,
Date

le jury a accepté le mémoire de Mme Stéphanie Codron dans sa version finale.

Membres du jury

M. André Mayers
Directeur
Département d'informatique

Mme Hélène Pigot
Codirectrice
Département d'informatique

M. Antoine Cornuejols
Membre
- Institut d'Informatique d'Entreprise

M. Marc Frappier
Président-rapporteur
Département d'informatique

Sommaire

Ce mémoire introduit le contexte et les objectifs du projet DOMUS et le situe dans le cadre de la recherche internationale. L'architecture de l'habitat intelligent proposée prend en compte les contraintes de la psychologie cognitive. Pour modéliser l'occupant, notre recherche s'est appuyée sur les théories suivantes : les théories de Baddeley, de Potter et de Norman et Shallice, et les architectures cognitives SOAR de Newell, EPIC de Meyer et Kieras, et ACT-R d'Anderson.

Un modèle global décrivant les composantes essentielles du système est présenté. Les trois principales composantes retenues sont l'*Assistance Cognitive*, la *Télésurveillance* et les *Urgences*. Ensuite, l'organisation des composantes d'*Assistance Cognitive* et de *Télésurveillance* est détaillée. Plus particulièrement, une sous-composante permettant la simulation d'une personne atteinte de troubles cognitifs est modélisée par une architecture cognitive basée sur la théorie de Norman et Shallice, et l'architecture cognitive d'Anderson : ACT-R. Nous espérons ici ouvrir la voie à des recherches visant à élaborer une nouvelle architecture fonctionnelle prenant en compte les mécanismes des processus superviseurs décrits dans la théorie de Norman et Shallice.

La réalisation informatique d'un prototype d'habitat intelligent en Java et en XML basé sur l'architecture décrite est ensuite abordé. Il s'agit d'un prototype réalisé en boîte noire dont le rôle essentiel est d'apporter une preuve du concept décrit et de mettre en relief les interactions entre les différentes composantes. Différents scénarios d'utilisation montrent les interactions entre les composantes de l'habitat et confirment la cohérence de l'architecture proposée.

Remerciements

Je tiens à exprimer ma gratitude à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.

Aussi je tiens à remercier M. André Mayers, mon directeur de recherche pour sa supervision et les nombreuses heures de discussions constructives au sujet de la conception de l'habitat.

Je tiens également à remercier Mme Hélène Pigot, ma codirectrice, pour ses nombreux conseils et sa vision toujours pragmatique du projet.

J'adresse également mes remerciements à l'ensemble du jury qui a accepté de consacrer du temps et du travail à la lecture de ce mémoire.

Je souhaite également remercier M. Marc Frappier, professeur de l'université de Sherbrooke, et M. Vincent Rialle, directeur du laboratoire TIMC-IMAG à l'université de Grenoble (France), pour leurs conseils sur la conception de l'habitat.

Je tiens également à remercier l'Université de Sherbrooke et le Département d'Informatique pour l'aide financière apportée.

Enfin, je remercie Mme Yolande Poupardin, ma mère, pour son soutien depuis la France et son aide dans la relecture de ce mémoire et M. Mustapha Hafsaoui, un étudiant en maîtrise, pour son aide en programmation.

Table des matières

Sommaire ii

Remerciements iii

Table des matières ix

Liste des abréviations x

Liste des figures xii

Introduction 1

I Etat de l'art 10

1 Les projets d'habitat intelligent 11

1.1 HIS, Habitat Intelligent et Santé 11

1.1.1 Les objectifs de leur projet 11

1.1.2 Les réalisations 13

1.1.3 HIS par rapport au projet DOMUS 14

1.2 *Assisted Cognition Project* 14

1.2.1 Leurs objectifs 15

1.2.2 Leurs réalisations 16

1.2.3 *Assisted Cognition Project* face au projet DOMUS 17

1.3 I.L.S.A : *Independant LifeStyle Assistant* 18

1.3.1 Leurs objectifs 18

1.3.2 Leurs réalisations 19
1.3.3 I.L.S.A vs DOMUS 20
1.4 AHRI : *The Aware Home Research Initiative* 20
1.4.1 Leurs objectifs 21
1.4.2 Leurs réalisations 21
1.4.3 AHRI et projet DOMUS 22
1.5 Prosafe : Projet du LAAS du CNRS de Toulouse 23
1.5.1 Leurs objectifs 23
1.5.2 Leurs réalisations 23
1.5.3 Le LAAS par rapport au projet DOMUS 25
1.6 Le projet DOMUS : une approche originale 26

2 Approche psychologique : les théories de la cognition 28
2.1 Qu'appelle-t-on connaissance en psychologie ? 28
2.2 Qu'appelle-t-on théorie, architecture cognitive ? 29
2.3 La théorie de Baddeley (1986) 31
2.4 La théorie de Potter (1990, 1993) : une théorie temporelle 32
2.5 La théorie de Norman et Shallice (1980, 1986) : présentation du contexte 34
2.5.1 Présentation générale de la théorie 34
2.5.2 Distinction de niveaux d'actions 36
2.5.3 Classification des erreurs cognitives dans le niveau intermédiaire . 38
2.6 La théorie de Norman et Shallice (1980, 1986) : fonctionnement du modèle computationnel de Cooper et Shallice 39
2.6.1 Principe général 39
2.6.2 L'organisation des schémas et des buts : le réseau de schémas . . 41
2.6.3 Le mécanisme d'activation du réseau de schémas 43
2.6.4 Le réseau d'objets et le réseau de ressources. 46
2.6.5 Les paramètres du système 48
2.6.6 Les résultats obtenus par Cooper et Shallice 48
2.6.7 Validité du modèle présenté par Cooper et Shallice 50
2.7 La théorie de Norman et Shallice (1980, 1986) : le module de supervision : S.A.S 51

3 Architectures cognitives **58**
3.1 Soar : une architecture cognitive de Newell, Laird et Rosenbloom 58
3.2 EPIC : Executive Process Interactive Control 61
3.2.1 Buts d'EPIC 61
3.2.2 Fonctionnement général 62
3.2.3 Fonctionnement du processeur cognitif : la mémoire de travail . . 64
3.2.4 Fonctionnement du processeur cognitif : le moteur d'inférence . . 65
3.2.5 Les modules périphériques 66
3.2.6 Conclusion 71
3.3 ACT-R : l'architecture cognitive d'Anderson 72
3.3.1 Principes généraux d'ACT-R. 73
3.3.2 La mémoire procédurale 74
3.3.3 Le module de la mémoire déclarative 78
3.3.4 Fonctionnement du système sensorimoteur. 81
3.3.5 Le module des buts 82
3.3.6 L'apprentissage dans ACT-R 83
3.3.7 Résultats et discussions sur l'approche d'Anderson 85

II Élaboration d'une architecture pour l'Habitat Intelligent 87

4 Architecture globale de l'habitat intelligent **88**
4.1 Une architecture en trois couches 88
4.2 Conception d'une architecture pour l'habitat intelligent 91
4.2.1 Identification des composantes 91
4.2.2 Un modèle global de l'architecture 95
4.2.3 Choix des composantes à spécifier 99
4.3 Le module de gestion des interventions 100
4.3.1 Scénario de fonctionnement 100
4.3.2 Rôle et fonctions du module de gestion des interventions 102

5 Assistance cognitive et Télésurveillance **104**
5.1 Identification des composantes et fonctions pour l'assistance cognitive . . 104

5.1.1 Types de données 105
5.1.2 La reconnaissance des activités 105
5.1.3 Calcul des probabilités 107
5.1.4 Le cycle de reconnaissance et de suivi dans son ensemble 109
5.1.5 Le diagnostic 111
5.2 Identification des composantes et fonctions pour la télésurveillance . . . 112
5.2.1 Traitement des données physiologiques 112
5.2.2 L'historique et les traitements associés 113
5.2.3 Télésurveillance des activités 114
5.3 Un modèle de fonctionnement pour les composantes d'Assistance Cognitive et de Télésurveillance 115
5.3.1 Nécessité d'un apprentissage 115
5.3.2 Nécessité d'une interaction avec l'occupant 117
5.3.3 Identification précise des composantes 117
5.3.4 Organisation du module Assistance Cognitive - Télésurveillance . 119
5.4 Problèmes et points à développer 121
5.4.1 Incertitude des données 121
5.4.2 Initialisation 122

6 Simulation de la personne : une nouvelle proposition d'architecture pour la cognition humaine 125
6.1 Vers une hybridation ACT-R - SAS 126
6.1.1 Un candidat pour le Contention Scheduling 127
6.1.2 Vers une spécification du SAS ? 128
6.2 Modèle de simulation de la personne 130
6.2.1 Rôle du module de simulation 130
6.2.2 La couche externe du module de simulation 131
6.3 Fonctionnement de ACT-R - SAS 132
6.3.1 Adaptations effectuées sur ACT-R 132
6.3.2 Adaptation du SAS 135
6.3.3 Fonctionnement global 138
6.4 Pistes de recherches futures 143

6.4.1 Le SAS - Planification 143
6.4.2 Points à développer 144

III Réalisation d'un prototype d'habitat intelligent 146

7 Réalisation et résultats 147
7.1 Description de l'implémentation 147
7.1.1 Simulation des données de l'habitat 147
7.1.2 Représentation des acteurs et plus particulièrement de l'occupant 150
7.1.3 La représentation des composantes du système 151
7.1.4 Communication entre les différentes composantes 154
7.1.5 Autres aspects de l'implémentation 161
7.2 Les résultats 162
7.2.1 Le scénario de la nuit 162
7.2.2 Le scénario du matin 163
7.2.3 Le scénario du midi et autres scénarios 164

8 Critiques et perspectives 167
8.1 Les conséquences de la mise en œuvre 167
8.1.1 Précision du modèle de la personne 167
8.1.2 Interactions entre la composante d'Analyse et de formulation des interventions et la simulation de la personne 168
8.1.3 La composante Apprentissage 168
8.1.4 Choix de l'effecteur 169
8.2 Pistes de développements et d'utilisation du prototype 169
8.2.1 Utilisation du prototype 169
8.2.2 Développements futurs 170

Conclusion **173**

Annexes **176**

A Glossaire **176**

B Plan d'un appartement intelligent **178**

C Composante Systèmes automatisés **193**

C.1 Systèmes automatisés 193

C.1.1 Description des Systèmes automatisés 193

C.1.2 Pistes de réalisations 195

Bibliographie **204**

Liste des abréviations

ACT-R : Adaptive Control of Thought - Rational

AHRI : Aware Home Research Initiative

AVQ : Activité de la Vie Quotidienne

CS : Contention Scheduling

COGENT : Cognitive Objects within a Graphical EnviroNmentT

DOMUS : DOmotique et informatique Mobile de l'Université de Sherbrooke

EPIC : Executive Process Interactive Control

HIS : Habitat Intelligent et Santé

ILSA : Independant LifeStyle Assistant

SAS : Supervisory Attentionnal System

Liste des figures

1 Le modèle de compétence de Rousseau 4
2 Schéma général de l'interaction entre les composants 6

1.1 Modèle de fonctionnement global du HIS 12
1.2 Plan du prototype d'habitat de Grenoble 13
1.3 Architecture de l'*Assisted Cognition Project* 16
1.4 Illustration du système I.L.S.A 19
1.5 Schéma général de principe du système de surveillance 24

2.1 Situation de diverses théories et architectures cognitives dans la perspective de dépendance au domaine et de la possibilité d'implémentation . . . 30
2.2 Structure générale du CS/SAS 35
2.3 Principales différences entre les trois niveaux d'actions 37
2.4 Les composantes du modèle de Cooper et Shallice 40
2.5 Exemple de réseau de schéma : la préparation du café 42
2.6 Schéma du CS/SAS Adaptation du schéma proposé par Shallice et Burgess 53
2.7 Correspondance entre le SAS et Domino proposé par Glasspool 56

3.1 Structure générale d'EPIC 62
3.2 Architecture d'ACT-R 75

4.1 Organisation des différents modèles et de leur instance 94
4.2 Légende pour le modèle global de l'architecture 95
4.3 Modèle global de l'architecture de l'habitat intelligent 97

5.1 Exemple de chaînes de Markov pour l'activité de toilette matinale 106

5.2 Structure des connaissances de la mémoire épisodique 108
5.3 Le cycle de reconnaissance et de suivi inclus dans la composante Assistance cognitive 110
5.4 Un modèle pour l'Assistance Cognitive et la Télésurveillance 120

6.1 Vue générale du module de simulation de la personne 131
6.2 L'architecture d'ACT-R adaptée 134
6.3 Proposition d'architecture pour le SAS 137
6.4 Structure de l'architecture hybride ACT-R - SAS 140
6.5 Structure du SAS - Planification 144

7.1 Fenêtre de l'interface usager permettant de configurer les différents scénarios 149
7.2 Fenêtre de l'interface usager pour le réglage des paramètres de l'occupant 151
7.3 Diagramme de classe des interfaces et classes abstraites pour représenter des composantes. 153
7.4 Modèle global de communication entre les composantes 155
7.5 Modèle de communication pour les composantes formant l'assistance cognitive et la télésurveillance 157
7.6 Modèle de communication pour la simulation de la personne 159
7.7 Lien de la composante assistance cognitive interactive avec le reste de l'architecture 160

C.1 Composante Systèmes automatisés 194

Introduction

Problématique du projet DOMUS

Aujourd'hui, dans les sociétés occidentales, nous vivons de plus en plus vieux mais aussi de plus en plus seuls. Les effets de ce vieillissement prolongé sont l'apparition de maladies spécifiques aux personnes âgées telle que la maladie d'Alzheimer. A l'heure actuelle, 5,1% de la population de plus de 65 ans est atteinte de cette maladie au Québec et 8% au Canada (Canadian Study of Health and Aging Working Group, 1994), 4 millions de personnes (National Institute for Aging) en 2000 pour les États Unis et 5% des plus de 65 ans pour la France (Touchon, 1999). En 2050, le nombre de malades d'Alzheimer atteindra 80 millions de personnes dans le monde entier selon les estimations de Shenk (2001).

Ceci n'est qu'un exemple. Les personnes atteintes de schizophrénie et celles souffrant de traumatismes crâniens sont d'autres exemples. De façon générale, de nombreuses personnes souffrent d'un dysfonctionnement des fonctions exécutives : attention, planification, décision, mémoire... Or, rien n'est prévu pour pouvoir maintenir des personnes atteintes de tels troubles à leur domicile. De ce fait, ces personnes sont souvent soit prises en charge par leurs proches qui sont rapidement débordés par les conséquences de ces troubles, soit hospitalisées, ce qui est coûteux à la société et à la famille. Le coût estimé aux Etats-Unis de la prise en charge des malades d'Alzheimer est de 100 milliards de dollars US (Ernst et al., 1997).

Objectifs du projet DOMUS

L'idée du projet DOMUS est donc de mettre en place un habitat intelligent capable de pallier les troubles cognitifs de la personne qui y vit. La création d'un tel habitat permettrait, d'une part, de soulager les familles de leur inquiétude et de la charge générée par quelqu'un souffrant de troubles cognitifs et, d'autre part, de maintenir plus longtemps la personne dans un environnement familier, ce qu'elle souhaite en général. D'un autre point de vue, la mise en place d'un tel habitat pourrait permettre de mieux suivre et de mieux comprendre les déficits cognitifs liés à la maladie d'Alzheimer, à la schizophrénie ou aux traumatismes crâniens.

Ce type d'habitat s'adresserait à des personnes présentant des troubles cognitifs qui les empêchent de mener à bien les activités de la vie quotidienne. Dans le cadre du projet MERCIS (Module d'Évaluation et de Remédiation Cognitives Implantable dans la Société), l'équipe de DOMUS collabore avec l'Université de Montréal et l'hôpital Louis H. Lafontaine. Ce projet s'adresse surtout à une clientèle du type schizophrène, mais aussi aux malades d'Alzheimer et à des adultes présentant des troubles cognitifs. Le projet est également développé en collaboration avec l'équipe du TIMC-IMAG, de l'Université de J. Fourier à Grenoble, France (Rialle, 2002). Leur projet «HIS» est décrit plus largement par la suite.

L'objectif du projet DOMUS est double : d'une part, aider les personnes atteintes de troubles cognitifs à vivre normalement en palliant leurs déficiences et, d'autre part, surveiller à long terme l'évolution de l'état de santé de la personne et prévenir rapidement le personnel qualifié en cas de problèmes (feu, chute, malaise ...). Pour répondre à ces objectifs, le système de l'habitat intelligent se décompose en deux modules distincts qui utilisent les mêmes bases de connaissances : le module d'assistance cognitive, chargé de l'aide à la réalisation des activités de la vie quotidienne (AVQ), et le module de télésurveillance, chargé de suivre l'état de santé de l'occupant.

En ce qui concerne la partie d'aide à la réalisation des AVQ, il s'agit de pallier les troubles cognitifs de l'occupant, de manière à ce qu'il puisse réaliser normalement l'en-

semble des tâches de la vie quotidienne. Cette aide nécessite une connaissance des habitudes de vie de la personne et l'élaboration d'un modèle de ses compétences en fonction de sa maladie. Il faut en effet que l'aide fournie soit adaptée. Une aide trop fréquente pourrait ennuyer l'occupant, voire aggraver ses pertes cognitives. En effet, le patient laisserait le système faire à sa place et n'exercerait plus ses capacités cognitives. Une aide trop rare laisserait la personne en situation de danger, ou du moins de risque. Il est donc important de doser l'aide, de la varier et de l'adapter.

En ce qui concerne le module de télésurveillance, il s'agit de pouvoir prévenir les personnes qualifiées de l'évolution de l'état de santé de l'occupant, mais aussi en cas de besoin ou de danger. Si, par exemple, il y a un début d'incendie causé par une cigarette mal éteinte, alors une intervention rapide est nécessaire. D'un autre coté, si l'état de santé du patient se dégrade, augmentation de la tension par exemple, alors une visite du médecin pour en déceler la cause est nécessaire. Il s'agit dans les deux cas d'une action passive de surveillance qui nécessite une connaissance des habitudes et de l'état de santé normal de la personne. Ces connaissances sont indispensables pour ne pas provoquer une intervention erronée qui ennuierait l'occupant ou le personnel aidant, voire qui provoquerait un rejet complet du système.

Au vu de ses objectifs, on voit qu'il est essentiel d'adapter le plus possible notre système aux spécificités de chacun. Cette adaptation assure d'une part l'efficacité du système et d'autre part l'acceptation du système par l'occupant.

Première réponse aux objectifs

Avant d'expliquer plus avant comment nous comptons réaliser ce système, nous allons parler du modèle développé par Rousseau et al. (2002) pour expliquer l'interaction personne/environnement. Le modèle de compétence de Rousseau et al. rassemble six concepts : personne, environnement, activité, rôle, compétence, handicap. Ils sont articulés dans le schéma la figure 1 (page suivante).

L'originalité de ce modèle vient du fait qu'il considère les handicaps ou compétences

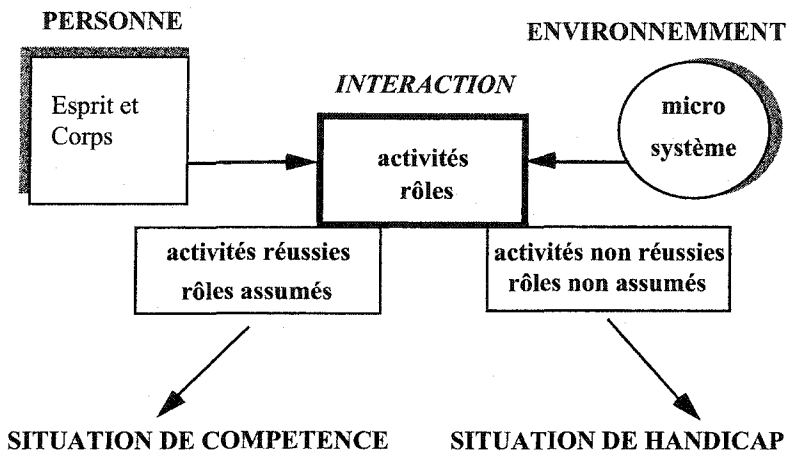


FIG. 1 – Le modèle de compétence de Rousseau et al. (2002)

de la personne par rapport à une situation dans un environnement donné. Les handicaps ou compétences ne sont plus génériques mais liés à l'interaction personne/environnement. Ainsi, lorsqu'il s'agit d'interagir avec un environnement humain, on parle de rôle et lorsqu'il s'agit d'interactions avec l'environnement physique, on parle d'activité.

Le concept d'activité représente une série de comportements. Plus précisément, elle se décompose selon la hiérarchie suivante : activité, tâches et opérations. L'activité de la vie quotidienne est représentée comme un ensemble de tâches, comme par exemple, préparer du café et des tartines le matin. Une activité implique nécessairement une intention, une raison de l'accomplir. Une activité n'est pas forcément liée à la volonté propre de la personne mais peut être le résultat des contraintes de la société.

A partir de là, le concept de compétence ou de handicap n'est plus centré sur la personne mais sur la situation. On parle de situation de compétence ou situation de handicap. Une situation de compétence est une situation dans laquelle une personne est capable d'interagir de manière effective avec son environnement. Autrement dit, il y a

adéquation entre les capacités/limites de la personne et les opportunités/demandes de l'environnement. A l'inverse, une situation de handicap est une situation où la personne se voit dans l'incapacité d'interagir efficacement avec son environnement. Ceci va de la difficulté légère et facile à surmonter, à une impossibilité totale pour certaines situations.

Ainsi, on peut voir que notre système s'intéresse à augmenter le nombre de situations de compétence lors de la réalisation d'activités par l'occupant. Pour parvenir à ceci, nous allons donc agir sur l'environnement via des messages visuels et autres signaux sonores. Nous ne pouvons en effet ni agir sur la personne (il ne s'agit pas de mettre au point un nouveau traitement), ni sur ses activités (il s'agit de laisser la personne libre de ses actes tant que ceux-ci ne sont pas dangereux). L'idée est donc de modifier l'environnement. L'assistance cognitive et la télésurveillance vont nous permettre de déterminer de quelles manières nous devons le modifier pour augmenter le nombre de situations de compétence.

Afin de réaliser les modules d'assistance cognitive et de télésurveillance, il est nécessaire d'établir un modèle de la personne et de ses compétences, un modèle des activités de la vie quotidienne et de leur déroulement. Un modèle de l'environnement tel qu'il est perçu par le système au travers des capteurs est également nécessaire afin de pouvoir déduire le déroulement des activités de l'occupant. La figure 2 (page suivante) présente les interactions entre ces modèles.

Dans le schéma de la figure 2 (page suivante), on retrouve les trois modèles que nous avons cités et une partie intégrateur. Les deux modules d'assistance et de télésurveillance sont compris dans cette dernière partie. Cette partie est celle qui devra décider des actions et/ou interventions à effectuer à partir des données traitées issues des capteurs et des modélisations de la personne, des AVQ et de l'environnement.

Elle s'occupera d'analyser les données issues des capteurs et va, grâce aux modèles de la personne, de l'environnement et des activités, extraire les informations pertinentes sous forme symbolique (phrases ou prédicats avec un degré de probabilité ou de certitude). Par la suite à l'aide de ces informations traitées, le système va devoir être capable de :

- savoir quelle est la tâche en cours ;

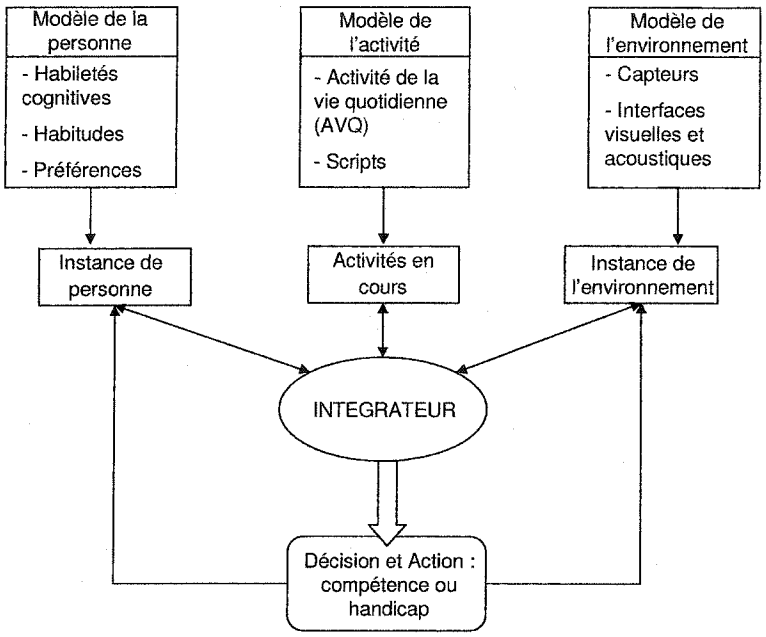


FIG. 2 – Schéma général de l'interaction entre les composants (Adapté de Pigot et al. (2003))

- savoir si la tâche en cours présente un danger ;
- savoir si la personne a besoin d'une aide ;
- simuler les comportements possibles de la personne pour prévoir les interventions utiles, le type d'intervention et le degré d'intervention ;
- décider d'intervenir ou non ;
- évaluer les capacités de la personne (suivi de la maladie).

Toutes ces inférences constituent la partie cognitive du système. On tente en quelque sorte de simuler un ergothérapeute [1] qui aurait comme sens les capteurs et comme mains les outils de communication du système : écrans, signaux sonores et contrôles sur un certain nombre d'appareils. Il est bien évident que le système informatique ne sera jamais capable de remplacer un ergothérapeute, mais il pourra appuyer le travail de celui-ci.

[1]**Ergothérapeute :** Personne qui, après avoir évalué les capacités physiques et mentales d'un patient aux prises avec des difficultés entravant ses activités quotidiennes, propose un programme individuel

Description du projet de maîtrise

La précédente description du système est très générale. Il n'en existe pas pour l'instant de plus précise. C'est ici qu'intervient ce projet de maîtrise. Le but de ce projet est de fournir un modèle général incluant les composantes nécessaires pour obtenir un système en mesure d'agir sur son environnement pour maximiser le développement de son occupant. La réalisation de ce modèle comprend la précision de son aspect cognitif, notamment la simulation de la personne pour prévoir ses comportements, et la précision des interactions entre chacune des composantes. L'accent est d'ailleurs mis sur la précision de ces interactions plutôt que sur l'élaboration d'un modèle pour chacune des composantes. Au final, l'objectif est de réaliser un prototype pour valider l'architecture cognitive globale du système sans, dans une certaine mesure, considérer la structure interne des composantes

Pour réaliser ce projet, il est prévu de s'appuyer sur différentes théories en psychologie cognitive, notamment celle de Norman et Shallice, et celle d'Anderson (ACT-R). Les effets de la maladie d'Alzheimer sur la cognition devront être compris et pris en compte, afin de mieux cerner les objectifs du projet et les problèmes qu'ils soulèvent. Ce travail bibliographique aura pour but d'acquérir suffisamment de connaissances pour élaborer un modèle d'architecture cognitive valide.

Par la suite, il s'agit de concevoir le prototype du système basé sur le modèle décrit. Comme il a été précisé, ce prototype aura pour fonction essentielle de mettre en avant les interactions entre les différentes composantes du système. Il ne s'agit pas, par contre, d'implémenter chacune des composantes. L'idée est donc de développer ce système en boîte noire. On simule le traitement effectué par chaque boîte en fonction de l'entrée fournie. D'un point de vue très simple, l'idée est de fournir une certaine chaîne de caractères en sortie en fonction des chaînes de caractères données en entrée. Dans le prototype réalisé, il ne devra pas s'agir seulement de chaînes de caractères mais plutôt d'objets (contenant un certain nombre d'attributs) qui circuleront entre les différentes composantes.

---

d'activités pour renforcer les habiletés du patient et, au besoin, des aides techniques et un aménagement de son environnement pour pallier ses difficultés.

Par la suite, nous allons décrire la méthodologie générale utilisée pour réaliser ce projet. Elle est précisée pour chacune des différentes parties en temps utile.

Méthodologie

L'idée était de commencer par étudier le projet DOMUS dans son ensemble et de bien comprendre ses objectifs. Pour cela, il était nécessaire de lire des présentations générales et des documents fournis par l'équipe de recherche. Dans le même temps, des rencontres avec tous les membres de l'équipe, que ce soit les professeurs ou les étudiants, ont été organisées, afin d'apprécier l'état d'avancement du projet.

Par la suite, le projet a débuté avec des recherches bibliographiques pour établir un état de l'art des projets du type habitat intelligent. Ce travail est présenté dans la première partie du mémoire. Cela a permis de mieux situer le projet DOMUS et ses avancées dans le contexte de la recherche internationale et de souligner que ce projet a une approche originale par sa façon de mêler psychologie cognitive et intelligence artificielle. La seconde partie des recherches bibliographiques a porté sur les théories de psychologie cognitive (Anderson et al. (2002), Norman et Shallice (1986)). Le résultat de ces lectures est présenté également dans la première partie de ce mémoire.

Dans un deuxième temps, il a fallu réfléchir sur un modèle global pour l'habitat intelligent où étaient seulement détaillées les composantes essentielles du système. Ce premier modèle a été le sujet de nombreuses réunions et discussions avec les différents membres de l'équipe. Par la suite, une fois ce premier modèle validé, l'intérêt s'est porté sur la composante chargée de fournir les fonctions d'assistance cognitive et de télésurveillance. Ceci a donné lieu à l'élaboration d'un sous-modèle précisant ces deux composantes. Enfin dans la composante chargée de l'assistance cognitive, l'accent a été mis sur la composante chargée de simuler la personne dans son environnement. Un modèle pour la simulation d'une personne a donc été conçu, basé sur les différentes théories en psychologie cognitive. Ce travail de modélisation de l'habitat et de ses composantes est entièrement présenté dans la deuxième partie de ce mémoire.

Dans un troisième temps, il fallait donner un corps informatique aux modèles théoriques expliqués dans la deuxième partie. Il y a donc eu une phase de conception plutôt courte puisqu'il s'agissait essentiellement de reprendre les composantes modélisées. Toutefois lors de la réalisation à proprement parler des composantes et surtout de leurs interactions, des oublis ou des erreurs de modélisation ont été identifiés. Soit il s'agissait d'aspects qui avaient été négligés dans la modélisation, soit d'interactions qui n'étaient en fait pas valides.

La modélisation proposée a donc été modifiée de façon à rectifier les erreurs ou à réparer les oublis. La modélisation proposée dans la deuxième partie est donc celle qui a été utilisée pour réaliser le prototype. En réalité, ces modifications n'ont pas été faites d'un bloc mais plutôt par petites touches au fur et à mesure de l'avancée de l'implémentation du prototype. Ce processus ainsi que les grandes étapes de réalisation du prototype sont détaillées dans la troisième partie de ce mémoire.

Nous allons donc maintenant aborder la première partie de ce mémoire qui rassemble sous forme d'un état de l'art, le travail bibliographique réalisé.

Première partie

Etat de l'art

Chapitre 1

Les projets d'habitat intelligent

Dans ce chapitre, seront abordés tout d'abord les projets similaires au projet DOMUS, qui sont à l'étude à travers le monde. Pour chacun, il sera fait un bref résumé de la façon dont la problématique de l'habitat intelligent est abordée et de l'avancée des travaux par rapport aux objectifs de chaque équipe.

1.1 HIS, Habitat Intelligent pour la Santé : Projet de Grenoble

Ce projet lancé en 2001 est sous la direction du laboratoire TIMC-IMAG de l'Université J. Fourier de Grenoble en collaboration avec le Centre Hospitalier Universitaire et la Faculté de médecine de Grenoble. Nous collaborons d'ailleurs également à ce projet par l'intermédiaire de visioconférences et d'échanges d'étudiants.

1.1.1 Les objectifs de leur projet

Le but du HIS est de permettre une meilleure autonomie à domicile pour les personnes âgées et pour les personnes atteintes de troubles cognitifs. L'accent est mis sur l'aspect télésurveillance qui va permettre aux personnes de demeurer chez elles en toute sécurité.

Le système de télésurveillance du HIS mettra en œuvre une chaîne complète de traitement d'informations allant de leur perception à leur analyse en termes :

- médicaux (diagnostic, évaluation fonctionnelle de la personne, télémonitoring) ;
- décisionnels (fusion de données, détection d'alarmes) ;
- statistiques (actimétrie diurne/nocturne) ;
- de gestion (droits d'accès, archivage, analyses rétrospectives, etc...).

En début de chaîne, se trouvent les nombreux capteurs installés en divers endroits de l'habitat. Ces capteurs sont reliés, via un réseau local filaire et/ou non filaire, à un logiciel d'acquisition et de traitement des signaux. Ce logiciel, local à l'habitat, est chargé d'analyser les événements, de surveiller l'évolution des paramètres et de détecter des situations à risques, des chutes, des appels vocaux... Il communique, par réseau téléphonique ou câblé, avec un poste de télévigilance médicale distant chargé de la réception et du traitement des informations et des alarmes. Des algorithmes permettent par exemple d'apprendre le comportement et les habitudes de la personne et de signaler d'éventuelles déviations importantes de ce modèle. Ce mode de fonctionnement est illustré par la figure 1.1 extraite du site du HIS (Laboratoire TIMC-IMAG, 2003).

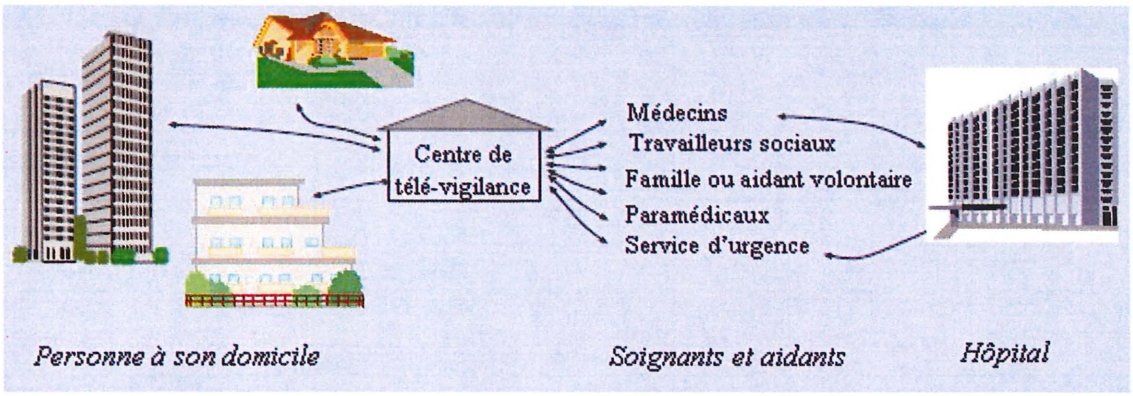


FIG. 1.1 – Modèle de fonctionnement global du HIS

L'objectif est donc de télésurveiller la personne pour éviter les risques liés au maintien à domicile tels que les chutes, les risques d'incendies, d'inondations. En cas d'urgence, le système est capable de prévenir le personnel concerné qui peut intervenir rapidement. Nous allons maintenant voir l'avancée des travaux de cette équipe, quelles sont leurs réalisations par rapport aux objectifs que nous venons de décrire.

1.1.2 Les réalisations

D'un point de vue physique, un prototype d'habitat a été mis en place dans les locaux du CHU de Grenoble (cf. figure 1.2 (Laboratoire TIMC-IMAG, 2003)). Il permet à l'équipe de tester les capteurs et les divers systèmes nécessaires à un tel habitat, mais aussi de collecter des données brutes pour tester les méthodes d'apprentissage, de reconnaissance de comportements... Il s'agit d'un prototype de logement de type T1 (50

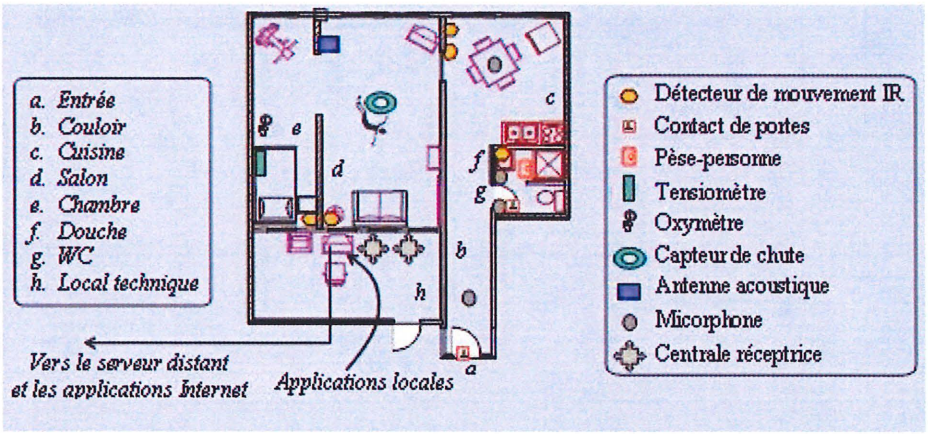


FIG. 1.2 – Plan du prototype d'habitat de Grenoble

m^2 environ), comprenant les zones d'habitat classiques que sont la chambre, le séjour, la cuisine, les toilettes, la douche et un couloir. Une zone technique attenante à l'appartement a été ajoutée afin de recevoir le système informatique d'expérimentation du projet.

D'un point de vue informatique, une première version de système distribué multi-utilisateurs a été réalisée selon une architecture orientée-objet et en utilisant les techniques de représentation UML et les technologies développement Java (RMI, Java Web Start, etc.) et XML. Ce logiciel couvre la chaîne complète d'informations depuis leur perception (capteurs), jusqu'à leur présentation au médecin (télésurveillance) et à leur analyse en termes décisionnels (fusion de données, détection d'alarmes), médicaux (diagnostic) et statistiques (actimétrie diurne/nocturne, etc...). Il constitue un prototype de système distribué sur Internet selon une architecture n-tiers. Le logiciel fonctionne simultanément sur autant d'ordinateurs que d'accédants autorisés, au moyen d'un navigateur

classique (HIS).

Le système d'information distribué mis en place est en phase de test ; il demande à être amélioré et ajusté pour répondre pleinement aux objectifs du projet. Il est vraisemblable que l'équipe a encore de nombreux problèmes à résoudre pour pouvoir appliquer dans la vie courante un tel système, mais rien de tout cela n'est précisé dans les références de ce projet (Rialle et al. (2001),(2002), Noury et al. (2000)).

1.1.3 HIS par rapport au projet DOMUS

Le HIS est un projet très proche du nôtre par la composante télésurveillance du système. De la même manière, le système du projet DOMUS sera capable de prévenir le personnel compétent en cas de danger et d'informer les médecins de l'évolution de la maladie. Par contre, nous sommes loin d'être aussi avancés que le HIS dans ce domaine.

Nous partageons des buts communs, notamment celui d'assurer une autonomie et une sécurité à domicile pour des personnes agées ou souffrant de déficiences. Il faut signaler que le HIS se préoccupe plus de l'aspect télésurveillance. Il n'est en effet pas destiné à une clientèle souffrant de troubles cognitifs. L'équipe de DOMUS se préoccupe plus de l'aspect assistance cognitive. Les deux projets sont donc complémentaires. Aussi, nous nous tenons régulièrement informés de leurs travaux et il existe un dialogue permanent entre les deux projets.

1.2 *Assisted Cognition Project* : Projet de l'université de Washington à Seattle

Ce projet, qui a débuté en 2001, est sous la direction d'une équipe du département d'informatique et de génie de l'Université de Washington à Seattle composée de Henry Kautz, Oren Etzioni et Dieter Fox en collaboration avec des médecins de diverses associations s'occupant de la maladie d'Alzheimer (notamment *Alzheimer's Disease Research Center*) (Kautz et al., 2001).

1.2.1 Leurs objectifs

Le but principal de ce projet est d'aider les personnes souffrant de limitations cognitives. Pour cela, la situation de la personne est interprétée à l'aide de capteurs aussi bien à son domicile qu'à l'extérieur (Kautz et al., 2003).

Le système du *Assisted Cognition Project* apprend à interpréter les comportements lors des Activités de la Vie Quotidienne et à reconnaître les erreurs de l'utilisateur, son éventuelle confusion ou détresse. Ceci est réalisé à l'aide de techniques de statistiques, de reconnaissance, et d'apprentissage. Le système doit être également capable d'offrir une aide avant l'action pour aider au moment approprié l'utilisateur grâce à des avertissements visuels, alarmes ou d'autres types d'interventions.

Comme pour notre projet, ce système s'adresserait principalement aux malades d'Alzheimer qui sont de plus en plus nombreux aux Etats-Unis : il s'agit de soulager la société et la famille des charges entraînées par cette maladie.

Divers systèmes sont actuellement à l'étude (Kautz et al., 2003) :

- *Activity Compass* : Système de guidage personnel qui apprend les caractéristiques de l'utilisateur dans la réalisation de ses AVQ, prédit les situations de handicap et aide à la réorientation du patient. Il s'agit d'un système portatif à la manière d'un GPS qui fonctionne à l'intérieur comme à l'extérieur.
- *AVQ Monitor* : Système situé dans la maison qui évalue les performances de l'utilisateur et détecte les anomalies, erreurs, et comportements anormaux.
- *AVQ Prompter* : guide l'utilisateur lors de la réalisation de tâches complexes, il peut être distribué dans la maison ou sur le système portatif.

Comme on peut le voir sur la figure 1.3 (page suivante), Kautz décrit une architecture en couches. Les données sont récupérées puis fusionnées grâce aux différents modèles. Par la suite, une analyse permet la reconnaissance de l'action puis une prise de décision pour l'aide à effectuer. L'échange du serveur avec les modèles permet l'adaptation du système à l'occupant. L'échange entre le serveur et l'interface fournit une aide effective en cas de besoin à travers les *prompters* et autres guides tel le *AVQ Compass*. Les auteurs ne sont

pas plus précis quant au fonctionnement de cette architecture en couche.

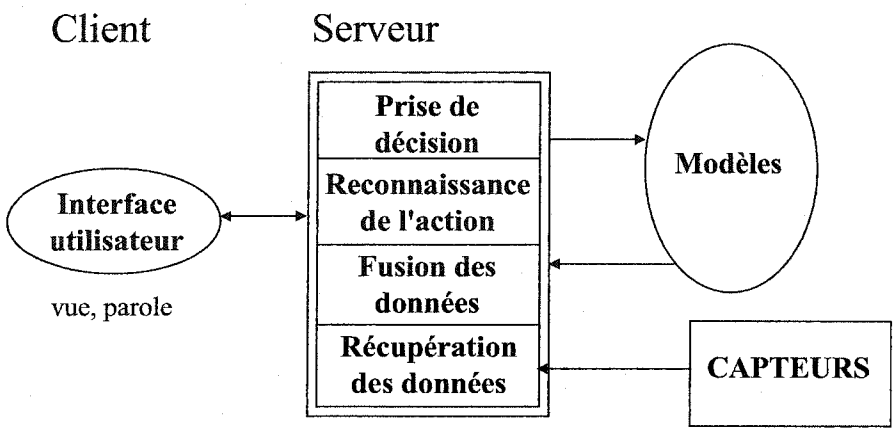


FIG. 1.3 – Architecture de l'*Assisted Cognition Project*, traduction libre (Kautz et al., 2001)

D'autre part, l'idée principale de Kautz pour modéliser le système cognitif en lui-même est d'utiliser une extension des réseaux Bayésiens associée à des modèles statistiques complexes (Kautz et al., 2003). Il espère grâce à ces représentations et aux recherches de Shastri (1993) dans le domaine neuronal pouvoir implémenter une théorie des performances et erreurs cognitives, et, par la suite, baser la décision d'intervention sur cette théorie tout en prenant en compte le coût et le bénéfice d'une intervention à travers le temps.

1.2.2 Leurs réalisations

Pour la localisation de la personne à l'intérieur ou à l'extérieur, plusieurs expériences ont été réalisées et les chercheurs s'orientent vers une méthode de localisation similaire

à celle utilisée pour les robots mobiles. Le problème majeur est la représentation de l'incertitude dans la localisation due à l'imprécision des capteurs et du système GPS.

Pour la modélisation du déroulement de la journée, le formalisme retenu est celui des semi-chaînes de Markov associées à des probabilités de transitions évoluant dans le temps (on note ici une orientation similaire à la nôtre). Pour la modélisation de l'utilisateur, ils ont retenu le formalisme bayésien, afin de mieux modéliser les changements de buts de l'utilisateur. Une fois la modélisation effectuée, la reconnaissance du but général de l'utilisateur, quant à elle, s'appuiera sur des travaux effectués précédemment par Kautz dans le cadre d'autres projets (Kautz et al., 2003) et le modèle cognitif s'appuiera sur les travaux de L. Shastri sur les modèles plausibles de fonctionnement des neurones humains. Mais il ne s'agit ici que d'annonces, ces idées n'ont pas encore été mises en pratique.

D'autres parties du projet sont à l'étude, notamment ils étudient comment modéliser les erreurs de l'utilisateur mais il ne s'agit encore que d'idées et ne font pas mention de réalisations particulières.

1.2.3 *Assisted Cognition Project* face au projet DOMUS

La problématique de ce projet est donc identique à la nôtre, il s'agit de fournir une assistance aux personnes limitées cognitivement, qu'elles soient à l'intérieur ou à l'extérieur. La partie assistance à l'extérieur est une différence avec le projet DOMUS, mais dans la description du projet, il est envisagé de fournir une aide à l'extérieur dans le cadre du projet MERCIS, une fois que le système concernant le domicile sera bien avancé.

Pour la partie cognitive de leur système, ils énoncent des idées, mais elles sont peu précises et n'ont donné lieu à aucune réalisation. La partie la plus développée de leur projet constitue la localisation d'une personne à l'extérieur. Pour les autres parties, Kautz reste très vague sur les réalisations.

1.3 I.L.S.A : *Independant LifeStyle Assistant*, Projet de l'entreprise Honeywell au Minnesota

Il s'agit d'un projet du laboratoire : *National Institute of Standards and Technology and Honeywell Laboratories* financé par Honeywell, entreprise qui est connue dans le domaine de l'aérospatiale, du transport et de la conception de systèmes automatisés (Entreprise Honeywell (Honeywell (2003))).

1.3.1 Leurs objectifs

Le but du I.L.S.A est de concevoir un système permettant aux personnes âgées de rester plus longtemps à leur domicile en utilisant les ressources de l'internet et d'une télésurveillance de la personne. De façon générale, ce système est conçu pour être très interactif, c'est-à-dire que la personne âgée pourra par exemple faire ses courses par internet, ou encore dialoguer avec le système. De plus, l'aspect surveillance est présent par l'utilisation de nombreux capteurs afin de suivre le comportement de la personne et éviter des comportements fâcheux.

La volonté du I.L.S.A est d'intégrer un ensemble de techniques et de systèmes électroniques comme des détecteurs de chutes, des détecteurs de mouvements, etc..., en un seul système capable à partir des données issues de différents capteurs et des différents systèmes de raisonner, d'analyser et de reconnaître les comportements de l'occupant puis de prendre les décisions nécessaires. Il est évident que les décisions prises respectent les préférences de l'occupant. Le projet est d'ailleurs qualifié de «hautement paramétrable» par ses concepteurs (Haigh et al., 2002a).

Pratiquement, le système du I.L.S.A comprend des capteurs qui sont disséminés dans l'appartement et des agents (signaux visuels...) qui permettent d'interagir avec l'occupant pour l'aider. Ils utilisent un modèle probabiliste pour reconnaître et déduire les buts de l'utilisateur dans ses activités de tous les jours. Par la suite, une autre partie du système va décider quels messages envoyer par rapport aux activités reconnues et aux compétences de l'occupant. Les messages sont classés pour déterminer leur priorité. En-

suite ils sont affichés ou énoncés là où l'occupant est susceptible de se trouver. De cette manière, le système entend pouvoir fournir une aide à l'occupant en le guidant dans sa vie quotidienne.

1.3.2 Leurs réalisations

D'un point de vue recherche, l'équipe a procédé à une analyse des besoins et des technologies disponibles à l'aide de méthodes d'analyse de données. Ils ont retenu une série de vingt points cruciaux qui devront être implémentés dans le système. Sans les détailler un par un, on peut citer les principales catégories : vérification de la prise de médicaments, capacité de surveiller les soins personnels (toilette), capacité de surveiller la mobilité (allées et venues, chutes), sécurité (bouton panique ...), facilité d'utilisation du système, mise en place de rapports (état de santé, activité...) et des fonctions utiles au personnel soignant. Ces diverses fonctions sont des sujets d'études à part entières dont on retrouve les publications sur le site du ILSA.

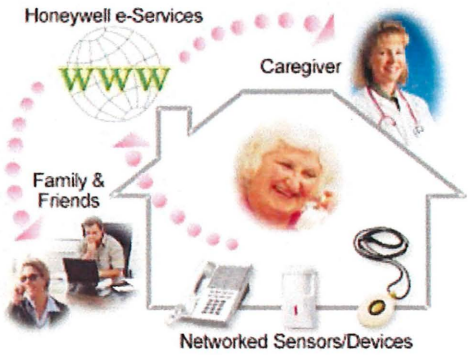


FIG. 1.4 – (Extrait du site I.L.S.A)

Des résultats intéressants ont été obtenus en ce qui concerne la reconnaissance des activités au long d'une journée (Guralnik et Haigh, 2002). Notamment, elles reconnaissent des motifs de comportement à partir des données issues des capteurs et tentent d'interpréter ensuite ces motifs pour établir le comportement général de la personne.

D'autre part, en ce qui concerne le système complet, une architecture et une série de fonctionnalités ont déjà été établies (Haigh et al., 2002a). Les diverses fonctionnalités comme le module de reconnaissance des événements, la reconnaissance des intentions, le système de design de la réponse (en fonction des préférences de l'utilisateur, de ses habitudes etc...) ont des rôles bien définis et sont clairement expliqués par les auteurs (Haigh et al., 2002a). Actuellement, ils sont en train d'implémenter le système. Ils mentionnent également leur volonté de créer une base de connaissances associée au système qui permettrait de définir des fonctionnalités supplémentaires pour le système.

1.3.3 I.L.S.A vs DOMUS

Ce projet est proche du nôtre par sa problématique qui est de faciliter la vie des personnes à domicile en leur proposant une aide, un aspect télésurveillance et des services internet. La composante «assistance» est donc présente, mais elle n'est pas envisagée d'un point de vue cognitif. La description du système fait bien état d'une aide via des affichages visuels et à l'aide de messages personnalisés, mais pas d'une aide pour des personnes atteinte de troubles cognitifs. D'ailleurs, la population cible de ce projet est la tranche des plus de 60 ans et non une clientèle spécifique présentant des troubles cognitifs. Toutefois, ils sont confrontés à des problèmes similaires aux nôtres pour ce qui est de la reconnaissance d'activités et des types d'interventions à dispenser. Enfin, d'un point de vue technologique, comme ce projet travaille avec l'entreprise Honeywell, il est possible que cette équipe fasse état d'appareils ou de systèmes qui pourraient être utiles au projet DOMUS.

1.4 AHRI : *The Aware Home Research Initiative*, Projet du Georgia institue of technologies à Atlanta

Ce projet a été mis en place depuis 1999 par l'institut de technologie de Géorgie basé à Atlanta. Il est dirigé par Gregory D. Abowd, et Irfan Essa, Beth Mynatt. A l'origine le groupe de recherche FCE travaillait déjà sur des projets de ce type mais à échelle plus

réduite.

1.4.1 Leurs objectifs

De façon générale, ils ont des objectifs communs au I.L.S.A. de Honeywell. Il s'agit de rassembler les différentes technologies de domotique pour les intégrer dans un seul système. Leur volonté est de faire passer l'ordinateur du bureau à l'ensemble de la maison. Ils suivent ainsi la tendance générale de l'*ubiquitous computing* ou informatique diffuse en français. Plus précisément, ils ont décomposé le projet en quatre objectifs :

- Design de systèmes légers et/ou portatifs pour aider les personnes dans leur quotidien, par exemple un aide-mémoire, et une sorte de pendentif-télécommande du système ;
- Développement et création des locaux physiques pour tester et mettre en place les différents systèmes et capteurs ;
- Construction du système informatique adapté ;
- Examen des implications sociales et éthiques de leurs travaux.

Chaque objectif est décomposé en plusieurs sous-projets, qui sont à l'étude. On peut signaler notamment une étude sur un système de localisation de la personne dans un appartement et sur un autre système capable de reconnaître les activités de la personne, ou encore un système capable de reconnaître une personne. L'aide-mémoire est un sous-projet important qui fait partie de leur volonté d'aider les personnes âgées à vivre mieux quotidiennement. Il est question de leur fournir un support cognitif via l'aide mémoire. Si, par exemple la personne est interrompue par le téléphone lors de la préparation d'un repas, l'aide mémoire serait alors là pour lui rappeler l'activité commencée. Il y a donc un objectif supplémentaire d'aide aux activités de la vie quotidienne. De plus, ils envisagent également la composante télésurveillance à distance (Kidd et al., 1999).

1.4.2 Leurs réalisations

Depuis 1999, ils ont à disposition une maison de deux étages : l'*Aware Home* qui leur permet d'expérimenter leurs systèmes. Chaque étage est aménagé de la même manière, l'un sert d'habitat permanent pour des étudiants (à terme ils veulent qu'une personne

âgée ou une famille y vive), l'autre sert de laboratoire à échelle réelle.

Les différents projets ont donnés lieu à plusieurs publications qui sont toutes ou presque disponibles sur leur site internet (Abowd et al., 2003). Ils ont actuellement réalisé plusieurs systèmes intéressants. On peut citer un système pour retrouver des objets égarés comme des clés de maison, un système de portrait digital dynamique qui permet aux proches de suivre l'intensité des activités de la personne via un affichage graphique...

En ce qui concerne l'aide mémoire, ils sont aux prises avec les mêmes difficultés que les autres projets, c'est-à-dire la reconnaissance des activités. Mais ils cherchent à contourner ce genre de difficultés en filmant certaines activités par exemple et s'il y a eu interruption en repassant à la personne les dernières images enregistrées pour qu'elle se souvienne de ce qu'elle était en train de faire. Ils tentent aussi de faire de la reconnaissance à travers ces vidéos.

1.4.3 AHRI et projet DOMUS

Leur voie de recherche est plus orientée vers l'intégration d'un ensemble de techniques pour une maison informatisée et intelligente que vers l'assistance aux personnes souffrant de limitations cognitives. Il s'agit pour nous d'un autre aspect de notre projet qui est pris en charge au sein de l'équipe DOMUS par Sylvain Giroux et Philippe Mabilleau. Toutefois, ils ont des objectifs communs, lorsqu'il s'agit pour eux de reconnaître des activités et d'offrir un support cognitif aux personnes âgées. Cette approche n'est pas envisagée de la même manière puisqu'il s'agit avant tout pour ce projet de développer un système populaire et non un système pour une clientèle spécifique. A ce sujet, Kidd et al. (1999) mentionnent qu'il ne doit pas «s'agir de personnes qui demanderaient des soins médicaux réguliers». Leur problématique est donc quelque peu éloignée de la nôtre de ce point de vue. Le projet DOMUS a un objectif beaucoup plus précis en ce qui concerne l'assistance cognitive.

1.5 Prosafe : Projet du LAAS du CNRS de Toulouse

Ce projet est développé par le LAAS : Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, à Toulouse sous la direction de Éric Campo, Daniel Estève et Marie Chan.

1.5.1 Leurs objectifs

Prosafe vise à fournir une assistance automatisée aux personnes âgées. L'objectif de ce projet est de «suivre une personne âgée vivant seule dans son milieu en ne perturbant en rien ses habitudes de vie et en instrumentant l'habitat par des capteurs classiques : détecteurs de passage, de mouvement, lumière, présence (...) de manière à réaliser par fusion multisensorielle un diagnostic de toute situation anormale» (LAAS, Laboratoire du CNRS de Toulouse, 2003). Ces capteurs sont reliés à un PC lui-même connecté à un serveur et à un surveillant (personnel médical, centre de surveillance...) ou à l'entourage de la personne âgée. Le système multicapteur assurerait une continuité dans la surveillance et le suivi de la personne. La différence avec la téléassistance traditionnelle pour une personne âgée est le déclenchement automatisé de l'aide, c'est-à-dire que ce n'est pas la personne qui va faire appel aux surveillants via un système de communication particulier, mais le système lui-même.

Le concept de ce système est basé sur l'utilisation de méthodes d'apprentissage et suivant le schéma de principe de la figure 1.5 (page suivante). Le principe de cette surveillance est de ne pas instrumenter la personne concernée mais simplement de détecter des actes précis caractéristiques de leur mode de vie, de les traiter statistiquement lors d'une phase dite d'apprentissage, de considérer les éléments obtenus comme des éléments de référence pour diagnostiquer, par la suite, tout écart à ces références.

1.5.2 Leurs réalisations

Un site expérimental simulant le lieu de vie d'une personne âgée dans une institution (maison de retraite, hôpital long séjour) a été mis en place au laboratoire de recherche ICARE de l'IUT de Blagnac. Le site comprend des capteurs infrarouges passifs placés au

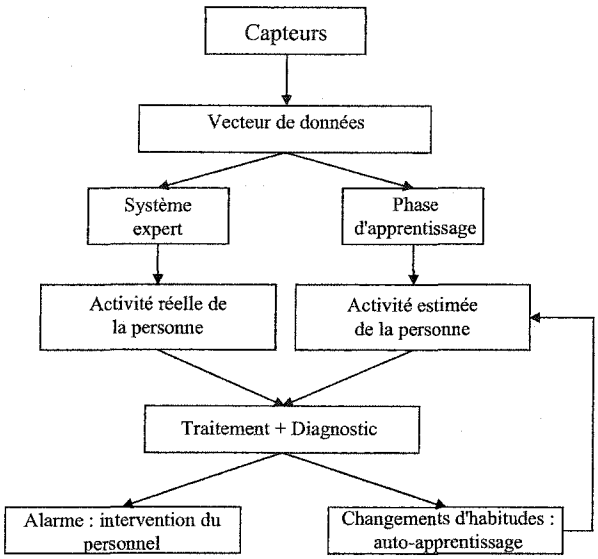


FIG. 1.5 – Schéma général de principe du système de surveillance. (Site du LAAS)

plafond, un réseau de communication filaire et un PC (à ce jour, une version sans fil a été expérimentée et validée ; elle est en cours d'optimisation). Les informations délivrées par les capteurs sont recueillies par des modules d'acquisition numériques qui les retransmettent au PC qui sert à stocker les informations collectées et à les traiter.

Ils ont également mis en place des systèmes en milieu hospitalier :

- à l'Hôpital Long Séjour de Muret (CHS Robert-Debré) : Le système est installé depuis juillet 98 dans une chambre hébergeant pour observation des personnes âgées atteintes de la maladie d'Alzheimer. Ce site sert à mettre au point les logiciels de traitement et à constituer une base de données importante ;
- au Centre de Médecine Gériatrique de Casselardit : un système multicapteurs similaire à celui de Muret. Ce site expérimental sert à valider le système d'un point de vue technique et logiciel.

Pour la mise au point du système sécuritaire, deux logiciels de traitements de données ont été développés, relatifs, d'une part au diagnostic des situations anormales et d'autre

part au suivi de personnes âgées pour des études sur leurs habitudes.

Pour le diagnostic des situations anormales, les logiciels vont dépendre des capteurs utilisés (pression, infrarouges...) et du réseau de communication utilisé (avec ou sans fil). L'aboutissement de cette partie est la mise en place du système relié à un serveur par l'intermédiaire d'un poste informatique et à un service prenant en charge les urgences pour assurer la sécurité des personnes âgées pouvant être maintenues à domicile. Le système doit être dans ce cas capable de localiser la personne surveillée et de déclencher le plus rapidement possible une alarme en cas de situations anormales : chute, fugue, malaise... L'algorithme en cours de développement doit nécessairement fonctionner en temps réel et doit tenir compte du comportement passé du patient pour fiabiliser au mieux le déclenchement d'alarmes.

Pour les études sur l'habitude, le logiciel est décrit pour obtenir les différents paramètres d'occupation des lieux par la personne. Cette partie s'apparente à une mise en évidence des différents paramètres de mobilité de la personne. On trouve pour chaque individu et pour chaque jour, une courbe des activités du jour et de la nuit. Une comparaison et une classification des différentes courbes permettent de dire si des courbes types existent pour les personnes et si elles sont utilisables pour une prévention de situation anormale. La mise en œuvre des méthodes d'apprentissage des habitudes peut se faire si les courbes types de mobilité ont été identifiées. Ces méthodes peuvent être testées pour le déclenchement d'alarme en cas de situations anormales et pour leur prévention.

Sur le site de démonstration du système Prosafe, il y a plusieurs exemples de fonctionnement et des captures d'écrans de l'interface utilisateur. C'est un projet qui est bien avancé puisque leur prototype de système est déjà en place.

1.5.3 Le LAAS par rapport au projet DOMUS

Ce projet s'adresse aux personnes âgées et aux personnes qui présentent des handicaps cognitifs. Toutefois, il s'agit uniquement de l'aspect télésurveillance de la personne âgée et non d'une réelle aide dans le quotidien. La composante assistance est complètement ignorée. Mais encore une fois, ce projet rencontre une problématique identique pour ce

qui est de l'identification des activités en cours et le degré d'intervention du système. Celui-ci devra-t-il donner l'alarme si la personne n'a toujours pas mangé à 14h de l'après midi ? De notre point de vue, il sera donc intéressant de suivre l'avancée de leurs travaux en ce qui concerne la phase de traitement et de diagnostic, mais aussi pour la phase d'apprentissage du système.

1.6 Le projet DOMUS : une approche originale

De façon générale, les cinq projets cités présentent tous la composante «télésurveillance». Ils ont tous la volonté de soulager les familles de l'inquiétude de laisser seule une personne âgée ou limitée cognitivement chez elle. Diverses idées sont envisagées, d'un simple portrait indicateur pour le AHRI à un système de surveillance complet comme le HIS ou le projet Prosafe.

En ce qui concerne la composante «assistance cognitive à la réalisation des AVQ», seul le projet de Kautz et de ses collègues en fait mention de manière explicite. Mais leur orientation diffère de celle du projet DOMUS. Nous comptons, en effet, nous baser sur des théories de psychologie cognitive, tandis que Kautz et ses collègues pensent utiliser le modèle de Shastri sur le fonctionnement neuronal. En ce qui concerne les autres projets, l'idée d'assistance à l'occupant est présente sous diverses formes et à divers degrés. Dans le LAAS, cette composante est absente. Pour le HIS, la collaboration avec notre projet assurera l'aspect assistance cognitive. Pour le I.L.S.A, le système est développé de manière à pouvoir fournir une aide à la personne qui se rapproche beaucoup d'une aide cognitive, mais ils n'envisagent pas d'utiliser les théories de psychologie cognitive pour arriver à leurs fins. Enfin, pour le AHRI, l'aide à l'occupant se fera via un aide mémoire qui permettra à l'occupant de garder le fil de ses activités. Ce n'est pas à proprement parler une aide cognitive, mais cela s'en rapproche, car il s'agit d'une aide sur une fonction cognitive : la mémoire.

Le projet DOMUS est donc pour l'instant le seul projet à avoir une approche s'appuyant sur la psychologie cognitive. L'idée est d'intégrer au système de l'habitat intelligent, une composante capable de prévoir les comportements (handicap ou compétence)

de la personne atteinte de troubles cognitifs et de lui fournir l'aide avant qu'elle ne soit en difficulté.

Dans les chapitres suivants de la première partie, nous allons nous intéresser aux théories et avancées en psychologie cognitive et en intelligence artificielle. Cette partie permettra de surcroît de définir les concepts essentiels utilisés dans la conception d'un modèle pour l'habitat intelligent.

Chapitre 2

Approche psychologique : les théories de la cognition

Dans ce chapitre, nous allons présenter quelques concepts liés à la psychologie cognitive, puis nous expliquerons les diverses théories de psychologie cognitive qui ont retenus notre attention.

2.1 Qu'appelle-t-on connaissance en psychologie ?

En psychologie, il faut distinguer deux types de connaissance selon leur rôle et leur utilisation dans les processus de cognition :

a) *Les connaissances déclaratives* : il s'agit des connaissances dites génériques qui servent à la représentation du monde. Dans les connaissances déclaratives on distingue :

- Les connaissances sémantiques indépendantes du vécu de la personne qui prennent souvent la forme d'une affirmation : le chien de ma grand-mère s'appelle Médor ;
- Les connaissances épisodiques : ce sont des connaissances qui représentent notre perception des événements et notre vécu. Par exemple, aujourd'hui, je suis allée voir mon ami d'enfance.

b) *Les connaissances procédurales* : connaissances qui peuvent s'apparenter au savoir-faire qu'il soit manuel ou intellectuel. Un exemple de connaissance procédurale sera :

savoir résoudre une équation du second degré. La définition exacte est «manipulations cognitives ou comportementales maîtrisées pour atteindre un but». La notion de but atteint est très importante, c'est ce qui va être mis en avant dans la plupart des architectures cognitives dont nous allons parler.

2.2 Qu'appelle-t-on théorie, architecture cognitive ?

Premièrement, commençons par définir le mot cognition : la cognition est une fonction complexe multiple regroupant l'ensemble des activités mentales (pensée, perception, action, volonté, mémorisation, rappel, apprentissage) impliquées dans la relation de l'être humain avec son environnement et qui lui permettent d'acquérir et de manipuler des connaissances (associations, rétroaction, traitement de l'information, résolution de problèmes, prise de décision, etc.).

Une théorie de la cognition a donc pour but de modéliser et d'expliquer les processus de la cognition humaine. La cognition est une fonction très complexe, aussi il n'existe pas de théorie absolue qui expliquerait tous les mécanismes des activités mentales humaines. Les psychologues eux-mêmes ne se consacrent souvent qu'à un seul aspect particulier de la cognition.

Il est nécessaire ici de bien faire la distinction entre une théorie de la cognition, une architecture cognitive, un outil de modélisation de la cognition et leurs implémentations respectives. Une théorie tente d'expliquer les processus de la cognition humaine et de les organiser comme je viens de le dire plus haut. Il s'agit ici d'un travail d'explication direct, et dont en général, il existe peu ou pas d'implémentation informatique. Une architecture cognitive est, si on reprend la définition donnée par Grant (1995), un ensemble particulier de structures, d'outils, de techniques et de méthodes qui vont permettre de construire et de tester des modèles de la cognition. Selon Anderson (1993), une architecture cognitive est un ensemble de propositions relativement complet à propos de la structure de la cognition humaine. Une architecture cognitive contient donc des éléments de théorie sur la cognition et permet de construire et tester des modèles de la cognition. Enfin un outil de modélisation de la cognition contient lui le moins possible d'à priori psychologiques

pour permettre au concepteur d'apporter sa propre façon de voir. Différents exemples en sont donnés par la suite. Pour résumer, on peut considérer le graphique suivant inspiré de Grant (1995), où s'organisent les diverses théories selon deux axes : la dépendance par rapport au domaine et le niveau de spécification pour l'implémentation (cf. figure 2.1).

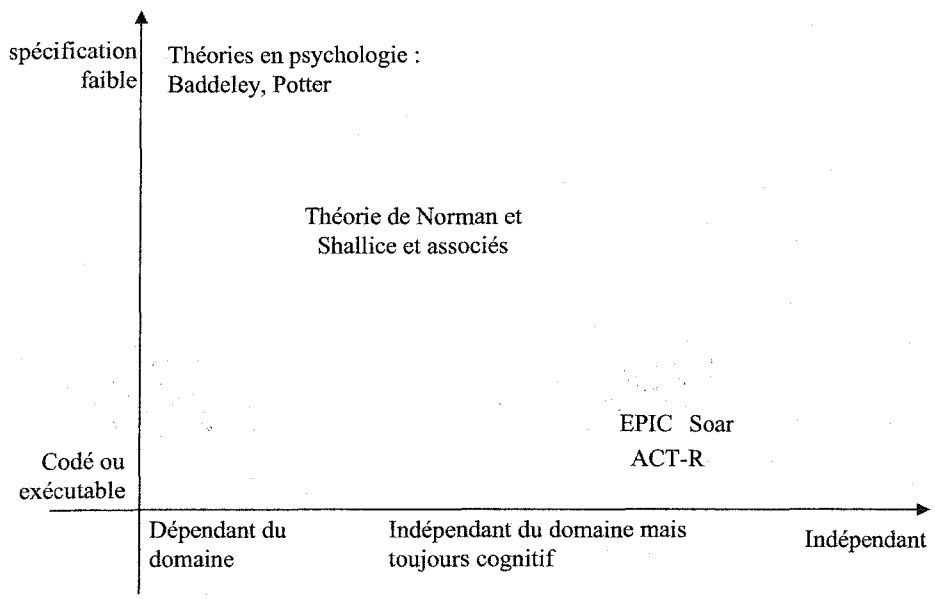


FIG. 2.1 – Situation de diverses théories et architectures cognitives dans la perspective de dépendance au domaine et de la possibilité d'implémentation (Inspirée de Grant (1995))

Pour bien comprendre l'organisation des deux chapitres suivants, nous nous référons à ces définitions. Dans la suite de ce chapitre nous allons présenter certaines théories de la cognition : la théorie de Baddeley (1986) et celle de Potter (1990) sur la mémoire de travail, la théorie de Norman et Shallice (1986) sur les processus de contrôle de l'activité et les tentatives d'implémentations qui en ont été faites par Cooper et Shallice (2000) et Glasspool et Cooper (2002). Dans le chapitre 3, nous allons présenter plusieurs architectures cognitives : SOAR de Newell (1987), EPIC de Meyer et Kieras (1997a) et

ACT-R d'Anderson (2002). Ces architectures ont chacune leur implémentation informatique. Nous verrons comment ces différentes théories et architectures vont pouvoir nous aider à modéliser une personne atteinte de troubles cognitifs.

2.3 La théorie de Baddeley (1986)

La théorie de Baddeley (1986, 1990, 1992) traite de la mémoire de travail. Cette théorie a été testée de nombreuses fois. L'idée est que la mémoire de travail serait formée d'une unité de gestion centrale et de deux modules esclaves qui permettent d'encoder l'information. L'unité de gestion centrale peut s'assimiler aux processus de l'attention. Les deux modules esclaves sont constitués de la boucle articulatoire et de la tablette visuo-spatiale. La boucle articulatoire est une mémoire active qui enregistre un nombre fixé d'items non visuels (chiffres, syllabes...) et cela de façon récursive avec une méthode FIFO (First In First Out, le plus vieil élément est remplacé par le nouveau).

L'information déposée dans cette mémoire s'apparente à la petite voix intérieure que l'on entend quand on lit ou quand on compose un numéro de téléphone. La tablette visuo-spatiale est l'équivalent de la boucle articulatoire pour les images mentales de stimuli visuels ou auditifs. Elle contiendrait trois emplacements.

La tâche verbale est une tâche prise en charge par la boucle articulatoire, et la tâche spatiale (par exemple tracer, ou écrire) est prise en charge par la tablette visuo-spatiale. Cependant, il doit s'agir d'informations bien définies : quand il s'agit de donner par exemple un chiffre aléatoirement, c'est l'unité de gestion centrale qui est mise en œuvre et non la boucle articulatoire, car l'information n'est pas définie avant d'être utilisée.

Les deux modules esclaves ont été l'objet de beaucoup de recherches, tandis que l'unité de gestion centrale a été un peu délaissée dans le passé. Maintenant, les chercheurs se penchent de plus en plus sur ce problème. En effet, les deux modules esclaves ne suffisent pas à expliquer la compréhension du langage naturel et le raisonnement. Ils ne font que reproduire l'information qu'ils contiennent.

Des expériences ont montrées que le module exécutif (ou unité de gestion centrale) a une faible capacité de manipulation des données et de raisonnement, ce qui amène des difficultés lors des inférences. Un sujet doit maintenir une représentation du problème dans sa mémoire et effectuer les opérations mentales nécessaires à sa résolution. Ce processus d'inférence est facilité quand on permet à la personne d'écrire les opérations successives. Sans possibilité d'écrire, le sujet aura tendance à oublier les prémisses du problème, même si celles-ci ont été correctement intégrées dans l'unité de gestion centrale (Gilhooly et al., 1993).

Le modèle de Baddeley est donc intéressant du point de vue des modules esclaves et de leur utilisation dans la mémoire de travail. Il pourra être intéressant de reprendre une approche similaire dans la conception de la mémoire de travail de notre architecture. Un autre point de vue possible est celui de Potter, théorie que nous abordons dans le point suivant.

2.4 La théorie de Potter (1990, 1993) : une théorie temporelle

Dans la théorie de Potter (1990, 1993), les modules se distinguent par le temps où ils maintiennent leur contenu accessible pour l'activité cognitive. Elle distingue ainsi quatre modules :

- Mémoire iconique (permet la perception initiale) : elle permet de reconnaître un paysage, une scène. Le temps d'attention nécessaire pour un individu normal pour reconnaître une scène ou une image est de 200 à 300 ms.
- Mémoire conceptuelle à très court terme : elle permet de reconnaître rapidement des scènes (100 à 200 ms) ou des mots (débit d'un mot par 100 ms). Les informations déposées dans cette mémoire disparaissent presque au fur et à mesure (moins d'une seconde). Cette mémoire permet entre autres l'activation de l'information correspondante dans la mémoire à long terme. Elle est sémantique : c'est-à-dire qu'elle retient le sens plutôt que la syntaxe ou la forme.
- Mémoire verbale à court terme : elle correspond à la boucle articulatoire du mo-

dèle de Baddeley. Elle permet par exemple la ré-interprétation d'une phrase mal comprise. Il s'agit d'une mémoire auditive.

- Mémoire à long terme : elle contient des informations sous la forme visuelle, conceptuelle et auditive. Elle permet aussi d'établir ou de retrouver des raisonnements effectués par le passé.

Ces deux théories nous montre que pour simuler la personne, il faut prendre en compte sa capacité de traitement des informations et son degré d'attention en fonction de sa maladie (données qui seront contenues et modélisées dans une des instances du modèle de la personne).

La quantité d'information qu'un individu peut traiter à un instant donné est limitée. Les psychologues ont introduit la notion de charge de traitement d'une tâche. On peut définir la charge de traitement d'une tâche par le nombre de dimensions à traiter simultanément pour effectuer cette tâche. Par exemple, pour répondre à «5 est plus grand que 3 ?», on aura besoin de deux dimensions, car pour répondre il faut considérer les deux arguments simultanément.

Toutefois, la capacité de ses dimensions est variable. On définit donc le chunk (notion introduite pour la première fois par Miller (1956) : unité cognitive indépendante d'information de grandeur arbitraire. Un chiffre, une lettre, un mot constituent des chunks pour la majorité des adultes, tandis que pour un enfant qui ne sait pas encore lire, un A est perçu comme l'association de trois segments et mobilise donc trois chunks. Pour une personne normale, la capacité de traitement en parallèle est de quatre chunks (Halford, 1993). On identifie souvent la notion de dimensions aux chunks. Il en résulte que les problèmes à quatre dimensions sont les plus complexes que peuvent traiter les humains.

Suite à cela, on voit déjà qu'il va être très important d'évaluer les capacités cognitives de l'occupant. Celui-ci n'aura vraisemblablement pas quatre chunks à sa disposition pour résoudre des problèmes ou effectuer une tâche. Le système devra être capable de simuler ses comportements compte tenu de ce type de modifications par rapport à un individu sain, et être capable de prédire dans quels cas ces manques pourront amener l'occupant

à une situation de handicap.

2.5 La théorie de Norman et Shallice (1980, 1986) : présentation du contexte

Dans cette section, nous allons introduire le principe général de la théorie de Norman et Shallice (1980, 1986), puis nous parlerons de certains concepts utilisés dans cette théorie, notamment des niveaux d'actions et une classification des erreurs cognitives. Ensuite, dans les deux sections suivantes, nous développerons des points de cette théorie.

2.5.1 Présentation générale de la théorie

Dans leur théorie, Norman et Shallice proposent une structure pour la cognition humaine lors de la réalisation d'activités. Cette structure est composée de deux modules. Le premier, Contention Scheduling (CS), permet de s'occuper des tâches habituelles ou routinières qui ne nécessitent pas de nouvel apprentissage, ni beaucoup de concentration. Le second module, Supervisory Attentionnal System (SAS), est le système de supervision. Il gère les situations non routinières qui exigent de la concentration de la part de la personne. Selon Norman & Shallice (1980, 1986), il y a cinq types de situations non routinières :

- situations qui impliquent de la planification, ou des choix non automatisés ;
- situations où des erreurs doivent être corrigées ;
- situations où les réponses ne sont pas connues ou qui contiennent de nouvelles séquences d'actions ;
- situations dangereuses ou techniquement difficiles ;
- situations qui requièrent la modification d'habitudes bien acquises ou de la résistance face à une tentation.

Le SAS intervient en activant ou en inhibant certaines actions du module de CS qui seraient normalement exécutées automatiquement sans intervention de l'attention. La structure du système SAS/CS[2] imaginé par Norman et Shallice, est de manière simplifiée représentée dans la figure 2.2 (page suivante) (Glasspool, 2000).

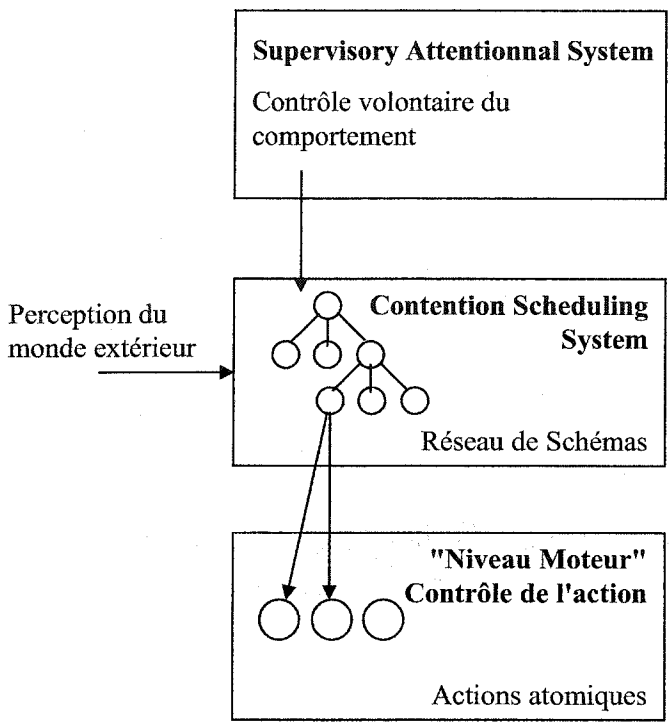


FIG. 2.2 – Structure générale du CS/SAS(Traduction libre) (Glasspool, 2000)

Le Contention Scheduling et le SAS seront l'objet des deux sections suivantes. Avant d'entrer dans le détail de leur fonctionnement, nous allons parler ici de quelques notions nécessaires à la bonne compréhension de la théorie.

[2]La couche «Système Moteur» est en réalité incluse dans le module de CS, comme on le verra par la suite. Il ne s'agit pas du système effecteur direct mais d'une version cognitive du système effecteur où les actions motrices sont représentées par des phrases (prendre, poser etc.) c'est-à-dire qu'il n'y a pas de bras effecteur, ni d'actions réelles.

2.5.2 Distinction de niveaux d'actions

Dans une action, on distingue des niveaux d'actions selon une échelle cognitive. Norman et Shallice propose trois niveaux distincts dont voici la description :

Niveau Bas Dans ce niveau, la compréhension de l'action se fait en termes biologiques des mouvements requis et de leurs propriétés physiques. Il s'agit de l'aspect sensori-moteur. Des schémas ont été développés par Schmidt (1975) à ce sujet (Motor Reponse Schema). Ce niveau d'action n'est pas sujet à un contrôle volontaire, il relève de la coordination des gestes et des influx nerveux nécessaires à la contraction des muscles.

Niveau Intermédiaire Dans ce niveau, la structure utilisée est celle des schémas, différente de celle introduite par Schmidt. La compréhension des activités ou actions se fait en termes de scripts : une succession de schémas décrit les actions élémentaires. Par exemple, l'action «se brosser les dents» peut être décrite comme : prendre sa brosse à dents, prendre du dentifrice, mettre du dentifrice sur la brosse à dents, poser le dentifrice, se laver les dents...

Conceptuellement, la différence essentielle entre ce niveau et le niveau bas, c'est que la sélection de la sous action est le problème critique, contrairement, au niveau bas, où le problème critique est de l'ordre de la coordination des mouvements. Dans le niveau intermédiaire, le processus de sélection de l'action est sujet à un contrôle volontaire de la part du sujet, tandis que la coordination est indépendante de la volonté du sujet.

Niveau Haut Dans ce niveau, la compréhension des actions se fait de façon abstraite en termes de comportements généraux. Il s'agit ici d'une approche très globale qui voit les actions dans leur ensemble. On ne se préoccupe plus ni du temps, ni de l'environnement. Ainsi dans l'activité «aller chez le médecin», on ne se préoccupe pas du tout du comment (i.e. à pieds, en voiture...) ni du temps (i.e. il faut une heure ...), on s'attache uniquement à la finalité de l'action.

Les principales caractéristiques de ces niveaux sont résumées dans le tableau de la figure 2.3 (page suivante). La théorie de Norman et Shallice porte sur le processus de

	Importance de l'environnement physique	**Contrôle du temps essentiel**	**Exécution parallèle des sous actions**	**Sous actions combinables avec d'autres activités en parallèle**	**Interruptions possibles entre les sous actions**	**Sous actions représentées comme des unités discrètes**	**Sélection des sous actions spécifiques critiques**	**Sujet aux erreurs**	**Sujet au contrôle volontaire**
Niveau Bas (ex : prendre)	Oui de façon très précise	Oui de façon très précise	Arrive souvent et nécessite de la précision	Non	Non	Non	Non	Non	Non
Niveau Intermédiaire (ex : se laver les dents)	Seulement de façon générale	Seulement de façon générale	Peut souvent être évité (ou alors les actions ne s'y prêtent pas)	Non pas facilement	Oui	Oui	Oui	Oui	Oui
Niveau Haut (ex : aller chez le docteur)	Non	Non	Non	Très Souvent Oui	Oui très souvent	Oui	Non pas nécessairement	Oui	Oui

FIG. 2.3 – Principales différences entre les trois niveaux d'actions (Traduction libre) (Cooper et Shallice, 2000)

sélection de l'action et sa supervision dans le niveau intermédiaire. Ce processus est sujet aux erreurs et le paragraphe suivant en présente une classification.

2.5.3 Classification des erreurs cognitives dans le niveau intermédiaire

Cooper et Shallice décrivent une classification des erreurs cognitives survenant dans le niveau intermédiaire lors du processus de sélection de l'action (Cooper et Shallice, 2000). Cette catégorisation des erreurs cognitives a été établie à la suite des travaux de Reason (1979, 1984, 1990) et des travaux de Norman (1981). Elle représente les erreurs qui sont faites par des gens sains dans le processus de sélection de l'action. Ces erreurs apparaissent également chez des personnes atteintes de troubles psychologiques mais de manière amplifiée voire caricaturée. Cette catégorisation n'est ni définitive ni exhaustive.

Erreur de Capture : une séquence d'actions imprévue (involontaire) est faite à la place d'une séquence d'action intentionnelle. (ex : aller dans la salle de bains se moucher, alors que l'on voulait se brosser les dents).

Omission : oubli d'une étape ou d'une sous tâche dans une séquence d'actions.

Anticipation : une sous tâche ou une étape est réalisée trop tôt dans la séquence (sans volonté intentionnelle de modifier l'ordre).

Persévération : répétition involontaire d'une étape ou d'une sous tâche (ex : faire déborder un verre d'eau, parce que vous êtes absorbés dans une discussion).

Substitution d'objets : substitution d'objets dans une sous tâche ou une étape volontaire.

Il est important que dans le modèle présenté, ces erreurs apparaissent, pour constituer un modèle valide du processus de sélection de l'action. En effet, pour des patients normaux, les erreurs viennent de la distraction. D'après les études de Reason, une erreur apparaît lorsque le niveau de distraction atteint 6 sur une échelle de 1 à 7. On voit, de cette manière, qu'il est nécessaire d'inclure au modèle général, l'attention de la personne par rapport à l'action qu'elle est en train d'effectuer. De même, l'étude de patients atteints neurologiquement montre que les erreurs qui surviennent dans l'organisation des actions

sont dues à un manque d'attention supérieure à la normale. Ces patients sont en général très facilement distraits de leurs occupations.

Maintenant que nous avons défini ces quelques concepts, nous allons décrire le fonctionnement du Contention Scheduling décrit en théorie par Norman et Shallice, puis implanté par Cooper et Shallice.

2.6 La théorie de Norman et Shallice (1980, 1986) : fonctionnement du modèle computationnel de Cooper et Shallice

Le modèle computationnel développé et implémenté par Cooper et Shallice (2000) cherche à simuler le processus de sélection de l'action (Contention Scheduling) qui se déroule dans le niveau intermédiaire de l'action. Dans la théorie initiale, Norman et Shallice sont partis de l'hypothèse que ce niveau pouvait être représenté sous forme de «schémas», représentation particulière de l'action, déclenchés par des seuils d'activation (*activation-trigger-schema framework*) (Norman et Shallice, 1986). Toutefois, la théorie n'est assez pas spécifiée au premier abord pour permettre une implémentation complète, elle a donc été sujette à quelques modifications afin de pouvoir être implémentée. Aussi, nous décrirons le fonctionnement du module de Contention Scheduling, tel qu'il a été implémenté par Cooper et Shallice (2000).

Tout d'abord, nous allons introduire de manière générale, le fonctionnement du modèle et ses diverses composantes. Puis, nous expliquerons le fonctionnement de chacune des composantes et leur mécanisme de mise à jour des valeurs d'activation.

2.6.1 Principe général

La composante principale du système est le réseau de schémas. Les nœuds de ce réseau sont des schémas, et ont chacun leur activation. L'interaction entre les nœuds de

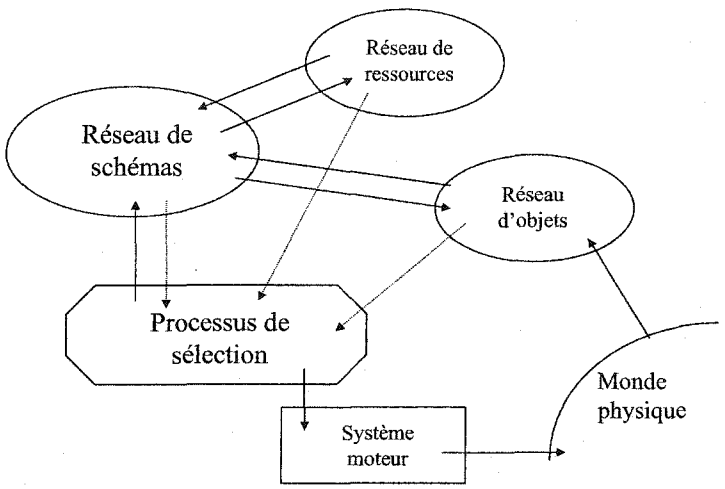


FIG. 2.4 – Les composantes du modèle de Cooper et Shallice (Traduction libre) (Cooper et Shallice, 2000)

ce réseau se fait à l'aide d'un mécanisme d'excitation/inhibition qui constitue une partie du mécanisme de mise à jour de l'activation. Sur la figure 2.4, apparaît un réseau de ressources qui fonctionne également avec un mécanisme d'activation, et qui représente les ressources disponibles pour la personne, qu'elles soient physiques (i.e. une main de libre...) ou cognitives. De la même manière, le réseau d'objets permet la représentation des objets du monde réel (i.e. une tasse, une cuillère etc...)

La composante «Processus de sélection» se situe entre le système moteur et les trois réseaux. Elle s'occupe de calculer les valeurs d'activation dans les différents réseaux, et d'effectuer les actions conséquentes à la sélection d'un schéma. La sélection d'un schéma a lieu quand l'activation de celui-ci dépasse un certain seuil fixé. Dans ce cas, il peut se produire deux choses : soit il y a répartition de l'activation de ce schéma dans le réseau, sur les composants du schéma activé, ce qui augmente la probabilité de leur sélection dans le futur (haut niveau dans le réseau) ; soit il y a action directe : affectation des objets et des ressources nécessaires, puis exécution effective de l'action grâce au système

moteur (bas niveau dans le réseau).

Pour la simulation, le processus est exécuté de façon cyclique et synchrone. A chaque cycle du processus, toutes les valeurs de l'activation sont mises à jour, et si c'est le cas, des actions directes sont effectuées. Toutes ces opérations sont effectuées en parallèle.

2.6.2 L'organisation des schémas et des buts : le réseau de schémas

Avant de décrire le fonctionnement du réseau de schémas, revenons sur le concept de *schéma*. L'unité fondamentale de description d'un comportement organisé du niveau intermédiaire d'action est le schéma. Celui-ci est une manière de représenter l'action et son but. Au sein du niveau intermédiaire, il peut à la fois être de bas niveau, c'est à dire simple comme : «Ouvrir », ou former une séquence de haut niveau représentant l'action «Préparer son café»(cf. figure d'un réseau de schéma 2.5 (page suivante)).

Chaque schéma est associé à un but qu'il permet d'accomplir. Formellement, un but est un état qui peut ou ne peut pas être satisfait par l'environnement. Le but est satisfait quand l'état qui le caractérise est atteint, c'est-à-dire que les conditions de l'état sont réalisées au sein de l'environnement courant. Les schémas sont donc des méthodes pour accéder à un but en modifiant l'environnement.

Un exemple d'association but/schéma :

But : sucrer une tasse de café

Un schéma : prendre une cuillère, la plonger dans le sucre, la remplir de sucre, amener la cuillère au dessus de la tasse, verser le sucre dans la tasse, remuer.

Le schéma proposé dans l'exemple, n'est pas le seul possible. Il existe un certain nombre de schémas qui permettent de réaliser le même but. Le choix du meilleur schéma dépend de plusieurs facteurs notamment la disponibilité des objets dans l'environnement (ex : absence ou non de petite cuillère) ou encore des préférences de la personne.

La relation entre les schémas et les buts n'avait pas été décrite dans la première

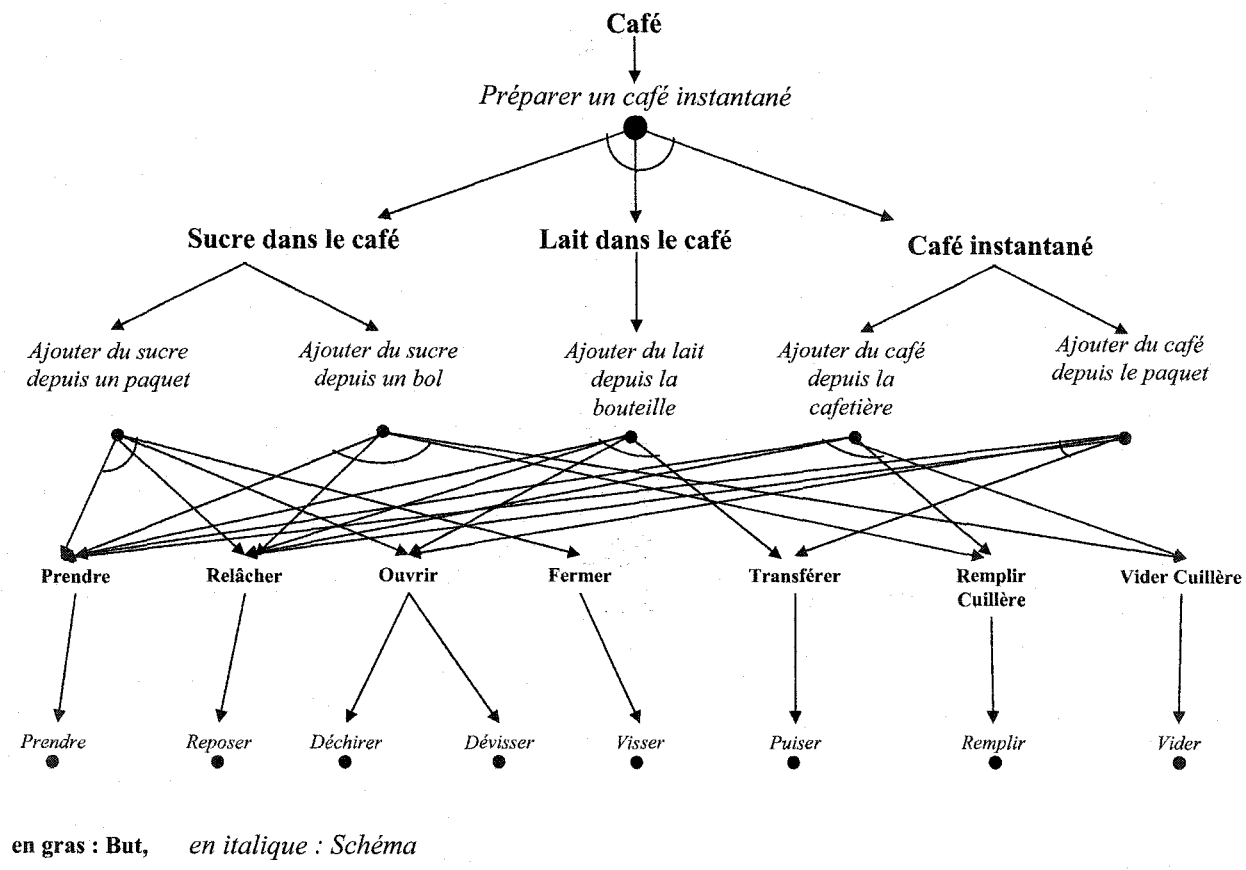


FIG. 2.5 – Exemple de réseau de schéma : la préparation du café(Traduction libre) (Cooper et Shallice 2000)

version de la théorie en 1986. Le module de Contention Scheduling n'était qu'un réseau de schémas. Par la suite, il est devenu évident qu'il était nécessaire d'intégrer la notion de but au sein de ce module. Les schémas deviennent alors des ensembles partiellement ordonnés de sous buts.

Ainsi notre exemple devient :

But : sucrer une tasse de café

Schéma du nom de «Sucre dans le café», composé des sous buts : avoir une cuillère dans la main, cuillère dans le pot de sucre, cuillère remplie de sucre, cuillère au dessus de la tasse, sucre dans la tasse, café remué.

A chacun de ses sous buts correspond alors le sous schéma cité dans l'exemple précédent. Les buts sont donc intégrés au réseau de schémas comme nous le montre la figure 2.5 (page précédente).

Pour résumer, un schéma est représentable par un arbre ou alternent des niveaux de buts et de schémas. Les fils d'un but sont reliés par la relation ou (un schéma ou autre doit être effectué pour réaliser le but), et les fils d'un schéma sont reliés par la relation et (tous les buts doivent être réalisés pour compléter le schéma). Maintenant que le concept de schéma est correctement défini, nous allons aborder le mécanisme d'activation au sein du réseau de schémas.

2.6.3 Le mécanisme d'activation du réseau de schémas

A chaque schéma est associée une valeur d'activation qui varie avec le temps. Si l'activation d'un schéma dépasse un certain seuil, alors ce schéma est sélectionné c'est-à-dire choisi comme comportement. Un schéma possède une activation de base fixe à laquelle est ajoutée une activation fluctuante : l'influence nette. Cette influence nette peut être une quantité négative, elle dépend des cinq facteurs suivants :

Influence verticale : activation qui provient du choix d'un schéma de plus haut niveau.

Influence de l'environnement : présence ou non des objets ou de ressources dans l'environnement.

Persistance : en l'absence d'une quelconque influence, l'activation d'un schéma a ten-

dance à persister, c'est-à-dire qu'elle diminue lentement jusqu'à arriver à son activation de base.

Influence horizontale : interaction entre les schémas de même niveau dans le réseau.

Bruit aléatoire : l'introduction d'un bruit est nécessaire si on veut ne sélectionner qu'un schéma à la fois et provoquer des variations dans le comportement.

L'influence nette sur l'activation est une somme pondérée des influences précédentes. Enfin, le système superviseur (SAS) peut lui aussi modifier les valeurs d'activation, c'est d'ailleurs un de ses moyens d'action sur le réseau de schémas c'est-à-dire sur le comportement.

L'influence verticale représente l'activation qui va être diffusée du haut vers le bas suite à la sélection d'un schéma de haut niveau. Au départ, la diffusion de cette activation est de la forme Activation de la source / Nombre de sous-buts. Mais, si on veut respecter les contraintes d'ordre, cette diffusion va être pondérée selon l'ordre des sous-buts. En effet, il y a trois types de contraintes d'ordre dans la réalisation de schémas :

- Contrainte d'environnement (pour prendre du sucre dans un pot, le pot doit être ouvert) ;
- Contrainte logique (pour sucrer le café, il faut d'abord prendre le sucre avant de ranger le paquet) ;
- Contrainte due à des préférences (certains préfèrent manger leur tartine avant de boire leur café).

La contrainte physique est réalisée grâce à l'interaction de l'environnement avec le réseau de schémas, via le calcul de l'influence de l'environnement. Les contraintes logiques et de préférences sont représentées par des préconditions sur les schémas. Ces préconditions influent sur la circulation du haut vers le bas de l'activation. Un sous-but dont la précondition ne serait pas réalisée empêchera la sélection de tout schéma permettant la réalisation de ce but.

L'environnement influence l'activation d'un schéma. Il est évident que si la situation courante ne se prête pas à un comportement, celui-ci ne sera pas sélectionné. Par exemple, pour pouvoir prendre un objet, celui-ci doit être présent dans l'environnement courant.

Donc, quand la situation est favorable à un schéma de bas niveau dans le réseau, l'influence de l'environnement sera positive. De plus, dans un mouvement du bas vers le haut, les schémas de plus haut niveau dans le réseau qui utilisent ce sous-schéma recevront eux aussi de l'activation (sous forme d'une influence).

Dans un mécanisme inverse, un schéma de bas niveau dans le réseau qui n'est pas réalisable dans l'environnement courant recevra une influence négative de l'environnement. Pour des schémas de bas niveau, le fait que l'environnement soit dans le bon état est la condition nécessaire au déroulement d'une action et à sa sélection[3].

L'influence horizontale, quant à elle, constitue un mécanisme de résolution de conflits, c'est-à-dire qu'elle évite que deux schémas de même niveau dans le réseau soient sélectionnés au même moment. En effet, il y a compétition lorsque les schémas représentent deux façons de réaliser le même but ou lorsque les deux schémas partagent un même sous but. En effet, si deux schémas partagent le même sous but, ils risquent d'avoir besoin des mêmes ressources dans l'environnement d'où un conflit potentiel.

Pour régler ce problème de conflit, on utilise surtout l'influence latérale (horizontale) afin d'être sûr de ne sélectionner qu'un seul schéma à chaque fois. Le principe est simple : on calcule l'influence latérale d'un schéma sur les autres, en examinant la différence entre son activation actuelle et son activation de base. Si l'activation actuelle du schéma est inférieure à son activation de base, alors l'influence latérale sur les autres schémas sera positive. En d'autres termes, on favorise les autres schémas car le schéma est lui-même déjà très inhibé. Si l'activation actuelle du schéma est supérieure à son activation de base alors l'influence sur les autres schémas sera négative, c'est-à-dire qu'un schéma fortement excité va avoir tendance à inhiber les autres. Enfin, les schémas qui sont à leur activation de base n'influencent pas les autres. Par ce processus d'accentuation des différences, on évite la sélection simultanée de schémas de même niveau dans le réseau.

[3]Ici, les auteurs ne considèrent pas qu'un sous-but peut justement être commun à deux tâches sans que cela ne porte à conséquence. Par exemple, on peut sortir une bouteille de lait à la fois pour en boire et pour faire un gâteau.

Après le mécanisme de mise à jour de l'activation, le système va désélectionner le schéma en cours, premièrement si l'action se termine, ou deuxièmement, si un schéma concurrent a une meilleure activation ou troisièmement, si le schéma source change (interruption d'une suite d'actions).

Lorsqu'un schéma est activé, on crée une liste de ses sous-buts et on l'ajoute à la pile (on suppose ici qu'il s'agit d'une pile, les auteurs ne donnent aucune indication à ce sujet). On cherche à compléter les sous-buts de cette liste (à l'aide d'autres schémas) et lorsqu'un des sous-buts est réalisé, il est alors simplement «retiré de la pile». De même, si on désélectionne un schéma, la liste de ses sous buts est retirée de la pile. De cette manière, on voit comment interviennent les buts et l'ordre des buts (réalisé avec les préconditions), dans le mécanisme de sélection de l'action.

Nous avons expliqué ici le mécanisme de l'activation du réseau de schémas. Cooper et Shallice (2000) donnent notamment en annexe de leur texte, le détail des formules utilisées. Nous allons maintenant expliquer le fonctionnement des autres composantes et les paramètres associés au système de Contention Scheduling.

2.6.4 Le réseau d'objets et le réseau de ressources.

Le réseau d'objets est lui aussi doté d'un mécanisme d'activation. Les nœuds du réseau sont constitués par les objets et les valeurs d'activation associées à chaque objet. En effet, chaque objet possède des valeurs d'activation selon les rôles qu'il peut remplir. Pour la tâche de préparation du café, on distingue les trois rôles suivants : source, cible, moyen (par exemple un contenant).

L'attribution d'une activation à chaque objet permet de calculer l'influence de l'environnement sur le réseau de schémas. Un objet bien en évidence dans l'environnement (i.e. avec une forte activation) aura tendance à être utilisé plutôt qu'un autre caché (i.e. avec une faible activation).

Contrairement au réseau de schémas, il n'y a pas de calcul d'influence verticale. Tous les objets sont au même niveau. Par contre, il y a un mécanisme de résolution des conflits

sous la forme d'un calcul d'influence latérale selon les rôles. Il ne faut pas qu'il y ait compétition entre deux objets qui n'ont pas le même rôle, si par exemple l'un est source dans l'action et l'autre cible. Par contre, il faut pouvoir choisir entre deux objets s'ils ont le même rôle, s'ils sont tous les deux sources d'une action par exemple. Le mécanisme d'influence latérale est donc fonction du rôle des objets. Le calcul de cette influence obéit aux mêmes principes que celui du réseau de schémas.

D'autre part, de la même manière que le réseau d'objets va influencer le réseau de schémas, le réseau de schémas influence le réseau d'objets. Si un schéma de bas niveau dans le réseau a une valeur d'activation très forte, il va avoir tendance à influer sur les objets dont il a besoin et augmenter leur valeur d'activation selon leur rôle. Inversement, un schéma fortement inhibé va inhiber les objets qui lui sont associés (toujours selon leur rôle).

Enfin, la persistance est également définie pour les objets. En résumé, le calcul de l'influence sur les valeurs d'activation des objets se fait selon leur rôle et dépend de quatre facteurs : l'influence horizontale, l'influence des schémas, la persistance et un bruit aléatoire. Il faut noter que dans la simulation faite par Cooper et Shallice, la création de nouveaux objets n'est pas prévu et l'environnement reste donc fixe.

De la même manière, le réseau de ressources représente les ressources disponibles. Chaque ressource possède une valeur d'activation. Il n'y pas ici de distinction de rôles. Le calcul de l'influence dépend des mêmes paramètres : l'influence horizontale, l'influence du réseau de schémas, la persistance et le bruit.

Toutefois, contrairement aux objets, les ressources peuvent être allouées à des schémas de haut niveau dans le réseau. On peut, par exemple, décider à un niveau assez élevé dans le réseau de mettre du sucre dans le café avec la main droite. Par conséquent, le calcul de l'influence du réseau de schémas sur le réseau de ressources ne dépend pas uniquement des schémas de bas niveaux dans le réseau mais aussi de ceux de plus hauts niveaux.

2.6.5 Les paramètres du système

Dans cette partie, nous allons décrire les paramètres du système de contrôle de l'action CS, sur lesquels les auteurs vont jouer par la suite pour simuler différents troubles cognitifs. Au total, il y a sept paramètres pour le système.

Deux paramètres contrôlent l'aspect général de la dynamique des réseaux : l'activation de base, c'est-à-dire la limite de décroissance de l'activation, et le degré de persistance qui contrôle la rapidité de ce retour à l'activation de base en l'absence de toute autre influence. Un troisième paramètre gère l'aspect aléatoire du bruit. Ensuite, trois autres paramètres contrôlent l'importance des influences les unes par rapport aux autres. Ces paramètres sont sous forme de rapports :

- Persistance/Influence horizontale : ce paramètre est relié notamment au mécanisme de résolution de conflits ;
- Influence Interne/ Influence Externe : (influence interne = influence verticale pour le réseau de schémas, influence externe = influence de l'environnement) ;
- Un troisième paramètre contrôle la proportion entre, persistance et influence horizontale d'un coté, et, influence interne et externe de l'autre.

Enfin, il y a le seuil d'activation à fixer, c'est-à-dire la valeur qui détermine la sélection d'un schéma ou non.

Nous allons maintenant voir quels sont les résultats obtenus par Cooper et Shallice lors des simulations et de quelle manière ils ont modifié ces paramètres pour simuler des troubles cognitifs connus.

2.6.6 Les résultats obtenus par Cooper et Shallice

Cooper et Shallice ont lancé le module de sélection de l'action sur la tâche spécifique de la préparation du café, dont nous avons donné une partie du réseau de schéma en figure 2.5 page 42. Ils ont choisi cette activité pour sa complexité moyenne (les tâches ne sont ni trop simples ni trop compliquées), pour la variété des sous actions (prendre, poser, visser...), pour la variété des objets, et pour le fait très important qu'une étude

a été réalisée sur cette activité auprès de malades par Schwartz et al. (1991). Il est donc possible de comparer les résultats obtenus avec cette simulation aux résultats médicaux. Les simulations lancées contenaient treize objets positionnés dans une grille et vingt-deux schémas d'actions, une partie étant des schémas de distraction (par exemple : manger le contenu de la cuillère).

Dans le mode normal de fonctionnement, le modèle est capable de reproduire une séquence valide pour la préparation d'un café instantané sans aucune difficulté et sans aucune erreur.

Par la suite, les auteurs introduisent des perturbations via les paramètres du système pour simuler les erreurs cognitives classiques dont nous avons présenté la classification précédemment. Ils ont notamment simulé des erreurs de capture en augmentant le degré de persistance de façon artificielle, ce qui empêche le mécanisme de résolution de conflits de fonctionner normalement. De ce fait, des échanges de séquences d'actions de haut niveau sont possibles.

En ce qui concerne les erreurs d'omission et d'anticipation, il faut au contraire que le degré de persistance soit bas. Ainsi, des schémas qui auraient dû être déclenchés ne le sont pas, et des séquences d'actions sont omises ou simplement décalées dans le temps. Un effet similaire peut être obtenu en réduisant l'influence verticale et l'influence de l'environnement.

Pour les erreurs de persévération, le ratio Persistance/Influence horizontale est augmenté artificiellement, ce qui provoque une non désélection du schéma en cours et une erreur de persévération simple. Quand aux substitutions d'objets, elles apparaissent si le bruit est trop grand, ou encore si l'influence du réseau de schémas sur le réseau d'objets est trop faible. Ceci entraîne que le bon objet n'est pas sélectionné au bon moment, du fait du bruit perturbateur, ou du manque d'activation de l'objet.

De façon générale en perturbant le système, Cooper et Shallice parviennent à obtenir les erreurs classiques dont nous avons parlé pour le niveau intermédiaire de l'action, pour

une personne saine. Toutefois, il semble que leurs perturbations s'effectuent dans des sens opposés pour des types d'erreurs différents : degré de persistance bas pour les omissions et anticipations et au contraire haut pour les erreurs de captures. Nous verrons en quoi ceci peut poser problème.

Par la suite, les auteurs ont cherché à simuler le syndrome de désorganisation de l'action de patients atteints neurologiquement. Pour cela, ils utilisent deux méthodes : l'addition de bruits et/ou la suppression de liens ou d'unités au sein du réseau de schémas. Pour tester leurs résultats, ils ont utilisé diverses méthodes d'évaluations. Les résultats obtenus au vu de ces tests sont encourageants dans le sens où ils simulent le comportement de personnes atteintes du syndrome de désorganisation. Toutefois, certains types d'erreurs sont totalement absents du résultat de ces simulations : il s'agit des erreurs de substitutions et des persévérations récurrentes, qui sont pourtant des erreurs caractéristiques de ce type de syndrome.

Les auteurs veulent tenter par la suite de simuler d'autres troubles neurologiques. Dans le futur, il est question notamment de simuler des troubles comme la bradykinésie, associée à la maladie de Parkinson et la répétition d'un stéréotype associée à certaines psychoses.

2.6.7 Validité du modèle présenté par Cooper et Shallice

Le module de Contention Scheduling et son implémentation présentés par Cooper et Shallice sont convaincants. Les résultats des simulations en mode normal sont encourageants : elles reproduisent le comportement humain. Toutefois, cela demande à être testé à une plus grande échelle. D'après les auteurs, le cadre de définition de l'activité est encore trop rigide pour qu'il puisse être étendu à des simulations de plus grande envergure.

D'autre part, on est obligé de perturber les paramètres du Contention Scheduling en sens contraire pour faire apparaître des erreurs cognitives distinctes. Ceci implique, si on veut pouvoir obtenir la simulation d'une personne atteinte de troubles cognitifs, l'existence d'un module qui modifierait à travers le temps les paramètres du CS. En effet, ce module permettrait à tous les types d'erreurs d'apparaître. Si on ne fait pas varier les

paramètres, il n'est pas possible de faire apparaître successivement des erreurs de persévération et d'omissions (car il faudrait que le degré de persistance soit à la fois très haut et très bas). Or il n'est fait mention nulle part d'un tel module. Donc, de ce point de vue, tant qu'il faudra modifier les paramètres du système pour lui faire commettre des erreurs, le CS sera «trop parfait» pour réellement reproduire un comportement humain.

Si l'idée est de reproduire le comportement de personnes atteintes neurologiquement alors le CS remplit de nombreux objectifs au vu des résultats obtenus. Toutefois, les méthodes suggérées par les auteurs peuvent être sujettes à controverse. En effet, une addition trop grande de bruit reviendrait à sélectionner aléatoirement les schémas. Pourtant dans le comportement de personnes atteintes, on retrouve une certaine logique propre à la personne atteinte. Leur façon de simuler les erreurs est donc discutable.

Ceci montre que le module de Contention Scheduling n'en est qu'à ses débuts. Il s'agit de la première tentative d'implémentation à partir de la théorie de Norman et Shallice. Il reste de nombreux problèmes à résoudre, notamment les interactions de ce module avec le S.A.S qui, on va le voir, est peu spécifié.

2.7 La théorie de Norman et Shallice (1980, 1986) : le module de supervision : S.A.S

Le SAS (Supervisory Attentionnal System) intervient lors de situations non routinières comme nous l'avons expliqué dans l'introduction. Il exerce un contrôle sur le Contention Scheduling, soit en activant directement les actions de bas niveau dans le réseau de schémas du CS, soit en causant la sélection d'un schéma existant qui ne serait pas sélectionné autrement dans cette situation. D'autre part, pour gérer des situations nouvelles, il peut créer un schéma temporaire qui contrôle les schémas du module de CS, ou permet de fournir une procédure pouvant gérer de manière effective une nouvelle situation.

D'un point de vue théorique, le SAS n'est pas encore assez spécifié. Le problème vient de la difficulté d'obtenir des données sur le type de processus concernés, car ils se dé-

roulent à un très haut niveau cognitif. Plusieurs études de Shallice et Burgess (1993, 1996) tendent à montrer que le lobe frontal est le siège des activités cognitives de haut niveau, et que le S.A.S lui-même peut être décomposé en plusieurs sous processus se déroulant dans le lobe frontal.

Shallice et Burgess (1996) détaillent trois étapes importantes du S.A.S :

Étape 1 : la construction d'un nouveau schéma temporaire :

L'élément clé de la gestion d'une situation nouvelle est la création et la construction d'un nouveau schéma qui peut prendre la place du schéma source déclenché dans une situation routinière. Ce schéma sera capable de contrôler des schémas de bas niveau et/ou de fournir une procédure plausible pour atteindre les buts de la situation non routinière. Le nouveau schéma peut être un schéma qui existe déjà, qui ne serait pas déclenché autrement, ou être une adaptation d'un ou plusieurs schémas existants. Les buts générés pour répondre à la situation sont définis durant la phase d'orientation du problème. Cette phase est suivie de la génération d'un ou plusieurs schémas candidats à l'achèvement de ces buts.

Étape 2 : Implémentation du schéma temporaire :

Cette implémentation requiert l'activation de schémas existant dans le CS correspondant aux actions qui composent le schéma temporaire.

Étape 3 : La surveillance de l'exécution du schéma :

Il faut bien vérifier que le schéma correspond à la situation, et est efficace quant au but qu'il est censé atteindre.

Par la suite, Shallice et Burgess (1996) ont encore décomposé ces étapes en plusieurs processus en s'appuyant sur des expériences médicales réalisées auprès de personnes présentant des lésions du lobe frontal. Ils ont notamment utilisé pour cela les tests de Hayling (exercice de complétion de phrases) et le test de Brixton (test d'anticipation).

Pour bien suivre le cheminement du SAS, nous n'allons pas présenter les processus dans le même ordre que les auteurs, et il est conseillé de se référer souvent à la figure 2.6.

De plus, les explications suivantes sont le fruit de mon interprétation et des quelques éléments donnés par Shallice et Burgess (1996).

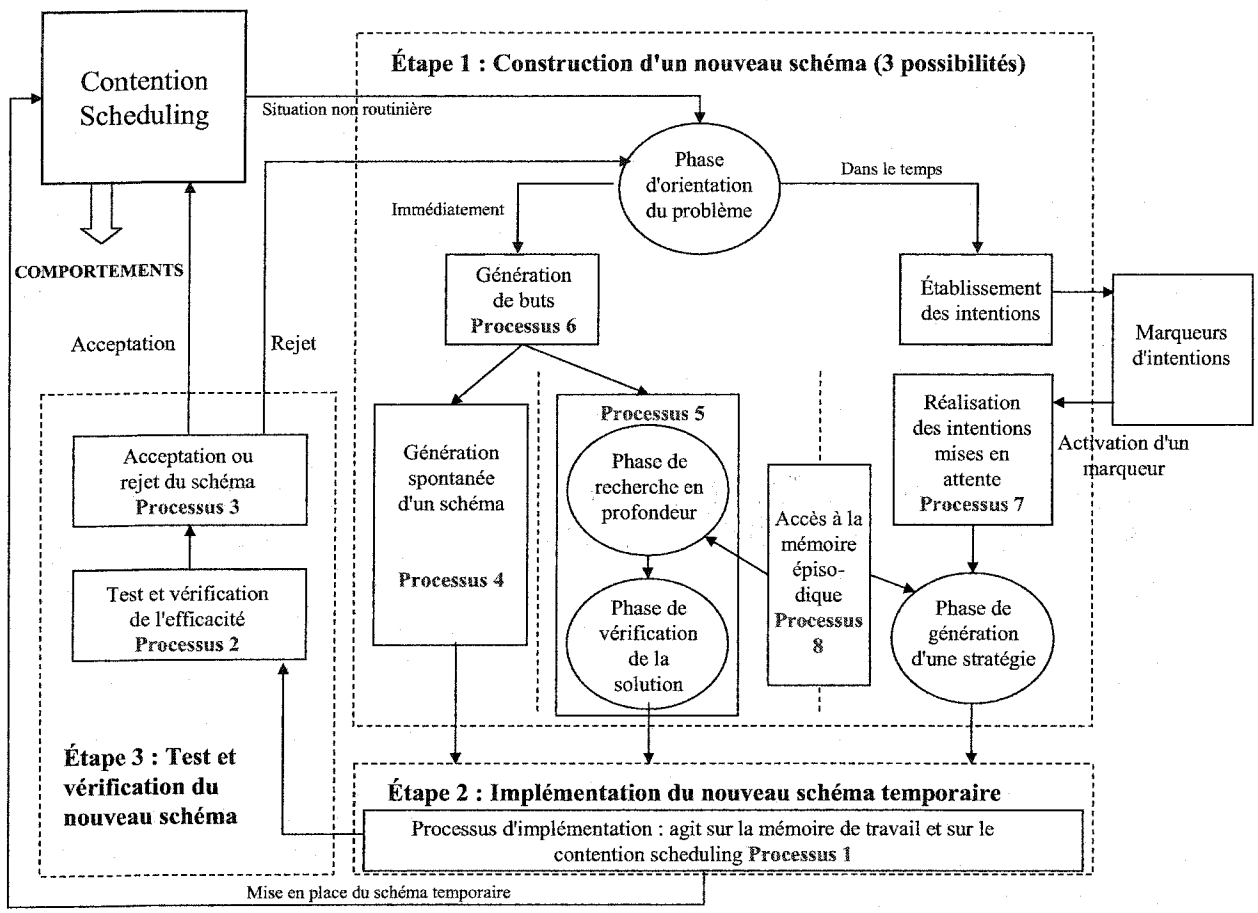


FIG. 2.6 – Schéma du CS/SAS Adaptation du schéma proposé par Shallice et Burgess (1996)

Tout d'abord, le module CS aboutit à une situation non routinière, il passe donc le relais au SAS, qui commence par une phase d'orientation du problème (début de l'étape 1). Cette phase consiste en quelque sorte à bien formuler le problème pour pouvoir le

résoudre (comme on poserait une division avant de la faire). Par la suite, l'étape 1 doit générer une solution au problème posé. Soit ceci est fait immédiatement ce qui déclenche le processus 6 de génération de buts, soit la résolution est décalée dans le temps, alors des «intentions» sont établies.

Dans le cas où la résolution doit être immédiate, le processus 6 de génération des buts s'occupe d'établir les buts à atteindre en fonction du problème posé (ex : faire la division maintenant qu'elle est posée). Ensuite, il faut générer un schéma (une procédure) répondant aux buts. Cette génération peut être spontanée (processus 4), c'est le cas quand la solution apparaît d'elle-même, et n'est pas le fruit d'un raisonnement. Sinon, cette génération peut être issue d'une stratégie de résolution de problème : c'est le processus 5. Dans ce processus, on peut distinguer deux phases qui sont propres à toutes les méthodes de résolution de problème : la phase de recherche en profondeur (solutions de plus en plus complexes) et la phase de vérification de la solution (Il s'agit de voir si elle est valide du point de vue du problème posé).

Dans le cas où la résolution du problème posé par la situation non routinière est décalée dans le temps, on crée ce qu'on appelle un marqueur d'intention. Lorsque celui-ci est activé, ce qui peut arriver beaucoup plus tard dans le temps, le processus 7 de réalisation des intentions mises en attente est activé, et donne lieu à la génération d'une stratégie pour répondre aux intentions.

Dans les deux cas précédents, Shallice et Burgess prévoient un accès à la mémoire épisodique pour faciliter la résolution de problèmes et la génération d'une stratégie. L'idée est de se baser sur des événements passés pour trouver une solution. Ils ne sont pas plus explicites à ce sujet et font uniquement intervenir le processus 8 dans leur schéma, de la même manière que sur la figure 2.6 (page précédente).

Dans les trois cas, on aboutit à la construction d'un schéma temporaire (plus ou moins complet) qui va passer par l'étape 2. Elle va implémenter temporairement le schéma dans le CS et/ou dans la mémoire de travail (Processus 1). Ensuite, dans l'étape 3, le schéma temporaire est testé (Processus 2). Il peut être gardé, altéré ou rejeté (Processus 3).

S'il est gardé, le CS reprend la main, et on passe à une exécution routinière à l'aide du nouveau schéma. S'il doit être altéré ou s'il est rejeté, un nouveau cycle du SAS commence.

Au niveau du détail de ses processus, Shallice et Burgess restent muets. Le sujet principal de leur article étant des considérations d'ordre neurologique sur le lobe frontal et le lieu de déroulement des différents processus. Nous n'avons ici qu'une esquisse de fonctionnement du SAS, qui est loin de permettre une implémentation directe. C'est un des points que nous tentons de clarifier dans la suite de ce mémoire.

Leur idée a ensuite été développée par Glasspool (2000), puis par Glasspool et Cooper (2002) qui en proposent une implémentation très simplifiée. Par une approche multidisciplinaire, Glasspool a remarqué une certaine similarité de formes entre le SAS décrit dans la théorie en neuropsychologie et une architecture d'agent intelligent pour la prise de décision, du nom de Domino, développée par Das et al. (1996). Domino est capable de prendre des décisions de façon argumentée, et de surveiller l'impact de ces décisions. En se référant à la figure 2.7 (page suivante), on peut voir qu'il y a une correspondance entre les deux, si on généralise le schéma du SAS de la figure 2.6 page 53, comme l'a fait Glasspool.

Glasspool (2000), puis Glasspool et Cooper (2002) ont réalisé une version simplifiée du CS/SAS à l'aide de l'environnement de modélisation COGENT réalisé par Cooper et al. (1998). Ils ont simulé grâce à ce modèle le Wisconsin Card Sorting Test (WCST) qui est un test psychiatrique pour évaluer le fonctionnement du lobe frontal. Il s'agit pour le patient de trier des cartes selon leur couleur, leur valeur, ou leur signe. L'expérimentateur dit à chaque fois au patient s'il a bien réussi à classer la carte, et peut au cours du test changer de critère sans prévenir le patient, qui doit alors se réadapter.

La simulation de ce test permet de tester les interactions simplifiées entre le SAS et le CS. En effet, lors de ce test, le SAS intervient d'abord pour déterminer quelle est la bonne stratégie de tri, puis il passe le relais au CS (l'activité devient routinière). Quand l'expérimentateur change de stratégie, le SAS doit de nouveau intervenir pour déterminer la nouvelle stratégie, puis il passe de nouveau le relais au CS.

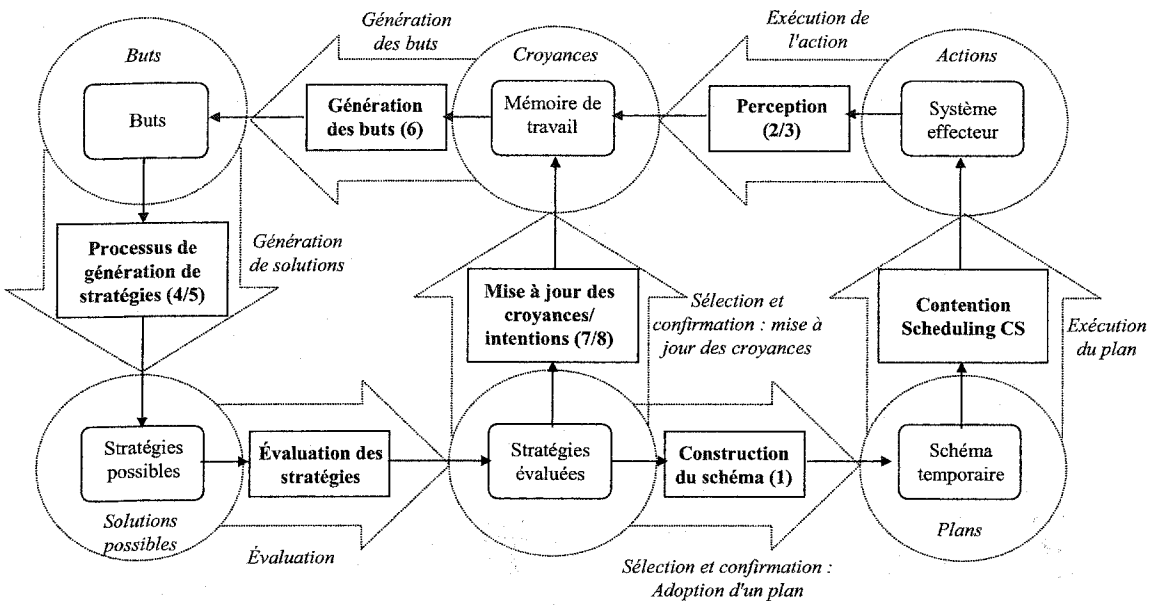


FIG. 2.7 – Correspondance entre le SAS et Domino proposé par Glasspool (2000) (Traduction libre) Les éléments en pointillés et italiques correspondent à l'architecture Domino et les lignes pleines et non italiques au SAS.

Toutefois, le SAS n'est pas encore assez spécifié pour permettre de réels tests avec le module de CS décrit par Cooper et Shallice (2000), et la version implémentée du SAS dans COGENT est beaucoup trop simplifiée. En effet, Glasspool et Cooper ont également implémenté le SAS pour un autre test, et ils ont dû modifier une grande partie de l'implémentation par rapport au premier test. Leur modèle est donc beaucoup trop dépendant de la tâche modélisée.

D'autre part, dans la figure 2.7 (page précédente), on peut voir que la mémoire épisodique a été supprimée (en comparaison avec la figure 2.6 page 53), alors que cet élément semble au contraire très important pour générer des solutions faces à des situations inhabituelles. L'être humain arrive à gérer des situations nouvelles grâce à son expérience qui constitue le contenu de la mémoire épisodique. Aussi, il semble essentiel que la mémoire épisodique ou un équivalent soit utilisé pour générer des solutions aux problèmes rencontrés.

Le système de supervision et de contrôle SAS reste peu spécifié. Nous ne disposons que d'un squelette du système et les implémentations tentées jusqu'à maintenant sont trop simples pour fournir une réelle spécification. Le travail que nous aurons à faire sera de spécifier le SAS si nous voulons l'utiliser, et de proposer des solutions nouvelles pour les différentes étapes du processus de contrôle.

Dans ce chapitre, nous avons présenté la théorie de Baddeley et celle de Potter qui nous ont permis d'introduire un certain nombre de concepts psychologiques. Puis, nous nous sommes intéressés à la théorie de Norman et Shallice. Le module de sélection de l'action CS est plutôt bien spécifié, et a été implémenté avec un certain succès. Toutefois, cette implémentation présente des limites, et il n'est pas évident que l'on puisse s'en servir pour modéliser une personne atteinte de troubles cognitifs. D'autre part, les spécifications du module de supervision SAS sont grandement insuffisantes pour être utilisées de manière brute. Nous verrons dans la partie concernant la conception, comment nous nous accommodons de ces diverses constatations. Dans le chapitre suivant, nous allons présenter plusieurs architectures cognitives.

Chapitre 3

Architectures cognitives

Comme nous l'avons dit une architecture cognitive est un ensemble de propositions relativement complet, à propos de la structure de la cognition humaine selon la définition d'Anderson. Ici les trois architectures cognitives qui sont présentées correspondent plus à cette définition qu'à celle de Simon Grant, qui ne voit dans une architecture qu'un outil de modélisation. SOAR, EPIC et ACT-R contiennent toutes trois d'importantes bases théoriques sur la structure et le fonctionnement de la cognition. Mais elles sont également des outils de modélisation de tâches cognitives.

3.1 Soar : une architecture cognitive de Newell, Laird et Rosenbloom

Soar est une architecture cognitive dédiée à la résolution de problèmes. Elle a été élaborée principalement par Laird et al. (1987) et Newell (1990). Dans un domaine donné, des analyses ont montré qu'on pouvait toujours se ramener à la résolution d'un ensemble générique de sous problèmes.

Les connaissances décrites sous forme de règles de production sont liées à un espace de problème. Un espace de problème décrit le but à atteindre, les états possibles, les actions pour passer d'un état à un autre. De cette manière, la réalisation d'une tâche cognitive est représentée comme le passage d'un état initial à un état final à l'aide d'une

suite d'opérateurs.

L'architecture de Soar est composée de deux modules pour encoder et manipuler les connaissances :

- la mémoire de travail : définit l'état actuel de Soar ;
- la mémoire à long terme : l'ensemble des règles de production.

La mémoire de travail est composée de 3 éléments :

Une pile de contexte : un contexte décrit le problème à résoudre à un moment donné. Le contexte est composé de 4 cases (but, espace problème, état et opérateurs) qui peuvent, soit contenir un objet, soit être notées comme «indécis».

Un ensemble d'augmentations : une augmentation est une structure déclarative qui permet de décrire tous les objets qui interviennent dans la tâche cognitive sous la forme (nom de classe, identificateur, attribut, valeurs). En fait, cela permet de définir par récurrence toutes les structures de données, et de décrire les composantes du contexte courant.

Un ensemble de préférences : une préférence est une structure interprétable par l'architecture. Elle encode une affirmation sur la valeur heuristique d'un objet lors des différentes étapes de la résolution du problème. Par exemple, une préférence permet de dire si un objet est meilleur qu'un autre.

La mémoire à long terme est un ensemble de règles de production. Il s'agit en fait d'une représentation des connaissances sous la forme : If Conditions Then Actions. L'information est contenue dans la partie Action de la règle. Ces informations sont par la suite rappelées au niveau de la mémoire de travail lorsque les conditions requises y sont présentes. Dans Soar, les règles de production ne peuvent qu'ajouter des augmentations ou des préférences à la mémoire de travail.

Le système des règles de production permet de changer les «indécis» en objets. Pour cela, Soar passe par deux phases :

Phase d'élaboration : les productions ajoutent des préférences et de nouveaux attributs aux objets existants, et ceci de manière cyclique.

Phase de décision : Analyse des préférences et de la pile des contextes en partant du plus vieux. Cette phase se termine par la substitution dans l'une des cases d'un contexte ou débouche sur une impasse.

S'il y a substitution, tous les objets présents dans la mémoire de travail liés à des contextes ultérieurs sont éliminés. S'il y a une impasse, un nouveau contexte est élaboré de façon à la contourner. Il y a 4 types d'impasses possibles :

- l'égalité de deux objets du point de vue des préférences ;
- lorsqu'il y a conflit parmi les objets possibles (toutes les règles de productions sont exécutées, on peut obtenir des contradictions) ;
- l'absence de changement ;
- le rejet : toutes les valeurs sont écartées, il n'y a pas de candidat possible.

On peut noter que contrairement à d'autres architectures, Soar ne possède pas de mécanisme de résolution de conflits. Lorsque la mémoire de travail permet de satisfaire les conditions de plusieurs productions, celles-ci sont toutes réalisées. Les impasses sont résolues par des stratégies spécifiques à la tâche cognitive. Soar peut aussi trouver une solution en examinant les choix adoptés dans une impasse similaire rencontrée par le passé. Enfin, lorsqu'un but est atteint, tous les sous buts associés sont éliminés de la pile des contextes.

Les capacités d'apprentissage de Soar se résument au «Chunking». Le chunking consiste à apprendre les caractéristiques de l'impasse, sous la forme d'une règle de production plus concise afin qu'elle puisse être exécutée dans des situations similaires à celles qui ont mené à l'impasse. Cela permet de transformer les impasses rencontrées en routines. Soar offre trois modes de fonctionnement :

- pas de chunking : donc aucun apprentissage, Soar refera les mêmes erreurs à chaque fois ;
- un chunk est créé à chaque but atteint : donc dès qu'il y a une erreur, la résolution est immédiate ;
- un chunk est créé uniquement lorsque le but est terminal : ce mode simule un apprentissage similaire à l'apprentissage humain.

D'un point de vue psychologique, tous les aspects de la cognition sont présents dans cette

architecture à un degré plus ou moins élevé, et s'ils ne sont pas présents, il est possible de les réaliser. Les connaissances procédurales prennent la forme des opérateurs. Par l'apprentissage, le système peut également augmenter ses connaissances procédurales. Les connaissances épisodiques sont représentées par la connaissance des événements antérieurs. Elles sont encodées dans la mémoire sous forme de règles de production dont les conditions dépendent du contexte de l'événement et dont l'action représente l'événement en lui-même. Les connaissances déclaratives, quant à elles, sont exprimées dans le langage des prédicats et retraduites pour Soar en terme d'augmentations, ou en terme d'action au sein d'une règle de production.

Toutefois, les fondements psychologiques de Soar sont faibles. Les principales critiques formulées à ce sujet sont la difficulté de manipulation des connaissances épisodiques (qui n'est pas du tout évidente dans la pratique), et l'accès sans restriction à la mémoire à long terme (i.e. il n'est pas possible par exemple de simuler des oublis). Pourtant, Soar a donné lieu à de nombreuses applications intéressantes, et il s'agit d'une architecture stable de la cognition humaine. Il convenait donc d'en faire mention.

3.2 EPIC : Executive Process Interactive Control

EPIC est une architecture cognitive qui a été développée par Meyer et Kieras (1997a). EPIC est l'acronyme de Executive Process/Interactive Control. La théorie de cette architecture est basée en partie sur celle de Newell (1990) et sur celle d'Anderson (1983), dont nous allons parler dans le point suivant. Le travail des concepteurs s'est concentré sur l'explication des processus mentaux du type exécutif, c'est-à-dire ceux qui interviennent lors de la réalisation d'une activité.

3.2.1 Buts d'EPIC

Le but des auteurs est de simuler le fonctionnement des processus exécutifs lors de la réalisation de plusieurs tâches en simultané. Il y a de nombreuses situations dans lesquelles une personne exécute plus d'une tâche à la fois (comme tourner le bouton de la radio en conduisant). Il est intéressant d'essayer de simuler ce genre de situation, car

il semble que plus le nombre de tâches augmente moins la personne est efficace. Dans cette architecture, l'accent est mis sur l'aspect sensori-moteur de l'architecture. Ils ont beaucoup travaillé sur les modules visuels, auditif et manuel. Nous allons commencer par présenter le fonctionnement général de l'architecture puis nous nous intéresserons au détail des différents processeurs impliqués dans le système.

3.2.2 Fonctionnement général

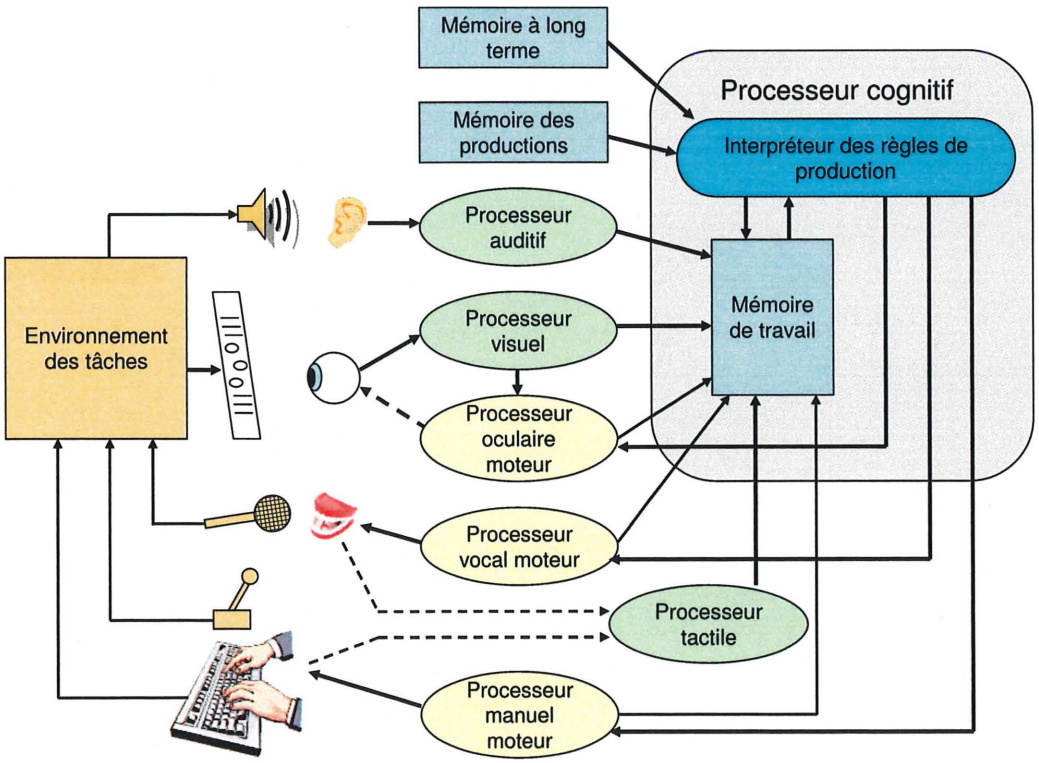


FIG. 3.1 – Structure générale d'EPIC (extrait de Meyer et Kieras (1997a))(Traduction Libre)

EPIC a un processeur cognitif central qui comprend un moteur d'inférence opérant

sur des règles de production et une mémoire de travail répartie. Il y a également des processeurs périphériques chargés de la perception de l'environnement et de l'exécution des actions ordonnées par le processeur cognitif central. Ces différents processeurs opèrent en parallèle.

Le système effectue des cycles de 50ms. Durant un cycle, il y a trois types d'opérations :

- la mémoire de travail est mise à jour avec le résultat des cycles précédents ;
- le moteur d'inférence sur les règles de production teste les conditions de toutes les règles de la mémoire procédurale ;
- les actions de toutes les règles dont les conditions sont satisfaites dans le contexte courant de la mémoire de travail sont exécutées.

Les cycles d'EPIC sont asynchrones par rapport aux stimuli extérieurs. Il n'y a pas de limite aux nombres de règles qui peuvent être exécutées à un moment donné. Il n'y a donc pas d'impasses ou de conflits dus à la nécessité de sélectionner une action. De cette façon, les processus cognitifs invoquant des ensembles de règles différents peuvent progresser simultanément et partager les ressources du système. Les actions résultantes de règles déclenchées en même temps sont exécutées en parallèle, tant qu'elles ne monopolisent pas des ressources identiques.

Les tâches sont modélisées en programmant le processeur cognitif avec des règles de productions, qui effectuent des actions selon le contenu de la mémoire de travail. Les règles de productions, stimulus, et les actions peuvent varier selon la tâche que l'on cherche à modéliser. Pour gérer la simultanéité des tâches, il faut également introduire des règles de supervision qui s'occupent de passer d'une tâche à l'autre (à la manière de la gestion des threads en informatique).

3.2.3 Fonctionnement du processeur cognitif : la mémoire de travail

La mémoire de travail est constituée d'items provenant de la mémoire visuelle (mémoire associée à la vue), d'éléments provenant du toucher, de l'ouïe. Elle contient aussi des items indiquant l'état des différents processeurs sensorimoteurs. Pour identifier les différents items de la mémoire de travail, les auteurs utilisent les tags : WM pour mémoire de travail, VISUAL pour la mémoire des processeurs visuels, AUDITORY pour le processeur auditif, TACTILE pour le processeur du toucher, MOTOR pour les différents processeurs moteurs. On peut ajouter ou supprimer des items lors de l'exécution du modèle.

La mémoire de travail est structurée de la manière suivante :

Des modules de stockage visuel, auditif et tactile, qui permettent de stocker les informations perçues au niveau des différents éléments périphériques. Ces modules sont réparties au sein des divers processus sensori-moteur (tags VISUAL, AUDITORY...).

Une pile d'étiquette qui contient les labels ou tags qui assignent un rôle spécifique aux éléments des modules de stockage.

Une pile de contrôle qui contient :

- les buts qui apparaissent dans les conditions des règles de production qui aident à achever une tâche particulière (but : tag GOAL) ;
- des étapes qui déclenchent certaines règles dans le respect d'une séquence définie, (étape : tag STEP) ;
- des informations sur la stratégie qui permet ou interdit à certaines règles de se déclencher (stratégie : tag STRATEGY) ;
- des informations sur l'état du système qui indique l'état de divers processus. (tag WM).

Cette pile de contrôle permet de faire fonctionner le moteur d'inférence correctement en définissant but, étapes et stratégies.

Les divers éléments dans la mémoire de travail peuvent se dégrader après un temps aléatoire, ainsi, à moins qu'on y fasse de nouveau appel, ils deviennent inutilisables. D'autre part, la capacité en termes de nombre d'éléments est illimitée. A priori, sans le phénomène de détérioration, la mémoire de travail d'EPIC serait infinie.

3.2.4 Fonctionnement du processeur cognitif : le moteur d'inférence

Le moteur d'inférence utilise la mémoire procédurale (production memory) qui contient toutes les règles de production nécessaires à l'exécution du modèle. Ces règles de productions prennent la forme IF conditions THEN actions. Les conditions comme les actions sont écrites sous la forme symbolique. Lors de chaque cycle d'EPIC, les conditions de chaque règle de production sont testées sur l'ensemble des items présents dans la mémoire de travail. Les actions des règles sont des ajouts ou suppressions d'éléments de la mémoire de travail, ou encore des actions via les différents effecteurs : yeux, voix et mains.

Le moteur d'inférence utilise aussi la mémoire à long terme, constituée d'items de grandeur variable. Ils sont distingués par le tag clé : LTM (pour Long Term Memory). Les auteurs ne détaillent pas la structure de ces items. Aucun item ne peut être ajouté ou enlevé pendant l'exécution du modèle. Il s'agit donc d'une mémoire statique. Son rôle reste peu évident, ils n'expliquent pas non plus le rôle joué par ses items lors de l'exécution d'un modèle.

Les règles de production peuvent appeler d'autres règles de production qui forment ainsi des buts et sous-buts. Une règle de production peut être interrompue parce qu'on lui aura précisé explicitement d'attendre pour un événement futur avant de continuer. Les tâches sont donc codées à l'aide de règles de production qui se déclenchent au bon moment. Les auteurs ont défini des protocoles particuliers pour signifier le début d'une tâche, sa fin, une erreur.

Lors de la modélisation de plusieurs tâches en simultané, il est nécessaire de définir des règles qui ont un rôle de superviseur, c'est-à-dire qui coordonnent et contrôlent l'exé-

cution des différentes tâches ou sous-tâches. Ces règles «superviseur» peuvent suspendre ou relancer une tâche en supprimant ou en ajoutant à la mémoire de travail le but associé à la tâche. Elles peuvent également établir des stratégies d'ordonnancement des tâches. On retrouve ici une façon de résoudre des conflits liés à l'utilisation des ressources et à l'ordonnancement des tâches. Ce sont les règles chargées de la supervision qui devront gérer la suspension et la reprise d'une action, si on doit, par exemple, attendre que l'environnement évolue pour pouvoir terminer une action.

Tout ceci va permettre de simuler l'exécution de tâches concourantes, en spécifiant d'une part les règles du moteur d'inférence directement reliées aux tâches, et d'autre part, les règles «superviseur» pour définir l'ordonnancement des tâches.

3.2.5 Les modules périphériques

Les systèmes présentés ici sont périphériques, car ils permettent d'interagir avec l'extérieur. Les sens reproduits dans EPIC, sont la vue, l'ouïe et le toucher. Il y a ainsi un module de vision, d'audition et de toucher. EPIC peut également via le système moteur, parler et/ou effectuer des mouvements (actionner des leviers, presser les touches d'un clavier, bouger les yeux). Dans EPIC, toutes ses capacités sont simulées, c'est-à-dire qu'elles ont lieu virtuellement.

Les modules perceptuels

Les modules de perception dépendent d'une forme basique d'attention. Certains capteurs physiques ou virtuels peuvent être orientés pour faciliter la perception. De plus, pour les informations issues des modules de perception, on associe un temps de transfert et d'identification du stimulus jusqu'à la mémoire de travail. Il est possible de mesurer ce temps pour observer l'impact du système de perception sur le fonctionnement de la mémoire de travail (Kieras et Meyer, 1998).

Les modules perceptuels : le processeur visuel

EPIC ne fait pas de reconnaissance de forme à proprement parler : un objet est représenté par un symbole arbitraire avec une liste de valeurs spécifiant ses attributs visuels.

Le processeur visuel est capable d'interpréter ces symboles de même que le processeur cognitif.

Deux espaces de vue coexistent : l'espace physique qui contient la représentation des objets dans l'espace (ex : coordonnées des touches d'un clavier), et l'espace psychologique qui contient uniquement les objets sur lesquels portent l'attention. Les propriétés des objets physiques et celles des représentations psychologiques sont liées mais peuvent différer.

L'environnement est responsable de la création des objets visuels physiques qu'il place dans l'espace physique. Le processeur visuel se décompose en deux sous processeurs : le processeur visuel sensoriel qui s'occupe de voir (représente l'oeil et la rétine) et le processeur visuel perceptuel qui s'occupe de percevoir (i.e. processus de reconnaissance). Lorsqu'un nouvel objet est placé dans l'espace physique, il est vu par le processeur visuel sensoriel, s'il est dans le champ de vision. L'apparition de cet objet constitue un évènement visuel. Une fois que le processeur visuel sensoriel a «vu», il transmet l'information au processeur visuel perceptuel. Ensuite, le processeur visuel perceptuel crée un objet psychologique correspondant à l'objet physique perçu et le transmet aux processeurs cognitif et oculaire. Le même processus se reproduit lorsqu'il s'agit de percevoir les propriétés visuelles de l'objet. Celles-ci sont envoyées au processeur visuel perceptuel et sont alors mises à jour dans l'espace psychologique.

Les modules perceptuels : les processeurs auditif et tactile

De la même manière que les processeurs visuels, il n'y a pas de réels mécanismes de reconnaissance auditive. Les entrées auditives sont encodées sous forme de couples (propriété, valeur) qui représentent les attributs des signaux auditifs. Le processeur auditif implémente un modèle de mémoire auditive simple : chaque item auditif est envoyé au processeur cognitif avec l'instruction d'être effacé après un certain délai.

Le processeur auditif gère deux sortes d'évènements : les évènements vocaux (phrases) et sonores (sons). Pour les événements sonores, le processeur auditif informe le processeur cognitif qu'il a perçu un son après un délai de reconnaissance. Dans un second temps, il

lui envoie la reconnaissance de ce son. Chaque son possède un attribut qui indique s'il est en train d'être émis ou pas.

En ce qui concerne les événements vocaux, les séquences de phrases sont représentées sous la forme d'un contenu sémantique. Ce contenu est envoyé au processeur cognitif après un certain délai. Une série de phrases ou de mots est représentée en un ensemble séparé d'items auditifs envoyés au processeur cognitif de façon chaînée.

Le processeur tactile de son coté, fonctionne de manière similaire, il ne fait que recoder l'entrée tactile sous forme d'un item tactile et l'envoie au processeur cognitif après un certain délai de reconnaissance.

Les modules moteurs

Un processeur moteur prend ses ordres du processeur cognitif et prend en charge les mouvements demandés. Il informe le processeur cognitif de son état à chaque cycle du système. Les différents processeurs moteurs sont le processeur oculaire, le processeur manuel et le processeur vocal.

Le processeur moteur fonctionne en trois phases distinctes qui peuvent se chevaucher afin de permettre à une série de mouvements de s'effectuer plus rapidement que si chacun des mouvements devait attendre que le précédent soit terminé.

Les trois phases sont :

1. phase de préparation : le processeur moteur crée la liste des éléments nécessaires à la réalisation du mouvement. La création de chaque élément demande un certain temps. Le temps total constitue le temps de préparation du mouvement ;
2. phase d'initiation : les auteurs ne détaillent pas cette phase ;
3. phase d'exécution où le mouvement est réalisé (mouvement mécanique effectif (virtuel) ex : bouger la langue pour parler).

Un processeur moteur possède une mémoire qui contient l'ensemble des données sur

les mouvements. Cet ensemble persiste à moins qu'il soit explicitement effacé par la commande RESET. Ces données peuvent être réutilisées au cours du mouvement suivant avec une commande particulière. Le type et la quantité de données pouvant être réutilisées dépendent des différents processeurs moteurs ou du type de mouvements.

Lors de la phase 1, si le processeur moteur reçoit une instruction du processeur cognitif, elle est ignorée et le processeur envoie un warning. Ceci correspond à une résolution de conflit du point de vue du partage des ressources. D'autre part, les éléments qui doivent être créés sont comparés avec ceux créés précédemment pour identifier ceux qui peuvent être réutilisés. Seuls les nouveaux éléments sont alors créés. Le temps de création de ces nouveaux éléments constituent alors le temps total de préparation. Ils sont ensuite stockés dans la mémoire du processeur pour la prochaine phase de préparation. On peut ainsi gagner du temps lors de tâches répétitives. Plus on répète un mouvement, plus la phase d'initialisation est courte (c'est sur ce principe qu'est basé le travail à la chaîne).

Pour donner un exemple de phase préparation, imaginons que l'on doive presser la touche d'un clavier sur ordre du processeur cognitif. Pour cela, on doit tout d'abord déterminer quelle main utiliser, ensuite quel doigt, puis la direction du mouvement et enfin où se trouve la touche. Pour les dactylos, la phase de préparation correspondant à ce mouvement est extrêmement courte et elles peuvent même se permettre d'ignorer la localisation de la touche. On parle d'une mémoire manuelle (un autre exemple est la différence entre un pianiste débutant et un pianiste confirmé).

Lorsque la phase de préparation est achevée, les éléments créés sont déplacés dans la pile des mouvements prêts à être exécutés. S'il n'y a aucun mouvement physique en cours, ces éléments sont déplacés dans la pile d'exécution proprement dite et la phase d'initiation est déclenchée, pour que le mouvement soit finalement exécuté. Si le processeur cognitif désire annuler un mouvement ou le remplacer par un autre, il ne peut le faire qu'après la phase de préparation et avant le déplacement des éléments nécessaires au mouvement dans la pile d'exécution.

Une fois qu'un mouvement a commencé à être exécuté, il ne peut plus être inter-

rompu. Lorsqu'il a fini d'être exécuté, on passe au mouvement suivant dans la pile des mouvements préparés.

Les processeurs moteurs : le processeur oculaire (mouvement des yeux)

Le processeur oculaire reçoit des instructions du processeur cognitif et des informations du processeur visuel perceptuel. Il possède deux mécanismes principaux : l'un pour le mouvement volontaire des yeux, toujours des balayage du champ de vision en réponse aux instructions du processeur cognitif; l'autre est responsable du mouvement involontaire des yeux en réponse à des évènements visuels en provenance du processeur visuel. Le système volontaire a priorité sur l'involontaire. Toutefois un mouvement des yeux ne peut être interrompu une fois qu'il a été enclenché. Ainsi, si un mouvement involontaire a débuté, on doit attendre son terme pour effectuer un mouvement volontaire.

Les auteurs arrivent ainsi à simuler le mouvement des yeux, toutefois ils n'ont pas pris en compte le fait que l'on pouvait regarder quelque chose sans le voir réellement, l'attention étant occupée par autre chose. Ceci veut dire que l'attention visuelle est dirigée mais permanente. C'est ensuite au niveau de la mémoire de travail que l'élément perçu est pris en compte ou non.

Les processeurs moteurs : le processeur manuel

Le processeur manuel contrôle les mains et les doigts virtuels du système. Ce processeur contient un certain nombre de mouvement prédéfinis. Ces mouvements peuvent être simples comme le mouvement «PUNCH» qui consiste à presser une touche ou un objet qui se trouve directement en dessous du doigt ou être plus complexes tel «PECK» qui en prend en compte une direction de frappe et l'étendue du mouvement.

De façon générale, les auteurs se sont concentrés sur la représentation de mouvements consistant à se servir d'un clavier ou d'une souris. Ainsi, ils ont des paramètres de position des mains et des doigts. La position par défaut des mains est du dessus d'un clavier. Ainsi les auteurs peuvent simuler l'exécution de tâches manuelles et plus particulièrement de tâches liées à l'utilisation d'un ordinateur (Kieras et Meyer, 1998).

Les processeurs moteurs : le processeur vocal

Le processeur vocal émet des phrases (ce n'est pas un synthétiseur vocal) en réponse aux commandes du processeur cognitif. A l'heure actuelle, c'est un mécanisme très simple avec des possibilités limitées car les modèles développés n'ont pas de grandes exigences au niveau vocal.

3.2.6 Conclusion

L'une des applications majeure d'EPIC est la validation d'interfaces homme-machine. Aussi puissant que puisse être un environnement informatique, si l'interface usager est mal conçue le système sera fortement limité par les lenteurs de l'opérateur humain. L'idée est de développer un modèle qui simule les interactions entre l'homme et la machine, permettant ainsi d'évaluer l'interface plus rapidement que les procédés classiques de validation d'interfaces (Kieras et Meyer, 1995).

La théorie présentée par les auteurs est complètement implémentée en LISP. Leur conception de la mémoire à long terme est en contradiction avec la plupart des théories en psychologie cognitive. En effet, celle-ci semble accessible en permanence, ce qui va l'encontre de toutes les constatations sur les phénomènes d'oublis. D'autre part, EPIC ne possède qu'un mécanisme très simple de détérioration des données pour la mémoire de travail, qui ne semble pas suffisant pour simuler de manière satisfaisante des oublis, ou des phénomènes tels que l'effet éventail qui sera abordé dans le point suivant. Enfin, il n'y a priori chez EPIC, aucune capacité d'apprentissage, à part le mécanisme de réutilisation des mouvements préparés, que l'on peut comparer à une mémoire des gestes.

Concernant les règles de production, EPIC exécute toutes les règles dont les conditions sont remplies (déroulement en parallèle). Comme nous l'avons expliqué en décrivant le moteur d'inférence, cette exécution en parallèle a rendu inutile la résolution de conflit au niveau des règles de productions. Mais de ce fait, il devient plus difficile pour EPIC de mettre en œuvre des erreurs cognitives comme les oublis ou les anticipations de tâches. Pour simuler de telles erreurs, il faut alors que ce soit les règles jouant le rôle de superviseur qui soient responsables de ces erreurs. On ne fait donc que déplacer le problème au

niveau des processus de supervision.

Une autre limitation d'EPIC est qu'il est nécessaire de préciser à chaque fois comment doit se dérouler le processus de contrôle. Il est spécifique au type de tâches et à l'enchaînement voulu de tâches. Il serait donc nécessaire d'ajouter un processus de supervision général à la manière du Supervisory Attentionnal System de Norman et Shallice. Cet aspect ne semble pas avoir été envisagé par les auteurs.

Ces défauts nous montrent qu'il serait difficile d'utiliser EPIC pour simuler une personne atteinte de troubles cognitifs. Toutefois l'originalité d'EPIC repose sur leur intérêt pour le système moteur et le système de perception. Il serait peut être intéressant de ce point de vue d'utiliser leurs différents processeurs périphériques. On peut d'ailleurs souligner que l'équipe d'Anderson pour ACT-R s'est inspiré des idées d'EPIC notamment pour les modules visuels et le mouvement des yeux.

3.3 ACT-R : l'architecture cognitive d'Anderson

ACT-R est le fruit de l'évolution de plusieurs théories en psychologie cognitive, notamment la théorie ACT acronyme de Adaptive Control of Thought. Nous allons en esquisser un bref historique. Le système ACT prend ses racines dans la théorie HAM sur la mémoire déclarative (Anderson et Bower, 1973). ACT a été créé en mariant la théorie HAM à une théorie du système de production de la mémoire procédurale proposée par Newell (1973a). Par la suite, la théorie ACT n'a pas cessé d'évoluer grâce aux travaux d'Anderson : ACT-E puis ACT* (Anderson, 1983), puis par la suite, ACT-R (Anderson, 1993).

ACT-R 2.0 est la première implémentation complète de ACT. Réalisée en Lisp, elle a été possible grâce à l'évolution des capacités de calculs des ordinateurs. De nombreuses versions ont vu le jour, notamment ACT-R 4.0 (Anderson et Lebiere, 1998) qui est une version nettoyée et améliorée incluant un module de perception et un système effecteur de manière optionnelle.

Enfin ACT-R 5.0, dernière version en date, est le reflet du développement de la théorie et des réponses de la communauté scientifique qui utilise largement ACT-R et ses possibilités depuis la version 2.0. En effet, plus de 100 modèles publiés sont basés sur l'architecture ACT-R 4.0 ce qui a permis de nombreuses améliorations. Une liste des applications est d'ailleurs disponible à l'adresse Web de ACT-R (http ://act-r.psy.cmu.edu). Nous allons décrire ici le fonctionnement de la version 5.0 d'ACT-R tel qu'il est présenté par Anderson (Anderson et al., 2002)).

3.3.1 Principes généraux d'ACT-R.

L'architecture ACT-R a été élaborée dans le but explicite de modéliser l'acquisition des habiletés cognitives de l'être humain. Ses composantes et sa dynamique sont appuyées sur de multiples recherches dans le domaine de la psychologie cognitive et de la neurologie. L'affirmation fondamentale à la source d'ACT-R est que «la cognition est la conséquence des interactions entre des unités spécifiques de connaissances procédurales et des unités spécifiques de connaissances déclaratives».

Les unités de connaissances déclaratives sont les chunks et représentent les connaissances sémantiques et épisodiques (Anderson ne fait pas de distinction entre les deux). Ces chunks peuvent de plus représenter des intentions et permettent donc de représenter un but. On retrouve ici le sens de chunk donné par Miller (1956) qui le définit comme une unité cognitive d'information de grandeur arbitraire.

Les unités de connaissances procédurales encodent les processus et les capacités nécessaires à la réalisation d'un but donné. Il s'agit de productions, c'est-à-dire des règles du type conditions-actions, qui se déclenchent quand les conditions sont satisfaites, et exécutent alors les actions spécifiées. Les conditions peuvent dépendre du but courant à réaliser, de l'état de la mémoire déclarative (c'est-à-dire des chunks dont le rappel est possible), et/ou des données sensorielles en provenance de l'environnement. Les actions, quant à elles, peuvent altérer l'état de la mémoire déclarative, changer les buts (courants ou non), ou initier des actions effectives sur l'environnement extérieur. La structure de ses règles est classiquement : If conditions, then actions.

La nature des interactions entre ces deux unités de connaissances est double. D'une part, elle a lieu à un niveau symbolique par la vérification des conditions des règles de productions, ce qui fait appel à des chunks de l'espace courant. D'autre part, nous verrons qu'elle a lieu également à un niveau sub-symbolique. Ce niveau sub-symbolique va régir l'évolution et l'accès aux différentes unités de connaissances.

La structure d'ACT-R est donnée dans la figure 3.2 (page suivante). Elle est décomposée en un ensemble de modules organisés autour du système procédural central. Le système de production central n'est sensible à l'activité de ces modules qu'au travers des tampons qui leur sont associés. En effet, une personne ne perçoit pas l'intégralité de sa mémoire déclarative à chaque instant, mais uniquement «ce qu'elle a en tête». Chacun des modules fournit l'information nécessaire au système de production, en la plaçant dans le tampon correspondant (*buffer*) sous forme d'un chunk. Cet ensemble de chunk constitue l'environnement courant du système central, sur lequel sont testées les conditions des règles de productions.

Autour du système central, Anderson distingue quatre modules. Le module d'intentions permet de garder la trace des intentions et des buts courants, le module déclaratif permet d'aller chercher les informations dans la mémoire déclarative, le module de perception permet de percevoir de façon visuelle ou auditive l'environnement simulé, et le système moteur de contrôler les mouvements du corps, plus particulièrement les mains.

Pour expliquer le fonctionnement de ACT-R plus en détail, nous allons séparer l'architecture en quatre parties : le système procédural central, le système associé à la mémoire déclarative, le système sensorimoteur (Perceptual-Motor System) et le système de buts. Le fonctionnement des différents modules est celui de la version 5.0 d'ACT-R.

3.3.2 La mémoire procédurale

Le système procédural central fonctionne par cycle. La durée d'un cycle est par hypothèse de 50 millisecondes. Chaque cycle s'effectue en trois phases :

- la phase d'association ;
- la phase de sélection ;

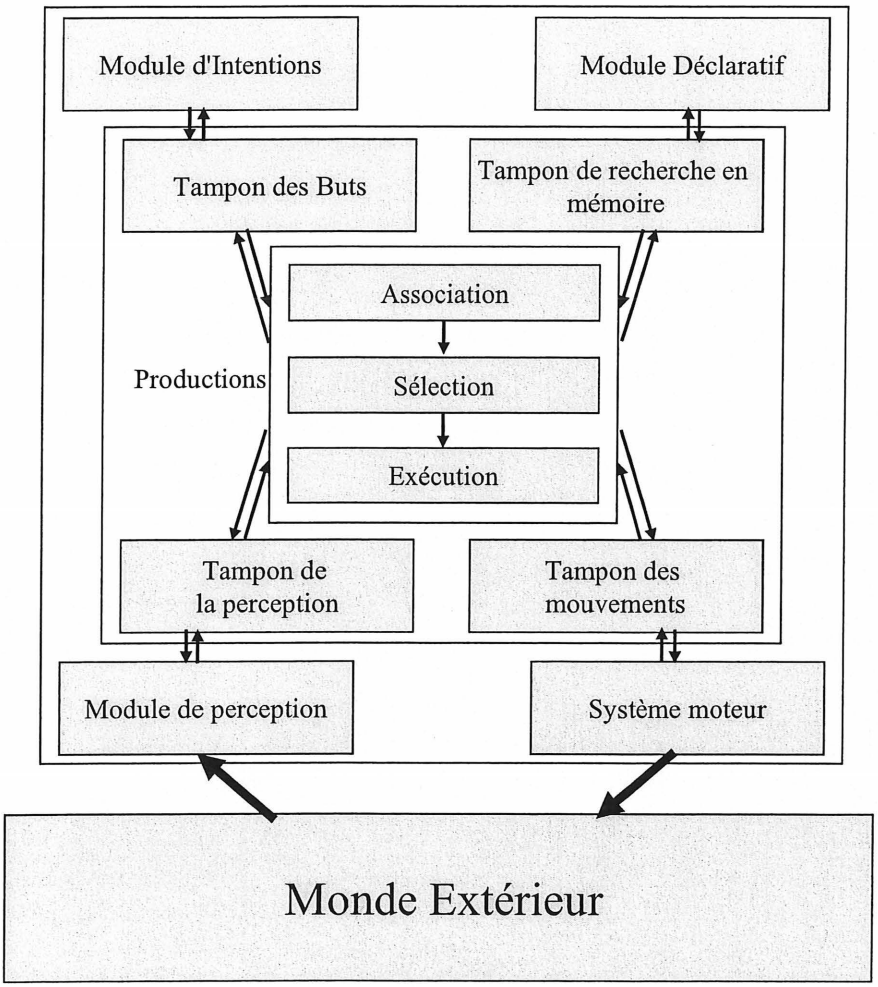


FIG. 3.2 – Architecture d'ACT-R (Traduction libre) (Anderson et al., 2002)

– la phase d'exécution.

Le processus d'association consiste à rassembler toutes les règles dont les conditions sont réalisées dans l'environnement courant. (Cet environnement courant est représenté par le contenu des différents tampons.) Il ne faut pas obligatoirement que les règles sélectionnées satisfassent toutes les conditions présentes dans l'environnement courant. Dès qu'une règle a toutes ses conditions vérifiées, elle est susceptible d'être déclenchée. Ceci signifie qu'on va avoir un ensemble de comportements possibles pour une situation, et il va falloir en choisir une seule.

Intervient alors la phase de sélection. Pour sélectionner la production qui va être exécutée, ACT-R contient un processus de résolution de conflits qui va choisir la meilleure production. Ce choix s'effectue à un niveau sub-symbolique. En effet, à chaque production est associée une valeur d'utilité qui évolue dans le temps. La production choisie sera celle qui aura la plus forte utilité. Cette façon de procéder n'est pas sans rappeler le mécanisme de sélection de l'action du Contention Scheduling de Cooper et Shallice (2000).

La valeur d'utilité d'une production cherche à rendre compte de l'utilité réelle d'une connaissance procédurale. Par exemple, cela permettrait de traduire le fait que pour ouvrir une boîte de conserve, il vaut mieux utiliser un ouvre-boîtes plutôt qu'un couteau. L'utilité de la production représentant la connaissance procédurale ouvrir une boîte avec un couteau, aura une utilité très faible voire nulle, par rapport à celle utilisant l'ouvre-boîtes.

Le calcul de l'utilité de chacune des productions est fait à chaque cycle du système. Elle est la différence entre :
- la probabilité P de réalisation du but à l'aide cette production ;
- le coût estimé C (en terme de temps) pour atteindre le but avec cette production.

Les valeurs de P et C varient pour chacune des productions et sont apprises avec l'expérience d'utilisation de cette production par le système. Plus précisément, P vaut le nombre de succès divisé par le nombre de tentatives (la somme des succès et des échecs).

La valeur de C est la valeur estimée du temps entre le déclenchement de la production et l'achèvement du but. Comme pour P, cette valeur dépend de l'expérience qu'aura le système de cette production. Pour donner un exemple concret, il est très difficile d'avoir une idée du temps que demande la résolution d'une équation lors de la première tentative. Mais avec de la pratique, il est possible d'estimer le temps nécessaire de plus en plus précisément. Celui-ci est en général traduit en termes de «facile» ou «difficile».

Ainsi une production qui a une faible probabilité de nous mener au but et un coût estimé élevé aura une utilité très faible. De façon opposée, si la probabilité est assez forte et le coût faible, l'utilité de la production sera élevée. Toutefois, comme ce n'est pas forcément l'approche la plus évidente qui permet de résoudre une solution, l'ajout de bruit dans le calcul de l'utilité permet que des productions de plus faibles utilités soient parfois sélectionnées. De cette manière, le système a un comportement indéterministe qui permet de reproduire plus fidèlement les choix fait par un être humain.

Enfin, la production sélectionnée est exécutée. L'exécution d'une règle se traduit par la modification du contenu d'un ou plusieurs des tampons. Ainsi, les actions d'une règle de productions peuvent par le biais des tampons :

- demander le rappel d'une connaissance en mémoire (tampon de la mémoire déclarative) ;
- modifier le but courant (tampon des intentions) ;
- demander de percevoir des éléments dans l'environnement (tampons de perceptions) ;
- effectuer une action (tampon du système moteur).

Une fois, les modifications des tampons effectuées, la phase d'association entre de nouveau en jeu. Dans les modules, les différents processus mis en jeu fonctionnent en parallèle et de manière asynchrone par rapport au système central. Ainsi, le système central n'attend pas le résultat d'un mouvement pour recommencer une phase d'association.

3.3.3 Le module de la mémoire déclarative

Le module déclaratif va contenir des unités de connaissances déclaratives stockées sous forme de chunks. Le module de la mémoire déclarative est ce qui correspond à notre mémoire des événements et des faits. Or, la capacité de rappel d'une information chez une personne dépend de nombreux paramètres, notamment du nombre d'utilisations de cette connaissance ou de la récence de la dernière utilisation. Pour rendre compte des phénomènes associés à la mémoire (oublis, effet bout-de-la-langue...), Anderson fait de nouveau intervenir le niveau sub-symbolique.

Chaque chunk de la mémoire déclarative est associé à une valeur d'activation qui varie avec le temps. Cette valeur d'activation contrôle à la fois sa probabilité de pouvoir être rappelée et son temps de rappel. Le calcul de l'activation est la somme de :

- son activation de base ;
- l'activation associative selon les sources ;
- l'activation due à une similarité avec d'autres chunks ;
- des bruits aléatoires.

L'activation de base d'un chunk dépend du nombre de fois où il a été utilisé et du temps écoulé depuis sa dernière utilisation. D'une façon générale, plus on utilise fréquemment une connaissance, plus on a tendance à s'en souvenir facilement. Cet effet est renforcé par la récence de la dernière utilisation. D'un autre coté, moins on utilise une connaissance et plus il y a de temps écoulé depuis la dernière utilisation, moins on va s'en souvenir.

Le calcul des associations du chunk avec les autres permet de rendre compte de plusieurs effets constatés en psychologie. Ce calcul fait référence au rôle du contexte dans le rappel d'une connaissance. Un exemple simple sont les associations du mot «pot» selon le contexte : si la conversation portait sur le petit-déjeuner, le pot va être associé à confiture, tandis que si la conversation portait sur les voitures, il va être associé à échappement.

D'autre part, le calcul des associations prend aussi en compte des effets tels que «l'effet éventail». Celui-ci montre que plus les sujets apprennent de faits relatifs à un concept,

plus le temps nécessaire pour récupérer un fait concernant ce concept augmente. L'étude de cet effet s'effectue en deux temps (Anderson, 1974) :

Phase 1. On demande au sujet d'étudier des phrases comme celles présentées ci-dessous :

Le médecin est dans la banque	*Le pompier est dans le parc*
L'avocat est dans l'église	*L'avocat est dans le parc*

On fait varier le nombre de personnes présentes dans un lieu (le parc est ainsi plus «fréquenté» que la banque) et l'on fait également varier le nombre de lieux visités par une personne (par ex., l'avocat visite plus de lieux que le médecin).

Phase 2 : Il s'agit de la phase de test. On présente des phrases au sujet qui doit décider si celles-ci ont été présentées durant la phase d'étude.

Résultats : Les résultats montrent que le temps de réaction augmente quand le nombre de faits associés aux personnes ou aux lieux augmente.

La formule de calcul de l'activation prend en compte ce genre d'effet et donne ainsi un fonctionnement très proche du fonctionnement de notre mémoire.

L'activation liée à la similarité permet quand à elle de prendre en compte un autre type de phénomène : la confusion entre certaines connaissances. Par exemple, chez des personnes saines, il arrive de confondre des noms propres dont la sonorité est proche. Un exemple, plus frappant chez les malades d'Alzheimer est la confusion entre de la farine et du sucre (même aspect granuleux blanc). Enfin, l'addition de bruit permet de faire varier les valeurs d'activations de façon aléatoire et ainsi d'introduire un facteur d'indéterminisme dans le système.

ACT-R considère qu'un chunk, c'est-à-dire une connaissance, a été oublié lorsque l'activation de celui-ci est en dessous d'un seuil fixé. Ce seuil est le seuil d'activation du système, il s'agit d'un paramètre réglable. Ainsi, si on place le seuil d'activation très bas, le système se rappellera de tout. Au contraire, si on le place très haut, il ne se souviendra

de rien.

La liaison avec le système procédural central s'effectue de la manière suivante : les règles de productions du système central peuvent utiliser des connaissances stockées dans la mémoire déclarative en y faisant directement référence. Toutefois, si l'activation des chunks concernés est insuffisante, c'est à dire en deçà du seuil d'activation, la production ne pourra pas être sélectionnée.

L'autre façon d'accéder aux connaissances de la mémoire déclarative est de demander explicitement le rappel d'une connaissance. Par le biais du tampon associé au module, le système central fait une requête de rappel. Si le chunk demandé a une activation suffisante, il est alors mis à disposition du système central dans le tampon. Le système de productions est ensuite libre de l'utiliser tant que l'activation de ce chunk ne chute pas en dessous du seuil.

Avec ce mécanisme de calcul de l'activation, le système peut simuler les différentes erreurs cognitives (oublis, confusions) et les effets de l'apprentissage (les connaissances sont d'autant plus activées, qu'elles sont répétées). La façon dont fonctionne le module de la mémoire déclarative d'Anderson est vraiment très intéressant et apparaît très complet. En effet, d'une part cette mémoire prend en compte la plupart des phénomènes psychologiques observés, mais en plus, elle modélise naturellement les deux aspects de la mémoire : explicite - implicite. La mémoire explicite réfère directement à un chunk de la mémoire déclarative. La mémoire implicite ne fait pas référence directement à un chunk mais met à disposition des connaissances associées. Par exemple, quand une personne dit «Je suis allé au cinéma», la mémoire implicite met à disposition des connaissances telles que «la construction d'une phrase est sujet verbe complément». Avec ces atouts, ACT-R apparaît d'ores et déjà comme l'architecture dont nous allons nous servir pour notre modèle de simulation de la personne, car elle semble beaucoup plus solide que le CS de Cooper et Shallice.

3.3.4 Fonctionnement du système sensorimoteur.

ACT-R a pour but originel de modéliser le fonctionnement de la cognition, aussi l'aspect sensorimoteur a-t-il été délaissé dans les premières versions de la théorie. Il apparaît maintenant de plus en plus nécessaire de faire intervenir le système sensorimoteur dans les processus de cognition, car il y est intimement lié. De nombreuses expériences cliniques le montrent. C'est pour cette raison que ce système a été beaucoup plus développé sur les deux dernières versions d'ACT-R et réellement considéré comme deux modules à part entière dans ACT-R 5.0.

Anderson et ses collègues se sont basés essentiellement sur le système proposé dans EPIC (Meyer et Kieras, 1997a) qui intègre des capteurs et de réels effecteurs. Ils l'ont en partie ré-implémenté. Le principe est de modéliser dans une forme approximative les comportements de base du système sensorimoteur, ainsi que leur durée. Comme il s'agit d'approximations, le système n'est pas toujours fiable, toutefois, il est amplement suffisant pour l'utilisation qui en est faite dans ACT-R.

Le système sensorimoteur est constitué du système de perception et du système moteur. Le système de perception comprend le système de vision de ACT-R, qui est séparé en deux modules, avec pour chacun un tampon associé. Il y un module dédié à la localisation des objets dans l'environnement, et un autre dédié à l'identification des objets. Il s'agit ici de modéliser ce qu'on appelle l'attention visuelle d'une personne et non de modéliser comment le processus de vision s'effectue. De plus, les deux types d'attention interviennent de manière équivalente : l'attention est à la fois basée sur la localisation de l'objet et sur son identification.

Quand une production fait une requête de localisation, elle spécifie une série de contraintes destinées à réduire le champ de recherche. Ses contraintes peuvent être des restrictions de l'aire de recherche ou des critères qualitatifs sur les propriétés de l'objet. Ensuite, le système renvoie un chunk représentant le résultat de la recherche dans le tampon de localisation.

Quand une production fait une requête d'identification sur un objet, elle doit four-

nir une localisation de l'objet, ce qui va déplacer l'attention du système sur cet objet et générer un chunk dans la mémoire déclarative représentant cette objet. Le système fournit deux niveaux de déplacement de l'attention. Un grossier où le déplacement prend un temps donné sans aucune considération par rapport à la distance entre l'objet et le précèdent point d'attention, et un plus fin qui modélise le mouvement des yeux.

Les auteurs ne précisent pas ensuite comment se déroule le processus d'identification de l'objet. Ils ne sont d'ailleurs pas plus précis sur les différentes fonctionnalités citées ci-dessus notamment en ce qui concerne la localisation de l'objet qui ne semble pas évidente à priori.

En plus, des tampons visuels, il existe un tampon auditif, par le biais duquel, le système central peut demander à percevoir des sons. En pratique, cet aspect a été développé pour implémenter des modèles impliquant le langage et sa reconnaissance. Ce tampon auditif fonctionne d'une manière similaire aux tampons visuels.

Enfin, ACT-R possède un système moteur qui lui permet d'effectuer des mouvements dans l'environnement simulé. Le système procédural central peut ainsi demander l'exécution d'un mouvement en plaçant un chunk contenant les bonnes informations dans le tampon associé au système moteur. Dans la pratique, les mouvements possibles sont plutôt limités et comme pour EPIC se concentrent sur des mouvements liés à l'utilisation d'une souris et d'un clavier.

Dans notre problématique, les modules de perceptions et le système moteur sont très importants puisqu'il est question de simuler une personne réalisant des activités quotidiennes. Il s'agit essentiellement de tâches manuelles comme les activités liées à la cuisine. Il serait donc intéressant de développer cet aspect pour le projet DOMUS.

3.3.5 Le module des buts

Le module des buts sert à créer des nouveaux chunks abstraits. Par exemple, imaginons qu'on ait demandé à un individu de calculer la somme 36+64. Pour faire cette somme, il utilise un processus complexe tel que poser le calcul des unités, puis le calcul

des dizaines. Il arrive alors au résultat «36+64=100». A ce moment là, le tampon des buts contient le fait que 36+64= 100. Le chunk correspondant, contenant cette information, va être par la suite stocké dans la mémoire déclarative, il constitue un nouveau chunk abstrait. Ce processus de création de chunks permet au système de retrouver les résultats et de lui éviter de les recalculer par un processus complexe. De cette façon, le système de buts sert à mettre en place une stratégie de résolution de problèmes efficace : il y a un apprentissage ou du moins une trace des solutions adoptées par le passé.

Un processus identique se déroule pour les autres tampons du système : n'importe quel chunk formé dans l'un des tampons (hors celui de la mémoire déclarative) est stocké dans la mémoire à long terme. Toutefois, il ne s'agit pas de chunks abstraits, car ils ne sont pas les résultats de la résolution d'un problème, c'est-à-dire qu'ils ne sont pas d'une plus grande abstraction. Les chunks du tampon de perception vont concerner des objets et leur propriétés, et ceux du tampon du système moteur les actions ou les mouvements. Le module de but est donc l'unique source de chunk abstraits.

Le tampon des buts permet de stocker le but courant du système. On peut y construire des chunks qui contiennent d'autres chunks comme sous éléments. Les auteurs ne sont toutefois pas plus précis quand à l'utilisation du module de buts. D'autant que l'existence à part entière d'un tel module est encore contestée, car les procédés concernant les buts ne sont pas encore localisés avec précision dans le cerveau. Cependant des travaux menés par Altmann et Trafton (2002) portent sur l'élaboration d'un niveau sub-symbolique pour le module d'intentions. Ainsi, le module d'intentions n'est pas encore bien défini et sa légitimité non plus.

3.3.6 L'apprentissage dans ACT-R

ACT-R possède un mécanisme d'apprentissage pour le système central. Il s'agit de la compilation de productions. Le principe est simple : deux productions qui s'enchaînent peuvent être contractées en une seule. Elles sont contractées au cas par cas, voici un exemple qui permet de mieux comprendre le procédé, examinons les deux productions :

```
If  but = lire un mot pour faire un test d'association
```

```
     et un mot a été donné
Then but = retrouver le mot associé

If  On a lancé une requête de recherche pour un test d'association
     et le mot associé a été retrouvé, la réponse est N
Then donner N
```

Dans le cas où on donne le mot Vanille, l'associé est Fraise. La première production va chercher le mot associé à vanille et attendre le résultat. La deuxième production va donner le résultat une fois retrouvé. Par la compilation de production, cette recherche en mémoire va être évitée la prochaine fois. En effet, la nouvelle production obtenue va être :

```
If  but = lire un mot pour faire un test d'association
    et mot donné est "Vanille"
Then donner "Fraise".
```

Ainsi, le système va pouvoir apprendre un certain nombre de connaissances. Par contre, on ne peut pas contracter directement les deux productions génériques précédentes puisque pour une association non connue, il sera nécessaire de faire la recherche en mémoire. On garde ainsi le coté générique tout en apprenant. Un parallèle serait les tables de multiplications que l'on connaît par cœur et le procédé pour multiplier deux nombres.

Avec ce processus, ACT-R peut compiler une série de productions. Toutefois, il y a une limitation : ce processus ne peut fusionner deux productions qui utilisent plusieurs ressources extérieures. Par exemple, on ne peut pas contracter ces deux règles :

```
If   but = ouvrir un menu
Then presser F1
      ET
If   but = ouvrir un menu et on a pressé F1
Then presser F2
      EN :
```

```
If   but = ouvrir un menu
Then presser F1 «puis» F2
```

Le système moteur exerce une contrainte sur le système procédural car il ne peut prendre en compte qu'une seule demande de mouvement à la fois. De la même manière, le système de perception va également empêcher la fusion de certaines règles.

Maintenant que nous avons expliqué le fonctionnement de l'architecture ACT-R 5.0, nous allons discuter de l'approche d'Anderson par rapport à notre problématique et voir quelle pourrait être son utilité.

3.3.7 Résultats et discussions sur l'approche d'Anderson

Les modèles ACT-R de la cognition ont eu de nombreuses applications depuis les premiers énoncés de la théorie. On peut citer des domaines aussi variés que la perception et l'attention, l'apprentissage et la mémoire, la résolution de problème et la prise de décision, le langage et d'autres encore. On voit ainsi qu'ACT-R est un système plus développé par rapport au SAS/CS de Norman et Shallice (1986).

De notre point de vue, ACT-R est plus intéressant que le CS de Cooper et Shallice (2000) car il est beaucoup plus proche du comportement réel humain. La possibilité de simuler le comportement de malades d'Alzheimer lors de la réalisation d'une AVQ est actuellement à l'étude. De plus, ACT-R est un système qu'il est possible d'adapter aux tâches qu'on veut lui faire faire. Ce système est disponible en ligne avec un tutoriel très complet sur la manière de procéder (Site ACT-R). De ce fait, il est très probable que le choix se porte sur ACT-R pour la modélisation d'une architecture cognitive pour le projet DOMUS. Ce système présente tous les avantages pour permettre de simuler le comportement routinier d'une personne atteinte de troubles cognitifs réalisant des activités dans l'habitat. Toutefois, pour pouvoir simuler la cognition chez un être humain, il manque encore un module de supervision à l'architecture d'ACT-R, pour les aspects non routiniers.

Dans ce chapitre, nous avons présenté trois architectures cognitives distinctes : SOAR, EPIC et ACT-R. Nous allons maintenant aborder la conception d'une architecture pour l'habitat intelligent.

Deuxième partie

Élaboration d'une architecture pour l'Habitat Intelligent

Chapitre 4

Architecture globale de l'habitat intelligent

Dans ce chapitre, nous allons décrire les premiers travaux réalisés dans le cadre de ce mémoire, avant d'aborder, la conception proprement dite de l'architecture de l'habitat intelligent. Il sera présenté dans ce chapitre le niveau le plus global de l'architecture en précisant les rôles et les fonctions de chaque composante.

4.1 Une architecture en trois couches

L'habitat s'organise en trois couches distinctes (Pigot et al., 2003). La première, classiquement appelée «HardWare», regroupe tous les aspects liés aux contraintes physiques de l'habitat. Notamment, le choix des capteurs, effecteurs, du matériel informatique, des appareils intelligent, l'organisation de tout ce matériel autour de différents réseaux... Cette couche est la base indispensable aux deux autres couches : le «MiddleWare» et la couche d'applications. La couche du «MiddleWare» va être la couche interface entre l'aspect matériel de l'habitat et la couche d'applications de l'habitat. Notamment, elle va permettre de collecter et d'interpréter en partie les informations issues des capteurs. D'un autre coté, elle fera le lien entre les interventions décidées par les couches supérieures et les effecteurs du système. La couche d'applications, quant à elle, regroupe l'ensemble des processus de décision et d'analyse que comprendront l'habitat pour fournir une assistance cognitive à son occupant. Dans le présent projet, nous nous intéressons uniquement à la couche

d'applications de l'habitat. Cependant, pour ne pas formuler d'hypothèses erronées sur les deux couches inférieures, un plan possible d'habitat et de ses capteurs a été réalisé. Ce plan est décrit dans les paragraphes suivants.

Dans ce plan de l'habitat intelligent, l'idée est de faire ressortir le rôle de chaque capteur au sein du système et de préciser quels risques ils vont permettre de pallier. Par exemple, un capteur de température placé dans la baignoire permet d'éviter que la personne ne prenne un bain bouillant ou à l'inverse un bain glacé. De cette manière, il est possible de pallier le risque de brûlures et le risque d'attraper froid associés à une mauvaise utilisation de la baignoire.

Le mémoire de maîtrise de Chevalier (2000), ainsi que des descriptions médicales des symptômes de la maladie d'Alzheimer (Bonder et Wagner, 1994), ont permis de lister les risques liés au maintien à domicile de personnes atteintes de la maladie d'Alzheimer ou de personnes présentant des handicaps cognitifs lors de la réalisation des AVQ. Selon Chevalier, sept risques sont déterminants dans la décision de maintien à domicile d'une personne atteinte de démence de type Alzheimer :

- Développer, accentuer des problèmes de santé ;
- Errer, fuguer ;
- Être sous-alimenté ;
- Provoquer un feu/ un incendie ;
- Se brûler ;
- Se perdre ;
- S'intoxiquer.

Pour élaborer un plan d'appartement intelligent, l'idée était donc d'utiliser cette catégorisation des risques pour voir quels capteurs pourraient aider à connaître les activités de la personne, pour pallier tel ou tel risque. Ce travail comprenant le plan général de l'appartement, ainsi que les capteurs et les effecteurs avec les tableaux explicatifs associés est présenté l'annexe B page 178.

Certains autres capteurs ne sont pas destinés à pallier des risques particuliers, mais

vont permettre de fournir l'assistance à la réalisation des AVQ. Par exemple, les capteurs de poids sur les plans de travail permettront de suivre la manipulation d'objets lors de la confection d'un repas, et ainsi permettre de suggérer les actions suivantes au moment voulu, si la personne semble ne pas pouvoir mener à terme cette activité. L'idée n'est pas de pallier un risque immédiat ici, mais plutôt d'aider la personne à long terme : l'aide lors de la préparation d'un repas est finalement destinée à pallier des risques de sous-alimentation.

Enfin dans ce modèle, figurent les organes de communication du système vers l'utilisateur. Ces organes sont des moyens d'affichages visuels et autres signaux sonores. Les autres moyens d'intervention du système ne seront pas directement destinés à l'occupant (par exemple, la possibilité d'éteindre certains appareils électriques via des contrôleurs). Ces moyens d'interventions indirects ne sont utilisés que lorsque l'occupant semble ne pas pouvoir intervenir lui-même. Si une plaque électrique reste allumée sans rien dessus pendant plusieurs minutes, le système va d'abord tenter d'avertir l'occupant pour qu'il l'éteigne lui-même, avant de prendre des mesures plus radicales si rien ne se passe.

Aux plans de l'appartement et des différentes pièces, s'ajoutent quelques précisions sur les capteurs portés par la personne et les différentes fonctions qui y sont associées. L'équipe n'a pas encore eu le temps de vraiment réfléchir sur le type de capteurs qui seront portés par l'occupant et ce qu'il sera possible de faire ou non, selon la taille de ces capteurs. Toutefois, ces capteurs devront respecter trois critères importants : la discrétion, le minimum d'encombrement et la légèreté.

Le plan présenté en annexe constitue un ensemble d'hypothèses de départ sur les organes sensorimoteurs du système. Cela permettra de créer des scénarios réalistes lors de l'implémentation en boîte noire de l'architecture modélisée. Ce plan permet aussi de mettre en lumière les compétences nécessaire au système. Par exemple, si le système doit être capable de pallier les risques liés à la préparation d'un repas, il va falloir non seulement qu'il ait les bons capteurs, mais aussi un modèle des activités de la vie quotidienne, et la possibilité de reconnaître et identifier les buts de l'occupant pour l'aider efficacement.

Nous allons maintenant nous intéresser uniquement aux processus mis en jeu dans la couche supérieure de l'habitat intelligent. Nous désignerons l'organisation de ces processus par architecture globale de manière quelque peu abusive, puisqu'il ne s'agira que de la couche supérieure de l'habitat. Ainsi l'architecture proposée ici n'est ni matérielle, ni logicielle (les deux étant des architectures réparties au sein de l'habitat).

4.2 Conception d'une architecture pour l'habitat intelligent

Dans cette partie, nous allons décrire le travail de conception effectué pour le modèle le plus global de l'architecture. Il s'agit de présenter une vue générale du système dont on aura justifié les composantes et les interactions.

4.2.1 Identification des composantes

Comme nous l'avons expliqué dans l'introduction, l'objectif du projet DOMUS est double. Il s'agit, d'une part, d'aider les personnes atteintes de troubles cognitifs à vivre normalement en palliant leurs déficiences, et d'autre part, de les surveiller à long terme afin de prévenir rapidement le personnel qualifié en cas de problèmes (feu, chute, malaise ...). Pour répondre à ces objectifs, nous avions identifié deux modules, celui concernant l'*Assistance cognitive* et celui pour la *Télésurveillance.*

Cependant, cette distinction n'est pas suffisante, il est apparu utile de distinguer deux composantes au sein du module de la télésurveillance en fonction du facteur temps :

- une composante *Urgences* qui s'occupera de la télésurveillance à court terme, c'est-à-dire du traitement des urgences comme les malaises, arrêts cardiaque, débuts d'incendie... ;
- une composante *Télésurveillance* à long terme destinée à suivre l'évolution de l'état de santé de l'occupant.

Le système se décompose ainsi en trois composantes : *Assistance cognitive*, *Télésurveillance* et *Urgences.* Nous expliquerons le rôle de chacune de ces composantes par la suite.

En plus de ces trois composantes majeures, nous avons identifié deux composantes périphériques. La première porte sur l'aspect matériel : un service de détection de panne qui devra être fourni par le système afin de prévenir les pannes de ses capteurs et effecteurs et d'avertir les techniciens de ce type de problème. Le système comportera également un aspect réseau non négligeable. L'idée est qu'un nouvel appareil électro-ménager devra se mettre en réseau automatiquement avec les autres. Ceci est désigné par le terme «réseautage spontané». Ces différents aspects qui comprendront une couche d'applications ont été regroupés dans une composante nommée *Gestion de l'environnement.* En réalité, cette composante n'est là que pour rappeler ces aspects nécessaires au système, il est en effet évident que le réseautage spontané ou la détection des pannes seront des composantes réparties au sein du système. La seconde porte sur la gestion de systèmes automatisés qui ne nécessitent qu'une intervention limitée du système global (composante *Systèmes automatisés*). On pense notamment à des systèmes qui existent déjà dans les maisons comme le chauffage ou la climatisation. Cette composante a été le sujet de quelques réflexions rapportées en annexe C page 193 de ce mémoire.

Dans l'introduction, nous avions déjà évoqué la nécessité de munir le système de modèles de la personne, des activités de la vie quotidienne et de l'environnement selon de modèle de Rousseau et al. (2002). Si l'habitat doit être capable de fournir une aide appropriée, il doit pouvoir se référer à un modèle des compétences, préférences et habitudes de la personne. Un modèle des activités est également nécessaire pour leur reconnaissance. Le système doit pouvoir identifier l'activité en cours. Par exemple, faire chauffer de l'eau le matin ou le soir n'aura pas la même utilité (le matin pour du café, le soir pour faire cuire des pâtes), donc ne nécessitera pas le même type d'aide. Quant au modèle de l'environnement, le système en a besoin pour savoir d'où proviennent les informations (localisation des capteurs), et comment l'environnement peut pallier aux handicaps (localisation des effecteurs).

Avant de lier les composantes identifiées entre elles, il est nécessaire d'apporter des précisions sur les dépendances entre les différents modèles évoqués. En effet, dans le modèle de la personne, se trouveront des données sur ses préférences, ses compétences cognitives et physiques, son vécu, ses habitudes et divers paramètres supplémentaires.

Les habitudes de la personne permettent de rassembler un ensemble de connaissances sur la façon générique de réaliser une certaine activité. Cette habitude peut être datée ou non, par exemple, l'habitude de déjeuner devant la télévision, ou encore le 25 décembre, de dîner chez ses enfants. Le lien avec le modèle des activités apparaît aussitôt. Le modèle des activités décrit les différentes manières de réaliser les AVQ, les habitudes vont être les différentes manières de réaliser les AVQ pour une personne donnée. Ainsi les habitudes instancient le modèle des activités.

Les habitudes de vie sont constituées d'éléments variés : habitudes liées à la nutrition, à la condition physique et mentale (relaxation, nombre d'heure de sommeil), aux soins corporels (toilette, hygiène, habillement, médication), à la communication entre personnes et avec son environnement, à l'habitation (choix et aménagement du domicile, mode d'entretien ...), aux déplacements (mode de transports), aux responsabilités (budget, citoyenneté), au travail et aux loisirs. La CIDIH (Classification Internationale des Déficiences, Incapacités et Handicaps) propose une nomenclature complète pour les habitudes de vie, les risques, les compétences, incapacités d'une personne (Fougeyrollas et al., 1993). Il pourra être utile lors de la conception du modèle de la personne et de celui des activités.

De plus, nous avons décidé de séparer compétences physiques et compétences cognitives au sein du modèle de la personne, puisque notre habitat met l'accent sur l'aspect cognitif. Les compétences cognitives porteront sur l'ensemble des aspects cognitifs : attention, concentration, mémoire, langage... et les compétences physiques sur l'ensemble des aspects physiques ou physiologiques.

Enfin, il est nécessaire de munir le système d'une représentation de la mémoire épisodique de la personne qui constituerait son vécu. Cette mémoire épisodique nous permettrait notamment de déduire les habitudes de la personne qu'elles soient normales ou non. Les habitudes de vie font alors référence aux habitudes saines de la personne pour servir de point de référence lors de l'évolution de la maladie. En examinant la mémoire épisodique, les habitudes peuvent être déduites ainsi que les déviations de ces habitudes.

La mémoire épisodique et les capacités cognitives de l'occupant nous le verrons jouent un rôle essentiel dans la détermination de l'assistance cognitive.

Suite à ces diverses constatations, les différents modèles et leur contenu s'organisent selon le schéma de la figure 4.1.

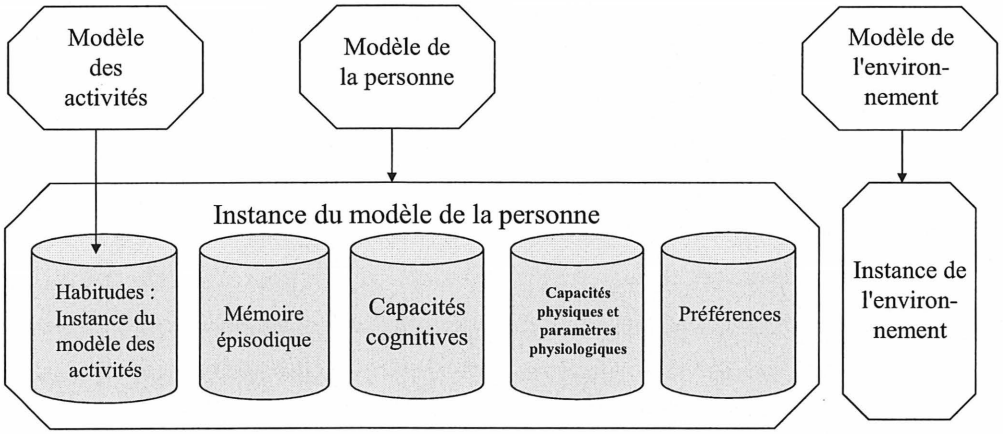


FIG. 4.1 – Organisation des différents modèles et de leur instance

Dans cette figure, est décrit, d'une part, le modèle de la personne et son instance, contenant les habitudes, qui est une instance du modèle des activités, la mémoire épisodique, les capacités cognitives, les capacités physiques et les paramètres physiologiques et enfin les préférences. D'autre part, est aussi représenté le modèle de l'environnement et son instance.

Maintenant que nous avons identifié les composantes principales du système et précisé

les modèles qu'elles utiliseront, nous allons détailler leurs interactions.

4.2.2 Un modèle global de l'architecture

Pour résumer, le système comprend trois composantes majeures : les *Urgences*, la *Télé-surveillance*, l'*Assistance Cognitive* ainsi que deux composantes périphériques : la *gestion de l'environnement* et les *systèmes automatisés*. Ses composantes interagissent avec une *instance du modèle de la personne*, le *modèle des activités* et une *instance du modèle de l'environnement*. C'est grâce au contenu de ces instances et du modèle que les composantes auront les informations nécessaires pour interpréter les données en provenance de l'habitat. D'autre part, une partie des composantes interagit avec l'environnement extérieur constitué par l'occupant, l'habitat et le personnel spécialisé via une composante qui prend en charge les modalités de communication. Ces interactions sont présentées dans le schéma de la figure 4.3 page 97. Pour bien comprendre ce schéma, il convient de se référer à la légende (cf figure 4.2).

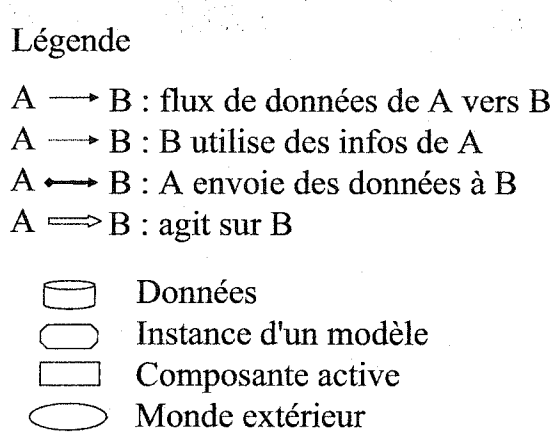


FIG. 4.2 – Légende pour le modèle global de l'architecture

Sur la figure 4.3 page 97, on voit aussi apparaître une composante dont nous n'avons pas encore parlé : le module de *gestion des interventions*. Celui-ci va permettre au système de choisir entre les diverses interventions qu'il aura à faire et de les hiérarchiser par ordre de priorité. Il va également permettre de prendre en compte les préférences

de la personne pour l'intervention et les possibilités d'intervention via les instances de la personne et de l'environnement.

Pour expliquer les différentes interactions de la figure 4.3 (page suivante), nous partons du monde extérieur pour y revenir. Premièrement, des données (représentées par un cylindre) sont collectées dans le monde extérieur : dans des systèmes distants, au sein de notre habitat et sur l'occupant. Il y a alors plusieurs possibilités. Selon le type de données, le traitement sera différent. Les données du type température ou luminosité sont directement interprétées par la composante *systèmes automatisés*. Ceux-ci agissent sur l'environnement réel en ajustant la climatisation, le chauffage ou l'éclairage. Ils utilisent des paramètres contenus dans le modèle de la personne comme «de telle à telle heure il faut chauffer à 20°C» à la manière des thermostats électroniques. Cette dépendance n'est pas représentée sur la figure par souci de clarté. Des détails sur cette composante sont disponibles dans l'annexe C page 193.

Certaines données sont traitées par la *gestion de l'environnement*. Ce module s'occupe de l'aspect matériel de l'environnement, notamment du réseautage spontané et de la détection des pannes matérielles : défaillance d'un capteur, d'un prompteur, coupure d'un réseau ... Il répercute ces pannes dans l'instance du modèle de l'environnement pour que le module de *gestion des interventions* et la couche de communication ne cherchent pas à utiliser du matériel indisponible. De plus, la composante de *gestion de l'environnement* envoie des informations sur la ou les pannes au module de gestion des interventions pour prévenir le personnel technique.

D'autres données issues des capteurs sont prises en charges par la composante *Urgences*. Lorsqu'une urgence est déclenchée, typiquement, les battements du cœur viennent à ralentir de façon alarmante ou un début d'incendie a été détecté, le module d'Urgences caractérise le problème et envoie des informations qui sont prioritaires au module de gestion des interventions. Il faut noter que certaines décisions et actions seront prises localement au niveau des appareils eux-mêmes sans que cela atteigne le module de gestion des interventions. Mais dès lors qu'il faut accéder à l'extérieur, ou que l'urgence ne peut pas être traitée par un appareil simple, on passe par le module de gestion des

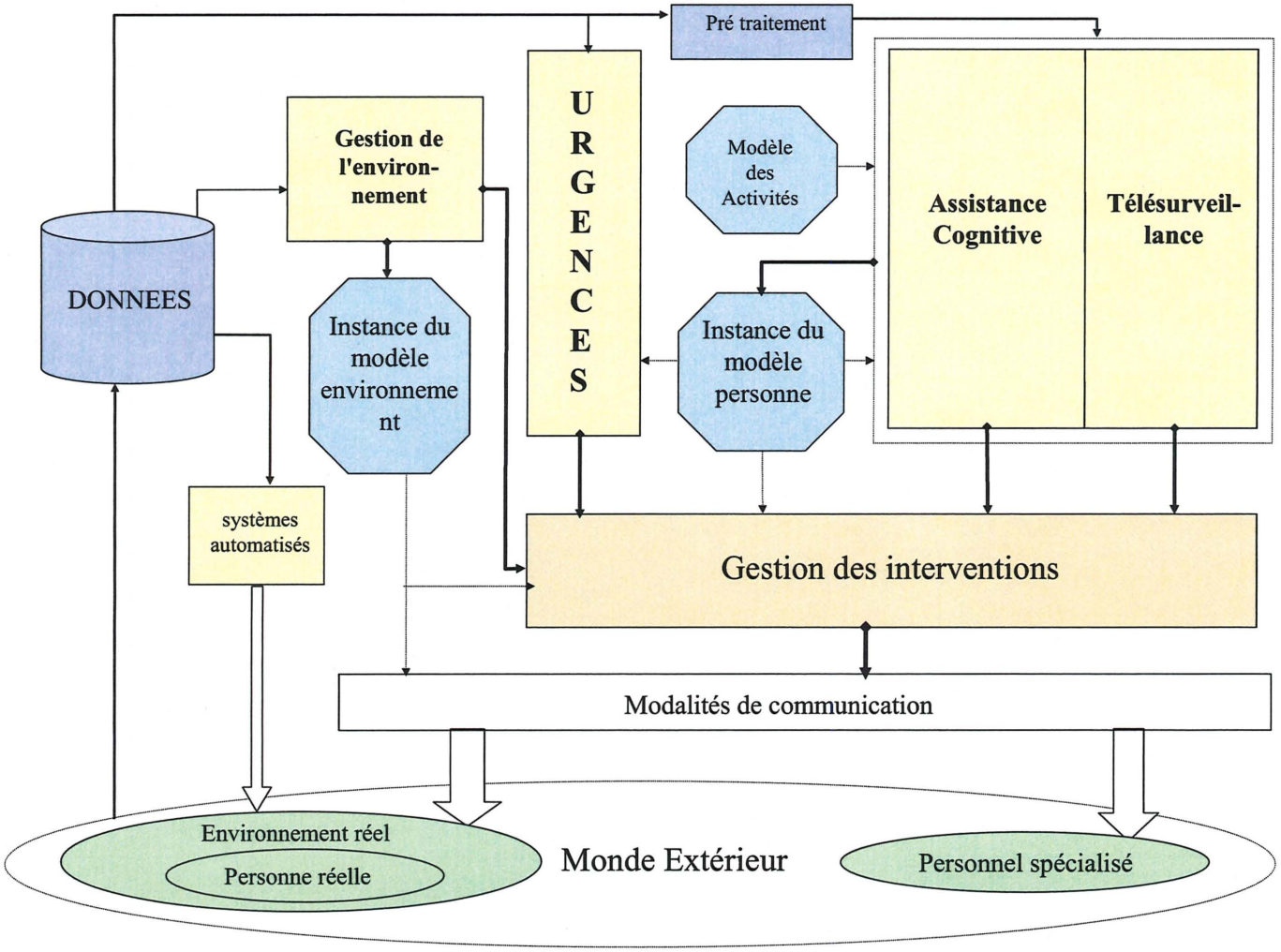


FIG. 4.3 – Modèle global de l'architecture de l'habitat intelligent

interventions qui va faire le relais et prévenir directement les personnes concernées et/ou déclencher les bonnes actions au niveau matériel. Ce module aura été optimisé pour minimiser le temps de réponse pour une information de type urgente. Le module *Urgences* utilise également des données contenues dans l'instance du modèle de la personne. Par exemple, quel médecin ou hôpital prévenir, ne pas s'alarmer si les battements du cœur sont irréguliers car l'occupant fait de l'arythmie cardiaque...

Ensuite, il y a deux autres possibilités de traitement. Premièrement avant d'atteindre le module de *télésurveillance* et le module d'*assistance cognitive*, un pré-traitement est effectué sur les données. La teneur de ce pré-traitement n'est pas encore définie. Elle dépendra du type de données qui sera utile pour les deux modules. L'assistance cognitive est le cœur du système de l'habitat intelligent et la télésurveillance à long terme lui est extrêmement lié. Nous verrons que ces deux modules partagent de nombreuses fonctionnalités, c'est pour cette raison qu'ils ont été accolés sur la figure 4.3 (page précédente).

Pour le module d'*assistance cognitive*, le système doit être capable de reconnaître les buts et les actions de la personne, d'évaluer ses compétences dans la tâche pour décider s'il est nécessaire ou non d'intervenir. Le module d'assistance envoie au module de gestion des interventions une information sur le type d'aide à donner et son contenu, par exemple : «il faudrait rappeler à l'occupant de prendre ses médicaments». L'*assistance cognitive* utilise des connaissances contenues dans l'instance du modèle de la personne, notamment des données concernant ses compétences cognitives. Ces données seront utilisées pour simuler le comportement de la personne, et ainsi prévoir ses handicaps lors de la réalisation des AVQ afin de pouvoir l'aider au bon moment.

Dans le module de *télésurveillance*, l'état du patient va être surveillé à long terme, et des rapports périodiques sur son évolution vont être envoyés aux médecins et au personnel soignant, qui pourront demander des précisions au système. Ce module utilise des connaissances comprises dans l'instance du modèle de la personne, notamment pour suivre l'évolution des paramètres physiques et physiologiques ainsi que le comportement de la personne (habitudes et mémoire épisodique). De plus, le module de *télésurveillance* pourra mettre à jour l'instance de la personne pour répercuter des changements de l'état

de santé ou des habitudes, tant que celles-ci ne sont pas déviantes. Le module sera capable de faire ces changements de manière intelligente, c'est-à-dire de distinguer entre un changement d'habitudes sans conséquence et un comportement déviant dangereux pour la personne, qu'il ne faudra alors surtout pas répercuter dans la partie habitude du modèle de la personne, mais qui sera pris en compte dans la mémoire épisodique.

Les modules d'*assistance cognitive* et de *télésurveillance* décrits ici ne font état que d'un fonctionnement automatique, c'est-à-dire que dans l'élaboration des interventions ou des rapports, il n'y a pas d'intervention humaine de prévue. Cependant, on peut imaginer que ce soit le cas, par exemple avec une aide à distance fournie par un ergothérapeute ou l'ajout d'informations par le personnel médical. Dans ce mémoire est uniquement décrit une assistance cognitive élaborée sans aide humaine et dont les interventions se font également sans aide humaine. En ce qui concerne la télésurveillance, l'élaboration des rapports et la prise de décision les motivant se fait également sans intervention humaine. Par contre, les rapports sont à destination du personnel médical ou des proches et peuvent donc déclencher des interventions humaines (exemple, la tension est trop élevée depuis quelques jours, le médecin avance donc sa visite hebdomadaire).

Dans les quatre modules précédemment décrits, une décision est déjà prise quant au type d'intervention qu'il faut effectuer. Le module de *gestion des interventions* sert donc à hiérarchiser les divers types d'interventions par leur priorité et à ordonner leur exécution en choisissant les bons médias pour intervenir dans l'environnement réel.

Ceci décrit un modèle global de fonctionnement de l'habitat intelligent. Il faut bien garder à l'esprit que ce modèle ne reflète ni l'architecture matérielle du système (qui sera certainement répartie), ni son architecture logicielle. L'idée est maintenant de préciser une partie des composantes citées auparavant. Dans le paragraphe suivant, sont expliqués les choix quant aux composantes à spécifier.

4.2.3 Choix des composantes à spécifier

La décomposition du modèle global de l'architecture ne sera effectuée qu'en partie. Certaines composantes sont attachées à l'aspect matériel de l'informatique, ou elles sont

déjà le sujet de recherche d'autres étudiants du projet. Ces composantes ne seront donc pas détaillées dans le présent mémoire.

Premièrement, la spécification des modèles d'activités, de la personne, de l'environnement ne sera pas plus détaillée que ce qui a déjà été précisé. Ce travail a été confié à d'autres étudiants. Les spécifications déjà fournies sont donc celles qui seront utilisées (cf. figure 4.1 page 94).

D'autre part, l'aspect *modalités de communication* et *gestion de l'environnement* ne seront pas non plus détaillés. La composante *Urgences* est assez proche du matériel puisque celles-ci seront en partie gérées par les appareils eux-mêmes. A la manière d'un réflexe pour le corps humain, toute l'information ne remontera pas systématiquement au niveau du module de *gestion des interventions*. Aussi la composante *Urgences* ne sera pas non plus détaillée.

Le projet va donc s'attacher à concevoir et modéliser de façon précise les composantes d'*Assistance Cognitive* et de *Télésurveillance* à long terme, ainsi que préciser le rôle du module de *gestion des interventions*. Ce travail est l'objet des chapitres suivants dans cette partie. La section suivante porte sur le module de *gestion des interventions*. Les chapitres 5 page 104 et 6 page 125 sur les composantes d'*Assistance cognitive* et de *Télésurveillance*.

4.3 Le module de gestion des interventions

Pour bien comprendre l'idée sous-jacente de ce module, nous allons commencer par décrire un scénario d'intervention du système, puis nous définirons le rôle et les fonctions de ce module.

4.3.1 Scénario de fonctionnement

Madame Dubois est une dame âgée de 82 ans et réside dans un habitat intelligent comprenant une salle de bain, une chambre à coucher, une cuisine et la salle principale.

Elle mesure 1m50 et pèse 55kg. Elle est myope et n'entend plus très bien. Elle n'a pas de problème de mémoire à long terme, elle se souvient toujours des histoires de guerre de son défunt mari, d'ailleurs elle adore raconter ses souvenirs. Toutefois, depuis sept ans, ses deux enfants remarquent que les tâches ménagères semblent prendre plus de temps que d'habitude et qu'elle oublie plus de choses qu'à l'accoutumée.

Ce jour-là, madame Dubois se prépare à manger. Une fois qu'elle a fini, elle enlève la casserole du feu et s'installe pour manger dans la salle principale. Hélas, elle laisse la plaque électrique allumée et un torchon se trouve à proximité. Le torchon prend feu. Madame Dubois ne s'aperçoit de rien car elle est dans une autre pièce et la ventilation de la cuisine absorbe le début de fumée. Le système détecte une série d'informations qui vont être traitées à différents niveaux. Au niveau des urgences, le système détecte le départ de feu et décide qu'il faudrait prévenir madame Dubois, les voisins et/ou les pompiers. L'assistance cognitive, elle, détecte un oubli : la plaque est restée allumée et il faut l'éteindre, donc elle décide qu'il faudrait prévenir Madame Dubois de son oubli. Deux informations arrivent donc au module de décision : prévenir Madame Dubois, les voisins et/ou les pompiers du départ de feu, et prévenir Madame Dubois de l'oubli d'une plaque électrique.

Dans le module de *gestion des interventions*, la priorité est telle que c'est la décision venant de la composante Urgences qui va être prise en compte en premier lieu. Le système décide alors d'allumer et d'éteindre toutes les lumières de l'appartement pour prévenir madame Dubois (elle préfère ceci à une sirène hurlante). C'est une mesure qui peut sembler radicale, mais il s'agit d'une urgence majeure, il ne faut donc pas tergiverser sur le moyen de prévenir l'occupant. Dans le même temps, le système prévient d'abord les voisins. Madame Dubois voit alors la coupure de lumière et entend la sonnette. Elle réalise que quelque chose ne va pas. Elle retourne dans la cuisine et voit le feu. Elle panique et se hâte tant bien que mal pour ouvrir la porte. Les voisins sont déjà munis de l'extincteur et maîtrisent le feu en quelques secondes. La plaque électrique avait déjà été coupée par la cuisinière elle-même. Dans la pile d'intentions du module de gestion des interventions, on avait placé la prévention des pompiers si le feu persistait plus d'un certain temps, ou si les voisins n'intervenaient pas. Mais, tout est rentré dans l'ordre et la nécessité de

prévenir les pompiers est alors supprimée des intentions, car elle est devenue inutile.

Ce scénario illustre le rôle du module de gestion des interventions au sein de l'architecture. Il va établir les ordres de priorités sur les actions à faire et choisir les moyens d'action au niveau haut. Nous allons décrire plus précisément les rôles et les fonctions de ce module dans le paragraphe suivant.

4.3.2 Rôle et fonctions du module de gestion des interventions

Le module de *gestion des interventions* est celui qui va définir ce que va faire le système. Il va avoir à choisir entre les diverses possibilités d'interventions qu'il aura : l'ensemble des «Il faudrait que ...» issus des différentes composantes (*urgences, télésurveillance, assistance cognitive et gestion de l'environnement*). Si on se réfère à notre scénario, il s'agit des deux informations «Prévenir Madame Dubois, les voisins et/ou les pompiers pour le feu» et «Prévenir Madame Dubois pour la plaque allumée».

De façon plus détaillée, le module de *gestion des interventions* a un rôle de sélection de l'action à effectuer. Il va permettre de hiérarchiser les aides. Une aide ou une action issue de la composante *Urgences* sera prioritaire par rapport à une action de l'*assistance cognitive* qui sera prioritaire sur un envoi de rapport de la part de la *télésurveillance.* La prévention des techniciens lors d'une panne sera faite le plus tôt possible, dès que cette action n'en gênera pas une autre. Cet ordre de priorité est arbitraire, il est vraisemblable que cela dépendra aussi de la situation. La prévention des techniciens au sujet d'une panne dans le système peut s'avérer cruciale, si cette panne paralyse tout ou une partie du système. Mais par contre la priorité sera toujours donnée aux urgences, car par définition ce sont les actions à traiter le plus rapidement.

En plus de classer les différents messages et actions par ordre de priorité, le module de *gestion des interventions* déterminera par quels moyens l'action sera faite. Il s'agit ici de définir le contenu du message ou le type d'actions selon les préférences de l'occupant quant aux moyens, à la forme et par rapport à ses capacités physiques notamment la vue, l'ouïe. De plus, en précisant ce genre de chose, il donnera des informations sur le type d'effecteur à utiliser : affichages visuels, signaux sonores ou autres sans toutefois aller

jusqu'à préciser lequel.

Le module de *gestion des interventions* devra prendre des décisions réfléchies sur les interventions du système. Par exemple, il devra être capable de détecter les conséquences de l'action et/ou ses interactions dans l'environnement réel. Pour donner un exemple, si madame Dubois regarde la télévision à un certain volume puisqu'elle est un peu sourde, il est fort probable qu'elle n'entende pas un signal sonore. Alors, même si sa préférence allait à la prévention sonore, lorsque la minuterie du four arrive à son terme, le module de gestion des interventions peut utiliser un affichage visuel pour lui dire que son repas est prêt pour renforcer le signal sonore.

Ce module est en quelque sorte la clé de voûte du système et finalise les décisions prises dans les différents modules. C'est en dernier lieu lui qui va décider quelle action faire et quand. Ce module va devoir disposer d'une base de connaissance et d'un accès aux instances du modèle de la personne et de l'environnement comme indiqué sur le modèle global de l'architecture (cf figure 4.3 page 97). Nous ne donnons pas ici de moyens pour réaliser ce module, nous nous sommes plus intéressés aux modules d'*assistance cognitive* et de *télésurveillance.*

Chapitre 5

Assistance cognitive et Télésurveillance

Dans ce chapitre, nous allons donner la description d'un modèle pour les composantes *Assistance cognitive* et *Télésurveillance.* Avant de proposer une représentation globale de l'assistance cognitive et de la télésurveillance, nous allons décrire les différentes composantes, fonctions et rôles de ces deux modules. Le but de cette démarche est de permettre une compréhension accrue du modèle qui sera proposé.

5.1 Identification des composantes et fonctions pour l'assistance cognitive

Dans le chapitre 4 page 88, nous avons établi un modèle de fonctionnement global pour l'habitat intelligent. Nous avons vu notamment qu'on effectuait un pré-traitement sur l'ensemble des informations parvenant du monde extérieur. Ce pré-traitement ne s'occupe que de considérations numériques : lissage des signaux, filtrage... On peut donc considérer que les données obtenues à la sortie de ce pré-traitement sont toujours les données brutes, mais de meilleure qualité. Les données évoquées par la suite font référence aux données issus de ce pré-traitement

5.1.1 Types de données

Pour pouvoir réaliser la tâche d'assistance cognitive et de télésurveillance à long terme, il va falloir analyser les données reçues et en retirer les caractéristiques essentielles. Tout d'abord, on distingue plusieurs types de données parvenant du monde extérieur :

- les données issues des capteurs de l'habitat, qui vont renseigner le système sur les activités de la personne ;
- les données dites physiologiques, qui seront obtenues grâce à des capteurs portés par la personne ou par des appareils de mesure installés dans l'habitat ;
- les données extérieures à l'habitat, qui seront sous forme de requêtes des différents acteurs : demande de renseignement sur l'état du patient ou ajout d'informations au système...

Ces types de données vont être traités différemment par les composantes de l'*assistance cognitive*. Nous allons d'abord nous intéresser au traitement des données issues des capteurs.

5.1.2 La reconnaissance des activités

Pour pouvoir fournir une aide appropriée, il a déjà été établi qu'il était nécessaire de pouvoir reconnaître les activités de la personne dans l'habitat. Le système va tout d'abord procéder à une reconnaissance atomique, c'est-à-dire reconnaître les opérations[4] de la personne au sein de l'habitat. Cette *reconnaissance atomique* va permettre à une composante de *reconnaissance et de suivi des activités* d'obtenir des renseignements sur l'activité en cours de réalisation et sur les activités futures.

La composante de *reconnaissance et de suivi* utiliserait un automate hiérarchique probabiliste. Les états de cet automate représentent les opérations d'une activité. On se déplace alors d'un état à un autre au fur et à mesure de la reconnaissance d'activités atomiques. Cependant, ceci a besoin d'être complété si on désire obtenir plus d'informations que la simple séquence d'opérations. L'idée est donc d'introduire un aspect hiérarchique

[4]Une activité se décompose en tâches, sous tâches ... et pour finir en opérations. Une opération est une partie de l'activité qui ne peut plus être décomposée. Par exemple, dans l'activité se laver les dents, prendre sa brosse à dents est une opération

pour les activités, qui traduirait la hiérarchie activités, tâches, sous-tâches, opérations. Cet aspect hiérarchique s'inspire des chaînes de Markov hiérarchiques (Fine et al., 1998) dont nous allons présenter un exemple. Ainsi, une chaîne d'opérations constitue une tâche ou une sous-tâche, une chaîne de tâches ou de sous-tâches forme une activité. De cette manière, lorsqu'une opération est reconnue, la reconnaissance des activités peut situer l'opération par rapport à une activité.

Nous venons d'évoquer l'aspect hiérarchique de l'automate. En ce qui concerne l'aspect probabiliste, il s'agit ici de traduire les habitudes de vie d'une personne. Le fait que tous les matins, l'occupant se brosse les dents en premier, puis se douche. Ceci est traduit en terme de probabilités : au début de la toilette, la probabilité de se brosser les dents sera plus importante que celle de prendre une douche. La figure 5.1 présente la décomposition de la toilette matinale en chaînes de Markov hiérarchiques. Cette décomposition-ci ne va pas jusqu'aux opérations élémentaires de l'activité, elle s'arrête aux sous-tâches.

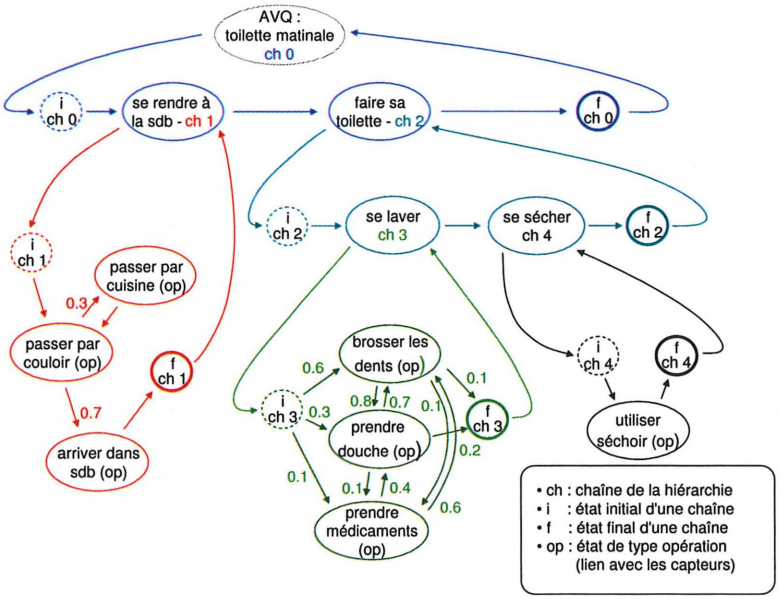


FIG. 5.1 – Exemple de chaînes de Markov pour l'activité de toilette matinale (Exemple tiré du travail de Bauchet J.)

Les arcs de la figure 5.1 (page précédente) qui ne portent pas de probabilités sont des arcs dont les transitions sont immédiates. Pour les autres, les probabilités de transition traduisent les habitudes et la suite probable des activités à un moment donné. Par exemple, si on examine la chaîne 3 : «se laver», à cet instant, la suite d'activité probable sera : «se brosser les dents (0.6)» puis «prendre sa douche (0.8)». Ensuite, si la personne a commencé par se brosser les dents puis prendre sa douche, alors la probabilité qu'elle retourne «se brosser les dents» après sa douche, qui est actuellement de 0.7 devra avoir changé et être très basse. Les suites d'activités probables seront «prendre un médicament» ou «se sécher».

En plus des probabilités sur les transitions, les transitions vont être conditionnées, c'est-à-dire que pour passer d'un état à un autre un ensemble de conditions devront être satisfaites. Ceci permet de prendre en compte le contexte de l'activité. Par exemple, l'idée que pour se brosser les dents de manière efficace, il faut auparavant avoir pris sa brosse à dent et avoir mis du dentifrice dessus. Ces préconditions vont permettre dans une certaine mesure de déceler des erreurs de la part de l'occupant.

De plus, l'utilisation de cette décomposition hiérarchique et des probabilités va permettre de détecter des comportements anormaux ou des changements d'activité. Par exemple, si la personne se lave les dents dans la salle de bain puis décide d'aller dans la cuisine pour préparer son petit déjeuner sans prendre sa douche, la composante signale un abandon de tâche avec l'information que la personne aurait dû prendre sa douche avec une probabilité de 0.8. Si la personne semble persister à ne pas prendre sa douche, par exemple en allant s'habiller, on peut alors lui suggérer de prendre sa douche (on imagine ici qu'il s'agit d'un matin ordinaire pour une personne âgée, mais qui a tendance à perdre la mémoire). Cette observation des probabilités d'actions futures va donc nous permettre de détecter des ruptures dans la routine quotidienne, et nous permettre d'aider la personne à ne pas oublier des tâches importantes dans une activité.

5.1.3 Calcul des probabilités

La *reconnaissance des activités* utilise un modèle probabiliste dont les probabilités évoluent dans le temps. Le calcul de ces probabilités n'est pas évident. En effet, comment

les faire évoluer dans le temps ? L'idée est d'utiliser une simulation des activités de la personne dans l'environnement, afin de déterminer les probabilités d'actions au temps suivant. Pour simuler ces activités, nous allons utiliser une architecture cognitive hybride ACT-R - SAS. Cette architecture est l'objet du chapitre 6. Afin de pouvoir simuler la personne et ses actions futures, l'architecture cognitive aura besoin de connaître les actions passées, c'est-à-dire le vécu de la personne, qui est représenté par la mémoire épisodique dans le modèle de la personne. En effet, il faut rappeler que la valeur d'activation de base d'une connaissance, dans la mémoire déclarative d'ACT-R, est calculée en fonction du nombre de fois où la connaissance a été utilisée. Ainsi, pour simuler avec succès les activités d'une personne, il faut que le système ait accès au nombre de fois effectif, c'est-à-dire réel, où telle ou telle connaissance a été utilisée, ou du moins qu'il soit capable, à partir de la mémoire épisodique, de déduire que telle ou telle connaissance a été utilisée à ce moment-là pour réaliser une action. L'idée est donc de construire les épisodes de la mémoire épisodique à partir des informations issues de la *reconnaissance des activités.*

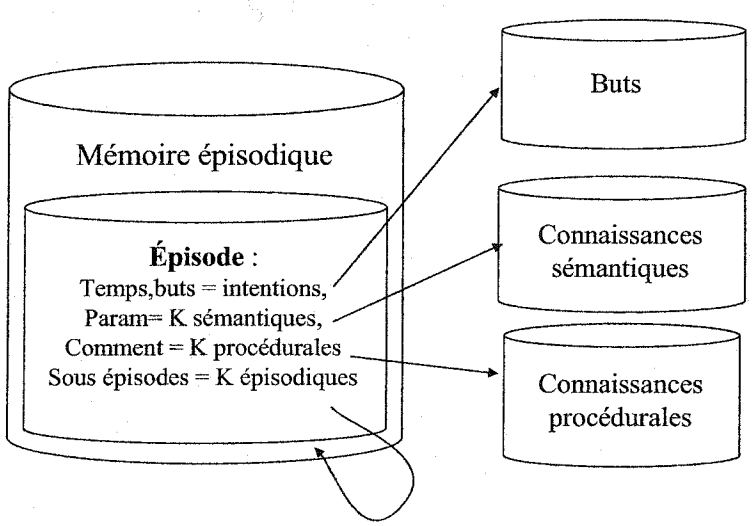


FIG. 5.2 – Structure des connaissances de la mémoire épisodique

La structure d'un épisode est présentée dans la figure 5.2. Une connaissance épisodique

(Mayers, 1997) est une connaissance déclarative particulière qui est composée :

d'un but , qui indique pourquoi cette action est effectuée ;

de paramètres , qui sont un ensemble de connaissances sémantiques déterminant le contexte de l'activité ;

d'une façon de faire , qui est une connaissance procédurale spécifiant de quelle manière l'activité a été réalisée ;

de sous épisodes , qui sont les sous-événements qui ont permis de réaliser l'action.

Une fois les épisodes construits, la composante de *simulation de la personne* va être capable de savoir ce que la personne a fait auparavant. Il devrait alors pouvoir calculer les probabilités d'actions futures à partir de la mémoire épisodique, et ainsi permettre l'évolution dans le temps des probabilités au sein des chaînes de Markov. Cette idée suggère un fonctionnement cyclique qui est l'objet de la section suivante.

5.1.4 Le cycle de reconnaissance et de suivi dans son ensemble

Les composantes *Reconnaissance et suivi des activités*, *Mémoire épisodique* et *Simulation de la personne* forment un cycle au sein de la composante d'*assistance cognitive*. Pour mieux comprendre le déroulement de ce cycle, il est présenté sur la figure 5.3 (page suivante) avec les différentes étapes qui interviennent.

Au temps t, des informations parviennent au module de *reconnaissance des activités* qui y reconnaît une activité. A partir de cette activité, une composante de traitement (*construction et répartition*) va construire ou compléter des épisodes dans la mémoire épisodique. Le module de *simulation* utilise entre autres les connaissances de la mémoire épisodique pour simuler les actions de la personne et fournit ainsi au module de reconnaissance les probabilités associées aux transitions pour le temps t+1. La composante chargée de la construction des épisodes va de plus passer des informations à d'autres composantes pour permettre le diagnostic. Nous préciserons cet aspect dans les sections suivantes.

Dans la section précédente, nous avons donné un exemple de chaîne de Markov concernant la toilette matinale 5.1 page 106. En reprenant cette exemple, nous allons illustrer

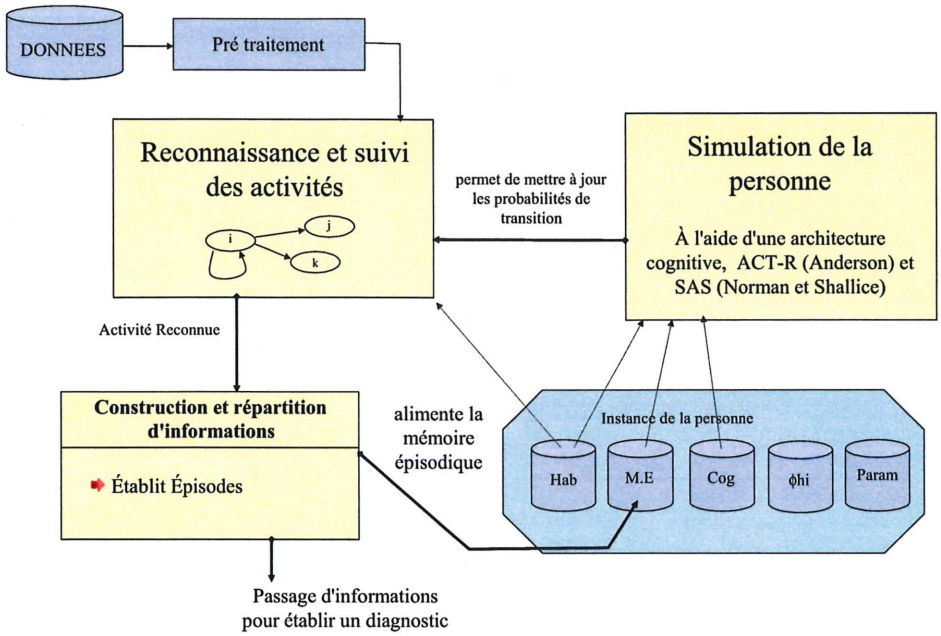


FIG. 5.3 – Le cycle de reconnaissance et de suivi inclus dans la composante Assistance cognitive

le cycle. Considérons la chaîne 3 «Se Laver», la personne vient donc d'arriver dans la salle de bains, et les activités probables sont par ordre décroissant : «Se brosser les dents», «Prendre sa douche» et «Prendre ses médicaments». La *reconnaissance* détecte que la personne commence par prendre sa douche. Un épisode concernant cette activité est construit par la composante de *construction et répartition.* L'architecture cognitive simule alors la personne dans la situation courante et obtient de nouvelles probabilités pour les transitions. Ainsi, la probabilité de se brosser les dents devient 0,6, celle de prendre ses médicaments devient 0,2, et celle de terminer l'activité devient 0,2. On peut supposer que le calcul et la mise à jour des probabilités puissent se faire avant le début d'une autre activité, si les temps de calculs sont suffisamment courts.

En examinant le fonctionnement de ce cycle, on se demande très vite comment il a

été initialisé, puisque pour faire le diagnostic et construire les épisodes au temps t, il nous faut les probabilités de transition au temps t, qui auront été obtenues à partir des informations au temps t-1 grâce à la *simulation de la personne*, etc... L'initialisation des probabilités de transition du module de *reconnaissance* et celle de la *mémoire épisodique* sont problématiques. L'idée pour répondre à ce problème va être de partir des habitudes de vie de la personne qui auront été entrées préalablement dans le système (grâce à un questionnaire par exemple) puis de faire passer le système par une phase d'adaptation à l'occupant où l'apprentissage du système sera très important. Cet apprentissage sera différent de l'apprentissage permanent que fera le système pour suivre l'évolution de l'occupant. Ces problèmes ne seront pas plus détaillés ici, ils seront l'objet de points suivants. Nous allons maintenant aborder l'aspect diagnostic.

5.1.5 Le diagnostic

La composante de *reconnaissance des activités* va fournir au système des renseignements sur les actions de la personne, passées et futures, ainsi que des informations qualitatives du type «la personne a abandonné la tâche en cours» ou «la personne a débuté une autre tâche en parallèle». A l'aide des informations sur les actions passées et des informations qualitatives, le système pourra souvent aider la personne sur ce qu'elle était en train de faire. Il va pouvoir décider si une intervention est nécessaire, et si c'est le cas, en déterminer la nature. Par exemple, le système a reçu les informations suivantes du module de *reconnaissance* : la personne a abandonné la tâche en cours qui était la préparation du repas, elle s'est dirigée vers la salle à manger. Le système décide dans un premier temps de ne pas intervenir. Ensuite, une information arrive : la personne est dans le fauteuil devant la télévision, elle est potentiellement en train de regarder la télévision, mais son repas est toujours en train de cuire. Le système sait grâce au modèle de la personne qu'elle a de grandes chances d'oublier son repas sur le feu. Le système décide alors de lui rappeler qu'elle a commencé à préparer son repas. Il s'agit ici d'une aide sur des actions déjà effectuées et dans cet exemple, le système agit comme un aide mémoire.

Avec les probabilités d'actions futures de la personne, le système pourra souvent prévoir une aide pour l'activité à venir. Dans une séquence d'actions, on peut suggérer l'action suivante à la personne avant qu'elle n'oublie de la faire. Le diagnostic sur les

actions futures permet d'anticiper un certain nombre de handicaps connus de l'occupant, et lui permettre ainsi de réaliser diverses tâches sans interruption ou sans erreur. D'autre part, le système peut aussi lui suggérer des activités probables à un plus haut niveau. Par exemple, si la personne semble errer ou persévérer dans une activité, le système interviendrait en lui suggérant l'activité future probable qu'elle devrait réaliser. Ici, le système remplit un rôle de planificateur pour l'occupant en s'occupant de le guider à travers ses diverses tâches au cours d'une journée.

De cette manière, le système va être capable d'aider cognitivement la personne, en intervenant au moment adéquat. Le système va agir à la fois comme un aide mémoire et un planificateur, permettant à l'occupant de vivre normalement chez lui malgré ses troubles cognitifs. Nous allons maintenant nous intéresser aux fonctions et composantes nécessaires à l'aspect télésurveillance du système.

5.2 Identification des composantes et fonctions pour la télésurveillance

Dans la partie Télésurveillance, nous allons nous intéresser au traitement des données de type physiologique, puis à l'aspect télésurveillance sur les activités.

5.2.1 Traitement des données physiologiques

Les données physiologiques pourront être issues de capteurs portés en permanence par l'occupant ou de capteurs fixes, tel qu'un appareil pour mesurer la tension. A l'aide des données récoltées, l'idée sera d'établir des courbes d'évolution dans le temps de ces différents paramètres, ainsi que des courbes de tendances, par exemple, évolution de la moyenne faite sur dix minutes de mesures. Pour des données collectés par des appareils externes, beaucoup moins de données seront disponibles, elles seront donc vraisemblablement exploitées directement.

A partir des valeurs, courbes et moyennes de référence, le système pourra souvent détecter des changements anormaux ou une évolution de certains paramètres, puis émettre

un rapport pour prévenir le médecin et les personnes concernées s'il y a lieu. Par exemple, la tension de la personne est plus faible que la normale et ceci depuis deux jours, le système prévient alors le médecin par un rapport. Celui-ci pourra alors avancer sa visite et diagnostiquer l'origine de cette baisse de tension. Il s'agit ici de télésurveillance à long terme. Si les paramètres physiologiques se dégradent trop violemment, c'est la composante *Urgences* qui intervient tout de suite, notamment en cas de chute ou de malaise.

Le traitement des données physiologiques n'est actuellement pas à l'étude dans le projet DOMUS, mais il pourra éventuellement être couplé avec les données envoyées sur les activités lors des rapports de *Télésurveillance.*

5.2.2 L'historique et les traitements associés

Avec le nombre de données qui seront collectées et les traitements effectués dessus, il est vite apparu nécessaire de doter l'habitat intelligent d'un historique. Ainsi, l'ensemble des courbes et moyennes, et autres données récoltées dans l'habitat vont être stockées dans l'*historique* du système. Il va permettre, d'une part, de ne pas refaire des calculs déjà fait, et d'autre part, de fournir des données de références à long terme pour les différents modules d'analyse. Il va constituer une base de données que les médecins et le personnel autorisé pourront interroger à distance pour suivre l'évolution de leur patient.

Toutefois, vu l'importance de la masse de données que l'habitat va percevoir, il sera nécessaire d'effectuer un traitement sur les données de cet historique. Ce traitement va majoritairement résumer les données aux informations pertinentes. Dans la mesure du possible, on tentera de garder une partie de l'information de manière qualitative.

Les données contenues dans cet *historique* seront accessibles aux médecins et aux personnes autorisées. Il est évident que les mesures de sécurité informatique seront très fortes pour éviter que des données privées ne deviennent publiques. Les données accessibles seront différentes selon l'accédant. De façon générale, le module de *télésurveillance* pourra générer un rapport à la demande de l'accédant. La mise en forme de ce rapport sera, elle aussi, différente selon l'accédant ou selon la personne à laquelle il sera destiné. Ainsi, il sera nécessaire d'avoir une composante capable de remettre en forme les données issues

de l'historique. Un médecin recevra par exemple un rapport plus technique, contenant des courbes chiffrées, tandis que les proches recevront plutôt un rapport qualitatif, avec des courbes d'évolution indiquant la tendance générale de la santé de l'occupant. On peut aussi, par exemple, convenir d'un code couleur ou typographique avec les médecins, pour que les informations essentielles ressortent d'elles-mêmes dans les rapports.

Que ce soit la remise en forme de rapports, ou la concaténation des données pour l'*historique*, il s'agit de composantes qui ne font que du traitement et aucune analyse réelle. L'analyse aura été faite en amont ou en aval de ces composantes de traitement. Il est probable que l'aspect concaténation et l'aspect édition de rapports soient regroupés dans une seule et même composante liée à l'historique.

Nous avons identifié ici plusieurs composantes et fonctions supplémentaires, mais dans le cadre de la télésurveillance sur les données physiologiques. Nous allons maintenant nous intéresser aux données issues des capteurs et faire de l'analyse de comportement.

5.2.3 Télésurveillance des activités

La *reconnaissance et le suivi d'activités* va fournir les données nécessaire à la construction de courbes retraçant les activités de la personne au long de la journée. Ces courbes et celles enregistrées dans l'*historique* vont être comparées, afin de détecter d'éventuelles déviations du comportement sur le long terme. Pour réaliser ceci, l'idée est d'utiliser des techniques statistiques consistant à reconnaître des motifs de comportements dans les courbes (Duchêne et al., 2002). L'idée est de surveiller le comportement de la personne et de signaler des comportements anormaux (malgré les conseils du système) comme l'oubli de plusieurs repas, une mauvaise hygiène etc... Ce signalement s'effectuera de deux manières :

- des rapports périodiques, permettant de suivre en continu l'évolution de la personne ;
- des rapports occasionnels, qui seront motivés par des comportements anormaux ou des problèmes physiologiques qu'il faut signaler rapidement.

D'autre part, les informations données par le module de *reconnaissance* seront stockées, elles aussi, dans l'*historique*. Le système pourra ainsi envoyer des rapports contenant le nombre d'oublis ou d'abandons de tâches que la personne a pu faire. Le médecin pourra alors mieux se rendre compte si le comportement de la personne est normal, ou au contraire erratique ou dangereux. De la même manière, dès que le système interviendra, cette intervention sera consignée pour être en mesure de voir si globalement le nombre d'interventions du système a augmenté ou non. Ce sera aussi une façon de mesurer s'il y a dégradation de l'état de santé du patient ou non.

Pour résumer la composante télésurveillance enverra des rapports périodiques, par exemple toutes les semaines, à l'équipe médicale pour la tenir au courant de l'évolution de l'état de santé du patient, et de l'évolution de ces compétences cognitives. D'autre part, si la télésurveillance à long terme détecte une anomalie et que l'assistance cognitive ne peut pas y pallier (exemple, cela fait plusieurs fois que malgré les rappels la personne ne prend pas ses médicaments) alors un rapport sera envoyé au moment opportun. Ces rapports sont les «rapports occasionnels» déjà mentionnés. Nous allons maintenant nous intéresser au fonctionnement général du module d'*Assistance cognitive et de Télésurveillance*.

5.3 Un modèle de fonctionnement pour les composantes d'Assistance Cognitive et de Télésurveillance

Avant d'esquisser le fonctionnement global des composantes d'*Assistance Cognitive et Télésurveillance*, nous allons nous intéresser à l'aspect apprentissage et à l'identification des fonctions communes aux deux composantes.

5.3.1 Nécessité d'un apprentissage

Nous avons décrit un certain nombre de rôles et de fonctionnalités pour la composante d'*Assistance cognitive et de Télésurveillance*. Toutefois, ces composantes ne sont pas suffisantes. Il est nécessaire d'ajouter une composante chargée de l'apprentissage du système. En effet, si on se contente de mettre en place un système où tous les paramètres sont fixés au préalable, notamment les habitudes de vie et les compétences de la personne,

alors le système deviendra vite incapable de réagir correctement par rapport à son environnement. Par exemple, si la personne modifie ses habitudes alimentaires et ne prend plus de sucre dans son café le matin, le système va continuer à lui rappeler de mettre du sucre, alors que cela n'a plus lieu d'être. La capacité d'apprentissage est donc essentielle pour notre système.

Il faut distinguer deux phases d'apprentissage. D'une part, il va y avoir l'apprentissage initial, qui va consister à apprendre les comportements, habitudes et compétences de la personne. Cet apprentissage va en grande partie contribuer à l'initialisation du système. D'autre part, le système doit être également capable de s'adapter en tout temps à des changements de compétences cognitives ou physiques, d'habitudes de vie ou de préférences de la personne, tant que ces changements sont sans conséquence pour la santé du patient. Pour cela, nous avons introduit une composante apprentissage qui sera chargée de répercuter les changements de façon intelligente dans le modèle de la personne.

Il est nécessaire de recourir à un apprentissage intelligent, car il est évident qu'il ne faudra pas répercuter un changement dès qu'il se produira. Par exemple, si la personne se réveille habituellement vers 7h30, mais que ce matin pour une raison ou pour une autre elle s'est rendormie et réveillée à 8h30, il ne faudra pas répercuter dans le modèle de la personne un changement d'habitude de vie «elle se lève le matin à 8h30». Par contre, si la personne se lève à 8h30 depuis une semaine, le système peut décider le modifier les habitudes de vie. Ce simple exemple montre que l'apprentissage sera une composante assez complexe, qui devra prendre en compte un certain de nombre de facteurs avant de répercuter les changements constatés. L'apprentissage sera réalisé en fonction des tendances à long terme d'un ensemble de paramètres.

Les composantes qui pourront être modifiées par cet apprentissage seront essentiellement les habitudes de vie, les compétences et les préférences de la personne. Il s'agirait

- d'habitudes de vie globales (plage horaire des différentes activités, organisation générale) ;
- des compétences cognitives de la personne (mémoire, organisation...) ;
- des paramètres physiologiques tels que la vue, l'ouïe, la capacité motrice générale ;

- de préférences tels que les moyens d'interventions préférés, les activités favorites, la température de la maison etc...

5.3.2 Nécessité d'une interaction avec l'occupant

Dans les scénarios d'assistance envisagés jusqu'à présent, il n'est pas fait mention d'une interaction entre l'occupant et le système. Cependant, il semble évident que celle-ci sera nécessaire pour permettre des interventions par étapes (aide à la réalisation d'une activité) ou encore pour permettre à l'occupant de dire au système ses intentions. Par exemple, la personne prend sa douche en pleine nuit, le système lui signifie que ce n'est pas normal, mais la personne pourrait indiquer au système qu'elle doit se lever en pleine nuit, car elle part en voyage.

Cette assistance cognitive interactive est un aspect qui n'a pas été considéré dans la description qui va suivre. On considère uniquement des interventions simples ne nécessitant pas d'interaction avec l'occupant. Cependant, cette composante a été prise en compte lors de scénarios dans l'implémentation pour mettre en valeur de rôle de la *simulation de la personne.*

5.3.3 Identification précise des composantes

Dans les deux sections précédentes, nous avons identifié un certain nombre de rôles et de fonctions nécessaires aux composantes d'*assistance et de télésurveillance.* Or il existe des similarités permettant d'opérer des regroupements entre certaines composantes. En effet, soit elles partagent des rôles communs, soit elles utilisent les mêmes données.

Notamment, nous avons regroupé l'aspect traitement de données des activités et des paramètres physiologiques. En effet, nous avons exprimé dans les paragraphes précédents la nécessité d'avoir un traitement sur les données issues du module de reconnaissance pour établir des courbes d'activités. D'autre part, nous avons parlé du traitement effectué sur les données physiologiques, qui consiste aussi à les transformer en courbes et à retirer les informations pertinentes. Nous avons ensuite parlé de la nécessité de traiter les données de l'*historique*, de les concaténer, mais aussi de pouvoir les remettre en forme

pour envoyer des rapports lisibles et construits. Tous ces traitements présentés de manière indépendante utilisent les données de l'historique. Il semble donc assez logique de les regrouper dans une même composante qui portera le nom de *traitement de l'historique.*

De plus, nous avons montré qu'il fallait une composante capable d'analyser les résultats collectés dans le module de *reconnaissance et de suivi*, afin de permettre au système d'intervenir sur les actions déjà effectuées par l'occupant, et sur celles qu'il s'apprête à effectuer. Nous allons désigner cette composante par *Analyse et Formulation des interventions.* Cette composante s'occupe donc de déterminer s'il y a lieu d'intervenir, et la nature de cette intervention pour l'assistance cognitive. L'*analyse et la formulation* aura également recours à la composante de *simulation de la personne* comme aide à l'analyse. En effet, nous avons évoqué les aides sur les actions futures de la personne avec les informations fournies par la *reconnaissance des activités.* Or, l'analyse de la situation pourra être aidée par une simulation des activités futures de la personne, pour voir si celle-ci est susceptible de se tromper ou d'oublier une activité. Par exemple, la personne est en train de préparer son repas et le téléphone sonne. Elle va répondre. L'*analyse* demande alors une simulation de la personne sur la suite de cette action. La simulation montre qu'au vu des compétences cognitives de la personne, elle va oublier qu'elle préparait son repas auparavant. L'*analyse* décide donc de formuler une intervention lui rappelant qu'elle était en train de préparer un repas. La différence avec un scénario similaire décrit précédemment, est que le système anticipe ici les erreurs probables de l'occupant, au lieu d'attendre et de constater l'erreur.

Enfin, la *simulation de la personne* sera également utilisée par la composante d'*analyse*, afin de tester les interventions qu'elle a formulée. L'idée est de simuler la personne, mais en prenant en compte une intervention du système, pour voir si celle-ci est efficace. De cette manière, l'analyse éviterait de fournir des aides inefficaces à l'occupant.

Nous avons parlé de tous les aspects importants que nous devions retrouver dans notre composante *assistance cognitive/télésurveillance.* Nous avons parlé d'un module de *reconnaissance d'activité*, d'un module de *construction des épisodes*, d'un module de *traitement de l'historique.* Nous avons ensuite parlé du module de *simulation de la per-*

sonne qui sera l'objet du chapitre suivant, d'un module d'*analyse et formulation des interventions* qui va décider la nature d'une éventuelle intervention et enfin d'un module d'*apprentissage* qui agira sur l'instance du modèle de la personne. Dans le point suivant, nous allons voir comment toutes ces composantes s'organisent autour d'un modèle présentant le fonctionnement de la composante *assistance cognitive et télésurveillance.*

5.3.4 Organisation du module Assistance Cognitive - Télésurveillance

Le modèle de fonctionnement est présenté dans la figure 5.4 (page suivante). Les données représentées sur cette figure sont celles qui sont collectées par le système, qu'elles proviennent de l'habitat ou de l'extérieur. Un premier traitement est effectué sur ces données, qui, par la suite, sont utilisées de diverses manières selon leur nature. Ainsi, les données issues des capteurs de l'habitat sont envoyées à la composante de reconnaissance des activités, les données physiologiques sont, quant à elles, directement envoyées à la composante de *construction et répartition.*

Sur la figure 5.4, est représenté le cycle dont nous avons parlé pour le module de *reconnaissance.* La composante de *construction et répartition* s'occupe de construire des épisodes, d'envoyer les activités reconnues à la composantes d'analyse, d'envoyer les données physiologiques et celles sur les activités à l'*historique.* L'*analyse* s'occupe de déterminer si une intervention est nécessaire, et comment elle sera faite. Notamment, l'intervention peut être du type assistance cognitive, quand l'intervention se fait dans l'habitat et à destination de l'occupant, ou être du type télésurveillance quand l'intervention doit atteindre l'environnement extérieur à l'habitat. En cas d'intervention du type télésurveillance, le module d'*analyse* envoie les informations nécessaires au module de *traitement de l'historique* qui va s'occuper de mettre en forme le rapport ou le message. Le module d'*analyse* peut avoir recours aux informations contenues dans l'*historique* et dans l'instance du modèle de la personne pour son analyse. Enfin, le module de *simulation de la personne* envoie lui aussi des informations à ce module comme aide à l'analyse.

L'*apprentissage*, pour sa part, se fera à partir de données issues du module d'*analyse*

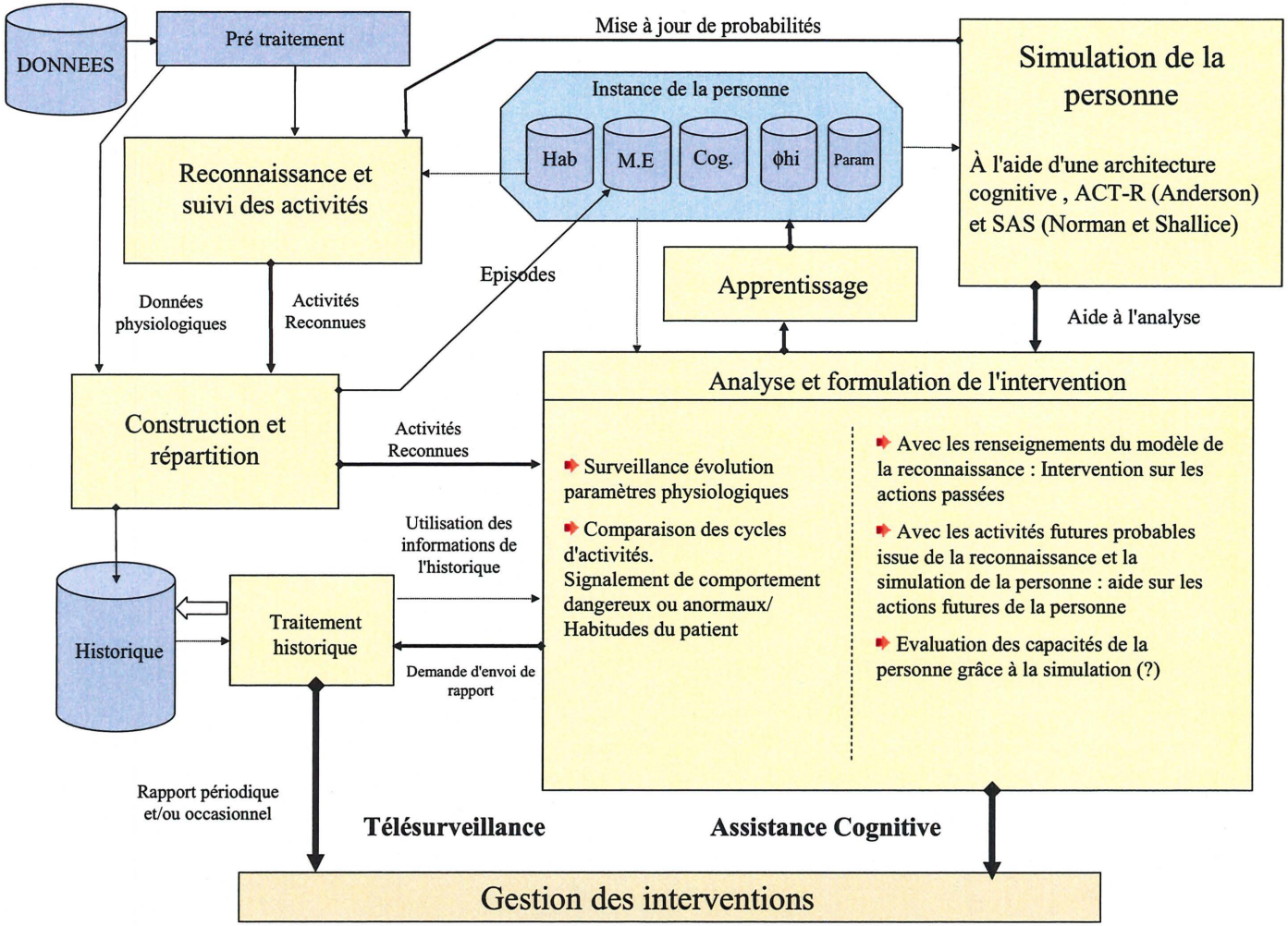


FIG. 5.4 – Un modèle pour l'Assistance Cognitive et la Télésurveillance

et formulation des interventions. En effet, ce n'est qu'après l'analyse des données que les changements d'habitudes ou de préférences auront été détectées. C'est donc le module d'*analyse* qui enverra à la composante *apprentissage* les données qui auront potentiellement à être apprises. L'apprentissage pourra aussi utiliser l'*historique* selon ses besoins, mais nous n'avons pas fait figurer ce lien sur le schéma pour plus de clarté. La composante *apprentissage* modifiera l'instance du modèle de la personne quand cela sera nécessaire, pour permettre au système de s'adapter aux évolutions de l'occupant.

Nous avons ainsi décrit un modèle de fonctionnement de la composante d'*Assistance Cognitive et de Télésurveillance.* Cette architecture est basée sur un certain nombre d'hypothèses et de suppositions, notamment sur la réalisation de certains modules. Il est possible que le fonctionnement de certains modules soit modifié. Ce n'est qu'en testant et en développant qu'il sera possible de s'en rendre compte. Dans la section suivante, nous allons expliquer quels sont les problèmes qui pourraient être rencontrés lors du développement de certaines composantes.

5.4 Problèmes et points à développer

En présentant cette architecture, il est vite apparu que cette vision des choses était plutôt optimiste. Le système ne disposera pas de toutes les informations dont nous avons besoin. Ceci risque d'engendrer de nombreux problèmes, et nous allons essayer ici d'en donner un aperçu.

5.4.1 Incertitude des données

Dans notre présentation d'une structure pour les composantes d'*assistance cognitive et de télésurveillance*, nous n'avons pas fait mention de la difficulté liée à l'incertitude des données qui seront collectées dans l'habitat. D'une part, le nombre de capteurs proposés dans le plan ne sera pas suffisant pour prétendre pouvoir suivre les activités de la personne en détail. D'autre part, les capteurs vont être très bruités, et il sera donc difficile d'en extraire une information significative de façon aussi détaillée. Par exemple, nous serons quasiment incapables de savoir quels types d'aliments la personne mange

quand elle cuisinera, ou même la quantité de ces aliments, ou encore de vérifier qu'elle nettoie bien son intérieur. Les exemples de situations où nous ne pourrons rien affirmer ne manquent pas.

L'incertitude liée au bruit des capteurs ajoute encore une autre dimension d'incertitude au système. En effet, la reconnaissance des activités atomiques ou opérations, quand elle est possible sera incertaine. Pour une séquence d'activation de capteurs donnée, nous aurons plusieurs possibilités d'activités atomiques à cause du bruit, et nous ne pourrons donc donner que des probabilités d'actions atomiques. Par exemple, si on détecte une entrée dans la chambre, le passage de la personne vers la salle de bains et rien après, on peut dire qu'elle est soit restée dans la chambre devant un placard, soit entrée dans la salle de bain, mais on ne l'a pas détectée. Ce n'est qu'après un certain temps (par exemple si la personne bouge dans la salle de bains) que nous pourrons confirmer ou infirmer l'une ou l'autre des hypothèses. Cette incertitude au niveau des actions atomiques va en entraîner d'autres à plus haut niveau. Nous ne pourrons plus être sûr de détecter une activité dans son ensemble. Il sera donc très important de prendre soigneusement les décisions d'assistance cognitive.

De façon générale, l'incertitude des données collectées dans l'habitat va être le plus gros problème à résoudre et à gérer. Il serait en effet utopique de penser que nous pourrions collecter des données sans incertitude. Toutefois, on peut imaginer par des recoupements successifs augmenter la confiance en certaines des données collectées, et ainsi obtenir des données presque certaines. Les composantes de ce système devront donc être conçues pour manipuler des connaissances avec des degrés de certitude.

5.4.2 Initialisation

Nous avons déjà soulevé le problème de l'initialisation dans d'autres sections, notamment en parlant du module de *gestion des interventions*. Ce problème va être d'autant plus crucial pour cette composante. En effet, nous avons déjà mentionné en parlant de la *reconnaissance des activités*, la difficulté qu'il y aura à initialiser le système, du point de vue de la mémoire épisodique, et du point de vue des probabilités de l'automate hiérarchique. La difficulté est liée à l'initialisation du modèle de la personne qui devra être

très poussée.

La solution actuelle à ce problème sera de récolter le maximum de données sur l'occupant potentiel de l'habitat, puis de les mettre en place dans le système. Ceci serait complété par une phase d'apprentissage plus ou moins longue des habitudes de vie de la personne, de ses capacités cognitives et de ses préférences. On soulève ici un autre problème, comment le système est supposé apprendre les comportements normaux de la personne en question, si celle-ci est déjà suffisamment malade pour avoir des comportements incohérents ? On présuppose peut-être trop ici que le système sera capable sans apprentissage de détecter les comportements normaux ou non. Il faut en effet bien voir que dans certains cas, la différence entre les deux est très faible. L'exemple d'une personne qui s'habille correctement et de celle qui s'habille avec une culotte par-dessus le pantalon est parlant. Le système ne peut tout simplement pas détecter l'un ou l'autre des comportements comme normaux ou pas. Il peut juste dire que la personne s'est habillée. Il faut donc avoir constaté au préalable (constatation des proches) que la personne peut avoir des problèmes pour s'habiller auquel cas ces données seront entrées manuellement dans le système. On voit ainsi qu'il sera nécessaire de recueillir le plus de renseignements possibles sur l'occupant.

La facilité d'initialisation va donc aussi dépendre de la certitude que nous aurons sur les données collectées dans l'habitat. Plus les données que nous pourrons détecter seront certaines, plus nous pourrons facilement effectuer la phase d'apprentissage initial. Il est vraisemblable toutefois que l'occupant aura à fournir lui-même ou aux techniciens un nombre non négligeable de connaissances le concernant pour que le système puisse commencer cette phase d'apprentissage correctement.

Dans ce chapitre, nous avons présenté un modèle pour la composante d'*Assistance cognitive et de Télésurveillance à long terme.* Nous avons identifié différentes composantes notamment la composante de *simulation de la personne.* Un modèle utilisant ACT-R - SAS pour cette composante est présenté dans le chapitre suivant. Nous ne détaillerons pas dans ce mémoire les autres composantes présentées. Il s'agissait ici de mettre en place

l'aspect cognitif de l'habitat intelligent. Aussi, nous avons décidé de concentrer nos efforts sur la simulation de la personne à l'aide d'une architecture cognitive.

Chapitre 6

Simulation de la personne : une nouvelle proposition d'architecture pour la cognition humaine

D'après Glasspool (2000),

> «Les théories en psychologie cognitive convergent vers une certaine forme d'architecture cognitive comprenant un système de contrôle organisé en couche. Les niveaux bas comprenant de multiples processus simples décrivant le comportement et les couches plus hautes caractérisées par des processus délibératifs plus lents qui exercent un contrôle global».

L'architecture cognitive proposée ici pour simuler les activités d'une personne atteinte de troubles cognitifs suit également ce principe. Au vu des différentes théories et architectures présentées dans la première partie, l'idée est d'utiliser ACT-R pour le contrôle des processus simples couplé au SAS pour l'aspect délibératif. Cette hybridation et son intégration au sein du système nécessite évidemment des adaptations. Dans ce chapitre, nous expliquerons le choix de cette solution et les adaptations apportées aux théories et architectures utilisées, puis nous aborderons le fonctionnement de l'architecture hybride obtenue.

6.1 Vers une hybridation ACT-R - SAS

Dans la première partie de ce mémoire, a été effectuée une revue plus ou moins exhaustive des théories et des architectures cognitives existantes. Il est apparu très vite que des théories comme celle de Baddeley (1986) ou de Potter (1990) sont grandement insuffisantes pour permettre de simuler une personne atteinte de troubles cognitifs. Elles ne couvrent en effet qu'un seul aspect de la cognition : la mémoire de travail. Toutefois, la théorie de Potter a permis de montrer qu'il est nécessaire d'introduire une forme d'attention dans la composante chargée de la simulation.

En ce qui concerne la théorie de Norman et Shallice (1986), nous avons déjà montré que le module de Contention Scheduling n'est pas assez robuste pour permettre de simuler les activités routinières d'une personne atteinte de troubles cognitifs. Toutefois, nous voulions garder l'idée d'architecture en couches, avec un système qui s'occupe des activités routinières et un système superviseur. Cette approche a été peu reprise, si ce n'est par ceux qui travaillent sur le SAS. En effet, dans les architectures que nous avons présentées, aucune, hormis EPIC, ne fait état d'un système superviseur. Cependant dans EPIC, il ne s'agit pas d'un processus à part entière, mais uniquement de règles chargées d'ordonnancer les tâches exécutées en parallèle. Pourtant, il apparaît important de faire intervenir un processus superviseur pour simuler l'attention du sujet par rapport à son environnement, et sa capacité à trouver des solutions nouvelles à des problèmes rencontrés dans la vie courante. Que ce soit SOAR, EPIC ou ACT-R, aucune architecture cognitive ne prévoit un processus de création. Les résolutions de problèmes se font de manière locale à la manière de SOAR, ou encore sont inexistantes dans ACT-R ou EPIC. Il faut bien voir que, dès que l'on sort de l'ensemble des règles et connaissances prévues et incluses par le programmeur, le système est incapable de réagir.

Nous avons donc décidé de garder l'idée CS/SAS. Les deux problèmes que posait alors ce choix étaient premièrement de trouver un remplaçant pour le CS de Norman et Shallice, et deuxièmement, de proposer une spécification du SAS. En effet, la théorie du SAS n'est pas suffisante pour donner lieu à une implémentation, il fallait donc développer cette théorie, la spécifier ou du moins trouver une solution pour l'habitat intelligent. Les

réponses que nous avons apportées à ces deux problèmes, sont présentées dans les deux points suivants.

6.1.1 Un candidat pour le Contention Scheduling

Pour remplacer le Contention Scheduling, il fallait trouver un système robuste, capable de simuler les troubles cognitifs d'une personne, et capable de simuler le déroulement d'activités quotidiennes. Dès lors, nous avons examiné les diverses architectures cognitives proposées dans la littérature. Nous avions alors le choix entre SOAR, EPIC et ACT-R qui ont toutes trois une version informatique.

Nous avons vite éliminé SOAR des possibilités, car comme cela a été souligné dans la première partie, sa représentation des connaissances respecte peu les contraintes de la psychologie cognitive. Il est difficile de modéliser le comportement d'une personne et sa mémoire épisodique à l'aide de SOAR. Or, la mémoire épisodique va jouer un rôle très important pour la création de solutions au niveau du processus superviseur, c'est en effet la plupart du temps grâce à son expérience et à son vécu que l'être humain est capable de s'adapter et de réagir aux situations nouvelles. Il est donc important que le processus superviseur puisse facilement y accéder et la représenter. Ce n'est pas le cas dans SOAR, les connaissances épisodiques ne sont pas facilement modélisables. D'autre part, un autre écueil de SOAR est l'accès sans limitation à la mémoire à long terme. Ceci va à l'encontre du fonctionnement du cerveau humain, qui a tendance à oublier. Ceci est d'autant plus vrai pour une personne atteinte de troubles cognitifs.

Le point positif d'EPIC par rapport à SOAR est que d'un côté, il y a bien un accès limité à la mémoire de travail grâce à un processus de détérioration. Mais d'un autre coté, la mémoire à long terme semble là encore accessible à tout moment. De plus, EPIC est peu adapté à la simulation d'une personne. Cette architecture est prévue pour simuler des tâches concourantes, mais le programmeur doit préciser lui-même la manière dont vont se superposer les tâches, et tout ce qui concerne les processus de contrôle, c'est-à-dire le début et la fin d'une tâche... Pour la spécification d'une ou deux tâches, EPIC semble adapté. Mais dès lors qu'il s'agirait de faire ce travail pour l'ensemble des tâches d'une journée, cela apparaît vite impossible, d'autant que les auteurs ne proposent pas

de processus de contrôle génériques, il faut donc les spécifier pour chacune des tâches. De plus, il ne semble pas évident que l'on puisse simuler les erreurs d'une personne présentant des troubles cognitifs, car les mécanismes gouvernant la mémoire de travail et moteur d'inférence sont très simples. Par exemple, il est difficile de simuler une erreur d'anticipation dans l'action alors que les règles de productions sont exécutées seulement lorsque leurs conditions sont satisfaites dans la mémoire de travail d'EPIC. Les objectifs d'EPIC ne sont pas de simuler une personne atteinte de troubles cognitifs, mais de simuler l'exécution de tâches en parallèle pour le test d'interfaces homme-machine. EPIC ne semble donc pas être un bon candidat pour jouer le rôle de Contention Scheduling. Cependant, cette architecture ne doit pas être rejetée totalement car elle possède un aspect sensori-moteur très développé.

Il reste donc ACT-R. Cette architecture a l'avantage par rapport aux deux autres, d'être très complète et facilement implantable. D'autre part, les fondements psychologiques et théoriques d'ACT-R sont très forts, il a été utilisé et testé de nombreuses fois, dans de nombreux types de systèmes. ACT-R semble être l'architecture la mieux adaptée pour fournir un système de Contention Scheduling fiable. Toutefois, il faudra certainement adapter cette architecture pour pouvoir la faire fonctionner avec un système de supervision et intégrer une simulation du contrôle sensorimoteur aussi complet que celui d'EPIC. Mais ces modifications sont minimes, elles sont détaillées dans la section 6.3.1 page 132.

6.1.2 Vers une spécification du SAS ?

Avant de tenter de répondre au problème de spécification du SAS, nous allons préciser son rôle. En effet, la *simulation* doit être capable, d'une part, de simuler la personne avec ses troubles cognitifs, et d'autre part, d'éventuellement fournir une solution valide (exempte d'erreurs) pour guider l'occupant dans la réalisation de ces activités. Dans les deux cas, le fonctionnement du SAS est le même, ce qui diffère sont les paramètres qui lui seront fournis. En effet, pour obtenir la «bonne solution», il faudra donner au SAS les paramètres représentant une personne saine et valide. Pour simuler la personne atteinte de troubles, il faudra lui donner les paramètres correspondant à ses compétences cognitives.

Comme nous l'avons constaté, la théorie du SAS est insuffisante pour permettre une implémentation quelconque. Il est donc nécessaire de la spécifier, pour cela, il y a plusieurs approches. En effet, Glasspool et Cooper (2002) ont essayé d'implanter des versions simples du SAS à l'aide de COGENT. Cette approche part du particulier pour aller vers le général. Elle permet d'avoir une version concrète et utilisable du SAS. La seule objection à cette approche, mais qui est de taille, c'est qu'ils risquent de ne pas pouvoir généraliser suffisamment, de rester dans le particulier, et ainsi obtenir un SAS qui sera toujours dépendant des types de test ou d'activités cognitives, que l'on voudrait le voir réaliser. Ainsi le choix s'est porté sur une autre approche qui consiste à rester dans la voie théorique, en suggérant des pistes de réalisations pour certains modules.

Toutefois, ceci ne règle pas le problème : pour l'habitat intelligent, il faut spécifier le SAS de façon à ce qu'il soit implémentable. Toute la difficulté vient du processus de création d'une solution. Comment reproduire le processus mental humain de création ? L'idée est de contourner cette difficulté en supposant que pour la simulation d'une personne saine, le système disposerait de toutes les solutions valides, c'est-à-dire de toutes les activités et toutes les manières de faire dites «normales». Cette solution ad hoc apporte de nouvelles difficultés, notamment, on peut discuter de la complexité spatiale d'une telle solution (comment stocker autant de données ?), et aussi de la faisabilité en termes combinatoires (est ce vraiment possible de tout énumérer ?). Cette solution soulève donc d'autres problèmes. Toutefois, il sera possible au prix d'un certain nombre d'heures de travail et d'essais d'y répondre facilement, ce qui n'était pas le cas lorsqu'il s'agissait de construire un processus de création de solutions.

Par contre en ce qui concerne la simulation d'une personne atteinte de troubles cognitifs, trouver une solution alternative pour le SAS est moins évident. Il est, en effet, hors de question d'énumérer l'ensemble des comportements possibles. Pour donner une idée, on a déjà vu des personnes tenter de se raser avec un biscuit ! Une solution pourrait être de simuler des comportements déviants à partir des comportements normaux. Cependant, ceci ne serait valide que pour des personnes dont les troubles cognitifs ne sont pas trop importants.

Ainsi, il s'offre à nous deux possibilités. La première qui consiste à poursuivre l'idéal du SAS capable de superviser et de créer de nouvelles solutions, est une voie de recherche en psychologie cognitive. La seconde plus pragmatique, qui consiste à tout énumérer pour la simulation d'une personne saine, est une solution probable pour le système de l'habitat intelligent en l'état actuel de la recherche. Nous parlerons donc de SAS pour désigner le SAS de manière générale et de «SAS - Planification» pour désigner la solution d'énumération.

Les premières réponses aux problèmes soulevés ont été présentées. Elles consistent à créer une architecture hybride utilisant ACT-R et SAS. Dans la section suivante, nous allons détailler un modèle de *simulation de la personne* afin de le relier aux composantes présentées dans le précédent chapitre. Par la suite, nous aborderons le fonctionnement de l'architecture cognitive proprement dite.

6.2 Modèle de simulation de la personne

Dans cette section, est présenté un modèle de *simulation de la personne*. Le cœur de ce module qui est l'hybridation de ACT-R et de SAS, est le sujet des sections suivantes. Nous nous contentons ici de faire la connexion de ce module avec le niveau supérieur de l'architecture.

6.2.1 Rôle du module de simulation

Dans le chapitre précédent, nous avons montré qu'il est nécessaire que le système soit capable de simuler les activités de la personne dans l'environnement. Il a donc été muni d'une composante chargée de la simulation. Plus précisément, le module de *simulation* doit permettre de fournir les probabilités d'actions futures de la personne au module de *reconnaissance des activités*. Pour cela, ACT-R et SAS doivent avoir accès à la mémoire épisodique de l'individu, mais aussi à différents paramètres contenus dans le modèle de la personne, comme les capacités physiques et cognitives, ainsi que les habitudes de vie. Nous verrons plus tard plus en détail comment ces paramètres interviennent.

De plus, le module de *simulation* va pouvoir donner des informations pour aider à l'analyse de la situation. Ces informations pourront concerner des erreurs cognitives probables dans la suite des actions, ou bien l'efficacité de telle ou telle intervention.

6.2.2 La couche externe du module de simulation

Suite aux constatations du paragraphe précédent, le module de *simulation* interagit avec l'instance du modèle de la personne, le module de *reconnaissance des activités* et le module d'*analyse et formulation de l'intervention.* Nous avons représenté ces interactions dans le schéma de la figure 6.1.

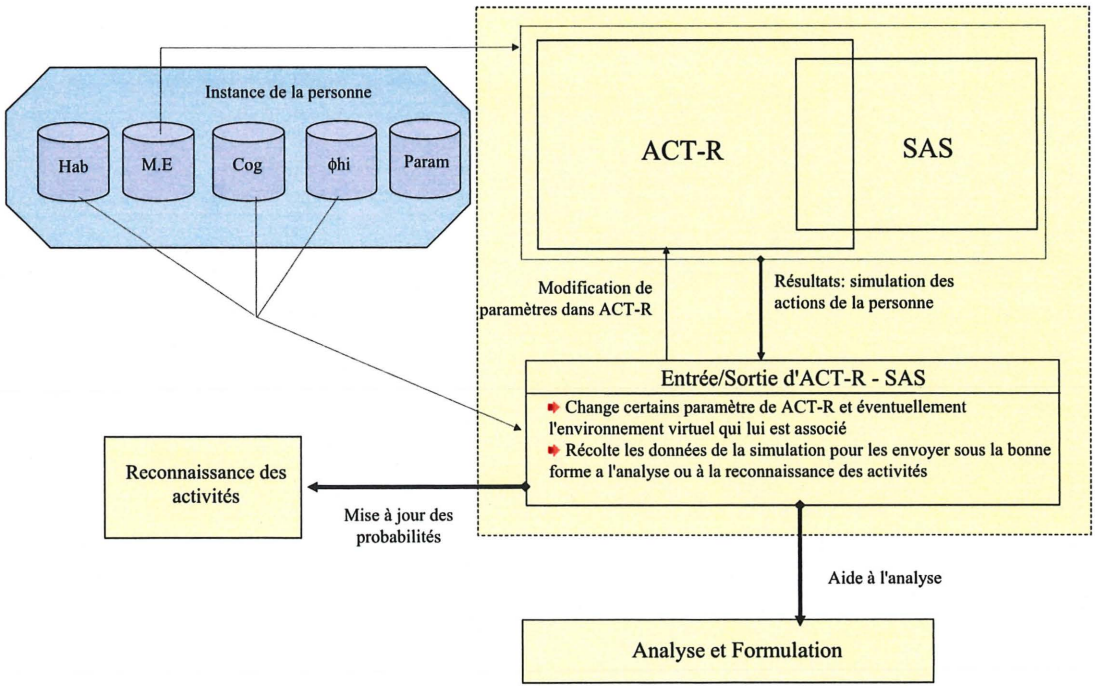


FIG. 6.1 – Vue générale du module de simulation de la personne

On peut remarquer qu'en plus d'ACT-R - SAS proprement dit, une composante de *gestion des entrées/sorties* a été ajoutée. Elle va permettre d'injecter dans le module de simulation les paramètres nécessaires à son fonctionnement, mais également de collecter les résultats. Cette composante va constituer l'interface entre l'architecture cognitive hybride et le reste du système. La seule exception est la composante de la mémoire épisodique qui est directement accessible à ACT-R - SAS, car il s'agit d'une composante de l'architecture hybride.

Maintenant que nous avons brièvement décrit la couche externe du module de *simulation*, nous allons décrire le fonctionnement et l'hybridation de ACT-R - SAS et notamment des adaptations de ACT-R et de SAS.

6.3 Fonctionnement de ACT-R - SAS

Dans cette section, nous présentons les adaptations effectuées sur l'architecture d'ACT-R et la théorie du SAS, puis nous expliquerons le fonctionnement global du système et ses interactions.

6.3.1 Adaptations effectuées sur ACT-R

Le fonctionnement de l'architecture ACT-R d'Anderson et al. (2002) a été présenté dans la section 3.3 page 72, figure 3.2 page 75. Par rapport à ce schéma de fonctionnement, nous avons dû procéder à des adaptations minimes. Tout d'abord, il fallait faire ressortir les types de connaissances utilisées dans l'architecture. Ainsi, les connaissances procédurales, les connaissances sémantiques et épisodiques, et les buts qui sont des connaissances sémantiques particulières, ont été mises en avant. On rappelle que les connaissances déclaratives se décomposent en deux connaissances : la connaissance sémantique qui correspond à des faits généraux et la connaissance épisodique qui correspond au vécu. Le noyau central d'ACT-R manipule des connaissances procédurales sous forme de règles de productions, dont l'une est sélectionnée puis exécutée. Dorénavant, les connaissances procédurales désigneront «une façon de faire» en général et les règles de production, les entités manipulées par le noyau d'ACT-R. Le module d'intentions manipule des buts et

le module déclaratif manipule des connaissances sémantiques et des connaissances issues de la mémoire épisodique. Ces quatre types de connaissances apparaissent sur le schéma de la figure 6.2 (page suivante).

Cette mise en relief des divers types de connaissances était nécessaire pour pouvoir utiliser ACT-R comme un Contention Scheduling, et faire le lien avec le modèle de la personne. Notamment, le fait de représenter la mémoire épisodique séparément, permet de l'inclure dans le modèle de la personne. Ainsi, la mémoire épisodique de la personne et celle de ACT-R modifié sont les mêmes. L'accès à la mémoire épisodique fournit un moyen de calculer les valeurs d'activation des connaissances dans la mémoire déclarative, et l'utilité des règles de production dans le noyau d'ACT-R. En effet, à partir de la mémoire épisodique, on peut obtenir le nombre d'utilisations d'une connaissance, sa récence... , autant de paramètres qui entrent dans le calcul de l'activation ou de l'utilité. Il faut rappeler que les connaissances de la mémoire épisodique sont construites à l'aide des connaissances sémantiques, procédurales et des buts (cf. figure 5.2 page 108). Sur la figure 6.2 (page suivante), ces liens ne sont pas représentés pour plus de lisibilité.

D'autre part, le monde extérieur pour ACT-R va être la représentation de l'habitat de façon virtuelle, où le système sera capable de localiser différents objets avec précision. Le module de perception va percevoir les objets placés dans cet environnement et le système moteur va agir sur cet environnement, virtuel, c'est-à-dire simuler les actions de la personne sur son environnement. Il est vraisemblable que nous garderons une trace du contenu des différents tampons pour suivre le processus de simulation de la personne précisément, et envoyer un certain nombre de résultats aux autres composantes.

D'un point de vue global, les mécanismes d'ACT-R n'ont pas changé. Ils ont été testés de nombreuses fois et validés d'un point de vue psychologique. Les changements ne sont que des changements de représentation, le principe et les calculs mathématiques associés restent les mêmes. Cette séparation des connaissances par rapport au système va permettre d'y greffer le SAS. Cet aspect sera présenté dans les paragraphes suivants.

Nous allons maintenant parler des adaptations et changements apportés à la théorie

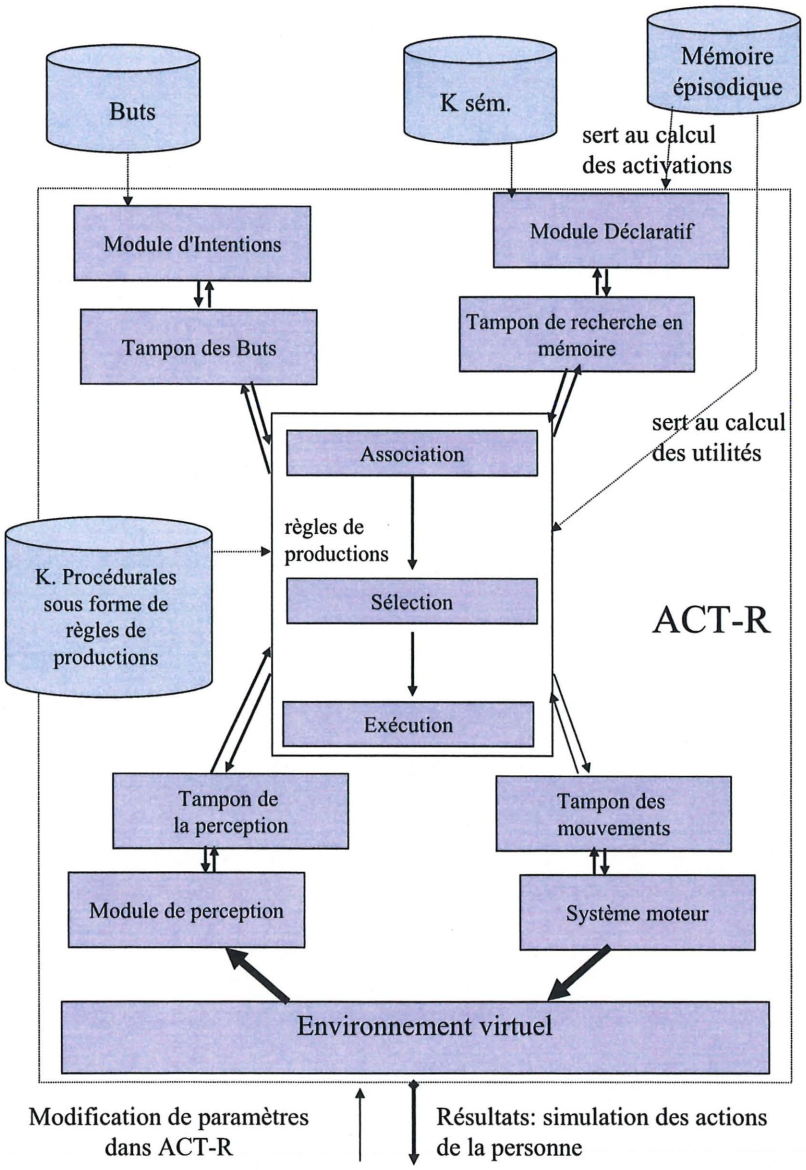


FIG. 6.2 – L'architecture d'ACT-R adaptée d'après ACT-R d'Anderson et al. (2002)

du SAS. Par la suite, nous nous intéresserons au fonctionnement global de l'architecture cognitive : ACT-R - SAS.

6.3.2 Adaptation du SAS

Dans cette section, nous allons parler des adaptations faites à partir du schéma de Shallice et Burgess (1996) (cf. figure 2.6 page 53). Pour adapter la théorie du SAS à ACT-R, il était nécessaire de modifier la notion de schéma. Les schémas ont donc été remplacés par des règles de production. Toutefois, nous avons gardé le terme générique de solution, car le SAS peut produire une solution simple c'est-à-dire une règle de production ou un plan c'est-à-dire une stratégie qui se présente sous la forme d'une connaissance sémantique décrivant les étapes de ce plan. Lorsqu'il s'agit d'un plan, la connaissance sémantique formée est considérée comme un ensemble d'intentions.

Par rapport au fonctionnement proposé par Shallice et Burgess (1996), la première étape a été conservée et renommée «Élaboration d'une solution». Dans le schéma de fonctionnement de la figure 6.3 page 137, le déroulement de ce processus n'est pas précisé contrairement au schéma de la figure 2.6 page 53. Ces précisions seront abordées ultérieurement.

Ensuite dans le schéma originel, il y avait deux étapes : une étape d'implémentation de la solution temporaire et une étape de test. L'ordre dans lequel ces deux phases se déroulent est discutable. En effet, d'une part, en cas de rejet de la solution, il n'apparaît pas judicieux d'avoir «implémenté» la solution auparavant, d'autre part, les types de solutions générés ne sont pas identiques. Le test ou l'implémentation d'une stratégie ou d'une solution simple ne seront certainement pas les mêmes. Aussi sur le schéma de fonctionnement du SAS, la phase de test donne lieu, soit à une stratégie, soit à une procédure simple en cas de succès, ou bien, en cas de rejet, le processus d'élaboration d'une solution est relancé (cf. figure 6.3 page 137). La phase de test correspond à vérifier mentalement la validité d'une solution, c'est-à-dire à simuler mentalement son exécution. Pour arriver à cela, le seul moyen à disposition est ACT-R. Le SAS utiliserait alors ACT-R dans sa phase de test pour vérifier la validité d'une solution. La phase d'implémentation fait alors suite à la validation de la solution ou non.

Dans la phase d'élaboration, le SAS peut avoir recours à des informations de la mémoire épisodique, conformément à la figure 2.6 page 53, qui inclut le processus 8. On a déjà justifié l'existence de cet accès auparavant, en expliquant que le processus de création peut faire appel à l'expérience et au vécu de la personne. Le SAS peut également modifier les intentions (buts futurs) au sein du module de Contention Scheduling, lorsque la résolution de la situation posant problème est repoussée dans le temps. On fait ainsi le lien avec les marqueurs d'intentions présentés dans la figure 2.6 page 53.

Dans le cas du SAS - Planification, nous avons expliqué que celui-ci accéderait à une base de connaissances génériques (représentées sous forme de règle de production ou sous forme de connaissances sémantiques décrivant des plans), contenant toutes les connaissances nécessaires pour réaliser des activités de la vie quotidienne. Le cylindre «K. génériques» du schéma 6.3 (page suivante) représente cette base générique. Elle est distincte des connaissances procédurales de la personne, représentées sur le schéma de ACT-R. Les connaissances génériques sont utilisées par le SAS - Planification et celles de la personne par ACT-R.

Nous avons parlé du cas où il n'y pas de solution possible et de l'intervention du SAS pour la création d'une solution. Toutefois, ceci ne constitue pas la seule intervention du SAS. Il ne faut pas oublier que le SAS peut jouer un rôle de superviseur, sans toutefois créer de nouvelles solutions. Il s'agit des cas de :

- situations où des erreurs doivent être corrigées ;
- situations dangereuses ou techniquement difficiles ;
- situations qui requièrent la modification d'habitudes bien acquises ou de la résistance face à une tentation.

Dans ces trois cas, il s'agit uniquement d'un effort d'attention et de concentration. Sur la figure 6.3 (page suivante), est donc représenté un module d'attention qui serait chargé de surveiller l'exécution des tâches au sein d'ACT-R. Ce module pourrait permettre à terme de simuler certaines erreurs cognitives en jouant sur les paramètres d'activation et d'utilité d'ACT-R, mais aussi de pouvoir simuler des situations d'attention, de concen-

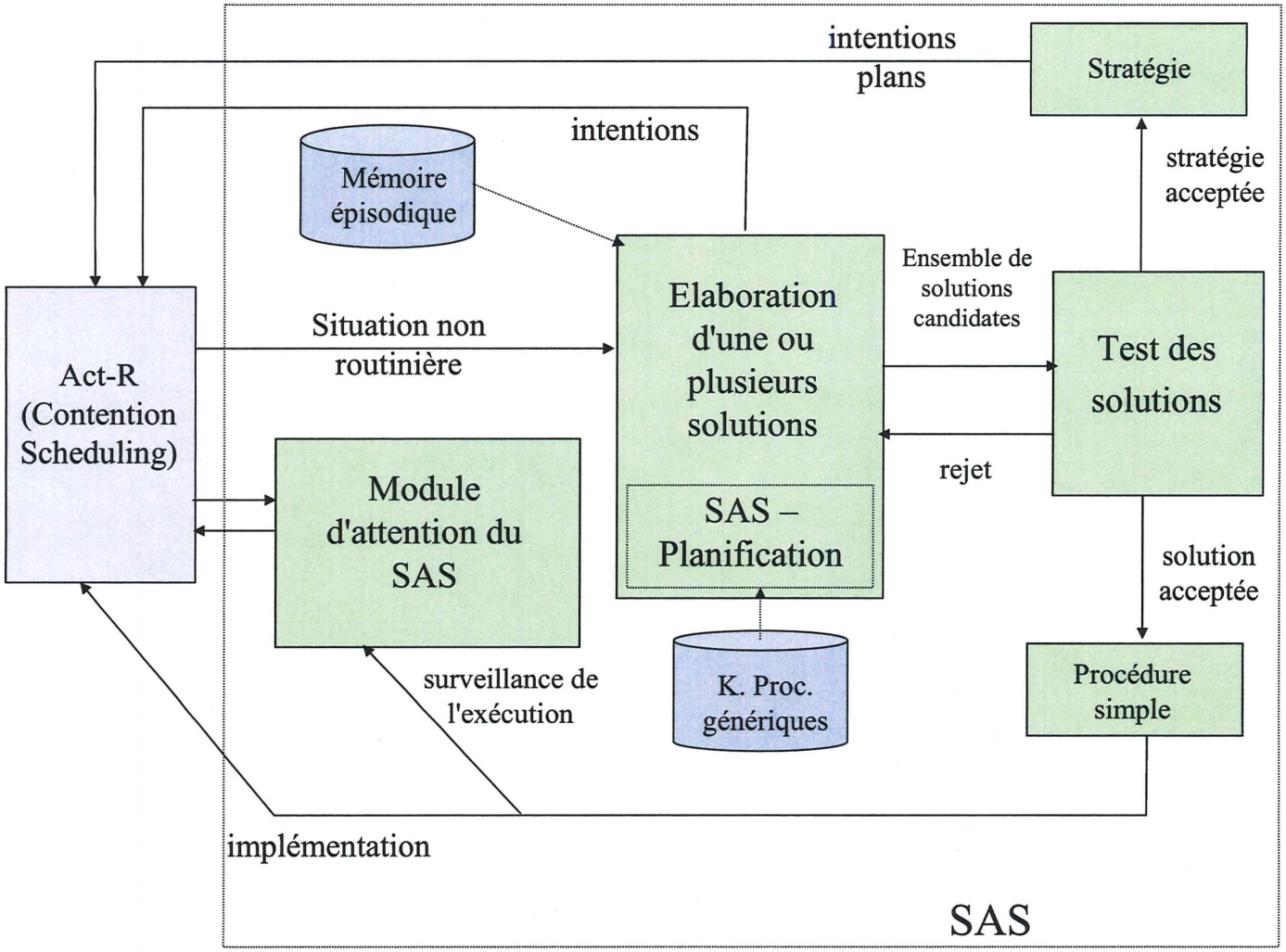


FIG. 6.3 – Proposition d'architecture pour le SAS

tration. Nous verrons plus en détail comment lorsque nous aborderons le fonctionnement global.

Après la phase de test, la procédure simple va être implémentée (ou simplement confirmée) dans ACT-R, et son exécution va être surveillée via le module d'attention du SAS. Même si la solution a été testée[5], on surveille son exécution pour simuler la concentration qui est associée à la réalisation d'une nouvelle tâche.

Dans le cas d'une stratégie, plans et intentions doivent être envoyés à ACT-R et certaines connaissances procédurales devront être implémentées puis exécutées. Nous allons voir comment tout ceci va être possible dans le paragraphe suivant qui présente le fonctionnement global de ACT-R et SAS.

6.3.3 Fonctionnement global

Dans les paragraphes précédents, nous avons décrit les modifications apportées à ACT-R et à SAS. Cette section va décrire l'intégration d'ACT-R et SAS. A l'heure actuelle, dans le processus central d'ACT-R, c'est-à-dire Association, Sélection, Exécution, rien n'est prévu lorsque la phase d'association échoue. La phase d'association consiste à faire la liste de toutes les règles de production dont les conditions sont remplies dans l'environnement courant. L'environnement courant est l'ensemble des connaissances contenues dans les tampons du système. En résumé, il s'agit de repérer toutes les «actions» possibles. Si on considère qu'ACT-R correspond à un système d'activités routinières, alors un échec de la phase d'association correspond à une situation nouvelle face à laquelle l'individu n'a pas encore acquis des connaissances procédurales et sémantiques routinières.

La question qui se pose alors est pourquoi jusqu'à présent rien n'a été prévu ? Dans les systèmes qui sont implémentés avec ACT-R, les programmeurs se chargent de remplir eux-mêmes les divers types de connaissances dont le système a besoin (procédurales,

[5]Dans notre cas, la surveillance de l'exécution et le test se feront de la même manière à l'aide d'ACT-R, pourtant sémantiquement les deux processus sont différents. Cette similitude vient du fait qu'on simule une personne et qu'on effectue pas manière réelle les actions

déclaratives). Donc la plupart du temps, ces connaissances sont entrées dans le système de manière à ce qu'il y ait toujours une règle à exécuter, et s'il n'en reste aucune c'est que ce qui devait être réalisé est terminé. Ceci marche tant que l'on cherche à simuler une suite d'actions particulières. Mais dans notre cas, on cherche à simuler le comportement d'une personne tout au long de la journée. Il va donc être nécessaire de prendre en compte le cas où il n'y a pas d'association possible[6].

L'idée est donc de recourir au SAS lorsqu'une situation nouvelle se présente, c'est-à-dire quand il n'y a pas d'association possible dans le contexte courant. Dès lors, on cherche à créer une solution permettant de réaliser le but courant et respectant les conditions formées par les différents éléments dans les tampons. La figure 6.4 (page suivante) reproduit côte à côte le schéma proposé pour ACT-R[7] et celui pour SAS. Cet appel au SAS est représenté sur cette figure par la flèche «pas d'association possible» partant de la phase d'association d'ACT-R vers la composante d'élaboration d'une solution du SAS.

Dans le paragraphe précédent, il a été souligné que le SAS pouvait changer les intentions dans ACT-R. Ce fait est représenté par une flèche qui part du SAS et qui va vers le module d'intentions d'ACT-R.

Lorsque le processus de résolution du SAS donne lieu à une stratégie, l'ajout d'intentions se fait également au niveau du module d'intentions, et les plans sont quant à eux implémentés dans le noyau d'ACT-R ou dans le module d'intentions si on les considère comme des connaissances sémantiques. L'ajout d'intentions au module éponyme permet de fixer un ensemble de buts à long terme c'est-à-dire une stratégie. De cette manière, les différentes règles qui auront été implantées seront exécutées au moment voulu. L'ajout d'une nouvelle règle au noyau est ensuite répercuté dans la base de connaissances procédurales de la personne, ceci est symbolisé par la flèche du noyau d'ACT-R vers la base de règles de production de la personne. Le module d'attention pour sa part, va suivre le

[6]Remarque : quand il n'y a pas de but dans le tampon des buts, il n'y a pas d'association au sein du noyau d'ACT-R, on ne fait alors pas appel au SAS, il s'agit de périodes pendant laquelle la personne ne fait rien (par exemple quand elle dort).

[7]Par rapport au schéma d'ACT-R proposé (cf. figure 6.2), les flèches indiquant le calcul des valeurs d'activations et d'utilité à partir de la mémoire épisodique ont été supprimées pour plus de lisibilité.

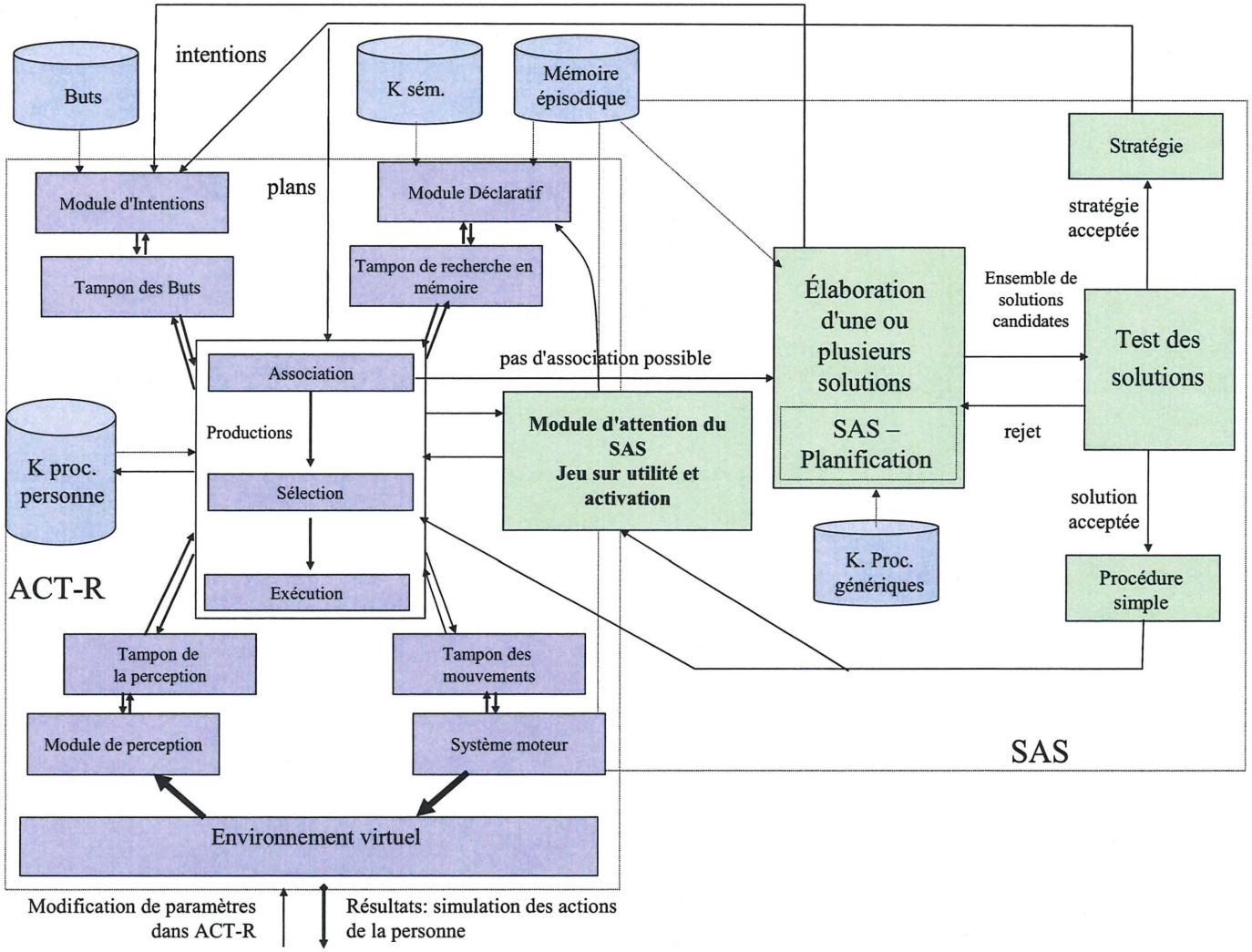


FIG. 6.4 – Structure de l'architecture hybride ACT-R - SAS

déroulement de la stratégie mais à haut niveau, sans regarder l'exécution de toutes les sous-actions du plan.

Lorsqu'il s'agit d'une procédure simple, celle-ci est directement implémentée dans le noyau d'ACT-R pour être exécutée immédiatement. Cette exécution est surveillée par le module d'attention du SAS qui peut jouer sur les valeurs d'utilité des productions et ainsi empêcher la sélection d'une autre règle de production.

D'autre part, le SAS a également un rôle d'attention sur le déroulement des processus d'ACT-R. Ce rôle est joué par le module d'attention du SAS. Celui-ci va permettre de simuler l'attention ou la concentration de la personne sur sa tâche. Pour cela, il peut modifier les valeurs d'activation des connaissances déclaratives et les valeurs d'utilité des règles de production. Ceci va constituer un moyen important pour simuler les troubles cognitifs. En effet, les erreurs cognitives dont nous avons parlé dans la section 2.5.3 page 38 sont les suivantes : erreur de capture, omission, anticipation, persévération, substitution d'objets.

L'idée est que le module d'attention ajoute en permanence aux utilités et aux activations une certaine valeur calculée en fonction de la mémoire épisodique et des compétences de la personne. Lorsque l'attention décroît, cette valeur décroît également mais pas de façon uniforme. Cette décroissance est différente selon les règles de production ou les connaissances déclaratives. Dès lors, il peut arriver qu'une séquence de règles de production soit choisie à la place d'une autre, on a alors une erreur de capture. Il peut arriver également qu'une règle de production soit sélectionnée trop tôt dans la séquence si son utilité est trop forte, ou soit oubliée si son utilité n'est pas assez forte, on obtient alors les erreurs d'omission et d'anticipation. Enfin la persévération peut s'expliquer par une valeur d'utilité qui serait trop forte donc la règle de production continuerait à être choisie alors qu'elle ne le devrait pas. Dans l'exemple de la personne qui continue à verser de l'eau dans un verre plein, la production qui continuerait à être sélectionnée aurait pour action «verser de l'eau dans le verre». En ce qui concerne les substitutions d'objets, il est possible de jouer sur les valeurs de similarités entre les objets dans ACT-R et ceci semble être un moyen efficace de simuler cette erreur. Lorsque la personne prêterait réellement

attention à ces objets alors le module d'attention réduirait la valeur de similarité entre les objets pour éviter toute erreur de substitution. Toutes ces possibilités d'actions sont représentés par les deux flèches partant du module d'attention, l'une allant vers le noyau d'ACT-R et l'autre vers le module déclaratif d'ACT-R.

D'autre part, le module d'attention va également intervenir pour des situations qui ne nécessitent pas la création d'une solution. Pour pouvoir agir au bon moment, le module d'attention doit disposer des mêmes informations que le noyau d'ACT-R. Ainsi le module d'attention a accès à tous les tampons, c'est ce que symbolise la flèche du noyau d'ACT-R vers le module.

Dans le cas où la personne doit résister à une tentation, l'intervention du module d'attention est nécessaire au moment de la sélection de la production : il va modifier les valeurs d'utilité des règles de production de manière à ne pas sélectionner la production qui devrait l'être normalement. Ce processus peut être déclenché par un but dans le tampon des buts qui serait explicitement «résister à la tentation de...». Il n'est pas sûr que ce soit réellement possible de réaliser ceci. Il sera nécessaire de réaliser de nombreux tests pour être capable de dire ce qui sera possible ou non dans cette hybridation.

Dans cette section, nous avons décrit l'architecture ACT-R - SAS. D'un point de vue général, l'hybridation se réalise plutôt bien. Mais il est possible que la nature de certaines interactions change par la suite, notamment lorsque les différentes composantes du SAS seront développées. Dans la partie suivante, nous allons évoquer les points à développer dans le futur, soit pour aboutir à une implémentation du système, soit pour faire avancer la théorie en psychologie cognitive.

6.4 Pistes de recherches futures

6.4.1 Le SAS - Planification

Le SAS - Planification va constituer une solution pratique au problème de manque de spécification du SAS pour la simulation des processus délibératifs chez une personne saine réalisant des AVQ. Dans les paragraphes suivants, on désignera le SAS - Planification par SAS - P.

Le principe de fonctionnement du SAS - P est très simple. On dispose d'une base complète des connaissances (représentées sous forme procédurales ou sémantiques) qui sont nécessaires et suffisantes pour réaliser toutes les activités de la vie quotidienne. Il suffit alors de sélectionner la ou les connaissances stockées dans cette base, qui vérifient les conditions formées par le contenu des tampons associés au noyau d'ACT-R. Le principe est exactement le même que celui de la phase d'association d'ACT-R, sauf que cette association se fait sur une base de connaissance plus large.

Ensuite, l'ensemble des connaissances vérifiant les conditions et formant ainsi un ensemble de solutions est testé afin de ne sélectionner qu'une seule règle de production ou un seul plan (série de règles). Dans le cas d'une règle de production seule, il suffit d'ajouter au noyau d'ACT-R la règle de production qui vient d'être sélectionnée. Le noyau d'ACT-R se chargera de mettre à jour la base de connaissances procédurales de la personne lorsque celle-ci utilisera pour la première fois cette connaissance. Dans le cas d'une stratégie ou plan, on ajoute la connaissance sémantique représentant le plan au module d'intentions, de manière à permettre le déclenchement des différentes règles en temps utile. Ces différentes phases sont présentées dans la figure 6.5 (page suivante).

Le SAS - P ne constitue qu'une solution de rechange pour le processus de création d'une nouvelle solution. Il reste à développer le module d'attention du SAS pour avoir une architecture valide et fonctionnelle.

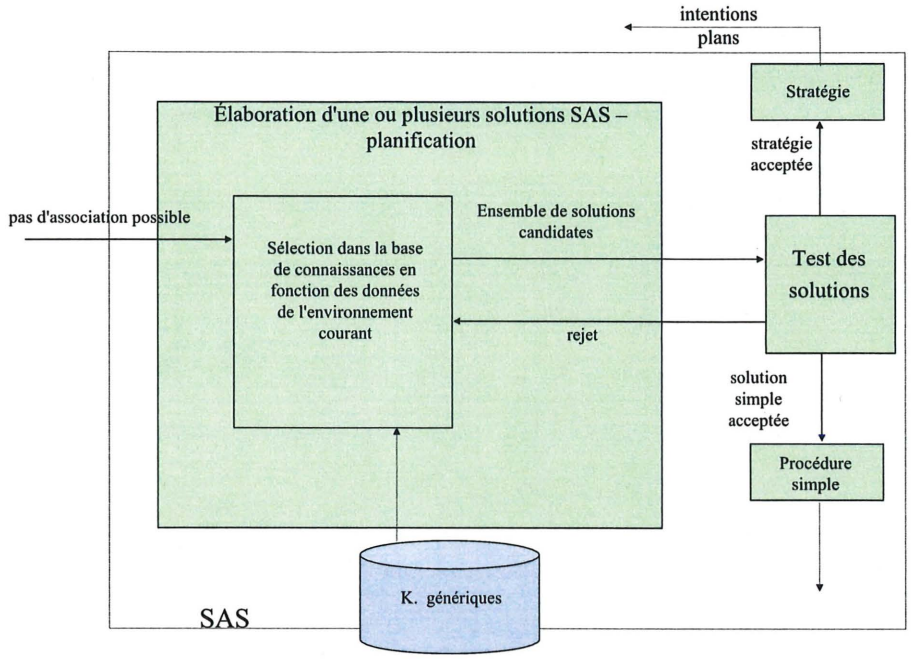


FIG. 6.5 – Structure du SAS - Planification

6.4.2 Points à développer

Les principaux problèmes vont découler de l'hybridation de ACT-R avec le SAS. Dans la section précédente, nous avons expliqué comment le SAS - Planification fonctionne. La constitution d'une base générique de connaissances sur la réalisation des activités n'est pas évidente. Nous avons déjà parlé des problèmes de complexité spatiale et des problèmes d'énumération que risque de poser une telle solution. Il sera certainement nécessaire de développer un modèle pour manipuler et remplir cette base de façon efficace. Là encore la réalisation peut amener de nombreux problèmes.

Il restera également à spécifier le module d'attention du SAS et à le développer. Il n'est pas évident a priori de trouver la manière dont le module d'attention devra influencer les activations et les utilités. Seules les grandes lignes de fonctionnement ont été données.

Pour avoir un module d'attention fiable, il sera nécessaire d'exprimer les équations mathématiques régissant l'évolution des différents paramètres et de prouver leur fondement théorique et psychologique.

Enfin, nous avons trouvé une solution quant à la simulation des processus délibératifs pour une personne saine (à l'aide du SAS - Planification), mais nous n'avons pas de solution pour le moment pour la réalisation d'un SAS capable de simuler une personne atteinte de troubles cognitifs. Il reste donc un certain nombre de problèmes et de points à développer avant d'arriver à une version informatique de l'architecture ACT-R - SAS. La solution du SAS - Planification constitue un palliatif au manque de théorie du SAS. L'objectif à long terme est toutefois de développer suffisamment la théorie du SAS pour permettre une implémentation, et ainsi obtenir une architecture générique pour simuler la cognition humaine.

Dans ce chapitre, nous avons présenté une architecture cognitive originale qui est une hybridation de l'architecture cognitive ACT-R d'Anderson et de la théorie du SAS de Norman et Shallice. Nous avons présenté les adaptations effectuées et le fonctionnement global de cette architecture. Il reste de nombreux points à préciser et à discuter, le travail le plus important sera de faire avancer la théorie du SAS.

Dans cette deuxième partie du mémoire est présenté un modèle d'architecture pour l'habitat intelligent dans lequel a été précisé le fonctionnement global des composantes de télésurveillance et d'assistance cognitive, ainsi que celui de la composante de simulation de la personne. La troisième partie de ce mémoire va porter sur la réalisation informatique du modèle décrit.

Troisième partie

Réalisation d'un prototype d'habitat intelligent

Chapitre 7

Réalisation et résultats

Dans ce chapitre, nous allons aborder la réalisation informatique d'un prototype d'habitat intelligent, basé sur l'architecture proposée dans la deuxième partie de ce mémoire. Il s'agit de réaliser un prototype en boîte noire qui met en valeur les interactions entre les différentes composantes du système. Dans un deuxième temps, nous décrirons les résultats obtenus suite à la mise en œuvre de ce prototype

7.1 Description de l'implémentation

Le prototype de système a été développé en Java, et certaines données ont été représentées à l'aide du langage XML. Nous allons décrire tout d'abord la manière dont sont simulées les données en provenance de l'habitat, puis nous expliquerons la représentation des acteurs du système, et enfin, les différentes composantes faisant référence à l'architecture présentée.

7.1.1 Simulation des données de l'habitat

Pour simuler les données en provenance de l'habitat, il a fallu faire un certain nombre d'hypothèses. En effet, le laboratoire DOMUS ne dispose pas encore de données capteurs réelles en provenance d'un habitat. Aussi, il a été convenu que les événements capteurs en provenance de l'habitat seront représentés par des objets en Java. Ces objets événements sont regroupés dans le package `evenements`. Un événement possède de façon générale un

nom, un libellé. Il peut avoir une durée, une localisation (généralement la pièce dans laquelle il a lieu) ou une destination (quand il s'agit d'événements d'intervention). De plus, il peut être typé ou non (par exemple : événement de type physiologique ou activité). Les données concernant les différents événements du système sont représentés dans un fichier XML. Ce fichier XML est parcouru au début du programme pour constituer une banque d'instances d'événements disponibles partout et à tout moment.

A partir de ces événements, l'habitat et ce qui s'y déroule est représenté par un script. Un script (objet représenté dans le package `script`) est constitué d'un moment de la journée (Nuit, Matin, Midi ou Autres), d'une description et d'une file d'événements constituant le cœur du script. Dans le package `script`, on trouve aussi des classes héritant d'une classe abstraite `ScriptCreator`, qui permettent de créer des scripts selon le moment de la journée et selon les informations données par l'interface du programme. Les scripts sont donc en partie paramétrables via l'interface utilisateur (cf. figure 7.1 (page suivante)).

Cependant, le script seul ne suffit pas, il faut en effet pouvoir envoyer les événements dans le système pour simuler l'arrivée de données capteurs. Pour cela, est utilisée la composante *Scénario*. Cette composante est un *thread* qui va parcourir le script et envoyer les événements au fur et à mesure dans le système. Plus précisement, les événements en provenance de l'habitat ont toujours une durée et en mode automatique, le *thread* dort le temps de cette durée avant d'envoyer l'évenement suivant.

Il y a deux modes d'envoi des données ou d'exécution :

- mode automatique : c'est le cas lorsque le thread dort le temps de la durée de l'événement precisée dans le fichier XML ;
- mode pas à pas : le *thread* `Scénario` se met en pause apres l'envoi d'un événement dans le système et c'est alors à l'utilisateur d'indiquer quand il souhaite que le scénario envoie l'événement suivant.

L'utilisateur peut passer indifféremment d'un mode à l'autre durant l'exécution d'un script. De plus, pour l'exécution en mode automatique, on peut régler l'échelle de temps concernant la durée. En effet, dans le fichier XML, la durée est simplement représen-

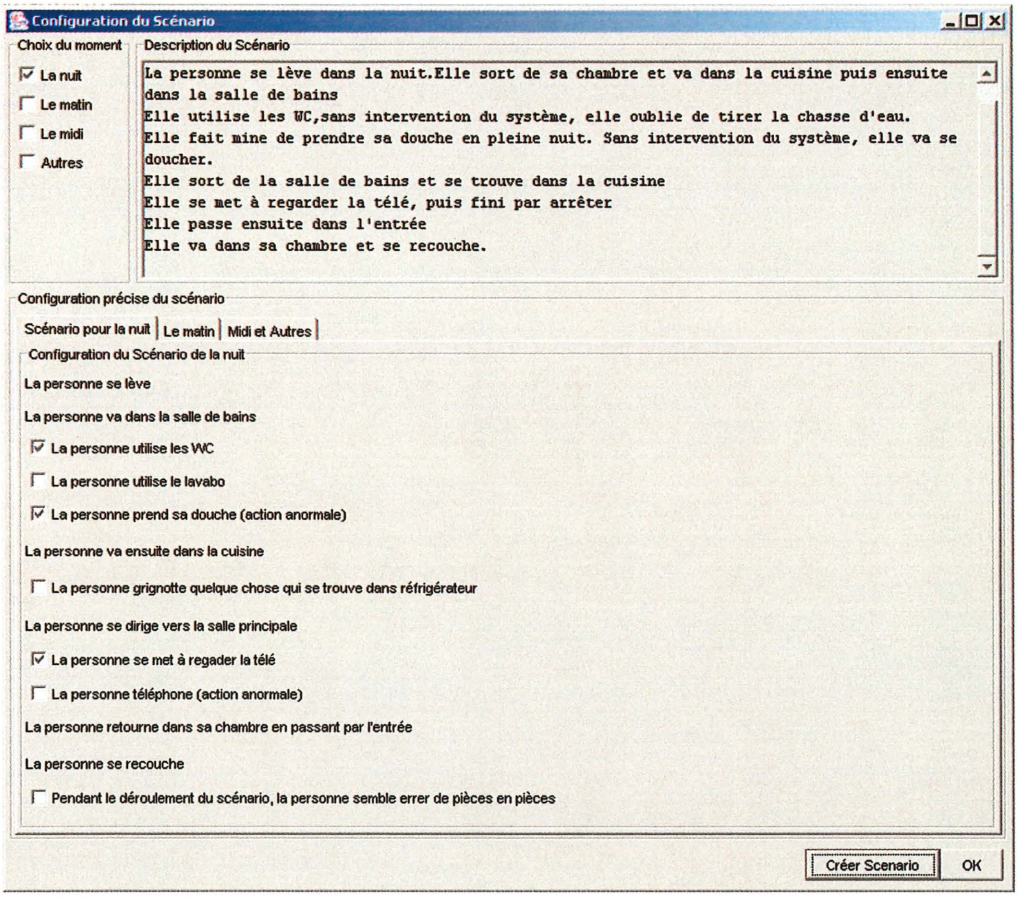


FIG. 7.1 – Fenêtre de l'interface usager permettant de configurer les différents scénarios

tée par un entier qui est ensuite converti en terme de secondes pour le système. On peut alors changer cette conversion et considérer que la durée est de l'ordre de 100ms, 1s ou 10s.

Ainsi avec ces divers packages et classes, le système peut simuler des données en provenance de l'habitat, jouer sur le script proposé et sur son mode d'exécution.

7.1.2 Représentation des acteurs et plus particulièrement de l'occupant

Le prototype prend également en compte des données concernant l'occupant de l'habitat. Le système charge lors de son lancement, un occupant par défaut dont les paramètres sont indiqués dans un fichier XML. Cet occupant est représenté dynamiquement en Java par une classe `Occupant` qui est décrite dans le package `acteurs`. Dans ce package, se trouvent notamment une classe permettant de représenter une personne. Une personne est identifiée par son nom, prénom, adresse, âge et numéro de téléphone.

Ensuite du point de vue de l'occupant, toute autre personne peut être une connaissance ou une relation. Ainsi, la classe `PersonneConnue` permet de représenter une personne et sa dénomination par rapport à l'occupant. Par exemple, l'occupant peut connaître un médecin qui sera alors une personne connue dont la dénomination est «médecin». Dans ce package, se trouvent aussi des classes qui permettent de représenter les compétences cognitives, les compétences physiques et les préférences de l'occupant. Donc, l'occupant est une personne qui possède des compétences cognitives, physiques et des préférences, ainsi qu'un ensemble de personnes connues.

En ce qui concerne les compétences et les préferences, le nombre de paramètres a été restreint de manière à permettre l'implémentation en boîte noire. Dans les compétences cognitives, on trouve des paramètres concernant :

- le stade de la maladie d'Alzheimer (entier de 1 à 7) ;
- la fréquence des oublis (très rares, rares, fréquents quasi-systématiques) ;
- la fréquence des errances (très rares, rares, quelques fois, fréquentes).

Pour les compétences physiques, on trouve un paramètre pour la vue, l'ouïe qui peuvent prendre les valeurs «bonne», «moyenne» ou «faible» et un paramètre pour la capacité de déplacement qui peut être «aisée», «quelques fois problématique» ou carrément «difficile». Enfin, pour les préférences, le système prend en compte le type d'interventions (sonore, visuelle ou indifférente) et la modulation de l'intervention (forte, moyenne ou faible). Ces différents paramètres ainsi que les renseignements concernant l'occupant sont réglables via une fenêtre spécifique de l'interface (cf. figure 7.2 (page suivante)).

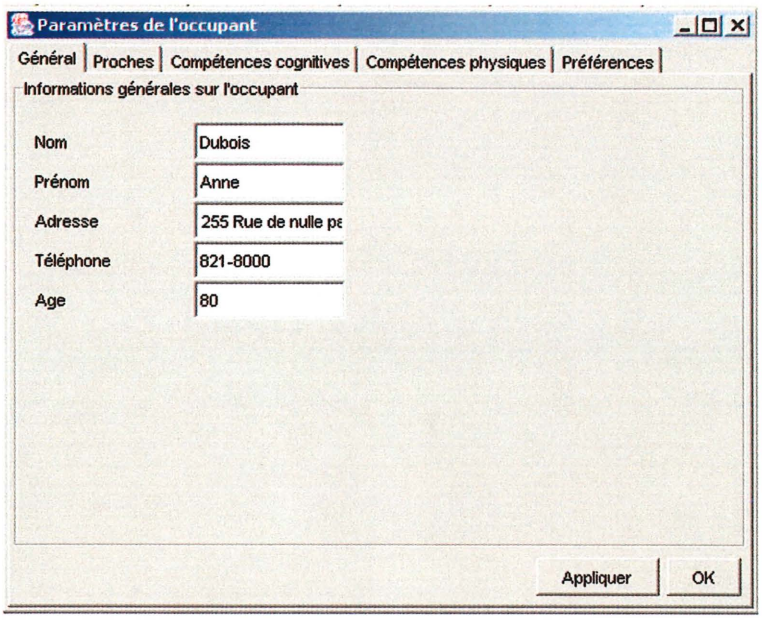


FIG. 7.2 – Fenêtre de l'interface usager pour le réglage des paramètres de l'occupant

7.1.3 La représentation des composantes du système

Chaque composante du système est représentée par une classe Java qui se trouve dans le package `composantes`. Pour respecter l'idée que le système de l'habitat intelligent sera un système réparti, chacune des composantes devaient être la plus indépendante possible des autres. Aussi les composantes communiquent par événements (selon le design pattern Observer (Gamma et al., 1995)). Lorsqu'une composante a terminé le traitement qu'elle devait faire, elle envoie les informations qu'elle estime nécessaires aux composantes qui l'écoutent. Ces informations peuvent être des événements dont la description est donnée dans le fichier XML, ou bien des objets représentatifs du type d'informations qui vont circuler d'une composante à une autre.

Les composantes représentées en Java correspondent à celles présentées dans la deuxième partie du mémoire. Ainsi, pour le modèle le plus global de l'architecture (cf figure 4.3 page 97) sont représentées :

- le `Scénario` qui représente l'habitat et les données en provenance de l'habitat ;
- une composante `Assistance cognitive` qui représente la composante *Assistance*

cognitive et télésurveillance ;
- une composante `Détection de panne` qui représente une fonctionnalité de la composante de *Gestion de l'environnement* ;
- une composante `Urgences` qui représente la composante de même nom ;
- une composante `Gestion des interventions` correspondant à la composante eponyme ;
- une composante `Assistance cognitive interactive` qui ne figure pas sur le modèle global mais que nous avons évoquée auparavant.

La composante d'assistance interactive a été realisée pour mettre en valeur les interactions entre ACT-R et SAS. Nous verrons dans la partie concernant les résultats de quelle manière.

Ensuite, si on considère le modèle concernant le fonctionnement de la composante d'assistance cognitive et de télésurveillance (cf figure 5.4 page 120), les composantes suivantes sont représentées :
- la *Reconnaissance des activités* ;
- le *Construction et répartition d'informations* (appelée `Traitement` dans le prototype) ;
- la *Mémoire épisodique* ;
- la *Simulation de la personne* ;
- l' *Analyse et la formulation des interventions* (composante simplement nommée `Analyse` dans le système) ;
- l'*Historique* et le traitement associé ;
- l'*Apprentissage.*

Enfin, si on considère le modèle donné pour la simulation de la personne (cf figure 6.1 page 131), la composante de *simulation* fait interface avec les deux composantes : *ACT-R* et *SAS.*

Chaque composante, sauf la mémoire épisodique et l'historique, sont des *threads* qui tournent en parallèle et exécutent des traitements selon les informations reçues. Au début de l'implémentation, chaque composante a été écrite selon les besoins qui lui était propre. Cependant, elles avaient des aspects communs qui ne sont apparus qu'après coup. Aussi, il y a eu une phase de *refactoring* qui a permis d'extraire les caractéristiques essentielles aux

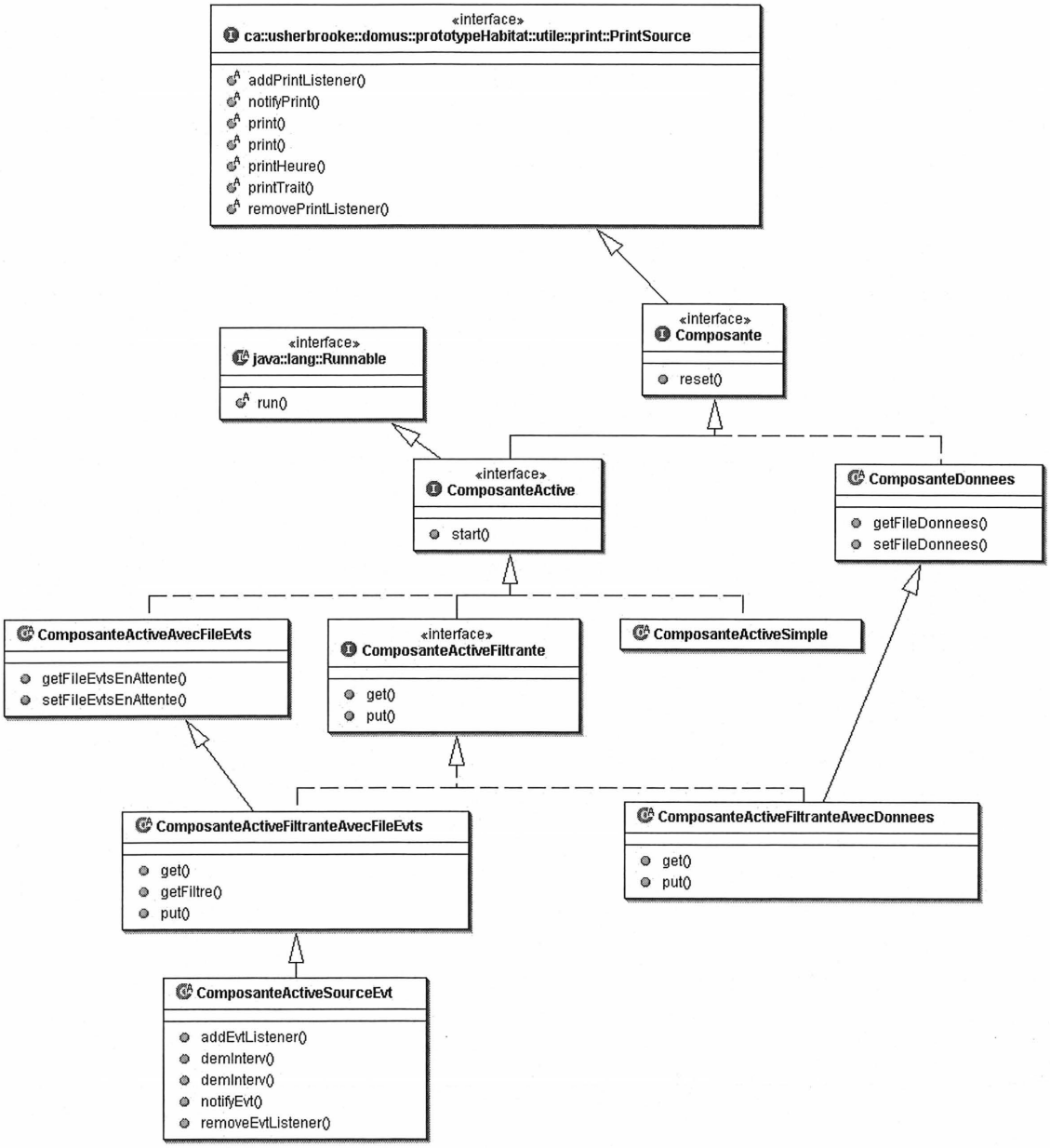


FIG. 7.3 – Diagramme de classe des interfaces et classes abstraites pour représenter des composantes.

différentes composantes. Ces caractéristiques sont hiérarchisées par des classes abstraites et des interfaces selon les types de composantes. Le diagramme de classe correspondant à cette hiérarchie est présenté dans la figure 7.3 (page précédente).

Ainsi, le type le plus général de composante doit pouvoir se réinitialiser et possède des fonctions d'impression. Ensuite, il faut distinguer les composantes selon leur activité et le type de données qu'elles manipulent. Ainsi, une composante active est une composante qui est un *thread* et qui possède donc une méthode start. Toutes les composantes sont actives hormis la mémoire épisodique et l'historique qui représentent des composantes de données. Ensuite la hiérarchie est établie en fonction des objets manipulés par les composantes, selon qu'il s'agit de files de données, de files d'événements, ou encore selon que la composante est capable d'effectuer un filtrage quelconque.

Enfin, on peut distinguer une catégorie de composantes particulières : les composantes qui sont sources d'événements (`Evt` générique). On regroupe dans cette catégorie la composante `Scénario` qui est source des événements de l'habitat, les composantes `Urgences`, `Détection de panne`, `Assistance cognitive` et `Assistance cognitive interactive` qui sont sources d'événements de type intervention.

Maintenant que nous avons rapidement décrit les différentes classes essentielles au système, nous allons aborder l'aspect communication entre les différentes composantes et préciser ce qui circule entre chacune d'elles.

7.1.4 Communication entre les différentes composantes

Pour décrire les données circulant entre les différentes composantes du système, nous allons garder la même démarche que celle adoptée lors de la description de l'architecture du système. Ainsi, nous allons nous intéresser aux composantes décrites dans le modèle le plus global, puis nous aborderons la composante d'assistance cognitive et de télésurveillance, et celle de la simulation de la personne. Enfin, nous ferons un point sur les interactions entre les composantes existantes et la composante d'*assistance cognitive interactive.*

Pour réaliser des classes communiquant par événements, il faut se munir d'un certain nombre d'objets qui sont les listeners (qui définissent le comportement de l'auditeur lors de la réception d'un événement), et les sources associées (qui permettent à une classe d'être source d'événements et auprès de laquelle les auditeurs peuvent s'enregistrer). Ainsi pour chaque type d'informations qui circulent dans le système, il existe une interface de listener et une classe source qui permet d'envoyer les événements aux classes qui implémentent ces listeners. L'ensemble de ces classes et interfaces sont regroupées dans le package `listeners`.

Modèle de fonctionnement global

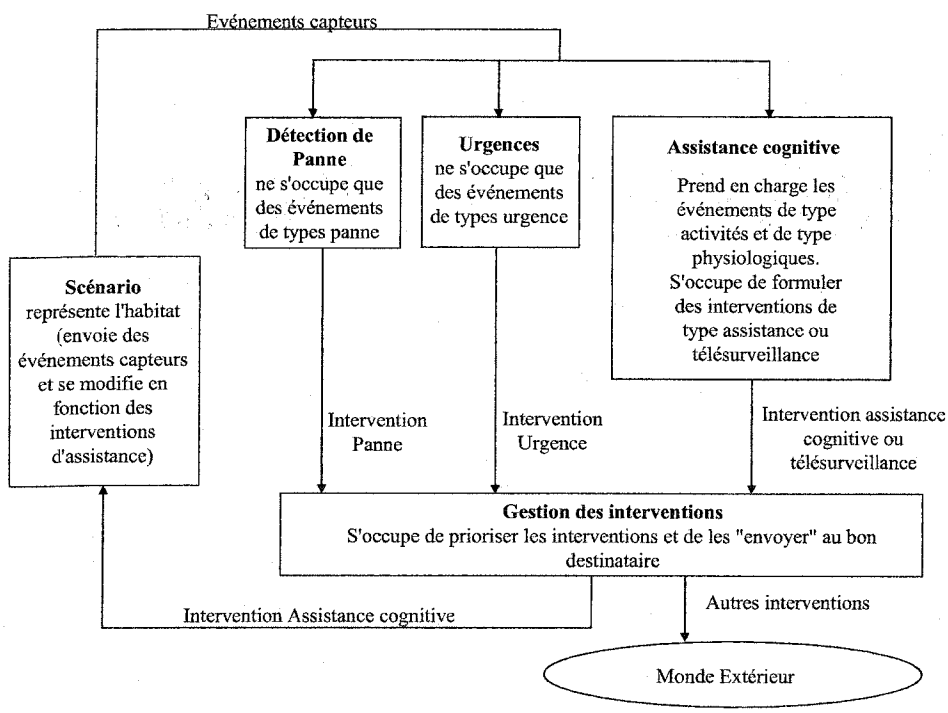


FIG. 7.4 – Modèle global de communication entre les composantes

Au niveau le plus global, le modèle de communication est très simple (cf. figure 7.4). La figure 7.4 correspond à la vue globale de l'interface utilisateur. La composante *Scé-*

nario envoie dans le système des événements du type `Evt` qui correspondent à des événements capteurs. Les auditeurs du scénario sont les classes `Assistance cognitive`, `Urgences et Détection de Panne`. Ceci respecte bien le modèle global donnée dans la figure 4.3 page 97. Ces trois composantes peuvent être source d'un événement d'intervention (intervention de type panne, urgences, assistance cognitive, télésurveillance). Ces interventions sont envoyées à la composante de *gestion des interventions*. Cette dernière envoie les interventions de type assistance cognitive au scénario (qui représente l'habitat) et qui se modifie alors pour simuler l'effet des interventions. Les autres interventions sont simplement envoyées «vers l'extérieur», c'est-à-dire qu'elles sont imprimées dans la partie de l'interface correspondant à la *gestion des interventions*.

Modèle de fonctionnement pour l'assistance cognitive et la télésurveillance

Le modèle de fonctionnement pour l'assistance cognitive et la télésurveillance est un peu plus complexe (cf figure 7.5 (page suivante)) que le précédent. Dans l'ensemble, il respecte le modèle donnée dans la figure 5.4 page 120.

La composante d'*assistance cognitive* du modèle global se comporte comme une interface avec le niveau inférieur. C'est elle qui réceptionne les événements en provenance de l'habitat et les envoie aux bonnes composantes. Notamment, elle prend en compte les événements de type activité qu'elle envoie à la *reconnaissance des activités*, et les événements de type physiologique qu'elle envoie à la composante de *construction et répartition*.

La *reconnaissance des activités* s'occupe d'apparier un certain nombre d'événements d'activités avec une activité reconnue qui est alors transmise à la composante de *construction et répartition*. Une activité reconnue est un objet Java qui contient un évenement représentant l'activité, un booléen pour rendre compte de sa normalité, un autre pour dire si elle est terminée ou non et l'ensemble des activités probables suivant l'activité reconnue. Ainsi, si on reconnaît que la personne est en train de préparer son café, s'il s'agit du matin, cette activité sera normale mais par contre incomplète et les actions probables suivantes seront de sucrer le café puis de le boire.

Ensuite, la composante de *construction et répartition* s'occupe de transmettre les ac-

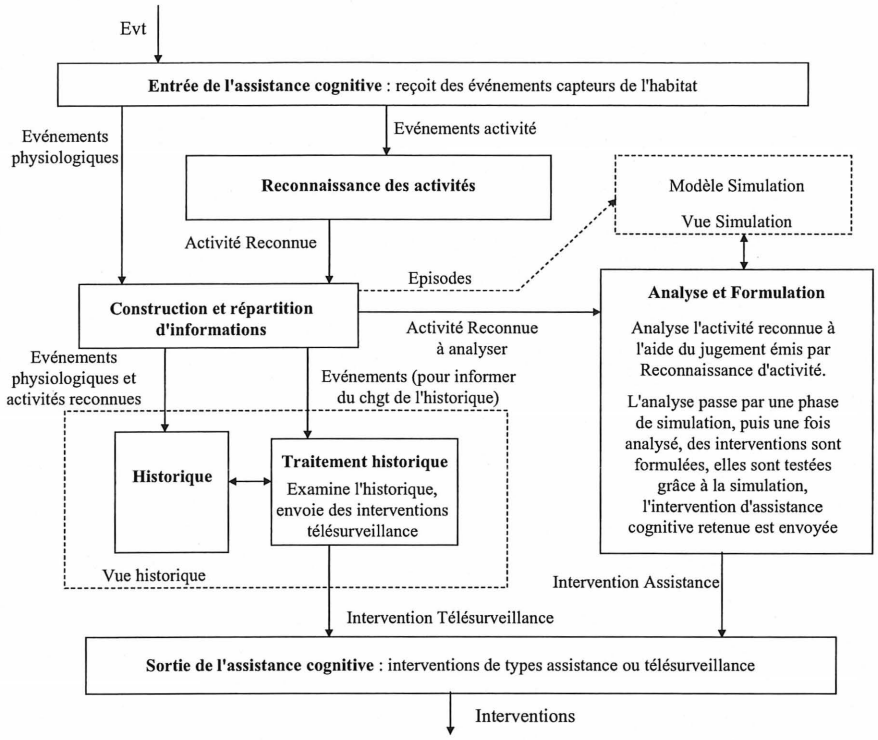


FIG. 7.5 – Modèle de communication pour les composantes formant l'assistance cognitive et la télésurveillance

tivités et les événements physiologiques à l'*historique* et au *traitement sur l'historique*, pour que celui-ci puisse éventuellement détecter des situations où une intervention de télésurveillance est nécessaire. La composante de *construction* s'occupe également de construire des épisodes (objet en Java) qui vont être envoyés à la mémoire épisodique. Enfin, l'activité reconnue sera envoyée à la composante d'*analyse*, qui va s'occuper, selon la normalité ou le caractère achevé de l'action, de formuler des interventions de type assistance cognitive, avec l'aide de la *simulation de la personne.*

Par rapport au modèle proposé pour la composante d'*assistance cognitive et de télésurveillance*, on remarque, d'une part, que l'aspect télésurveillance ne passe plus par la composante d'analyse et de formulation de l'intervention. Ce changement par rapport à la théorie n'est pas permanent. En effet, la mise en œuvre en boîte noire implique que les

traitements effectués sont assez simples. Ainsi pour déterminer la nécessité d'une intervention de télésurveillance, on se contente de compter le nombre d'occurences de tel type d'activité ou d'évenement physiologique dans l'historique récent, et, si cela dépasse un certain seuil, une intervention de télésurveillance est élaborée. Ce traitement simple n'a pas besoin des données dont dispose la composante *Analyse*, et n'a pas sa place en terme de concept objet dans cette composante. Aussi ce traitement est délégué temporairement à la composante de *traitement sur l'historique* qui centralise alors tous les aspects concernant la télésurveillance. Cependant, il est fort probable qu'à terme lorsque le traitement réel sera effectué, ce soit la composante d'*analyse* elle-même qui détecte la nécessité d'une intervention de télésurveillance.

Enfin, la composante *apprentissage* est ici absente du schéma, mais a bien été mise en œuvre. Elle n'a été développée que très brièvement, et l'absence de représentation précise des habitudes de vie et d'un réel traitement quant à la détection du changement des compétences de l'occupant, font qu'elle a peu d'impact dans le système. Cependant son rôle est brièvement illustré par un des scénarios.

Modèle de fonctionnement pour la simulation de la personne

En ce qui concerne la composante de *simulation de la personne*, ACT-R et SAS ont chacun été représenté par une classe. Elles communiquent selon le schéma de la figure 7.6 (page suivante). Ainsi, la *simulation* peut être appelée par l'*analyse* pour détecter des erreurs cognitives dans le futur proche ou pour tester les interventions formulées par rapport à une situation.

La composante ACT-R, quant à elle, peut être directement sollicitée par l'arrivée d'un nouvel épisode dans la mémoire épisodique. Dans ce cas, ACT-R calcule les nouvelles probabilités d'actions futures de l'activité à l'origine de l'épisode et envoie ces données à la composante de *Reconnaissance d'activité*. Les probabilités d'actions sont ici une série de chaînes de caractères associées à une activité, chacune des chaînes de caractères représentant une activité probable.

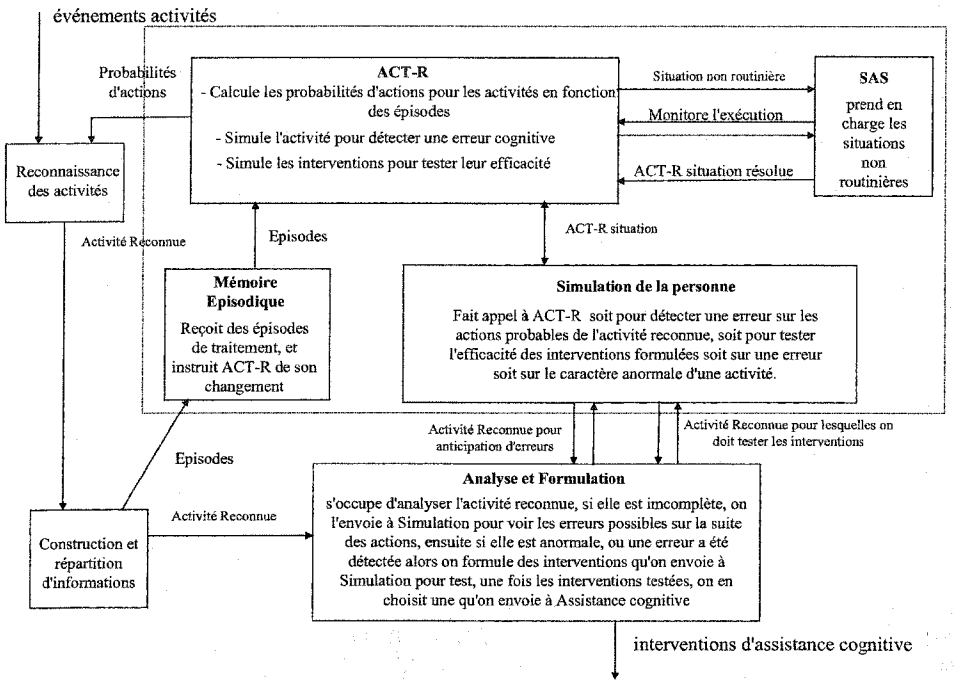


FIG. 7.6 – Modèle de communication pour la simulation de la personne

En ce qui concerne les interactions entre ACT-R et SAS, nous n'allons pas aussi loin que dans la partie théorique par rapport au modèle présenté dans la figure 6.4 page 140. Cependant les interactions essentielles ont été implémentées : l'appel d'ACT-R au SAS lors d'une situation non routinière, l'aspect surveillance de l'exécution (ou monitoring) et le renvoi d'une solution définitive à ACT-R. Ces aspects sont plus illustrés avec les scénarios d'aide interactive que nous décrirons dans la section traitant des résultats.

Modèle de fonctionnement pour l'assistance cognitive interactive

Comme il a été mentionné dans le point précédent, une composante chargée de l'*assistance cognitive interactive* a été ajoutée au système. Son rôle est de répondre aux demandes d'aide de la part de l'occupant, ou permettra de formuler des interventions d'assistance cognitive plus complexes qui mettront en jeu une interaction avec l'occupant.

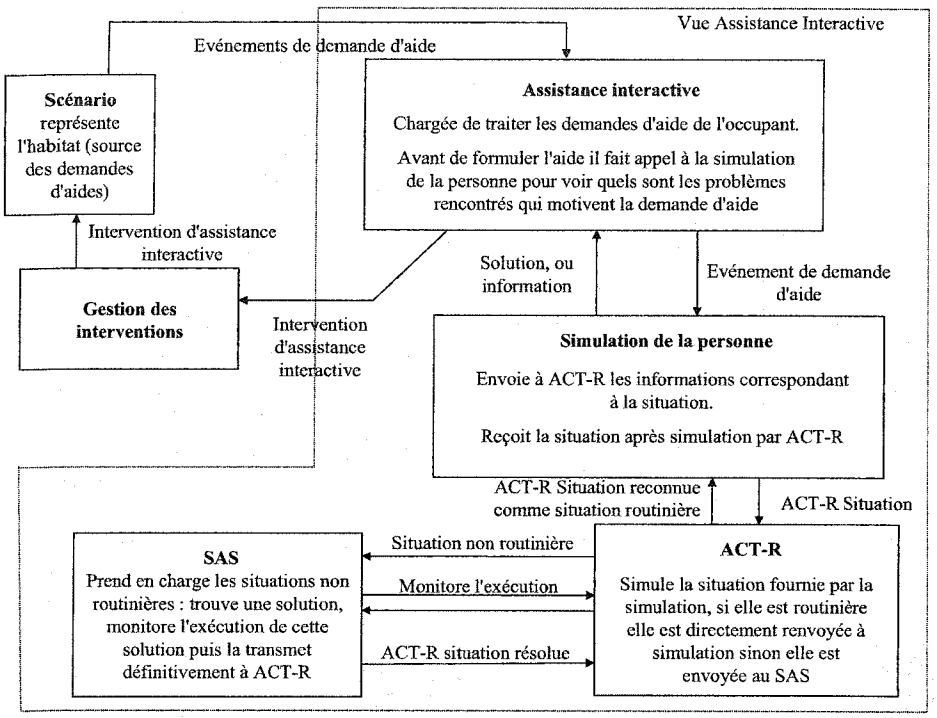


FIG. 7.7 – Lien de la composante assistance cognitive interactive avec le reste de l'architecture

Ici dans le modèle présenté dans la figure 7.7, l'aspect concernant la formulation d'interventions plus complexes n'apparaît pas, il n'a d'ailleurs pas été réalisé. Cependant, l'aspect réponse à la demande d'aide est pris en compte. Ainsi, si une demande d'aide provient de l'habitat, cette composante l'analyse pour déterminer, d'une part, si la demande d'aide est justifiée et, d'autre part, formule une aide interactive pour l'occupant. L'*assistance cognitive interactive* fait appel à la *simulation* pour savoir si la personne est réellement en difficulté ou non[8]. Si ce n'est pas le cas, on fournit simplement des indices de la solution à l'occupant. Cette politique est toujours motivée par le souhait de ne pas dégrader les compétences de l'occupant en laissant le système tout faire à sa place. Si par contre, la *simulation* montre que l'occupant est incapable de réaliser l'activité pour laquelle il demande une aide, l'architecture cognitive élabore une solution qui est ensuite

transmise à l'*assistance cognitive interactive* qui se charge de formuler les interventions correspondantes.

La *simulation de la personne* s'occcupe de donner les bons paramètres à ACT-R pour qu'il simule la personne. Lorsqu'il s'agit d'une situation routinière pour ACT-R, cela veut dire que la personne est en réalité capable de réaliser l'activité pour laquelle elle demande de l'aide, il faut donc juste lui donner des indices. Lorsqu'il s'agit d'une situation non routinière, le SAS prend le relais et élabore une solution qu'il va tester et pour finir renvoyer à ACT-R. Ensuite ACT-R transmet cette solution qui sera utilisée pour élaborer l'aide cognitive.

L'*assistance cognitive interactive* est une composante qu'il sera nécessaire de travailler et dont il faudra mieux préciser le rôle au sein du système. Deux scénarios permettent de mettre en valeur son rôle ainsi que les interactions entre ACT-R et SAS.

7.1.5 Autres aspects de l'implémentation

Pour permettre une meilleure compréhension du fonctionnement du prototype, une interface usager a été developpée (package `interfacePrototype`). Nous avons déjà vu qu'il est possible de régler les paramètres de la personne et de créer des scripts via cette interface. Il est également possible de démarrer, stopper ou reprendre le déroulement d'un scénario à la manière d'un débuggeur pour que l'utilisateur puisse observer à loisir le comportement de chacune des composantes. Pour cela, l'interface se découpe en un ensemble de vues sur les différentes composantes du système, qui informent l'interface de ce qu'elles font. Il est également possible d'observer plus précisémenent certaines composantes au moyen de fenêtres indépendantes, et également de consulter les modèles présentés dans le point précédent. Ceux-ci sont en effet inclus pour faciliter la compréhension de l'utilisateur.

D'autre part, d'autres packages concernent des aspects plus internes du système, tel que le package `xml` qui permet de parser les différents fichiers de données XML, ou le

[8]Cette décision est provisoire, il serait souhaitable de considérer la question en terme éthique.

package `utile` qui contient des objets et des fonctionnalités générales telles que des fonctions d'impression ou une banque de noms utilisée dans le système pour réduire le nombre de littéraux hardcodés dans les différentes classes.

Nous allons maintenant aborder les scénarios qui ont été décrits pour le prototype et les résultats associées à cette implémentation.

7.2 Les résultats

Afin de simuler le système et les rôles des composantes, des scénarios ont été établis. Ils sont divisés arbitrairement selon le moment de la journée. Ainsi, il y a un scénario pour la nuit, un pour le matin, un autre pour le midi et un dernier dit «autres». Plutôt que de détailler chacun des scénarios événements par événements, nous allons plutôt expliquer ce qu'ils permettent d'illustrer dans le système. Dans les scénarios présentés, on suppose qu'il s'agit d'une personne agée d'environ 80 ans atteinte de la maladie d'Alzheimer.

7.2.1 Le scénario de la nuit

Ce scénario se déroule la nuit vers 3h00 du matin. L'occupant se réveille et va effectuer une série d'activités qui sont paramétrables, puis à la fin du scénario, il retourne se coucher.

Parmi les activités qu'il est possible de lui faire faire, deux activités sont reconnues comme anormales. Ces deux activités sont «prendre une douche» et «téléphoner». Elles déclenchent toutes deux des interventions d'assistance cognitive qui, pour la première indique à l'occupant qu'il ne devrait pas prendre sa douche à cette heure-ci, et pour la seconde, indique qu'il risque de réveiller son interlocuteur (on imagine qu'il essaye d'appeler un ami ou un parent et pas les urgences ou le médecin).

Dans ce scénario, on peut aussi observer l'intervention du système en cas d'errance de la personne. Cette intervention dépend de la valeur paramètre concernant la fréquence

des errances. Le scénario contient un événement «aucun événement» qui simule l'inaction de la personne pendant un certain temps (c'est-à-dire une errance). A chaque fois qu'elle passe dans une pièce, cet événement est envoyé dans le système. La personne semble donc errer de pièces en pièces. Si la fréquence des errances est moyenne ou élevée, le système déclenche une intervention de télésurveillance qui sera envoyée au médecin.

Enfin, l'occupant peut passer par la salle de bains pour ses besoins naturels, boire un verre d'eau, aller dans la cuisine et grignoter, ce qui ne motive pas d'intervention particulière. S'il décide d'aller regarder la télévision, le système lui prodigue le conseil d'aller se recoucher. Même si cela semble sans effet, le système n'insiste pas sur ce point.

7.2.2 Le scénario du matin

Ce scénario commence à 8h00 du matin. La personne se lève. Ceci déclenche un apprentissage d'un changement d'habitude. En effet, dans l'historique, on a mis que cela fait plusieurs jours que la personne se lève à 8h00 au lieu de 7h00, la composante de *traitement de l'historique* le détecte et l'envoi à l'apprentissage. Cet aspect est destiné à mettre en relief le rôle de la composante *apprentissage*. Ici, il n'y a pas de réel changement des habitudes de l'occupant, car il n'y a actuellement pas de structure pour les représenter. La composante *apprentissage* se contente donc d'afficher au niveau de l'interface le changement d'habitude.

Ensuite, l'utilisateur du programme peut choisir de lui faire prendre sa tension au moyen d'un appareil situé dans la cuisine. Cette prise de tension déclenche une intervention de télésurveillance de la part de la composante *traitement de l'historique*. Là encore, l'*historique* contient le fait que les 6 jours précédents, sa tension était élevée. Comme c'est le septième jour où cela arrive, le système décide de lancer une intervention de télésurveillance pour prévenir le médecin.

Par la suite dans le scénario, la personne va dans la salle de bains et prend une douche. A la fin de cette activité, si le paramètre d'oublis a la valeur «fréquent» ou «quasi-systématique», la personne va oublier de se sécher. Le système va donc lui rappeller de se sécher. Ceci illustre le rôle de la composante d'*assistance cognitive* lorsque

l'action est incomplète. La composante d'*assistance cognitive* va demander à la *simulation de la personne* de simuler la suite d'actions probables pour détecter des erreurs cognitives sur les actions suivantes. Dans ce cas, la *simulation* renvoie que l'erreur possible que fera l'occupant sera d'oublier de se sécher. Le système anticipe donc l'erreur.

Après l'épisode de la douche, l'occupant va préparer son petit déjeuner. Cette activité peut se décomposer en la préparation de tartines et celle du café. Lors de la préparation du café, il risque d'oublier de le sucrer, si le paramètre concernant les oublis est suffisamment élevé. Dans ce cas, une autre intervention d'assistance a lieu pour lui rappeler de mettre du sucre dans son café.

Par la suite, l'occupant emporte son petit déjeuner dans la salle principale et s'installe à la table pour manger, le scénario s'arrête là.

7.2.3 Le scénario du midi et autres scénarios

Le scénario du midi est très court. Il est destiné à mettre en avant ce qu'il se passe lors du déroulement d'activités en parallèle. La personne est train de préparer son déjeuner et l'utilisateur de l'application peut décider de faire sonner le téléphone. Dans ce cas, la personne y répond et selon son paramètre d'oublis, elle se souviendra ou non de l'activité précédente. L'intervention du système devra donc lui rappeler qu'elle a commencé une activité dans la cuisine.

En ce qui concerne les autres types de scénarios, ils sont destinés à mettre en valeur certaines interactions précises du système. Ainsi, il est possible de déclencher une panne de capteur ou de réseau pour observer le rôle de la composante de *détection de panne*, ou encore déclencher un début d'incendie ou un malaise pour observer le rôle de la composante *Urgences*.

De plus, il est possible de mettre en scène deux demandes d'aides de l'occupant au système. La première est une demande d'aide simple concernant la réalisation d'une recette de cuisine pour trois personnes au lieu de quatre. Cette demande d'aide va être traitée différemment selon le stade de la maladie de l'occupant. Ainsi, si le système estime la per-

sonne capable (via la simulation à l'aide d'ACT-R), la composante d'*assistance cognitive interactive* va simplement fournir une calculatrice à l'occupant. Si la *simulation* montre qu'elle n'est pas capable de réaliser cette activité, alors le SAS va élaborer la solution qui consiste à prendre les diverses quantités d'ingrédients, les diviser par quatre et les multiplier par trois. Le SAS surveille ensuite l'exécution de cette solution dans ACT-R, puis finalement transmet la solution retenue à ACT-R. Celui-ci va la transmettre à la *simulation de la personne* qui va elle-même informer l'*assistance cognitive interactive.* L'*assistance cognitive interactive* va alors formuler une intervention, qui va consister à afficher sur un écran tactile, par exemple, la marche à suivre en plus de fournir une calculatrice à l'occupant.

La seconde demande d'aide concerne la programmation d'un magnétoscope. L'occupant désire le programmer pour enregistrer une émission tardive. Comme pour l'aide sur la recette, ACT-R va permettre de déterminer si l'occupant est capable de s'en sortir tout seul. Dans le cas où il n'est pas capable de faire cette programmation tout seul, le SAS va élaborer un plan en trois étapes pour aider l'occupant. Ce plan est monitoré dans ACT-R, puis transmis via ACT-R et par la composante de *simulation de la personne*, jusqu'à l'*assistance cognitive interactive.* Celle-ci va alors formuler des interventions conformément au plan et envoyer l'intervention concernant la première étape à l'habitat. Cette première étape demande à être validée par l'occupant. L'*assistance cognitive interactive* attend la validation de la première étape. Lorsqu'elle la reçoit, elle envoie la deuxième étape à l'habitat et ainsi de suite jusqu'à la fin du plan.

Les deux derniers scénarios sont destinés à mettre plus en avant les interactions entre le SAS et ACT-R. En effet, dans les autres scénarios, les activités mises en jeu étaient toujours routinières. Ceci ne permettait donc pas de mettre en avant le lien entre SAS et ACT-R. Ici l'aide peut concerner une activité non routinière ce qui permet de montrer l'appel au SAS et les deux aspects que peuvent prendre les solutions : solution simple ou bien un plan.

La description de ces scénarios et leur implantation dans les composantes ont amené certains changements dans la conception de l'architecture de l'habitat. Ces changements

et une analyse critique de l'implémentation sont présentés dans le chapitre suivant.

Chapitre 8

Critiques et perspectives

Dans ce chapitre, nous allons décrire les aspects précisés ou modifiés par la mise en œuvre du prototype informatique, puis nous développerons des idées pour développer et améliorer ce prototype.

8.1 Les conséquences de la mise en œuvre

Lors de la mise en œuvre du prototype, il est apparu que certaines composantes ou interactions manquaient de précision. La réalisation en boîte noire de ces composantes était alors difficile. De nouvelles discussions ont permis de clarifier certains points, d'en modifier d'autres ou simplement de souligner les zones floues dans les modèles proposés. Ces changements sont essentiellement intervenus au niveau des composantes d'Assistance cognitive et de Télésurveillance, le niveau global ayant été peu modifié. Nous allons détailler ces changements dans la suite de cette section.

8.1.1 Précision du modèle de la personne

Dans le modèle concernant les composantes d'*Assistance cognitive et de Télésurveillance*, le modèle de la personne était à l'origine peu précisé. Il avait déjà été établi que celui-ci contiendrait les compétences physiques, cognitives de l'occupant, ainsi que ses habitudes, sa mémoire épisodique et ses préférences. Toutefois, les relations et surtout la différence entre les habitudes et la mémoire épisodique étaient peu précises. Ainsi, lors de

la réalisation du cycle impliquant les composantes *Reconnaissance d'activités*, *Construction des épisodes*, *Mémoire Épisodique* et la *Simulation de la personne*, il a fallu préciser ses relations, qui ont été détaillées dans la section 5.1.4 page 109.

8.1.2 Interactions entre la composante d'Analyse et de formulation des interventions et la simulation de la personne

Une autre interaction qui manquait de précision était l'interaction entre la composante d'*Analyse et Formulation* et celle de la *simulation de la personne*. Il avait été établi que cette dernière fournirait une aide à l'analyse, mais la teneur de cette aide n'avait pas été précisée. Lors de l'implémentation, il a fallu détailler cet aspect afin de pouvoir mettre en place des interactions valides entre les deux composantes. Les précisions apportées ont clarifié le rôle de la composante de *simulation de la personne*. L'*analyse* a ainsi recours à la composante de *simulation*, d'une part, pour anticiper les erreurs cognitives sur les actions incomplètes de l'occupant, et d'autre part, pour tester l'efficacité des interventions d'assistance cognitive formulées.

8.1.3 La composante Apprentissage

L'implémentation a aussi révélé des points sur lesquels un travail supplémentaire de précision est nécessaire. Notamment les interactions du système avec la composante d'*apprentissage* sont encore peu définies. Dans le modèle proposé pour l'*Assistance cognitive et la Télésurveillance*, la composante d'*analyse et de formulation* est la seule à fournir des informations à la composante d'*apprentissage*. Mais, il n'est pas précisé quel type, ni à quelle fréquence. Dans le prototype réalisé, l'*apprentissage* reçoit l'information concernant un changement d'habitude significatif sur le lever de la part de la composante de *Traitement de l'historique*. On voit ici que l'*analyse et la formulation* ne serait certainement pas la seule composante à fournir des informations à l'*apprentissage*. Enfin, en ce qui concerne les actions de la composante d'*apprentissage*, il a été suggéré dans la partie théorique que l'apprentissage se ferait sur les habitudes et sur les compétences cognitives, mais ceci est flou, il sera nécessaire de préciser dans quelle mesure les changements pour-

ront être faits.

8.1.4 Choix de l'effecteur

Il sera nécessaire de préciser le moment où sera choisi l'effecteur permettant d'atteindre l'occupant. En effet, dans l'architecture proposée, il s'agit de composante *gestion des interventions* et des modalités de communication qui sont chargées de décider quel afficheur ou haut-parleur utiliser. Toutefois, le choix de cet effecteur dépend notamment de la localisation de l'occupant dans l'habitat. Or, les informations sur la localisation de la personne ne seront peut être pas ou plus disponibles au niveau de modalités de communication ou de la gestion des interventions. Aussi, le choix de l'effecteur interviendrait peut être avant au sein même de la composante d'*Assistance cognitive et de Télésurveillance* ou des autres composantes chargées d'élaborer l'intervention.

L'implémentation d'un prototype a donc permis à la fois de clarifier certains aspects du modèle proposé et à la fois de souligner les points restant flous ou peu précis. La précision de ces points dépendra notamment de la manière dont seront réalisées certaines composantes ou encore de l'accessibilité de certaines informations.

8.2 Pistes de développements et d'utilisation du prototype

Dans cette section, nous allons développer quelques idées permettant la réutilisation de ce prototype, son amélioration ou son extension.

8.2.1 Utilisation du prototype

Le prototype réalisé avait surtout pour rôle de fournir une preuve de la cohérence de l'architecture proposée. De ce point de vue, avec les modifications apportées à la théorie lors de la mise en œuvre, le prototype répond aux objectifs fixés. Cependant ce n'était pas son seul rôle.

L'idée est aussi de permettre aux membres de l'équipe et aux acteurs extérieurs de mieux comprendre et cerner les fonctionnalités de l'habitat au niveau cognitif. Comme l'interface actuelle est une interface Java simple mais non spécifique au projet DOMUS, il est prévu de fusionner ce prototype avec l'interface spécifique réalisée par un autre étudiant en maîtrise. Cette fusion permettra de fournir une démonstration de l'habitat intelligent et des objectifs du projet DOMUS en général.

Les autres idées d'utilisation sont de pouvoir remplacer une composante en boîte noire par une composante dont le traitement est réel afin d'observer son comportement lors des différents scénarios. Ce type d'utilisation implique néanmoins une certaine robustesse et une certaine indépendance des composantes les unes par rapport aux autres qui sont en partie réalisées avec l'utilisation du design pattern Observer (Gamma et al., 1995). Cependant, il n'est pas exclu qu'il faille modifier des aspects du prototype pour intégrer une nouvelle composante, notamment si les données en entrée ou en sortie sont modifiées, ou si de nouvelles interactions non décelées lors de la réalisation de ce travail apparaissent.

Les cas d'utilisation présentés ici demandent peu ou pas de développement supplémentaire si on exclut le développement complet d'une des composantes. Dans le paragraphe suivant, nous allons exposer des idées qui permettraient de développer le prototype pour d'autres utilisations, mais cette fois-ci, avec un travail conséquent d'un point de vue développement.

8.2.2 Développements futurs

A l'heure actuelle, les scénarios sont hardcodés dans les classes correspondant aux composantes. Les événements constituant le script sont séparés des composantes mais le traitement sur ces événements est harcodé. Par exemple, la composante de *reconnaissance d'activités* reconnaît une activité lorsqu'elle reçoit un certain nombre d'événements significatifs : l'activité `Personne_se_leve` est reconnue lors de la réception de `Capteur_poids_moins_lit` et `Capteur_mouvement_lit`. Donc, pour chaque action reconnue, il y a une liste d'événements correspondante. Or, cette liste est hardcodée dans la classe dans une structure de table de hashage, de même pour un certain nombre d'autres com-

posantes.

L'idée serait donc d'extraire ces informations et de les mettre dans un fichier XML qui préciserait le scénario et les entrées/sorties de chaque composante. De cette manière, après avoir rempli le fichier XML, il suffirait de le lire en début de programme et le comportement des composantes serait déterminé par celui décrit dans le fichier. Ceci permettrait d'améliorer le prototype, car le comportement ne serait plus hardcodé et permettrait de développer une interface de création de scénarios pour l'utilisateur.

Il est possible d'aller plus loin en utilisant le travail d'un autre étudiant en maîtrise. Celui-ci travaille sur un procédé pour générer des scénarios à partir des habitudes, compétences cognitives et compétences physiques de la personne. A partir des données d'entrée, il génère des scénarios de haut niveau contenant simplement les activités puis au fur à mesure, ce scénario est ramifié jusqu'à obtenir une séquence d'activations de capteurs. Cette génération est faite à partir d'un formalisme utilisant les réseaux de Pétri.

L'idée serait d'utiliser les données concernant les activités, tâches et séquences d'activations capteurs de façon inverse pour construire automatiquement le comportement des composantes du système. Par exemple, avec ce formalisme, les séquences d'activations capteurs sont déjà associées à une tâche lors de la création du scénario, il suffirait donc d'implanter cette association dans la composante de reconnaissance des activités pour simuler son traitement. Ce reverse engineering des scénarios permettrait de créer encore plus facilement des scénarios réalistes puisqu'ils prendraient en compte les paramètres de l'occupant de manière automatique. La partie manquante concernerait l'élaboration des interventions du système en fonction des tâches et activités. On pourrait alors considérer les interventions possibles comme des paramètres d'entrée parmi lesquels le système aurait le choix. Il s'agirait alors d'observer le comportement du système en fonction du scénario et des interventions possibles.

Les possibilités de développement proposées demandent un certain travail de mise en œuvre qui n'a pas été réalisé ici. Elles permettent à ceux qui le désirent de continuer le travail commencé.

Conclusion

Conclusion

La problématique du projet DOMUS est de concevoir un habitat intelligent dédié à l'assistance et à la télésurveillance de personnes souffrant de troubles cognitifs. Au sein de cette problématique, l'objectif de ce projet de maîtrise était de concevoir un modèle global de l'aspect cognitif de l'habitat intelligent respectant les contraintes de la psychologie cognitive.

Ainsi, dans ce mémoire, une revue de littérature importante a été faite comprenant les théories et architectures cognitives incontournables de la psychologie cognitive : les théories de Baddeley et Potter pour la mémoire de travail, la théorie du CS/SAS pour les processus de contrôle de l'activité, les architectures cognitives SOAR de Newell, EPIC de Meyer & Kieras et ACT-R d'Anderson. Cet état de l'art a permis d'évaluer les contraintes liées à la psychologie cognitive pour la conception d'une architecture capable de simuler une personne atteinte de troubles cognitifs.

Par la suite, nous avons présenté des modèles détaillant la nature et les interactions des différentes composantes nécessaires au fonctionnement d'un habitat intelligent, tout en respectant les contraintes de la psychologie cognitive. Un premier modèle global du système de l'habitat intelligent a été établi. Ensuite, un sous-modèle concernant les composantes d'*Assistance cognitive* et de *Télésurveillance* a été présenté. Enfin, un sous-modèle du précédent a été proposé en tant que nouvelle architecture cognitive pour la simulation d'une personne atteinte de troubles cognitifs.

La réalisation informatique en boîte noire d'un prototype du système a permis d'assurer une certaine cohérence pour les modèles proposés. La mise en œuvre informatique

a précisé certains aspects des modèles décrits en théorie et en a rectifié d'autres. La réalisation de scénarios a permis de plus de mettre en valeur les différentes interactions entre les composantes du système.

Cependant, nous avons constaté qu'il restait de nombreux points où un travail de précision est nécessaire, notamment les composantes liées à l'apprentissage ou encore celles liées à la simulation des processus délibératifs (SAS). D'un autre point de vue, le prototype réalisé est très simple et demande à être développé pour être réutilisé. Des pistes de développement ont été évoquées consistant à systématiser la représentation des scénarios et le traitement effectué par les composantes en boîte noire.

Enfin, les modèles proposés sont de façon générale des modèles boîtes noires. Il reste donc à proposer le fonctionnement d'un certain nombre de composantes du système. Nous avons vu notamment que la composante d'*Analyse et de Formulation de l'intervention* et celle concernant l'apprentissage sont encore mal définies. D'autres comme la *simulation de la personne* sont plus détaillées. Néanmoins, l'architecture cognitive proposée (l'hybridation de ACT-R et de SAS) pour simuler une personne atteinte de troubles cognitifs demandera un travail théorique important pour préciser et affiner la partie responsable des processus superviseurs.

Annexes

Annexe A

Glossaire

Activité, tâche, opération :
Une activité se décompose en tâches, sous tâches ... et pour finir en opérations. Une opération est une partie de l'activité qui ne peut plus être décomposée. Par exemple, dans l'activité se laver les dents, prendre sa brosse à dents est une opération.

Bradykinésie :
Ralentissement significatif de l'initiation volontaire d'une séquence de d'actions (la personne va mettre plusieurs minutes avant de commencer à se brosser les dents, puis par la suite effectuer la séquence d'actions à un rythme presque normal)

Chunk :
Unité cognitive indépendante d'information de grandeur arbitraire. Un chiffre, une lettre, un mot constituent des chunks pour la majorité des adultes.

Cognition :
Fonction complexe multiple regroupant l'ensemble des activités mentales (pensée, perception, action, volonté, mémorisation, rappel, apprentissage) impliquées dans la relation de l'être humain avec son environnement et qui lui permettent d'acquérir et de manipuler des connaissances (associations, rétroaction, traitement de l'information, résolution de problèmes, prise de décision etc...)

Ergothérapeute :

Personne qui, après avoir évalué les capacités physiques et mentales d'un patient aux prises avec des difficultés entravant ses activités quotidiennes, propose un programme individuel d'activités pour renforcer les habiletés du patient et, au besoin, des aides techniques et un aménagement de son environnement pour pallier ses difficultés.

Répétition d'un stéréotype :

Tendance à répéter des actions de manière stéréotypée, la répétition des actions interviendra toujours de la même manière, ou sera toujours liée à la même action.

Schéma (sens de Cooper et Shallice) :

Ensemble partiellement ordonné de sous buts constituant une méthode de réalisation d'un but global, méthode agissant sur l'environnement. Exemple : le schéma "préparation du café" est une méthode pour réaliser le but : préparer un café. Ce schéma est constitué des sous buts ici non ordonnés : "mettre du café instantanée dans tasse", "mettre eau dans tasse", " mettre sucre dans tasse".

Effet d'éventail (fan effect) :

Effet montrant que plus des sujets apprennent de faits relatifs à un concept, plus le temps nécessaire pour récupérer un fait concernant ce concept augmente.

Annexe B

Plan d'un appartement intelligent

Cette annexe présente le plan d'un appartement intelligent avec les spécifications des capteurs et des effecteurs présents dans les différentes pièces dans les différentes pièces.

Notes et précisions générales sur les capteurs

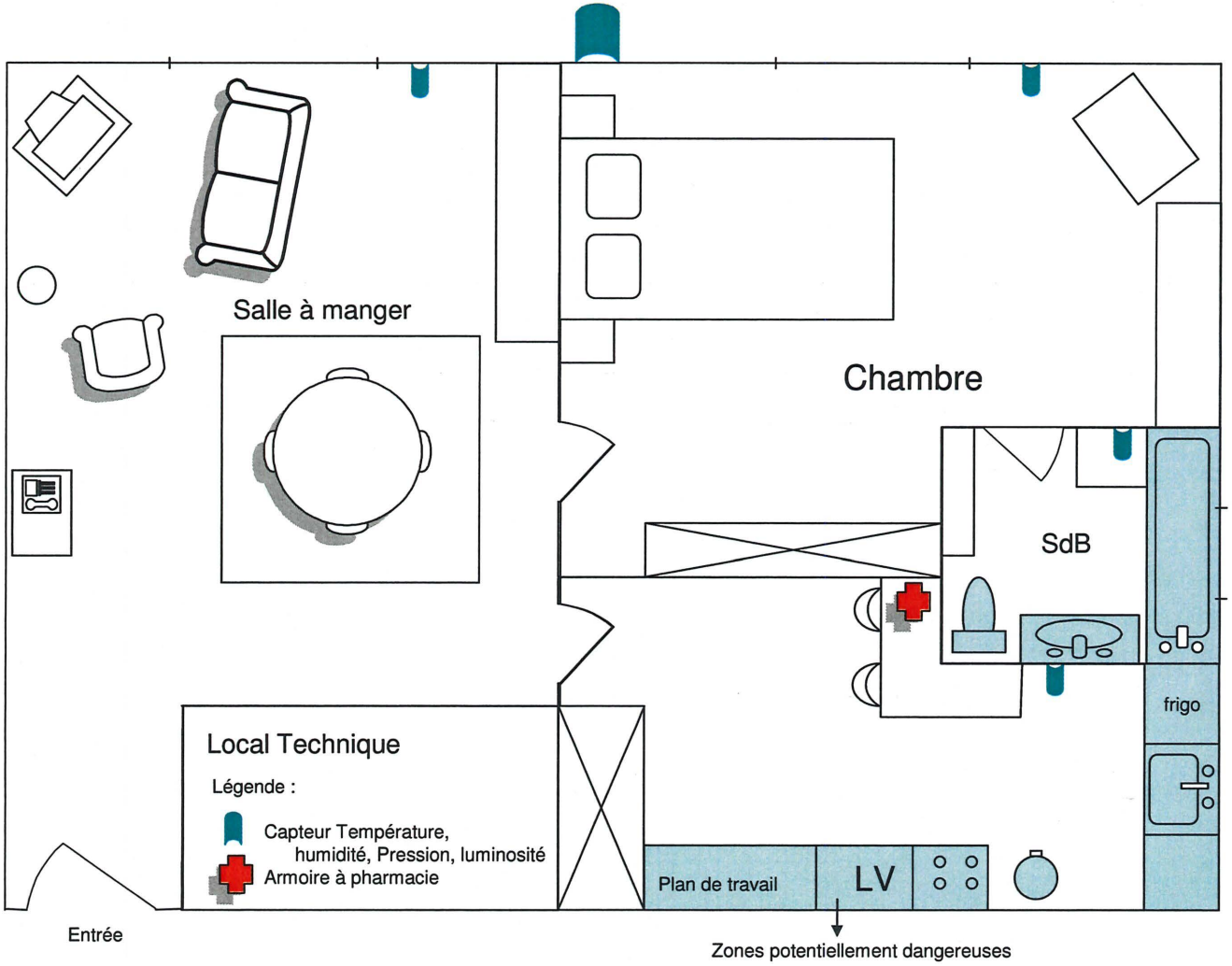
1) Les données techniques des capteurs ne sont pas encore disponibles, le choix des capteurs adéquat n'a pas encore été fait. Seuls les capteurs infrarouge sont à l'étude (projet d'Ezzat Demnati).

2) Ceci présente de façon générale, l'ensemble des capteurs qui seraient nécessaires à l'interaction avec l'occupant, la pertinence de chaque capteur pouvant être encore sujette à discussion. Pour chaque pièce est présenté un tableau expliquant le rôle et l'utilité des capteurs disposés dans l'appartement.

3) Les capteurs et moyens de communications ont été placés en se basant sur l'hypothèse que la personne s'occupe elle-même de toutes les activités liées à la vie quotidienne (lessive, ménage, etc. ...). Dans l'évolution des troubles cognitifs, l'intervention d'une personne extérieure pour la lessive ou le ménage peut par la suite être envisagée, ce qui rendrait certains capteurs inutiles.

4) Les capteurs présentés ici sont des capteurs qui sont disponibles dans le commerce ou auprès de fabricants spécialisés. Il pourrait être intéressant notamment de voir du coté des avancées en domotique (régulateur de lumière, de chaleur, déclenchement à distance ou programmé de certaines activités telle que se faire couler un bain...

1) Schéma général de l'appartement

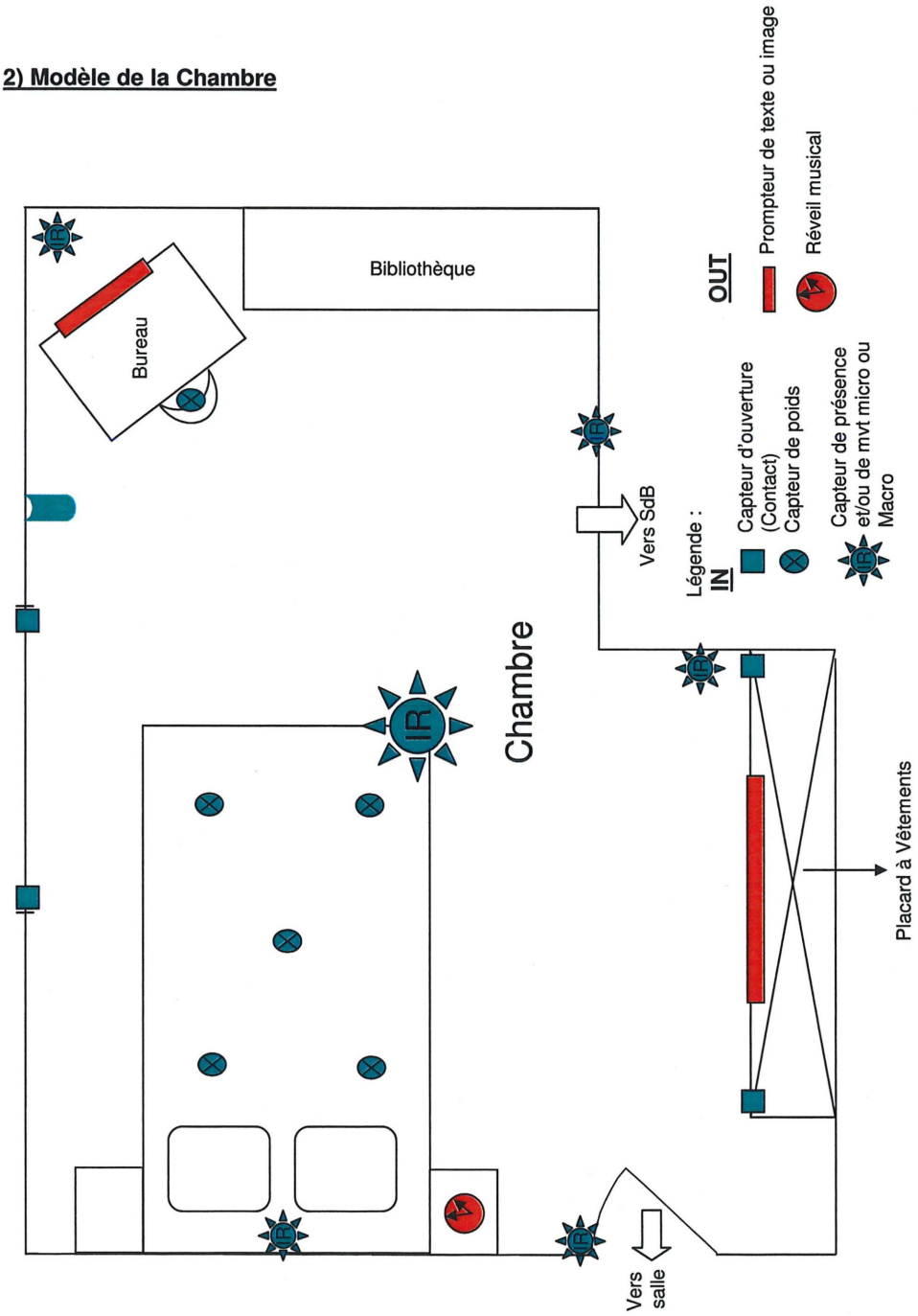


1) Schéma général de l'appartement : Description des capteurs

Type du Capteur	Localisation	Rôle	Destiné à pallier les risques
Capteur de Température, humidité, pression, luminosité	1 dans chaque pièce	Ces capteurs doivent permettre la régulation automatique de la climatisation, du chauffage et de l'éclairage	➢ de froid par mauvais chauffage ➢ de malaises : T° trop élevées ➢ d'aggraver des troubles de la vue par un mauvais éclairage ➢ prévenir les incendies ou les inondations
Capteur de Température, humidité, pression, luminosité	1 à l'extérieur	Permet de suggérer l'habillement et aussi de réguler les différences intérieur extérieur	➢ d'être malade par des changements de température importants ➢ d'être malade à cause d'un habillement inadapté

Notes : La régulation automatique prévue est toujours dépendante des volontés de l'occupant tant que le système estime que celui est capable de bien le faire. Pour des stades avancées de la maladie d'Alzheimer par exemple, l'occupant n'aura plus le droit de s'en occuper.

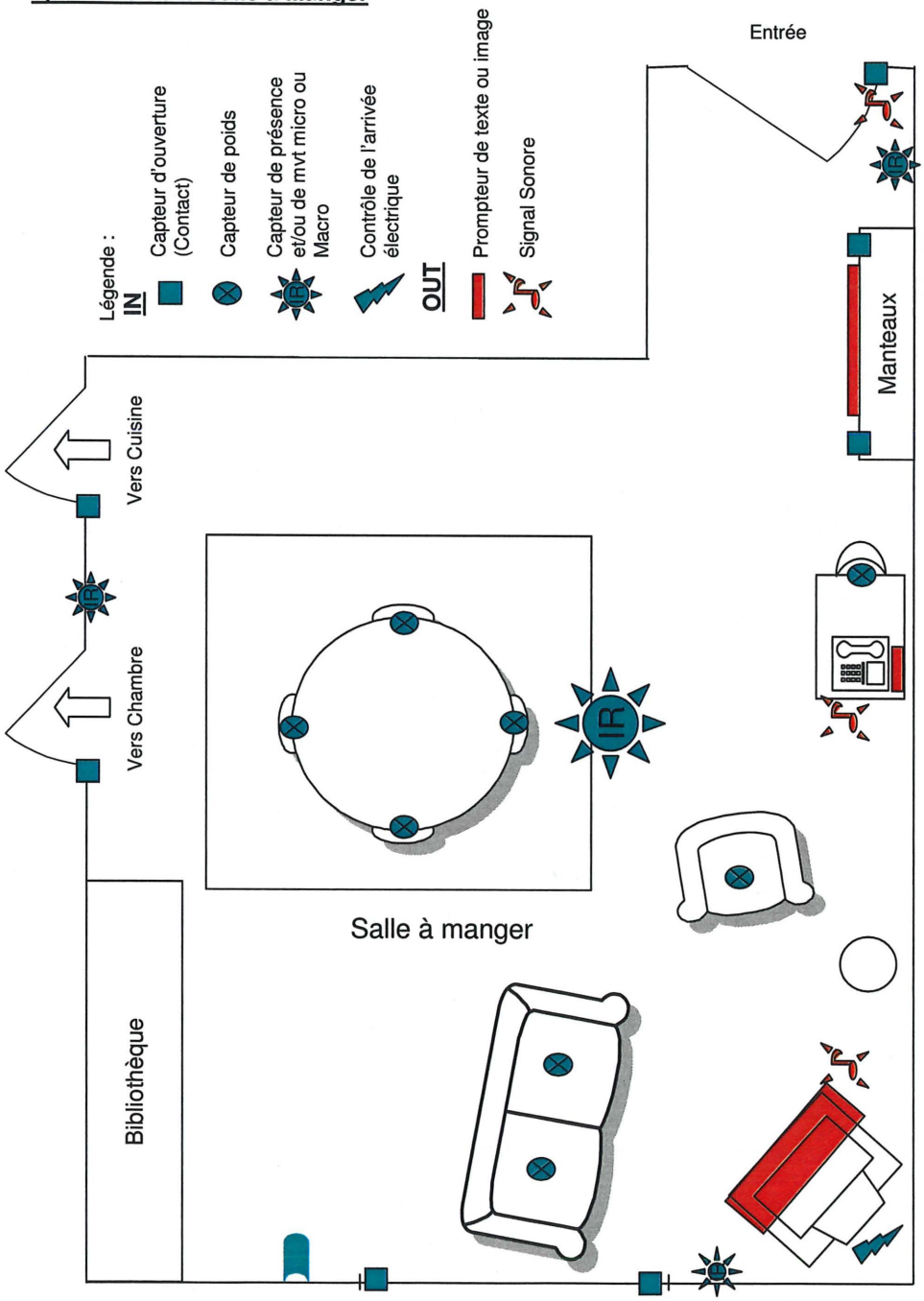
2) Modèle de la Chambre



2) Modèle de la Chambre : descriptions des capteurs

Type du Capteur	Localisation	Rôle	Destiné à pallier les risques
Capteur de poids	Lit	Savoir si la personne est sur le lit	➢Manque de sommeil
	Chaise de bureau	Savoir si la personne est assise à son bureau	
Capteur d'ouverture	Fenêtre	Éviter des ouvertures inopportunes	➢Froid extérieur, pluie ➢Défenestration
	Placard	Couplé au prompteur pour un habillement correct	➢Habillement incorrect
Capteur IR micro	Divers emplacements	Situer plus précisément la personne pour que le système puisse appréhender les taches en cours	➢Malaises ou chutes (action si la personne ne réagit pas …)
Capteur IR Macro	Plafond centre de la pièce	Situer la personne dans l'appartement	➢Malaises ou chutes (action si la personne ne réagit pas …)
Prompteur	Bureau	Suggestion d'actions	➢Oubli, désorganisation
	Placard	Suggestions d'habillement	➢Habillement incorrect
Réveil	Table de nuit	Réveil et suggestion de démarrage de la journée, premier guide (pourrait donner le tps, la date…)	

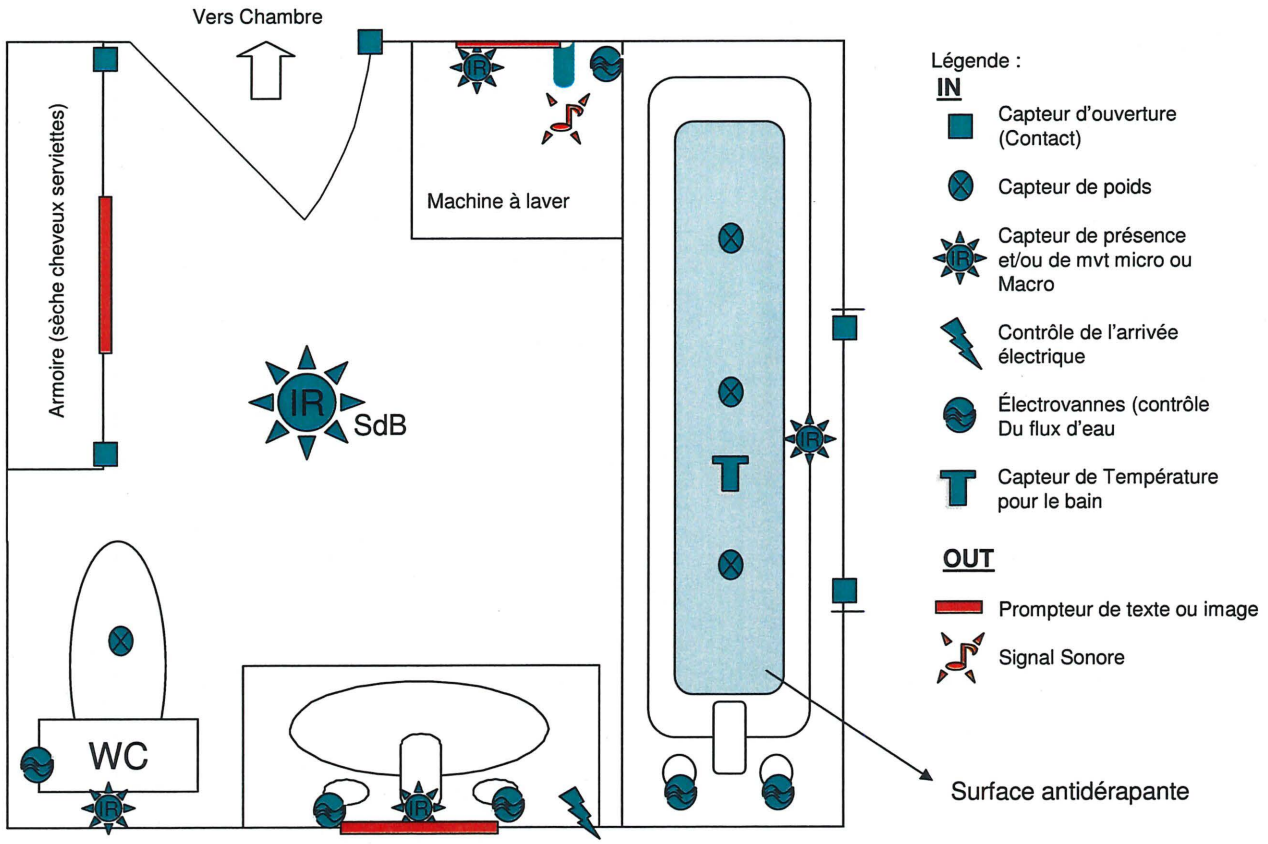
3) Modèle de la Salle à Manger



3) Modèle de la Salle à Manger : description des capteurs

Type du Capteur	Localisation	Rôle	Destiné à pallier les risques
Capteur de poids	Table	Localisation précise	
	Salon	Savoir si on peut utiliser la TV pour les messages…	
	Téléphone	Détecter si éventuellement actions concourantes	➢Oubli d'une autre tâche
Régulateur électrique	TV	Savoir si la télé est allumée, plus empêcher risques de court-circuit.	➢Prévenir risque d'électrocution, de départ de feu du à un court-circuit
Capteur d'ouverture	Fenêtre	Éviter des ouvertures inopportunes	➢Froid extérieur, pluie ➢Défenestration
	Placard à manteaux	Couplé au prompteur pour un habillement correct	➢Habillement incorrect
Capteur IR micro	Divers emplacements	Situer plus précisément la personne pour que le système puisse appréhender les tâches en cours	➢Malaises ou chutes (action si la personne ne réagit pas …)
Capteur IR Macro	Plafond centre de la pièce	Situer la personne dans l'appartement	➢Malaises ou chutes (action si la personne ne réagit pas …)
Prompteur	TV	Suggestion d'actions	➢Oubli, désorganisation
	Placard	Suggestions d'habillement	➢Habillement incorrect
	Téléphone	Rappel d'une action en cours Suggestion appel de service en cas de problèmes	➢Oubli d'une autre tâche
Signaux sonores	Entrée	Sonnette, suggestion d'habillement	➢Habillement incorrect
	Téléphone	Soutien au prompteur correspondant	➢Oubli d'une autre tâche

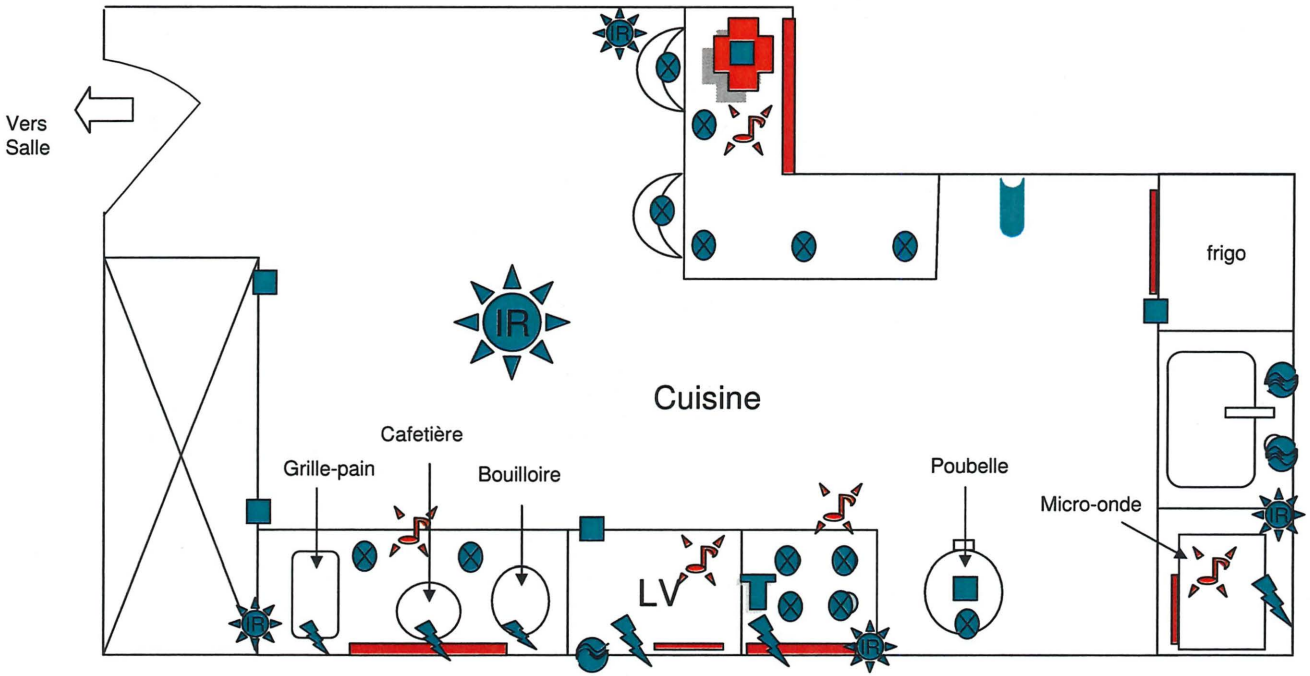
4) Modèle de la Salle de Bains



4) Modèle de la Salle de Bains : Description des capteurs

Type du Capteur	Localisation	Rôle	Destiné à pallier les risques
Capteur de poids	Baignoire	La personne prends un bain ou une douche	➢Mauvaise hygiène
	WC	Savoir si la personne utilise régulièrement les WC	➢Mauvaise utilisation des sanitaires
Régulateur électrique	Près du Lavabo	Éviter court-circuit du à une mauvaise utilisation sèche cheveux ou rasoir	➢Prévenir risque d'électrocution, de départ de feu du à un court-circuit
Capteur d'ouverture	Fenêtre	Cf. description précédente	➢Cf.. description précédente
	Placard	Couplé au prompteur	➢Mauvaise planification de la toilette…
IR Micro et Macro	Divers emplacements	Cf. description précédente	➢Cf... description précédente
Capteur de Température	Baignoire	Mesurer la température de l'eau	➢Brûlures (eau chaude), coup de froid (eau froide)
Électrovannes	Robinets Lavabo	Réguler le flux d'eau en terme de température et quantité	➢Inondations, brûlures
	Robinets Baignoire	Réguler le flux d'eau en terme de température et quantité	➢Inondations, brûlures
	Machine à laver	Réguler le flux d'eau en terme de température et quantité	➢Inondations
	WC	Savoir si la personne utilise régulièrement les WC	➢Mauvaise utilisation des sanitaires
Prompteur	Lavabo	Suggestion d'actions (brossage..)	➢Mauvaise planification de la toilette…
	Placard	Suggestions d'action (se sécher)	➢Mauvaise planification de la toilette…
	Machine à laver	Aide à une bonne utilisation	➢Mauvaise hygiène pour les vêtements
Signaux sonores	Machine à laver	Sonne par ex si la machine est finie…	

5) Modèle de la Cuisine



Légende :

IN

- Capteur d'ouverture (Contact)
- Capteur de poids
- Capteur de présence et/ou de mvt micro ou Macro
- Contrôle de l'arrivée électrique
- Électrovannes (contrôle Du flux d'eau
- Capteur de Température pour la cuisinière

OUT

- Prompteur de texte ou image
- Signal Sonore

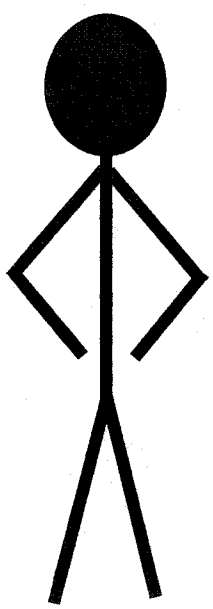
5) Modèle de la Cuisine : Description des Capteurs

Type du Capteur	Localisation	Rôle	Destiné à pallier les risques
Capteur de poids	Chaise cuisine	Localisation précise	
	Plaques électriques	Savoir si la personne utilise les plaques	➢Mauvaise utilisation de la cuisinière, feu
	Plan de Travail	Suivi des activités de la personne	➢Oubli, Désorganisation
	Poubelle	Savoir si la poubelle est pleine	➢Mauvaise hygiène
Régulateur électrique	Appareils électriques	Éviter court-circuit du à une mauvaise utilisation	➢Prévenir risque d'électrocution, de départ de feu du à un court-circuit
Capteur d'ouverture	Frigo	Fréquence d'utilisation, planification des repas	➢Sous-alimentation, mauvaise utilisation
	Placards	Si produit dangereux…	
	Lave-vaisselle	Utilisation correcte..	
	Placard à médicaments	Supervision des prises de médicaments	➢Intoxication, aggravation par manque de soins
	Micro-onde	Utilisation correcte	
IR Micro et Macro	Divers emplacements	Cf. description précédente	➢Cf. description précédente
Capteur de Température	Cuisinière	Surveillance	➢Brûlures, départ de feu
Électrovannes	Robinets Évier	Réguler le flux d'eau en terme de température et quantité	➢Inondations, brûlures
	lave-vaisselle	Réguler le flux d'eau en terme de température et quantité	➢Inondations

5) Modèle de la Cuisine : Description des Capteurs (Suite)

Type du Capteur	Localisation	Rôle	Destiné à pallier les risques
Prompteur	Plan de travail	Aide à l'utilisation des tous les appareils électriques	➢Brûlures, mauvaise utilisation
	lave-vaisselle	Aide à une bonne utilisation	➢Mauvaise hygiène pour la vaisselle
	Cuisinière	Suggestion de recettes, d'actions	➢Sous-alimentation, carence, départ de feu, brûlures
	Micro-onde	Suggestion d'utilisation, recommandations	➢Sous-alimentation, carence, départ de feu, brûlures
	Frigo	Suggestion de recettes, d'actions	➢Sous-alimentation, carence
	Armoire à pharmacie	Illustration des médicaments, aide à la prise	➢Intoxication, aggravation par manque de soins
Signaux sonores	Plan de travail	Aide à l'utilisation de tous les appareils électriques	
	lave-vaisselle	Sonne par ex si la vaiselle est propre …	➢Mauvaise hygiène pour la vaisselle
	Cuisinière	« ça chauffe, je suis prêt »	
	Micro-onde	« ça chauffe, je suis prêt »	
	Armoire à pharmacie	Rappel de la prise de médicaments	➢aggravation par manque de soins

Et l'occupant ?



Les capteurs portés par l'occupant

1) Les plus légers possibles
2) Les moins encombrants possibles
3) Les plus discrets possibles

IDEE :
Capteurs sous forme d'une **bague** portée par l'utilisateur, ou un **bracelet montre**

Quels Capteurs ?

- Tension(?), pression artérielle(?), rythme cardiaque : surveillance de l'état de santé
- Détection des chutes
- Identification de la personne
- Transfert d'information au sein de l'habitat
- Émotions (affective computing)

Références

- Hélène Pigot : Notes sur la description des capteurs
- Rialle V., Noury N. Hervé T., *An experimental Health Smart Home and its distributed Internet-based Information and Communication System : first stepsof a research project*, MEDINFO 2001, 10th World Congress on Medical Informatics. Amsterdam : IoPress. Pp. 1479-1483
- Bonder B. R. & Wagner M. B., *Functionnal Performance In Oder Adults*, Chapter 15 p.240-255, Édition F.A. Davis, 1994
- Chevalier Marie, *Les risques déterminants dans la décision de maintien à domicile d'une personne atteinte de démence de type Alzheimer*, Mémoire de maîtrise à l'université de Montréal, Mars 2000.

Annexe C

Composante Systèmes automatisés

C.1 Systèmes automatisés

Dans cette partie, nous allons expliquer le principe général de la composante et donner quelques pistes de réalisation. Il s'agit d'une composante sur laquelle il y a eu quelques réflexions et discussions. C'est pourquoi elles ont été placées en annexe.

C.1.1 Description des Systèmes automatisés

La figure représente les systèmes automatisés. L'idée de tels systèmes se rapproche de celle des thermostats automatiques qui existent à l'heure actuelle. Nous allons suivre un cycle du système pour bien appréhender son fonctionnement. Des données extérieures : température, luminosité, humidité ainsi que des données intérieures, température dans chaque pièce, luminosité, vont être collectées. Elles vont permettre au système de fonctionner en ajustant en permanence la température des chauffages ou de la climatisation pour atteindre les préférences programmées par l'occupant. En ce qui concerne la luminosité, il s'agira d'une répercussion encore plus directe puisqu'il s'agira d'augmenter la lumière ou de la diminuer.

Les préférences de l'occupant seront stockées dans l'instance du modèle de la personne. Les thermostats et autres variateurs de lumières vont ici servir d'interface avec l'utilisateur. Les données seront donc également des paramètres de température et de

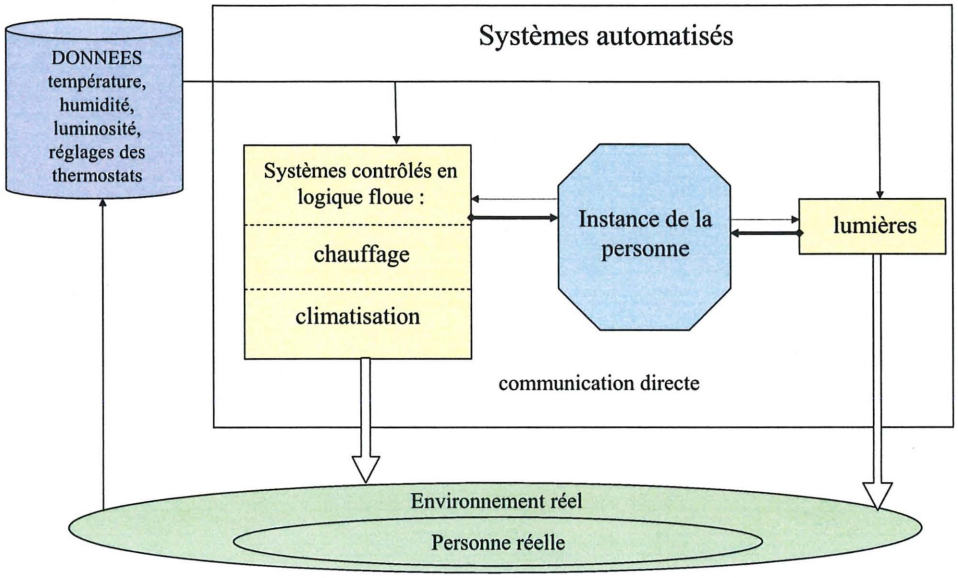


FIG. C.1 – Composante Systèmes automatisés

lumière réglés par l'occupant via le thermostat ou le variateur. Ces paramètres seront sauvegardés dans le modèle de la personne. On y trouvera également un drapeau qui permettra de dire si oui ou non on prend en compte les réglages réels faits par l'occupant via les thermostats. Pourquoi cette mesure ? Si les compétences de la personne viennent à se détériorer, la personne ne sera plus capable de régler elle-même son chauffage ou la climatisation de manière satisfaisante ou tout simplement, elle ne s'en souciera pas. Il arrive donc un moment où le réglage du chauffage et de la climatisation deviendra complètement automatique. Les seuls paramètres pris en compte seront ceux qui ont été enregistrés par le système auparavant.

Le réglage automatique des lumières sera un peu particulier, il consistera à augmenter l'éclairage de la pièce où se trouve l'occupant au fur et à mesure que la nuit tombe.

Bien évidemment, si la personne décide de demeurer dans la pénombre en éteignant certaines lampes, le système ne devra pas les rallumer aussitôt. Ici il est probable que le système global reprenne le dessus à partir du moment où le système automatique aura été contourné (i.e. la personne éteint les lumières). De plus, le système pourra gérer d'allumer légèrement la lumière ou une veilleuse lorsque le mouvement d'une personne est détectée dans une pièce plongée dans le noir.

Pourquoi tant d'intérêt pour l'aspect lumière de l'habitat ? Certaines études médicales ont montré que le noir ou le manque de lumière plus simplement, provoquent, notamment chez les schizophrènes, des angoisses qui peuvent déclencher des crises plus graves. Aussi l'ajustement automatique de la luminosité lors de la tombée de la nuit va permettre de diminuer certains risques. Pour d'autres patients, ceci peut permettre d'éviter les chutes ou d'aggraver des troubles de la vue, si par exemple, la personne lit son livre dans la pénombre grandissante. Les lumières seront contrôlées par le système global, elles permettront en dernier recours de prévenir de manière efficace l'occupant, par exemple en éteignant et rallumant toutes les lumières de la maison plusieurs fois.

C.1.2 Pistes de réalisations

Les systèmes de chauffage et de climatisation pourront être réalisés à l'aide de contrôleurs en logique floue car ces systèmes offrent l'avantage d'engendrer moins de variations de température qu'un thermostat simple, le contrôle est plus fin, et de plus, cela permet de prendre en compte plusieurs paramètres. Enfin, d'un point de vue réalisation, il y a eu de nombreuses applications de contrôleur de température dans l'industrie et la littérature, aussi il ne s'agira pas d'un procédé trop complexe à mettre en place ni trop coûteux.

En ce qui concerne les variations de lumière, le contrôle se fera via des variateurs qui seront en partie contrôlés par le système. La lumière est un système plus simple puisque sa diffusion est immédiate contrairement à la chaleur ou à la fraîcheur. De ce fait, le système peut directement indiquer qu'il veut plus ou moins de luminosité, la répercussion sera immédiate sur l'environnement. Du point de vue de l'occupant, le variateur de lumière suffit. Dès que l'intervention sur les lumières sera plus subtile, cela sera du ressort de l'assistance cognitive.

Bibliographie

Abowd, G. D., Battestini, A., et O'Connell, T. 2002. Location service : A framework for handling multiple location sensing technologies. PDF en ligne. `http://www.cc.gatech.edu/fce/ahri/publications/location_service.pdf`.

Abowd, G. D., Battestini, A., et O'Connell, T. 2003. Site web du Aware Home. Georgia institute of technologies. `http://www.cc.gatech.edu/fce/ahri/`.

Altmann, E. M. et Trafton, J. G. 2002. Memory for goals : An activation-based model. *Cognitive Science*, 26 :39–83.

Anderson, J. R. 1974. Retrieval of propositional information from long-term memory. *Cognitive Psychology*, 5 :451–474.

Anderson, J. R. 1983. *The architecture of cognition.* Havard University Press, Cambridge, MA.

Anderson, J. R. 1993. *Rules of the Mind.* Erlbaum, Hillsdale, NJ.

Anderson, J. R., Bothell, D., Byrne, M., et Lebiere, C. 2002. An Integrated Theory of the Mind. `http://act-r.psy.cmu.edu/papers/403/IntegratedTheory.pdf`.

Anderson, J. R. et Bower, G. H. 1973. *Human associative memory.* Winston and Sons, Washington.

Anderson, J. R. et Lebiere, C. 1998. *The atomic components of thought.* Erlbaum, Mahwah, NJ.

Baddeley, A. D. 1986. *Working Memory.* Clarendon Press, Oxford.

Baddeley, A. D. 1990. *Human Memory : Theory and Practice.* Erlbaum, Hillsdale, NJ.

Baddeley, A. D. jan 1992. Working Memory. *Science*, 255 :556–559.

Bonder, B. R. et Wagner, M. B. 1994. *Functionnal Performance In Oder Adults.* F.A. Davis.

Canadian Study of Health and Aging Working Group. 1994. Canadian Study of Health and Aging : Study methods and prevalence of dementia. *Canadian Medecine Association Journal*, 150(6) :899–913.

Chevalier, M. mars 2000. Les risques déterminants dans la décision de maintien à domicile d'une personne atteinte de démence de type Alzheimer. Master's thesis, Université de Montéal, Québec.

Cooper, R. et Shallice, T. 1995. Soar and the case for unified theories of cognition. *Cognition*, 55 :115–149.

Cooper, R. et Shallice, T. 2000. Contention Scheduling and the control of routine actvities. *Cognitive Neuropsychology*, 17(4) :397–338.

Cooper, R. P., Yule, P., Fox, J., et Sutton, D. 1998. COGENT : An environment for the development of cognitive models. In Schmid et al. (1998), 55–82.

Das, S. K., Fox, J., Elsdon, D., et Hammond, P. 1996. A flexible architecture for autonomous agents. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(4) : 407–440.

Duchêne, F., Rialle, V., et Noury, N. May 2002. Télésurveillance médicale à domicile : proposition d'une architecture pour un système de détection de situations critiques et de décisions sur l'état du patient. In *Neuvièmes Journées Francophones d'Informatique Médicale (JFIM)*, Québec. Springer-Verlag.

Ernst, R. L., Hay, J. W., Fenn, C., Tinklenberg, J., et Yesavage, J. 1997. Cognitive function and the costs of Alzheimer's disease. An Exploratory Study. *Archives of Neurology*, 54(6) :687–693.

Essa, I. 1999. Ubiquitous sensing for smart and aware environments : Technologies towards the building of an Aware Home. `http://www.cc.gatech.edu/gvu/perception/pubs/se99/pp.pdf`. Position Paper for the DARPA/NSF/NIST Workshop on Smart Environments.

Fine, S., Singer, Y., et Tishby, N. 1998. The Hierarchical Hidden Markov Model : Analysis and Applications. *Machine Learning*, 32 :41–62.

Fougeyrollas, P., Majeau, P., Tremblay, G., St Michel, G., Dallaire, P., Anjou, C., Lachance, M., et Cloutier, C. 1993. *Guide de formation sur les systèmes de classification des causes et des conséquences des maladies, traumatismes et autres troubles.* Réseau international sur le processus de production du handicap, Québec, Canada.

Fox, J. et Das, S. K. 1996. A unified framework for hypothetical and practical reasonning(2) : lessons from medical applications. In *Proocedings of International Conference on Formal and Applied Practical Reasoning.* Springer-Verlag.

Fox, J. et Das, S. K. 2000. *Safe and Sound : Artificial Intelligence in Hazardous Applications.* American Association of Artificial Intelligence and MIT Press, Cambrigde, MA.

Gamma, E., Helm, R., Johnson, R., et Vlissides, J. 1995. *Design Patterns.* Addison-Wesley.

Gilhooly, K. J., Logie, R. H., Wetherick, N. E., et Wynn, V. 1993. Working memory and strategies in syllogistic-reasoning tasks. In *Memory and Cognition*, 21, 115–124.

Glasspool, D. W. Avril 2000. The integration and control of behaviour : Insights from neuroscience and AI. AISB Convention. In Symposium How to design a functionnal mind.

Glasspool, D. W. et Cooper, R. P. 2002. Executive processes. In Cooper, R. P., éditeur, *Modeling High-Level Cognitive Process*, chapter 8, 313–362. LEA.

Grant, S. 1995. Developing cognitive architecture for modelling and simulation of cognition and error in complex tasks. `http:www.simongrant.org/pubs/val/text.html`.

Guralnik, V. et Haigh, K. Z. Juillet 2002. Models of Human Behaviour with Sequential Patterns. In *AAAI-02 Workshop on Automation as Caregiver : The Role of Intelligent Technology in Elder Care*, 24–30.

Haigh, K. Z., Geib, C. W., Miller, C. A., Phelps, J., et Wagner, T. Juillet 2002a. Agents for recognizing and responding to the behaviour of an elder. In *AAAI-02 Workshop on Automation as Caregiver : The Role of Intelligent Technology in Elder Care*, 31–38.

Haigh, K. Z., Phelps, J., et Geib, C. W. Juillet 2002b. An open agent architecture for assisting elder independence. In *The First International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 578–586.

Halford, G. S. 1993. *Children's understanding : The development of mental models.* Lawrence Erlbaum Associates, Hillsdale, NJ.

Honeywell. 2003. `http://www.honeywell.com/index.html`.

Huang, J., Jennings, N. R., et Fox, J. 1994. Cooperation in distributed medical care. In *Second international Conference on Cooperative Information Systems.*

Kautz, H. Novembre 2002. Assisted Cognition : Or, making sense of sensors. `http://www.cs.washington.edu/homes/kautz/talks/kautz-colloq.htm`.

Kautz, H. 2003. Assisted Cognition Project. Université Washington à Seattle. `http://www.cs.washington.edu/assistcog/`.

Kautz, H., Etzioni, O., Fox, D., Boriello, G., et Arnstein, L. 2001. An overview of the Assisted Cognition Project. `http://www.cs.washington.edu/homes/kautz/papers/acaaai02.pdf`.

Kautz, H., Etzioni, O., Fox, D., Weld, D., et Shastri, L. Mars 2003. Foundations of Assisted Cognition Systems. Technical Report CSE-03-AC-01, Université de Washington à Seattle. `http://www.cs.washington.edu/homes/kautz/papers/ac03tr.pdf`.

Kidd, C. D., Orr, R. J., Abowd, G. D., Atkeson, C. G., Essa, I. A., MacIntyre, B., Mynatt, E., Starner, T. E., et Newstetter, W. Octobre 1999. The Aware Home : A living laboratory for ubiquitous computing research. In *In The Proceedings of the*

Second International Workshop on Cooperative Buildings. http://www.cc.gatech.edu/fce/ahri/publications/cobuild99_final.PDF.

Kieras, D. E. et Meyer, D. E. 1995. An overview of the EPIC archiecture for cognitiion and performance with application to human-computeur interaction. Technical Report 5, University of Michigan.

Kieras, D. E. et Meyer, D. E. 1998. The epic architecture : principles of operation. ftp://www.eecs.umich.edu/people/kieras/EPIC/EPICArch.pdf.

Kieras, D. E., Meyer, D. E., Ballas, J. A., et Lauber, E. J. 2000. Modern computational perspectives on executive mental processes and cognitive control : Where to from here ? In Monsell, S. et Driver, J., éditeurs, *Attention and Performance, XVIII : Control of Cognitive Processes*, 681–712. MIT Press, Cambrigde, MA.

LAAS, Laboratoire du CNRS de Toulouse. 2003. Projet Prosafe. http://www.laas.fr/MIS/surveillance/Websurv.htm.

Laboratoire TIMC-IMAG. 2003. Habitat Intelligent pour la Santé (HIS). Faculté de Médecine et CHU de Grenoble. http://www-timc.imag.fr/AFIRM/SiteHIS/index.html.

Laird, J., Newell, A., et Rosenbloom, P. 1987. Soar : An architecture for general intelligence. *Artificial Intelligence*, 33.

Marcus, G. 2001. *The algebraic mind : Integrating connectionism and cognitive science.* MIT Press, Cambridge, MA.

Mayers, A. avril 1997. *Miace : Une architecture théorique et computationnelle de la cognition humaine pour étudier l'apprentissage.* PhD thesis, Montréal : Université de Montréal.

Meyer, D. E. et Kieras, D. E. 1997a. A computational theory of executive cognitive processes and multiple-task performance. part 1 : Basic mechanisms. *Psychological Review*, 104(1) :3–65.

Meyer, D. E. et Kieras, D. E. 1997b. A computational theory of executive cognitive processes and multiple-task performance. part 2 : Accounts of the psychological refractory-period phenomena. *Psychological Review*, 104(4) :749–791.

Miller, G. A. 1956. The magical number seven plus or minus two : Some limits on our capacity for processing information. *Psychological Review*, 63 :81–97.

Moore, D., Essa, I., et Hayes, M. Mars 1999. Exploiting human actions and object context for recognition tasks. In *Proceedings of IEEE International Conference on Computer Vision*, Corfou, Grèce.

Mynatt, E. D. et Rogers, W. A. 2002. Developing technology to support the functional independence of older adults. In *Aging International*. `http://www.cc.gatech.edu/fce/ecl/publications/functionalIndependence-agingInternational2002.pdf`.

National Institute for Aging. 2000. Progress Report on Alzheimer's disease. `http://www.alzheimers.org`.

National Institute of Standards and Technology and Honeywell Laboratories. I.L.S.A. `http://www.htc.honeywell.com/projects/ilsa/researchers.html`.

Newell, A. 1973a. Productions systems : Models of control structures. In Chase, W. G., éditeur, *Visual Information Processing*, 463–526. New York : Academic Press.

Newell, A. 1973b. You can't play 20 questions with nature and win : Projective comments on the papers of this symposium. In Chase, W. G., éditeur, *Visual Information Processing*. New York : Academic Press.

Newell, A. 1981. Reasoning, problem solving and decision processes : The problem space as a fundamental category. In Nickerson, R., éditeur, *Attention and Performance*, 8. Hillsdale, NJ : Erlbaum.

Newell, A. 1990. *Unified Theories of Cognition*. Harvard University Press, Cambridge, MA.

Newell, A. et Rosenbloom, P. S. 1981. Mechanisms of skill acquisition and the law of practice. In Anderson, J. R., éditeur, *Cognitive Skills and their Acquisition*. Hillsdale, NJ : Lawrence Erlbaum Associates.

Norman, D. A. 1981. Categorisation of action slips. *Psychological Review*, 88(1) :1–15.

Norman, D. A. et Shallice, T. 1980. *Attention to action : Willed and automatic control of behaviour.* University of California, San Diego, CA. Chip Report 99.

Norman, D. A. et Shallice, T. 1986. Attention to action : Willed and automatic control of behavior. In Davidson, R. J., Schwarts, G. E., et Shapiro, D., éditeurs, *Consciousness and self-regulation. Advances in research and theory*, 4, 1–18. New-York : Plenum Press.

Noury, N., Hervé, T., Rialle, V., Virone, G., et Mercier, E. Octobre 2000. Monitoring behavior in home using a smart fall sensor and position sensors. In *Proceedings of Microtechnologies in Medicine and Biology conference*, 607–610, Lyon, France.

Orr, R. J. et Abowd, G. D. 2000. The Smart Floor : A mechanism for natural user identification and tracking. In *Proceedings of the 2000 Conference on Human Factors in Computing Systems*, The Hagues, Pays-Bas.

Pigot, H., Mayers, A., et Giroux, S. avril 2003. The intelligent habitat and everyday life activity support. In *5ieme conférence internationale in Simulations in Biomedecine*, Slovénie.

Potter, M. C. 1990. Remembering. In Osherson, D. N. et Smith, E. E., éditeurs, *Thinking*, 3, 3–32. Cambridge, Massachusetts : The MIT Press.

Potter, M. C. 1993. Very short-term conceptual memory. In *Memory and Cognition*, 21, 156–161.

Reason, J. T. 1979. Actions not as planned : The price of automatization. In Underwood, G. et Stevens, R., éditeurs, *Aspects of consciousness*, chapter 4, 67–89. London : Academic Press.

Reason, J. T. 1984. Lapses of attention in everyday life. In Parasuraman, W. et Davies, R., éditeurs, *Varieties of attention*, chapter 14, 515–549. Orlando, FL : Academic Press.

Reason, J. T. 1990. *Human error.* Cambridge University Press.

Reichle, E. D., Pollatsek, A., Ficher, D. L., et Rayner, K. 1998. Toward a model of eye movement control in reading. *Psychological Review*, 105 :125–127.

Reichle, E. D., Rayner, K., et Pollatsek, A. 1999. Eye movement control in reading : Accouting for initial fixation locations and refixations within the E-Z Reader model. *Vision Research*, 39 :4403–4411.

Rialle, V. 2002. Health Smart Home, information technology for patients at home. *Telemedecine Journal et E-Health*, 8(4).

Rialle, V., Noury, N., et Hervé, T. 2001. An experimental health smart home and its distributed internet-based information and communication system : first steps of a research project. In et al, V. P., éditeur, *MEDINFO 2001, 10th World Congress on Medical Informatics*, 1479–1483. Amsterdam : IOPress.

Rosenbloom, P. S., Newell, A., et Laird, J. E. 1989. Toward the knowledge level in soar : The role of the architecture in the use of knowledge. In Van Lehn, K., éditeur, *Architectures for intelligence*, 75–111. Erlbaum, Hilsdale NJ.

Rousseau, J., Potvin, L., Dutil, E., et Falta, P. 2002. Model of competence : A conceptual framework for understanding the person-environment interaction for persons with motor disabilities. In *A New Addition to Occupational Therapy in Health Care*, 16.

Salvucci, D. D. 2001. An integrated model of eye movements and visual encoding. *Cognitive Systems Research*, 1 :201–202.

Schmid, U., Krems, J. F., et Wysotzki, F., éditeurs. 1998. *Mind Modelling : A Cognitive Science Approach to Reasoning, Learning and Discovery.* Pabst Science Publishers : Lengerich, Deutschland.

Schmidt, R. A. 1975. A schema theory of discrete motor skill learning. *Psychological Review*, 82(4) :225–260.

Schwartz, M. F., Reed, E. S., Montgomery, M. W., Palmer, C., et Mayer, N. H. 1991. The quantitative description of action disorganisation after brain damage : A case study. *Cognitive Neuropsychology*, 8(5) :381–414.

Shallice, T. et Burgess, P. 1993. Supervisory control of action and thought selection. In Baddeley, A. D. et Weiskrantz, L., éditeurs, *Attention : Selection, Awareness, and Control (A tribute to Donald Broadbent)*, 171–187. Clarendon Press : Oxford.

Shallice, T. et Burgess, P. 1996. The domain of supervisory processes and temporal organization of behaviour. In *Philosophical transactions of the Royal Society of London*, number B. 351, 1405–1412.

Shastri, L. et Ajjanagadde, V. 1993. From simple associations to systematic reasoning : a connectionist encoding of rules, variables and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences*, 16(3) :417–494.

Shenk, D. 2001. *The Forgetting : Alzheimer's : Portrait Of An Epidemic.* NY : Doubleday, New York.

Touchon, J. 1999. La maladie d'Alzheimer. `http://www.maladie-alzheimer.com/maladie/home.html`.

Wolfe, J. M. 1994. Guided search 2.0 : A revised model of visual search. *Psychonomic Bulletin and Review*, 1 :202–238.