

UNIVERSITÉ DE MONTRÉAL

Étude et modélisation des connaissances et
raisonnement de l'apprenant dans un STI

par

Mohammad-Ebrahim Shiri-Ahmadabadi

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures
en vue de l'obtention du grade de
Philosophiæ Doctor (Ph.D.)
en informatique

Septembre, 1999

© Mohammad-Ebrahim Shiri-Ahmadabadi, 1999



Université de Montréal

Faculté des études supérieures

Cette thèse intitulée :

Étude et modélisation des connaissances et
raisonnement de l'apprenant dans un STI

présentée par

Mohammad-Ebrahim Shiri-Ahmadabadi

a été évaluée par un jury composé des personnes suivantes :

Michel Boyer	Président rapporteur
Esmâ Aïmeur	Directeur de recherche
Claude Frasson	Co-directeur de recherche
François Lustman	Membre du jury
Ramon López de Mántaras	Examineur externe
Pierre Bordeleau	Représentant du doyen de la FES

Thèse acceptée le : 1999

Sommaire

Les Systèmes Tutoriels Intelligents (STI) ont pour objet de réaliser, à l'aide d'un ordinateur, un enseignement individualisé. Ces systèmes s'adaptent dynamiquement à la formation en utilisant diverses stratégies pédagogiques selon les performances de l'apprenant et les informations incluses dans le modèle de ce dernier. Bien que le domaine de modélisation de l'apprenant a fait l'objet de plusieurs études, les difficultés liées à la conception et au développement d'un tel domaine, et précisément à l'analyse du raisonnement d'un apprenant dans un contexte de formation restent encore des problèmes majeurs. Une des grandes problématiques de l'analyse des raisonnements repose sur la difficulté à identifier les différentes méthodes employées par l'apprenant lors de la résolution d'un problème. En appliquant les techniques du Raisonnement à Base de Cas (RBC, ou Case-Based Reasoning (CBR)), nous tentons de résoudre ce problème dans la présente thèse.

Notre démarche consiste à présenter un problème à l'étudiant, à fournir à sa demande des informations sur les cas similaires (au problème soumis) et à inférer sa méthode de résolution en se référant à celles de l'expert. Dans ce cadre, nous avons identifié deux modes d'interactions : *AS mode (Accessible Solution mode)* et *US mode (Unaccessible Solution mode)*. Dans le premier mode la base de cas est accessible par l'apprenant et il peut y chercher les cas similaires au problème donné. En effet, il peut effectuer les processus d'adaptation sur les solutions de cas afin d'atteindre une solution au problème courant. Dans le deuxième mode la base de cas ne lui est pas accessible, il doit donc résoudre le problème (sans voir les solutions des cas) en se rappelant les étapes de solution(s) d'un (ou des) cas similaire(s) qu'il a déjà résolu(s). La solution de l'apprenant sera finalement analysée pour déterminer son mode de raisonnement et en déduire ses connaissances.

Un formalisme appelé Graphe Classifié Concepts et Relations (GCR) a été développé pour représenter les connaissances du domaine qui sont sous forme de cas et deux

prototypes de STI appelés SARA (Système d'analyse des Raisonnements de l'Apprenant) et SARA-II sont développés pour valider et tester l'approche.

Mots-clés : modélisation de l'apprenant par cas, modélisation du raisonnement, agent intelligent pour l'aide au raisonnement, système à base de cas, systèmes tutoriels intelligents.

Table des matières

SOMMAIRE	III
TABLE DES MATIÈRES	V
LISTE DES FIGURES	X
LISTE DES TABLEAUX	XII
LISTE DES ABRÉVIATIONS	XIII
REMERCIEMENTS	XV
INTRODUCTION	1
OBJECTIFS	3
PRÉSENTATION DE L'APPROCHE	4
PLAN DE LA THÈSE	5
CHAPITRE 1 - LES SYSTÈMES TUTORIELS INTELLIGENTS	8
1.1 INTRODUCTION.....	8
1.2 DÉVELOPPEMENT DES SYSTÈMES TUTORIELS : DE L'EAO VERS L'EIAO.....	9
1.3 SYSTÈME TUTORIEL INTELLIGENT : DÉFINITION ET OBJECTIFS.....	11
1.3.1 Définition.....	11
1.3.2 Le but des STI.....	12
1.3.3 Les domaines utilisés pour la conception des STI.....	12
1.4 ARCHITECTURE DE BASE D'UN STI.....	13
1.4.1 Le module expert	13
1.4.2 Le modèle de l'apprenant	14
1.4.3 Le module tuteur.....	15
1.4.4 Le module interface	15
1.5 LES STRATÉGIES TUTORIELLES.....	16
1.5.1 Tuteur classique (ou mode directif).....	16
1.5.2 Co-apprenant.....	16
1.5.3 Compagnon d'apprentissage	17
1.5.4 Tuteur inversé.....	17
1.5.5 Perturbateur	17
1.5.6 Apprentissage par double test	18
1.6 TYPOLOGIE DES STI.....	19
1.6.1 Les STI socratiques.....	19
1.6.2 Les STI de type guidage (Coach).....	19
1.6.3 Les STI démonstrateurs	20
1.6.4 Les systèmes critiques.....	20
1.6.5 Les systèmes sociaux	20
1.6.6 Les systèmes planificateurs.....	21
1.7 QUELQUES EXEMPLES DE STI.....	21
1.7.1 Quelques systèmes représentatifs	21
1.7.2 Autres systèmes.....	23
1.8 VERS LES STI INTÉGRÉS.....	23
1.8.1 Architecture générale d'un STI.....	23
1.9 PROBLÉMATIQUE DE DÉVELOPPEMENT DES STI.....	27
1.9.1 Les processus de développement des systèmes	27

1.9.2	<i>Les difficultés liées au développement des STI</i>	27
1.9.3	<i>Limites des systèmes tutoriels intelligents</i>	28
1.10	CONCLUSION.....	28
CHAPITRE 2 - MODÈLE DE L'APPRENANT		29
2.1	INTRODUCTION.....	29
2.2	UTILITÉ DU MODÈLE DE L'APPRENANT.....	30
2.3	PROBLÈME DE MODÉLISATION DE L'APPRENANT.....	30
2.4	LA VUE D'UN MODÈLE DE L'APPRENANT PAR SES COMPOSANTS	31
2.4.1	<i>La connaissance de l'apprenant</i>	32
2.4.2	<i>Composants du modèle de l'apprenant</i>	34
2.5	PROCESSUS DE DIAGNOSTIC.....	36
2.6	LES APPROCHES DE MODÉLISATION DE L'APPRENANT.....	38
2.6.1	<i>Modèle de Recouvrement</i>	38
2.6.2	<i>Modèle Correctionnel</i>	39
2.6.3	<i>Modèle de Reconstruction</i>	39
2.6.4	<i>L'approche de Goldstein (Graphe Génétique)</i>	39
2.6.5	<i>L'approche de John Anderson (La théorie ACT* (Adaptive Control of Thought))</i>	41
2.6.6	<i>Autres approches</i>	43
2.7	PRÉSENTATION DU MODÈLE DE L'APPRENANT DANS QUELQUES STI.....	43
2.8	DISCUSSION.....	48
2.9	CONCLUSION.....	50
CHAPITRE 3 - LE RAISONNEMENT À BASE DE CAS		51
3.1	INTRODUCTION.....	51
3.2	CONCEPTION DU RAISONNEMENT À BASE DE CAS	52
3.3	LES ORIGINES DU RAISONNEMENT À BASE DE CAS	52
3.3.1	<i>La mémoire dans les sciences cognitives</i>	53
3.3.2	<i>RBC et raisonnement par analogie</i>	55
3.4	LA MÉTHODOLOGIE DU RAISONNEMENT À BASE DE CAS.....	55
3.5	LE CYCLE DU RBC	56
3.6	ÉLÉMENTS CONSTITUTIFS DU RBC.....	57
3.6.1	<i>Recherche</i>	58
3.6.2	<i>L'organisation et l'indexation de la mémoire</i>	58
3.6.3	<i>Mesure de similarité</i>	60
3.6.4	<i>Adaptation</i>	63
3.6.5	<i>Révision et Recyclage</i>	65
3.7	LES APPLICATIONS DU RBC	66
3.7.1	<i>Les principaux domaines d'utilisation</i>	66
3.7.2	<i>Récapitulation de quelques systèmes de RBC existants</i>	68
3.8	L'INTÉGRATION DU RBC AUX STI.....	68
3.8.1	<i>Le système TECHDOC-I</i>	69
3.8.2	<i>Le système Sarah</i>	70
3.8.3	<i>Le système Ocram-CBR</i>	70
3.9	COMPARAISON DU RBC AVEC D'AUTRES FORMES DE RAISONNEMENT	71
3.9.1	<i>Le RBC et le raisonnement à base de modèles (Model Based Reasoning (MBR))</i>	71
3.9.2	<i>Le RBC et le raisonnement à base de règles (Rule Based Reasoning (RBR))</i>	72
3.9.3	<i>Le RBC et le raisonnement par analogie</i>	72
3.10	AVANTAGES ET DÉSAVANTAGES DE RBC.....	73
3.11	CONCLUSION.....	73
CHAPITRE 4 - MODÉLISATION DE L'APPRENANT PAR L'ANALYSE DU RAISONNEMENT : PRINCIPES ET CONCEPTION		75
4.1	INTRODUCTION.....	75
4.2	INTÉGRATION DU RAISONNEMENT DANS LES STI	76
4.2.1	<i>Quelques définitions du raisonnement</i>	76

4.2.2	<i>Caractéristiques du raisonnement</i>	77
4.2.3	<i>Rôle du raisonnement dans la modélisation de l'apprenant</i>	78
4.3	TYPES DE RAISONNEMENT EN IA.....	78
4.4	MODÉLISATION DE L'APPRENANT PAR RAISONNEMENT À BASE DES CAS : PHILOSOPHIE GÉNÉRALE.....	79
4.4.1	<i>Le rôle de la remémoration dans l'apprentissage</i>	79
4.4.2	<i>Philosophie d'expériences progressives</i>	81
4.4.3	<i>Résolution des problèmes et modélisation des apprenants</i>	82
4.5	CONCEPTION DE NOTRE APPROCHE.....	83
4.5.1	<i>Principes de conception et présentation générale</i>	83
4.5.2	<i>Exemples</i>	86
4.6	CONCEVOIR UNE ARCHITECTURE POUR LE MODÈLE DE L'APPRENANT.....	87
4.5.1	<i>Les composants d'un modèle de l'apprenant</i>	87
4.5.2	<i>L'architecture du modèle de l'apprenant</i>	88
4.5.3	<i>Caractéristiques d'un sous-système modèle de l'apprenant</i>	90
4.5.4	<i>Les connaissances utiles pour modéliser l'apprenant</i>	90
4.6	MODÉLISATION DES CONNAISSANCES ET DES RAISONNEMENTS DE L'APPRENANT.....	91
4.6.1	<i>Modes d'interaction</i>	91
4.6.2	<i>Le besoin de différents modes d'interaction</i>	93
4.7	CONCLUSION DU CHAPITRE.....	94
CHAPITRE 5 - GCR : UN FORMALISME POUR LA MODÉLISATION ET LA REPRÉSENTATION DES CONNAISSANCES DANS LES SYSTÈMES À BASE DE CAS		95
5.1	INTRODUCTION.....	95
5.2	MODÉLISATION DES CONNAISSANCES DU DOMAINE.....	96
5.2.1	<i>Base de cas</i>	96
5.2.2	<i>Les cas</i>	96
5.2.3	<i>Base de problèmes</i>	97
5.2.4	<i>Les problèmes</i>	97
5.2.5	<i>Notion d'espace de recherche</i>	98
5.2.6	<i>Exemple</i>	98
5.3	L'APPROCHE DE REPRÉSENTATION DES CONNAISSANCES.....	100
5.3.1	<i>Représentation des connaissances dans les STI</i>	100
5.3.2	<i>Représentation des connaissances sous forme de graphes dans les systèmes à base de cas..</i>	105
5.4	NOTRE APPROCHE DE REPRÉSENTATION DES CONNAISSANCES.....	105
5.4.1	<i>Graphe Classifié Concepts et Relations (GCR)</i>	106
5.4.2	<i>Définition formelle</i>	107
5.4.3	<i>Les opérations sur les GCR</i>	108
5.5	REPRÉSENTATION DES CONNAISSANCES DANS NOTRE SYSTÈME.....	109
5.5.1	<i>Représentation d'un problème</i>	110
5.5.2	<i>Représentation d'un cas</i>	111
5.5.3	<i>Représentation de la base de connaissances</i>	112
5.6	QUELQUES ASPECTS FONDAMENTAUX.....	115
5.6.1	<i>Le cycle du RBC pour l'expert</i>	115
5.6.2	<i>Le choix des indices</i>	115
5.6.3	<i>La construction d'une base de cas</i>	115
5.6.4	<i>Le problème de couverture des cas</i>	116
5.6.5	<i>Déterminer les solutions (de l'expert) des problèmes</i>	116
5.7	CONCLUSION.....	117
CHAPITRE 6 - ANALYSE DU RAISONNEMENT DE L'APPRENANT DANS NOTRE MODÈLE		118
6.1	INTRODUCTION.....	118
6.2	LE PROCESSUS DE MODÉLISATION DE L'APPRENANT EN AS MODE.....	119
6.3	L'ANALYSE DU RAISONNEMENT EN AS MODE.....	124
6.4	MODÉLISATION DES ÉTUDIANTS EN US MODE.....	128
6.5	ANALYSE DU RAISONNEMENT EN US MODE.....	130

6.5.1	<i>Exemple 1</i>	131
6.5.2	<i>Exemple 2</i>	133
6.5.3	<i>Algorithme de modélisation</i>	135
6.5.4	<i>Les connaissances utilisées pour l'analyse et les connaissances émises de cette analyse</i>	136
6.6	FORMATION DU MODÈLE DE L'APPRENANT	137
6.6.1	<i>Mise à jour du modèle cognitif de l'apprenant</i>	138
6.6.2	<i>Mise à jour du modèle d'inférences</i>	139
6.7	LES AVANTAGES.....	141
6.8	CONCLUSION	141
CHAPITRE 7 - IMPLÉMENTATION		143
7.1	INTRODUCTION.....	143
7.2	LE PROTOTYPE SARA.....	144
7.2.1	<i>Introduction</i>	144
7.2.2	<i>Les idées principales de notre technique de modélisation des apprenants</i>	144
7.2.3	<i>L'architecture de SARA</i>	145
7.2.4	<i>L'utilisation de SARA</i>	150
7.2.5	<i>Mise à jour du modèle de l'apprenant</i>	156
7.2.6	<i>Avantages et limitations</i>	156
7.3	LE PROTOTYPE SARA-II	157
7.3.1	<i>Introduction</i>	157
7.3.2	<i>Le sous-système modèle de l'apprenant</i>	158
7.3.3	<i>L'analyse du raisonnement de l'apprenant (dans les deux modes d'interaction)</i>	159
7.3.4	<i>Discussion</i>	162
7.4	CONCLUSION.....	165
CHAPITRE 8 - MÉTHODOLOGIE DE CONSTRUCTION D'UN AGENT INTELLIGENT POUR L'AIDE AU RAISONNEMENT		167
8.1	INTRODUCTION.....	167
8.2	QU'EST-CE QU'UN AGENT ?	168
8.2.1	<i>Définition</i>	168
8.2.2	<i>Caractéristiques</i>	168
8.3	CLASSIFICATION.....	170
8.4	TYPOLOGIE D'AGENTS.....	172
8.4.1	<i>Les agents réactifs</i>	172
8.4.2	<i>Les agents délibératifs</i>	172
8.4.3	<i>Les agents coopératifs</i>	172
8.4.4	<i>Les agents cognitifs</i>	173
8.5	ARCHITECTURE D'UN AGENT INTELLIGENT	174
8.5.1	<i>L'architecture de BDI (Beliefs, Desires, Intentions)</i>	174
8.5.2	<i>Architecture d'un agent délibératif</i>	176
8.6	LES CARACTÉRISTIQUES DES AGENTS UTILISÉS DANS SARA ET SARA-II.....	177
8.7	POSSIBILITÉS DE PERFECTIONNEMENT DE NOS AGENTS INTELLIGENTS	179
8.8	ÉVOLUTION VERS LES SYSTÈMES MULTI-AGENTS COGNITIFS.....	179
8.8.1	<i>Organisation sociale d'agents cognitifs</i>	180
8.8.2	<i>Les différences entre SARA, SARA-II, et SARA-III</i>	182
8.9	AGENT INTELLIGENT POUR L'AIDE AU RAISONNEMENT (AIAR).....	183
8.9.1	<i>Tâches à exécuter par AIAR</i>	184
8.9.2	<i>Les propriétés de l'AIAR</i>	185
8.9.3	<i>Architecture de l'AIAR</i>	186
8.10	LA COOPÉRATION ENTRE AGENTS POUR L'ENSEIGNEMENT ET L'APPRENTISSAGE.....	189
8.10.1	<i>Supporter la résolution de problèmes</i>	189
8.10.2	<i>Exemple</i>	189
8.11	CONCLUSION.....	191
CHAPITRE 9 - CONCLUSION.....		192

9.1	LA RÉALISATION DE L'APPROCHE.....	192
9.2	COMPARAISON AVEC D'AUTRES APPROCHES	194
9.3	COMPARAISON AVEC DES TRAVAUX SIMILAIRES	195
9.4	AVANTAGES ET LIMITES DE L'APPROCHE THÉORIQUE	197
9.5	LES AVANTAGES DU FORMALISME DE GCR.....	197
9.6	EXPÉRIMENTATION.....	198
9.7	RECHERCHES FUTURES.....	199
9.7.1	<i>Automatiser l'acquisition des connaissances</i>	199
9.7.2	<i>Intégrer les résultats aux problèmes complexes</i>	199
9.7.3	<i>Améliorer notre prototype vers des systèmes utilisables dans plusieurs domaines</i>	200
9.7.4	<i>Intégrer l'idée de l'agent intelligent aux STI</i>	200
BIBLIOGRAPHIE		201
ANNEXE A - TYPES DES RAISONNEMENTS EN IA		218
A.1	LE RAISONNEMENT FORMEL.....	218
A.1.1	<i>Le raisonnement logique</i>	218
A.1.2	<i>Le raisonnement approximatif</i>	221
A.1.3	<i>Le raisonnement temporel et temps réel</i>	222
A.1.4	<i>Le raisonnement hypothétique et maintien de vérité</i>	222
A.2	LE RAISONNEMENT PROCÉDURAL.....	223
	<i>Le raisonnement par classification</i>	223
A.3	LE RAISONNEMENT PAR ANALOGIE.....	224
A.4	LE RAISONNEMENT PAR GÉNÉRALISATION ET ABSTRACTION.....	225
A.5	LE RAISONNEMENT GÉOMÉTRIQUE	225
ANNEXE B - BASE DE CONNAISSANCES		226
B.1	BASE DE CAS	226
B.2	BASE DE PROBLÈMES.....	227
ANNEXE C - MODÉLISATION LES BASES DE CONNAISSANCES		228
C.1	LE LANGAGE DE MODÉLISATION UML	228
C.2	LA MODÉLISATION DES BASES DE CONNAISSANCES.....	231
C.3	LA MODÉLISATION DU MODÈLE DE L'APPRENANT	232
C.4	LE DIGRAMME D'ÉTAT GLOBALE DE SARA-II.....	233

Liste des Figures

Figure 1. 1 : Domaine de Sciences Cognitives	13
Figure 1. 2 : Architecture classique de STI	14
Figure 1. 3 : Évolution des stratégies tutorielles	19
Figure 1. 4 : Architecture Safari du STI	25
Figure 2. 1 : Les trois phases du processus de diagnostic	37
Figure 2. 2 : Exemple de graphe génétique	40
Figure 3. 1 : Conception du RBC	53
Figure 3. 2 : Le cycle du RBC	57
Figure 3. 3 : Une hiérarchie simple d'abstraction	62
Figure 4. 1 : Relation entre remémoration et apprentissage	80
Figure 4. 2 : Le déroulement du raisonnement de l'apprenant et de l'expert dans notre approche.....	85
Figure 4. 3 : L'architecture du modèle de l'apprenant	89
Figure 5. 1 : Un espace de solutions.....	99
Figure 5. 2 : Exemple d'un graphe et/ou.	101
Figure 5. 3 : Un exemple de schéma (la description d'un lit du hôtel).....	102
Figure 5. 4 : Un exemple de réseau sémantique	103
Figure 5. 5 : Exemple de GC.....	104
Figure 5. 6 : Les éléments de base d'un GCR.	106
Figure 5. 7 : La représentation d'un problème	110
Figure 5. 8 : La représentation d'un cas.	111
Figure 5. 9 : La représentation de la base de connaissances.....	114
Figure 6. 1 : Le processus de modélisation de l'apprenant en <i>AS mode</i>	120
Figure 6. 2 : Génération et stockage de la solution du problème par le système.	123
Figure 6. 3 : Le mécanisme d'adaptation.....	125
Figure 6. 4 : Un scénario de la résolution de problème par l'apprenant (en <i>AS mode</i>).....	126
Figure 6. 5 : L'information nécessaire pour l'analyse du raisonnement et l'information.....	127
Figure 6. 6 : Le processus de modélisation de l'apprenant en <i>US mode</i>	129
Figure 6. 7 : L'ordre à suivre en cours de recherche.	131
Figure 6. 8 : L'évaluation de solution de l'apprenant en <i>US mode</i>	135
Figure 6. 9 : L'information nécessaire pour l'analyse du raisonnement et l'information.....	137
Figure 6. 10 : Le modèle cognitif de l'apprenant.	140
Figure 7. 1 : L'architecture de SARA	146
Figure 7. 2 : L'interface de SARA	147
Figure 7. 3 : Le modèle cognitif de l'apprenant dans SARA.....	148
Figure 7. 4 : Les problèmes et les cas dans le système.....	150
Figure 7. 5 : L'Interface de SARA et le ' <i>Cases Retriever Tool</i> '	152
Figure 7. 6 : Un exemple d'interaction.....	154
Figure 7. 7 : L'interface du mode <i>AS</i> et le ' <i>Cases Retriever Tool</i> '.	160
Figure 7. 8 : Un exemple d'analyse des actions de l'apprenant.	161
Figure 7. 9 : L'interface du mode <i>US</i>	162
Figure 7. 10 : Le modèle d'un étudiant qui utilise le système.....	164
Figure 8. 1 : Domaines d'influence	169

Figure 8. 2 : Classification des agents	171
Figure 8. 3 : Structure de BDI	174
Figure 8. 4 : Un exemple de <i>Croyances, Buts et Intentions</i>	175
Figure 8. 5 : Architecture d'un agent délibératif	176
Figure 8. 6 : La coopération entre les agents pour un apprentissage dans SARA-III	181
Figure 8. 7 : Architecture de l'AIAR.....	187
Figure 8. 8 : Exemple d'un plan	188
Figure 8. 9 : Exemple de dialogue entre agents.....	190
Figure C. 1: Digramme de classe.....	230
Figure C. 2: La base de connaissances intégré sous forme d'objets	231
Figure C. 3: Digramme du modèle de l'apprenant	232
Figure C. 4: Diagramme d'état global (SARA-II).....	233

Liste des tableaux

Tableau 2. 1 : Techniques de diagnostic	38
Tableau 3. 1 : Intervalles des âges dans CASY	63
Tableau 3. 2 : Exemple de systèmes à base de cas.	68
Tableau 4. 1 : Les événements d'enseignements	82
Tableau 6. 1 : Les problèmes précédemment résolus par l'apprenant	132
Tableau 6. 2 : Les nombre de problèmes résolus par l'apprenant	132
Tableau 6. 3 : Calcul des valeurs heuristiques.....	133
Tableau 6. 4 : L'échelle de Gagné	138
Tableau 8. 1 : Rapport entre les composants de SARA, SARA-II et SARA-III.....	182
Tableau 8. 2 : Rapport entre les caractéristiques de SARA, SARA-II et SARA-III.....	183
Tableau C. 1 : Notation UML.....	229

Liste des abréviations

AA	Agent Apprenant
ACT	Adaptive Control of Thought
AI	Agent Interface
AIAR	Agent Intelligent d'Aide au Raisonnement
AF	Agent d'inFormtion
AQ	Agent de Question
AS mode	Accessible Solution mode
AT	Agent Tuteur
BDI	Belief, Desire, Intention
CBR	Case-Based Reasoning
CBSM	Case-Based Student Modelling
DTL	Double Test Learning
EAO	Enseignement Assisté par Ordinateur
EIAO	Enseignement Intelligemment Assisté par Ordinateur
GC	Graphe Conceptuel
GCR	Graphe Classifié Concepts et Relations
GG	Graphe Génétique
GS	Gestionnaire de Session
IA	Intelligence Artificielle
MBR	Model Based Reasoning
MOP	Memory Organization Paquets
RBC	Raisonnement à Base de Cas
RBR	Rule Based Reasoning
RS	Réseau sémantique
PM	Pedagogic content Model
PLI	Programmation Logique Inductive
SARA	Système d'Analyse du Raisonnement de l'Apprenant

SM	Subject matter Model
STI	Système Tutoriel Intelligent
TOP	Thematic Organization Paquets
US mode	Unaccessible Solution mode

Remerciements

J'adresse ma profonde reconnaissance à ma directrice de recherche la professeure Esma Aïmeur pour sa patience, sa disponibilité, son aide, ses encouragements, ses conseils, le suivi de ce travail, et ses remarques précieuses.

J'aimerais également remercier vivement mon co-directeur de recherche le professeur Claude Frasson pour ses conseils, ses encouragements, ses commentaires qui ont contribué à l'amélioration de ce document et enfin pour le support financier de ma thèse.

Je remercie le professeur Michel Boyer d'avoir bien voulu présidé le jury de ma thèse.

Je remercie le professeur François Lustman pour avoir évalué ma thèse en tant que membre du jury.

J'ai l'honneur de remercier mon examinateur externe, le professeur Ramon López de Mántaras de l'université Autònoma de Barcelona (Bellaterra, Spain) qui m'a accordé son temps précieux pour lire ma thèse.

Je remercie le professeur Pierre Bordeleau pour l'honneur qu'il me fait en faisant partie du jury de ma thèse.

Je tiens à remercier tous les collègues du laboratoire Héron qui ont toujours créé un endroit chaleureux et agréable, ainsi que ceux qui ont su être présents quand j'avais besoin d'eux, en particulier Monsieur Jose Manuel Fernandez, Madame Wafa Niar et Madame Nacéra Melliti qui ont passé beaucoup de temps à relire ma thèse et mes articles.

Je voudrais exprimer un grand merci à tous mes amis pour avoir fait de mon séjour à Montréal, une période "très difficile à oublier".

Je remercie mon épouse Nahid Ghasemi-Rasa pour son soutien constant et surtout sa patience et ses encouragements. Je demande pardon à mes enfants Ali et Elnaz pour le temps que je leur ai volé.

Je remercie ma famille, principalement, mes parents, mes beaux-parents, mes sœurs et mon frère qui ont toujours su être "présents" malgré la distance.

Enfin, je tiens à remercier le Ministère des Études Supérieures du Gouvernement Irannien, le Ministère de l'Industrie, du Commerce, de la Science et de la Technologie du Gouvernement du Québec, le Centre d'Excellence Canadien de TeleLearning, l'Université de Montréal et l'Université d'Amirkabir de technologie pour leurs soutiens financiers durant mes études de doctorat.

À mon épouse Nahid Ghasemi-Rasa

À nos enfants Ali et Elnaz

Introduction

L'Enseignement Intelligemment Assisté par Ordinateur (EIAO) est un domaine de recherche très dynamique qui a pour objectif de développer des tuteurs informatiques utilisant des techniques d'Intelligence Artificielle (IA) pour dispenser l'enseignement. Ces derniers simulent la démarche pédagogique d'un tuteur humain et comportent certains aspects intelligents. L'EIAO a ajouté à l'EAO (Enseignement Assisté par Ordinateur), d'une part des capacités qui offrent une grande souplesse à l'interaction étudiant-système, et d'autre part une individualisation de l'enseignement qui entraîne un accroissement du rendement pédagogique [Barr et Feigenbaum, 1982]. Les Systèmes Tutoriels Intelligents (STI) [Frasson et Gauthier, 1989] constituent des systèmes mettant en œuvre un tel enseignement. Ils varient beaucoup dans les techniques utilisées, dans leur approche pédagogique, et bien sûr selon la matière enseignée. L'architecture de tels systèmes n'est pas standard. Néanmoins, elle comporte généralement les principales composantes suivantes : une base de connaissances du domaine à enseigner, un modèle de l'apprenant, un module tuteur et une interface.

Le modèle de l'apprenant constitue une partie fondamentale sur laquelle le STI va se baser pour déterminer ses actions. En effet, il permet au STI de connaître l'état des connaissances de l'apprenant, par conséquent d'adapter ses interventions et en particulier de choisir des stratégies pédagogiques adéquates. Il évolue constamment et représente la partie dynamique du système à partir de laquelle nous pouvons bâtir une interaction plus appropriée aux processus d'apprentissage de l'apprenant.

Le modèle de l'apprenant reste toujours la composante la plus compliquée dans un STI en raison de la difficulté de modéliser les connaissances et les raisonnements de l'apprenant. Plus particulièrement, les problèmes concernent le contenu du modèle de l'apprenant, la représentation des connaissances, ou encore la construction et mise à jour du modèle lui-même.

Le premier **problème** concerne donc le contenu du modèle. Ce dernier représente selon les auteurs l'état des connaissances maîtrisées par l'apprenant ou encore ses erreurs et ses déviations par rapport aux connaissances de l'expert, qui sont supposées représenter le modèle idéal. Bien que plusieurs modèles de l'apprenant soient développés, la plupart ne prennent en compte que les connaissances conceptuelles de l'apprenant. D'autres types de connaissances telles que ses capacités, ses préférences, ses caractéristiques, etc. ne sont malheureusement pas considérées. Ainsi, le modèle de recouvrement de Carr et Goldstein (1977) considère les connaissances de l'apprenant comme un sous-ensemble de celles de l'expert (qui 'recouvre' toutes les connaissances). Le modèle correctionnel [Burton et Brown, 1976] construit un modèle de l'apprenant comprenant les connaissances correctes et les connaissances incorrectes relevées. La modélisation basée sur le graphe génétique [Goldstein, 1982] représente la manière dont l'apprenant apprend en utilisant les liens génétiques (tels que l'analogie, la génération, etc.). Le traçage de modèle [Anderson et Pelletier, 1991] [Anderson *et al.*, 1995] permet d'interpréter le comportement d'un apprenant en le suivant et en associant chacun de ses mouvements à un état cognitif. Les connaissances contenues dans ce dernier représentent les informations pour la solution proposée par l'expert, ainsi que les erreurs les plus courantes observées chez l'apprenant.

Aucun de ces modèles ne contient de connaissances complètes sur l'apprenant, en particulier ils ne contiennent pas les modes de raisonnement que l'apprenant emploie dans ses résolutions de problèmes (sauf dans le modèle basé sur le graphe génétique qui représente seulement le raisonnement du type analogie). Ces derniers sont nécessaires et doivent être pris en compte par le STI enfin de fournir les explications appropriées à l'apprenant (c.-à-d., selon les modes de raisonnement qu'il maîtrise).

Le deuxième **problème** concerne la construction et mise à jour du modèle de l'apprenant qui est une tâche complexe et difficile. Le modèle de l'apprenant correspond à un sous-ensemble des connaissances de l'expert. L'enseignement consiste à faire augmenter ce sous-ensemble, qui n'est cependant pas suffisant pour représenter l'évolution des connaissances chez l'apprenant. En effet, cette connaissance évolue par les raisonnements que l'apprenant emploie. Par exemple, lorsque l'apprenant maîtrise bien le fonctionnement d'un moteur de voiture, il sera en mesure de trouver plus facilement les

pannes. Les mécanismes de raisonnement qu'il emploie sont différents de ceux qu'il employait lorsqu'il était apprenti mécanicien [Marcenac et Herin-Aime, 1988].

Une des grandes problématiques de l'analyse des raisonnements repose sur la difficulté à identifier les différentes méthodes employées par l'apprenant lors de la résolution d'un problème. Nous tentons dans cette thèse de résoudre ce problème.

Le troisième **problème** concerne la représentation des connaissances (aussi bien les connaissances du domaine que les connaissances du modèle de l'apprenant). Bien que plusieurs formalismes de représentation de connaissances, comme les règles de production [Davis *et al.*, 1977], les réseaux sémantiques [Collins et Quillian, 1969], les graphes génétiques [Goldstien, 1982], les schémas [Minsky, 1981] et les graphes conceptuels [Sowa, 1984] soient développés, il n'y pas de consensus pour trouver une représentation générale. Chaque groupe de chercheurs justifie son choix afin de développer son approche. Nous devons donc chercher le formalisme le mieux adapté à notre approche.

Objectifs

L'objectif principal de notre travail consiste à déterminer le mode de raisonnement employé par l'apprenant et d'en déduire ses connaissances. On souhaite créer un modèle de l'apprenant qui soit le plus proche des connaissances et du raisonnement de ce dernier. Pour cela on veut extraire lors d'une session d'apprentissage :

- les connaissances (complètes, partielles, ...) de l'apprenant;
- son mode de raisonnement, en fonction de différentes tâches qui lui sont proposées.

Pour réaliser cet objectif nous souhaitons :

- développer un mécanisme pour modéliser l'apprenant;
- développer un formalisme pour représenter les connaissances du domaine (et les connaissances du modèle de l'apprenant);
- concevoir une architecture pour le modèle de l'apprenant.

Notre deuxième objectif consiste à valider le modèle précédant, et en particulier permettre d'évaluer les performances des apprenants. Ce système doit pouvoir :

- produire des problèmes en fonction de la situation pédagogique de l'apprenant pour les lui proposer;
- produire toutes solutions possibles (tous les points de vue) à chaque problème donné afin d'être comparées à celles de l'apprenant;
- construire le modèle de l'apprenant lors de la résolution de problème.

Présentation de l'approche

Plusieurs auteurs [Polya, 1957] [Bandura, 1971] [Ross, 1989] [Kolodner, 1997] ont montré que les gens raisonnent à partir d'expériences et de cas déjà rencontrés pour résoudre de nouveaux problèmes. L'apprentissage humain est donc basé sur des expériences progressives, par conséquent les anciennes résolutions laissent une trace en mémoire qui s'active pour contribuer aux nouvelles résolutions. Nous concluons donc que la reconnaissance des raisonnements actuels de l'apprenant pendant la résolution d'un problème est à l'image de la représentation correcte de l'état de ses connaissances et de son mode de raisonnement habituel. Nous pensons qu'une façon naturelle de réaliser cette tâche est de donner la possibilité à l'apprenant de choisir un cas similaire au problème donné afin d'adapter sa solution pour résoudre le problème. En observant ensuite ses actions et ses intentions et en analysant sa solution au travers d'une représentation interne on peut alors, extraire le mode de son raisonnement.

En effet, le Raisonnement à Base de Cas (RBC) [Kolodner, 1993] [Leake, 1996] [Watson, 1997] qui retient l'expérience passée peut jouer un rôle important dans notre démarche. C'est pour cette raison que nous proposons une approche basée sur le RBC pour la modélisation de l'apprenant.

Dans cette optique, l'analyse du raisonnement de l'apprenant se fait en fonction de celle de l'expert; par conséquent nous avons besoin de relier à chaque problème toutes ses solutions possibles, tout en associant à chaque solution son mode de raisonnement. À cette fin, nous avons besoin d'un formalisme flexible et général qui nous permet d'une

part de représenter aussi bien les connaissances du modèle de l'apprenant que les connaissances du domaine, et qui d'autre part nous permet d'exprimer les relations entre les problèmes et les cas, ainsi que les relations entre un cas (ou un problème) et ses solutions, etc. Nous cherchons donc un formalisme similaire aux Graphes Conceptuels (GC) proposés par Sowa (1984) pour représenter les connaissances¹. Ceci parce que, les point forts de ces derniers sont nombreux [Aïmeur, 1994] : (1) ils constituent un modèle de représentation général très clair; (2) ils permettent d'intégrer une culture générale très vaste et apportent enfin une rigueur logique de la représentation ainsi que des aspects sémantique riches; (3) ils offrent une gamme variée d'opérations (telles que la projection, la restriction, la jointure, etc.), pouvant servir à la conception et à l'implantation aisées d'une analyse de raisonnement, etc.

En ce qui concerne les composantes du modèle de l'apprenant, nous nous basons sur le modèle développé par Nkambou (1996), car ce modèle est basé sur une organisation bien structurée avec un contenu d'informations suffisamment riche sur l'apprenant. Ce dernier est composé de trois sous-modèles : un *modèle cognitif* (information sur l'état des connaissances de l'apprenant), un *modèle d'inférences* (règles permettent d'inférer des nouvelles connaissances dans le modèle de l'apprenant), et un *modèle affectif* (informations sur les attitudes et les préférences de l'apprenant). Le sens du modèle d'inférences employé par Nkambou n'est pas identique au nôtre. Nous devons donc identifier dans notre modèle de l'apprenant une partie pour les informations sur les modes d'inférences. Enfin, nous nous limiterons dans notre tâche de modélisation de l'apprenant au modèle cognitif et inférentiel.

Plan de la Thèse

Cette thèse est divisé en neuf chapitres. Le chapitre 1 a pour but de faire un bref état de l'art sur les systèmes d'enseignement intelligemment assistés par ordinateur. Nous présentons dans ce chapitre l'historique des systèmes tutoriels intelligents, une architecture de base, une architecture générale et une typologie des STI. Nous allons voir

¹ Nous avons développé un formalisme basé sur les GC appelé GCR pour représenter nos connaissances.

les différents éléments que l'on retrouve dans un STI, énumérer les différentes stratégies d'enseignement, et montrer quelques faiblesses des STI.

Le chapitre 2 traite du modèle de l'apprenant. Nous présentons dans ce chapitre l'utilité du modèle de l'apprenant, une architecture possible, le problème de modélisation et les processus d'analyse des connaissances de l'apprenant (appelé méthodes de diagnostic). Ensuite, nous présentons quelques approches théoriques de modélisation du raisonnement, puis nous décrivons les modèles considérés dans quelques systèmes tutoriels. Cette étude nous permet ainsi de dégager les principales approches utilisées jusqu'à aujourd'hui concernant l'aspect modèle de l'apprenant dans les STI.

Le chapitre 3 présente l'évolution et la méthodologie du raisonnement à base de cas. Nous y présentons l'origine, la méthodologie, le cycle, les techniques et les applications du RBC. Nous présentons ensuite, quelques exemples des systèmes représentatifs, et finalement nous mettons l'accent sur les STI qui utilisent les techniques du RBC dans leur approche.

Le chapitre 4 consiste à présenter l'intégration du raisonnement dans les STI. Nous y présenterons l'utilité du raisonnement dans la modélisation des connaissances (et le raisonnement de l'apprenant). Nous présenterons ensuite les principes de notre approche (qui est basée sur le RBC, autrement dit sur le 'Case Based Reasoning (CBR)') pour réaliser cette dernière. Nous analysons finalement les différents problèmes qui peuvent entraver la réalisation d'un système de RBC.

Le chapitre 5 aborde la représentation des connaissances que notre système doit adopter. Nous présentons d'abord quelques formalismes traités dans la littérature. Nous détaillons ensuite, notre formalisme, appelé *Graphe Classifié Concepts et Relations (GCR)*, qui permet de représenter ainsi bien les connaissances du modèle de l'apprenant que les connaissances du domaine.

Le chapitre 6 détaille notre approche de modélisation de l'apprenant basée sur l'idée retenue dans le chapitre 4 et sur la base du formalisme de représentation des connaissances retenu

dans le chapitre 5. Nous y détaillons en outre, les algorithmes et les processus de modélisation de l'apprenant.

Le chapitre 7 décrit la réalisation et l'implantation de nos systèmes. Nous y présenterons deux prototypes SARA et SARA-II, développés exclusivement pour démontrer la validité de notre base théorique.

Le chapitre 8 explique comment concevoir et utiliser l'idée des agents intelligents pour développer un agent intelligent pour l'aide au raisonnement. Dans ce chapitre, nous commençons par introduire les agents intelligents et présenter quelques modèles de base de tels agents, ensuite nous examinons les aspects d'agents dans les systèmes SARA et SARA-II, et finalement nous mettons l'accent sur l'évolution de cette discipline vers les agents intelligents pour l'aide au raisonnement.

Le chapitre 9 résume nos résultats et présente les extensions possibles à notre recherche. Dans ce chapitre, nous évaluons aussi notre approche par rapport à celles qui existent déjà.

Chapitre 1

Les Systèmes Tutoriels Intelligents

... the main promise of computer tutors... lies in their potential for moment-by-moment adaptation of instructional content and form to the changing cognitive needs of the individual learner, and our task ... is to find principles which can guide the construction of tutors which fulfill that promise.

- Ohlsson, 1986.

1.1 Introduction

L'apprentissage peut être considéré comme étant l'ensemble des activités amenant un individu à acquérir de nouvelles connaissances et de nouveaux comportements qu'il est capable soit de mettre en œuvre plus tard, soit d'exploiter ou de redéfinir [Péninou, 1993].

L'utilisation de l'ordinateur pour l'éducation et la formation a connu un important essor pendant la dernière décennie. Les recherches effectuées ont visé à comprendre les comportements humains et à les reproduire dans l'ordinateur. Les premiers systèmes conçus et développés dans ce contexte sont appelés systèmes d'Enseignement Assisté par Ordinateur (EAO) [Bestougeff et Fargette, 1982] [Lefevre, 1984]. L'objectif de ces systèmes est de favoriser l'apprentissage d'un domaine de connaissances par un apprenant. L'EAO suppose que la relation entre apprenants et enseignants se trouve remplacée par une relation apprenant/machine, cette dernière joue alors le rôle de l'enseignant.

Une forme plus récente de l'instruction par ordinateur est celle des Systèmes Tutoriels Intelligents (STI) [Frasson et Gauthier, 1989]. Les STI sont un exemple d'application des technologies de l'Intelligence Artificielle (IA) à l'instruction. Bien que leur développement soit limité aux études de laboratoire, ils ont été offerts comme solutions prouvées aux problèmes techniques et fonctionnels dans la formation et la technologie éducative.

Dans ce chapitre, nous décrivons brièvement d'abord le développement historique de l'instruction assistée par ordinateur et les principaux aspects théoriques de l'Enseignement Intelligemment Assisté par Ordinateur (EIAO) [Sleeman et Brown, 1982]. Nous examinons en second lieu les composants des systèmes tutoriels intelligents. Troisièmement, nous passons en revue les stratégies tutorielles et les typologies de STI. Nous décrivons ensuite, le développement des systèmes représentatifs. Finalement, nous présentons une architecture générale des STI.

1.2 Développement des systèmes tutoriels : de L'EAO vers L'EIAO

L'EAO, qui est l'utilisation interactive de l'ordinateur comme support d'enseignement, est apparue dans les années 1950. Comme son nom le suggère, il s'agit de donner à l'ordinateur un rôle de tuteur, de précepteur. Un échange didactique va s'instaurer entre un apprenant et un enseignant simulé par l'ordinateur. L'EAO se présente en gros comme une succession de 'information-question-réponse-commentaire' entièrement prévus et rédigés à l'avance par les auteurs du didacticiel [Arsac, 1987].

La majorité des systèmes d'EAO sont critiqués du fait qu'ils sont développés sans aucune base théorique. Pour leur part, les programmes d'EAO incorporent quelques principes d'enseignement sous une forme ou une autre. L'approche suivie pour la conception et la réalisation des systèmes d'EAO consiste à voir comment un bon professeur enseigne sa matière et ensuite écrire un programme avec des réponses toutes prêtes à une série de problèmes ou d'exercices relatifs au cours.

Les systèmes d'EAO présentent certaines limites et faiblesses comme par exemple :

- le problème d'établir un dialogue avec l'apprenant (dans sa langue naturelle);
- l'incapacité d'améliorer la stratégie d'enseignement pour réduire la distance entre l'apprenant et l'expert du domaine;
- la difficulté de décider ce qui devrait être enseigné par la suite;
- l'incapacité de traiter les réponses de l'apprenant;
- le problème de diagnostiquer et de comprendre les erreurs et les incompréhensions de l'apprenant.

Depuis le début des années 70, les progrès réalisés en intelligence artificielle et le développement d'outils qui y sont associés, ont aidé les chercheurs à réaliser des systèmes d'EIAO, qui sont une nouvelle génération des systèmes d'EAO. Plusieurs techniques d'IA sont utilisées pour la réalisation de tels systèmes dont notamment, les méthodes de représentation des connaissances, le traitement du langage naturel, l'utilisation de processus inférentiels et aussi plusieurs autres techniques plus spécifiques à l'IA comme par exemple : l'intégration symbolique, les diagnostics médicaux, les preuves de théorèmes, etc.

En fait, les systèmes d'EIAO ont été conçus pour combler certaines lacunes des systèmes conventionnels d'EAO, par exemple : en plus d'avoir une représentation des connaissances de la matière à enseigner, en utilisant les techniques développées en IA, les systèmes d'EIAO sont capables d'engager un dialogue individuel avec l'étudiant et d'utiliser ses erreurs pour diagnostiquer les éléments incompris de la matière à enseigner. Ils tiennent compte des réactions de l'étudiant, de ses erreurs et de ses intérêts. Les réponses et les interventions de ces systèmes sont plus 'intelligentes' et plus individualisées. L'étudiant bénéficiera d'un cheminement personnalisé et d'une approche pédagogique plus adaptée à ses besoins. Ce comportement simule en fait celui d'un expert-pédagogique humain avec tous les aspects intelligents que cette tâche requiert. Ces systèmes répondent au moins à trois critères [Burns et Capps, 1988] :

1. ils comprennent et connaissent le domaine;

2. ils peuvent déduire l'état de connaissances de l'apprenant;
3. ils implantent des stratégies d'enseignement pour réduire la distance entre les connaissances de l'apprenant et celles de l'expert du domaine.

1.3 Système tutoriel intelligent : définition et objectifs

Nous allons dans la suite donner la définition de STI du point de vue de certains auteurs.

1.3.1 Définition

Plusieurs définitions des STI sont données dans la littérature, certaines définitions insistent sur le diagnostic cognitif des actions de l'apprenant, la correction adaptative de ses erreurs; d'autres, sur la présentation des activités pédagogiques, l'environnement d'apprentissage, etc.

VanLehn (1988) par exemple définit un STI comme un programme informatique qui instruit l'apprenant, d'une manière intelligente.

D'après Mandl et Lesgold (1988), un STI est un programme informatique qui peut faire des inférences sur les connaissances de l'apprenant et interagir intelligemment avec lui (sur la base d'une ou plusieurs représentations individuelles de ses connaissances).

D'après Greg Kearsley (1982) un STI est un système qui utilise les techniques d'IA pour aider une personne à apprendre.

Pour nous, le STI est un programme de formation, basé sur les techniques de l'IA qui a pour objet de réaliser, à l'aide d'un ordinateur, un enseignement individualisé.

L'individualisation de l'enseignement est alors une des caractéristiques et l'un des avantages les plus évidents des STI. Voyons ce qu'elle apporte à l'apprenant :

- *Rythme individuel* : Dans le cadre des STI chaque apprenant a à travailler à son rythme.

- *Participation active* : Tous les messages sont adressés à l'apprenant et toutes les questions lui sont posées. Il doit y répondre et alors ses réponses sont analysées et commentées.
- *Correction immédiate et systématique* : Dans le cadre des STI, la correction vient immédiatement. Le système permet également de diminuer la répétition systématiquement du même type d'erreur (tout au long d'un exercice) par l'apprenant.
- *Augmentation de la capacité de concentration de l'apprenant* : La nécessité de répondre à toutes les questions de l'apprenant et de lui envoyer systématiquement des commentaires par le système lui imposent de se confronter à toutes les étapes du processus didactique et de constater la qualité de sa performance.
- *Intimité et anonymat* : Dans un travail individualisé, l'apprenant se trouve seul et libre; il se sent donc à l'aise pour répondre aux questions.

1.3.2 Le but des STI

Le but d'un STI est de présenter la connaissance de façon claire, simple et accessible à l'étudiant, en faisant intervenir des éléments médiatiques et diverses stratégies pédagogiques, pour essayer de placer l'étudiant dans un environnement cognitif lié au domaine d'apprentissage.

1.3.3 Les domaines utilisés pour la conception des STI

La Figure 1.1 montre les domaines utilisés pour la conception des STI. La conception et le développement de tels systèmes se trouvent à l'intersection de l'*informatique*, de la *psychologie cognitive* et de la *recherche en éducation*. Ce domaine est désigné souvent sous le nom de *sciences cognitives* [Polson et Richardson, 1988].

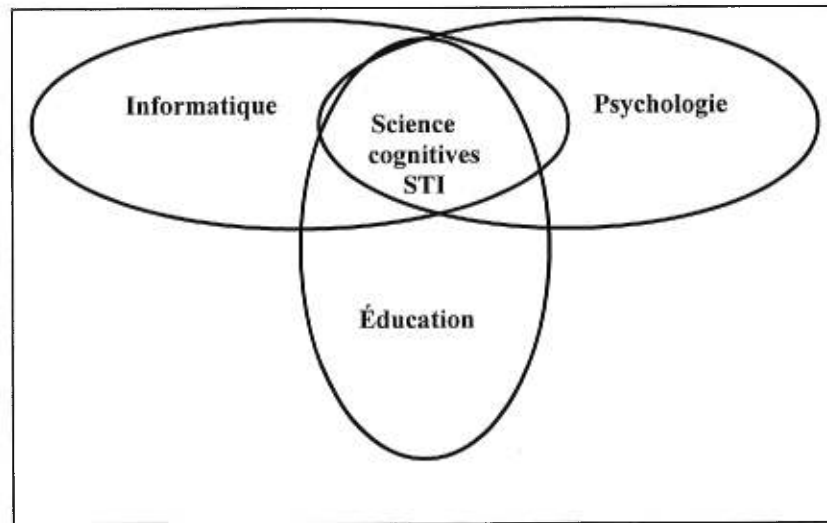


Figure 1. 1 : Domaine de Sciences Cognitives

1.4 Architecture de base d'un STI

L'architecture de base d'un STI fait ressortir quatre modules de base [Burns et Capps, 1988] : *le module expert, le modèle de l'apprenant, le module tuteur, et l'interface* (voir Figure 1.2).

1.4.1 Le module expert

Le *module expert* est le composant du STI qui modélise l'expertise du domaine. Il contient une représentation des connaissances du domaine à enseigner et un système qui permet de manipuler ces connaissances. Ce module permet au STI de résoudre les problèmes que le système pose à l'étudiant pour être ainsi en mesure d'interpréter les actions de ce dernier lorsqu'il résout un exercice. Plusieurs techniques sont utilisées pour représenter le domaine à enseigner [Barr et Feigenbaum, 1982]. Les principales sont les réseaux sémantiques, les graphes génétiques, les représentations procédurales et les représentations multiples.

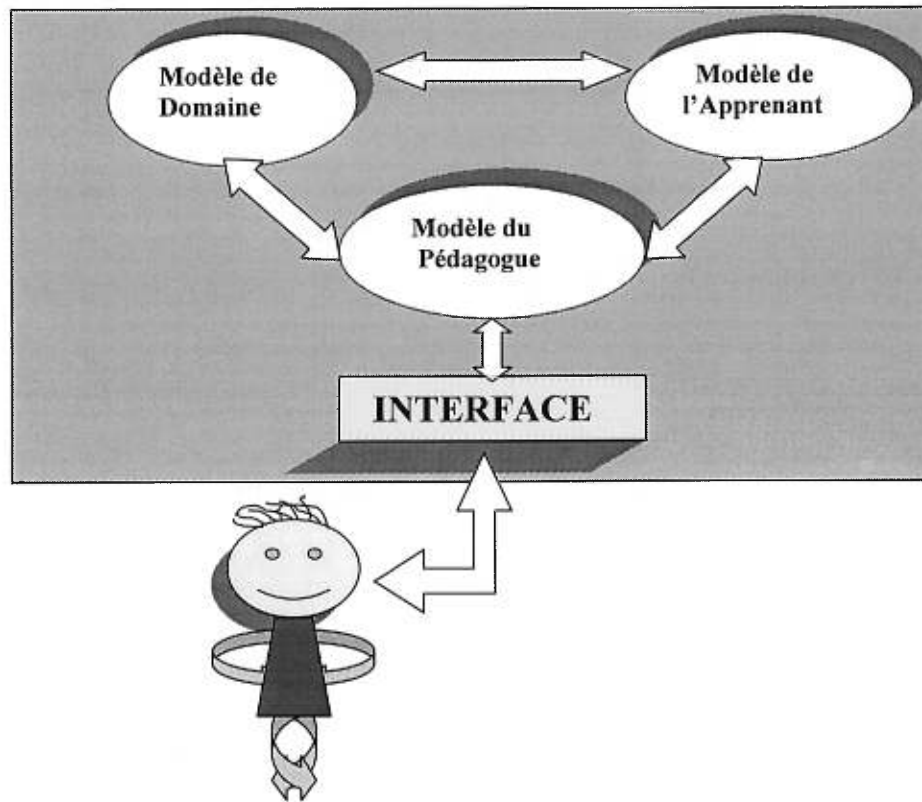


Figure 1. 2 : Architecture classique de STI

1.4.2 Le modèle de l'apprenant

Le *modèle de l'apprenant* est utilisé pour évaluer l'état des connaissances de l'étudiant et le STI utilise les informations contenues dans ce modèle pour ajuster ses stratégies pédagogiques.

En général, un *modèle de l'apprenant* contient une représentation de l'état des connaissances de l'étudiant et des règles permettant de diagnostiquer cet état. Il est à noter que, le processus qui forme et met à jour le modèle de l'apprenant en analysant des données rendues disponibles au système s'appelle souvent le diagnostic.

En effet, la séparation de deux composantes du modèle de l'apprenant permet au STI de présenter à l'étudiant une plus grande variété d'activités d'apprentissage, certaines

permettant l'acquisition de connaissances et d'autres stimulant l'acquisition de nouvelles compétences au regard de la mise en pratique des connaissances acquises. Nous décrirons en détail le modèle de l'apprenant dans le chapitre suivant.

1.4.3 Le module tuteur

Le *module tuteur* est la composante principale d'un STI qui gère le fonctionnement global du système. Il est chargé de communiquer avec l'étudiant, de sélectionner les problèmes qu'il va lui poser, d'analyser ses réponses, de fournir une rétroaction quand c'est nécessaire et de présenter à l'étudiant la solution des problèmes posés [Barr et Feigenbaum, 1982].

Le STI représente en fait, un tuteur humain expert, qui combine ses connaissances sur la matière à enseigner et sur l'état des connaissances de l'étudiant dans le domaine et qui est capable d'appliquer les techniques et les règles adéquates pour aider ce dernier à avancer dans son apprentissage [Sleeman et Brown, 1982].

C'est lui qui doit décider à chaque instant quelle action spécifique il y a lieu d'entreprendre dans la situation présente, en appliquant les techniques d'enseignement appropriées, au moment opportun durant la séance d'apprentissage. Ces techniques d'enseignement sont connues sous le nom de stratégies tutorielles du STI. Le tuteur doit guider l'enseignement sans pour autant paraître trop directif et/ou autoritaire.

1.4.4 Le module interface

Pour être en mesure de fournir une interaction diversifiée, un STI dispose d'une composante (appelée Interface) pour la communication. En général un environnement d'apprentissage comprend un micro-monde et/ou une Interface. Cet environnement d'apprentissage, comme son nom l'indique, supporte les apprentissages de l'étudiant en permettant la mise en application des concepts appris et des interventions pédagogiques de la part du STI. Des études ont montré que l'Interface Humain-Machine joue un rôle majeur dans l'apprentissage. En particulier, une interface pauvre et confuse influence négativement l'apprentissage et l'enseignement, en revanche une interface bien conçue

améliore la formation [Miller, 1988]. En ce sens, l'interface doit être la plus transparente possible et facile à maîtriser pour ne pas perturber l'apprentissage, car cette dernière influence directement le genre de stratégies pédagogiques que le STI peut employer. L'interface avec l'apprenant inclut plusieurs média, i.e. textes, images, graphiques, sons, vidéo, etc.

1.5 Les stratégies tutorielles

Dans le domaine des STI, plusieurs stratégies pédagogiques, couvrant une variété d'interactions entre le tuteur et l'apprenant, ont été développées.

Définition : Une stratégie pédagogique tente de reproduire le comportement d'un spécialiste en pédagogie qui adapte le rythme de son enseignement à celui de l'apprenant [Holt et Wood, 1990].

Nous allons décrire ici quelques stratégies tutorielles mises en œuvre par des systèmes existants. Il faut noter que ces stratégies permettent au tuteur de réagir de la bonne façon au bon moment [Frasson et Gauthier, 1994]. Pour chaque stratégie, nous allons énumérer les agents impliqués, leurs rôles respectifs, et les étapes éducatives touchées.

1.5.1 Tuteur classique (ou mode directif)

Dans cette stratégie on retrouve deux agents : un tuteur simulé par l'ordinateur et un étudiant humain, le tuteur démontre des concepts et des tâches, et l'apprenant résout des problèmes simples sous la supervision du tuteur qui le conseille. Enfin, le tuteur présente à l'apprenant une critique de sa performance.

1.5.2 Co-apprenant

Cette stratégie implique deux agents : une personne qui étudie et un co-apprenant simulé par ordinateur [Self, 1988]. Le co-apprenant devient un démonstrateur de niveau intermédiaire, d'un point de vue conceptuel, le co-apprenant a récemment acquis la présente leçon, il a donc le modèle explicite du processus encore en mémoire et il aide l'apprenant à suivre le même chemin de construction. L'emphase ici est mise sur la

construction de la connaissance et non sur sa transmission [Aïmeur et Frasson, 1995] [Aïmeur et Frasson, 1996].

1.5.3 Compagnon d'apprentissage

Cette stratégie implique trois agents : l'apprenant, un ou plusieurs compagnons d'apprentissage (qui peuvent être simulés ou humains), et un tuteur lui aussi simulé. En général, le compagnon possède à peu près le même niveau de connaissances que celui de l'apprenant, seuls ses processus de raisonnement diffèrent. L'apprenant et le compagnon travaillent ensemble sur la même tâche, sous la supervision du tuteur. Pendant la phase d'exercices, l'apprenant peut être conseillé par le compagnon, ou peut lui-même le conseiller, ce qui augmente sa motivation [Chan et Baskin, 1990].

1.5.4 Tuteur inversé

C'est une autre stratégie qui implique deux agents : l'apprenant et un compagnon simulé par l'ordinateur. Dans cette approche, c'est l'apprenant qui est supposé jouer le rôle du tuteur, explore ses connaissances de manière claire et précise pour que le compagnon simulé puisse les comprendre [VanLehn *et al.*, 1994]. De plus, il doit corriger les fautes du compagnon, ce qui l'aide à formuler clairement ses connaissances.

1.5.5 Perturbateur

Le perturbateur est un agent (qui a l'apparence d'un compagnon) qui conseille l'apprenant pendant son travail, ces conseils peuvent être valides ou non. En réalité c'est un tuteur déguisé qui est spécialisé dans la pédagogie et s'entend avec le tuteur sur les points d'intervention. Le but de cette stratégie est de tester la confiance de l'apprenant en ses connaissances [Aïmeur et Frasson, 1996]. Pour être pédagogiquement efficace, le perturbateur se doit de proposer des conseils-pièges qui serviront à mettre en valeur certaines subtilités de la tâche, qui passeraient sinon inaperçues.

1.5.6 Apprentissage par double test

La stratégie du ‘*Double Test Learning*’ (DTL) [Aïmeur et Fahmi, 1998] est une nouvelle stratégie tutorielle qui s’appuie sur la théorie de l’apprentissage social [Miller et Dollard, 1941]. Le but essentiel de cette stratégie est de permettre à l’apprenant d’apprendre à partir de ses propres erreurs et/ou des erreurs commises par autrui. Elle met alors l’emphase sur l’importance d’observer et de modéliser les comportements, les attitudes et les réactions émotionnelles des autres. Pour cela, elle nécessite l’intervention de trois agents : *le tuteur*, *le co-apprenant (CA)* et *l’apprenant*.

Cette stratégie comporte quatre étapes : le *Pré-Test*, la *Formation*, le *Post-Test1* et le *Post-Test2*. Dans la première étape, le tuteur pose à l’apprenant une série de questions qui lui permet d’évaluer ses réponses afin de créer le modèle de l’apprenant. Ensuite, lors de la formation, le tuteur présente à l’apprenant et au (CA) une série de problèmes et leurs solutions afin de permettre une acquisition de connaissances pour les deux agents. Dans la troisième étape, l’apprenant observe le CA qui interagit avec le tuteur et il écoute les questions du tuteur et les réponses du CA. Lors de la quatrième étape, l’apprenant répond aux questions (certaines de ces questions sont les mêmes que celles du post-test1) ce qui permet au système de détecter s’il a pu profiter des erreurs commises par le CA lors du Post-Test1. À la fin, le tuteur donne des explications aux questions auxquelles l’apprenant a échoué.

En conclusion, la stratégie du mode directif (utilisant le tutorat classique) dans laquelle le tuteur a un rôle central pour contrôler le déroulement du cours, fut la première approche utilisée dans les STI. Dans cette stratégie, le discours était généralement dirigé du tuteur vers l’apprenant.

Toutefois, dans les STI les plus récents, les chercheurs prennent en considération une approche coopérative entre l’apprenant et le système en utilisant le concept de ‘*co-apprenant*’, ou le ‘*compagnon*’ ou bien le ‘*perturbateur*’. L’idée principale d’utiliser le co-apprenant dans les STI par exemple, provient du fait que les connaissances acquises par l’apprenant devront être construites et non simplement transmises. Ceci est possible

grâce à l'interaction qui existe entre l'apprenant humain et le co-apprenant simulé. La Figure 1.3 montre l'évolution des stratégies tutorielles dans les STI.

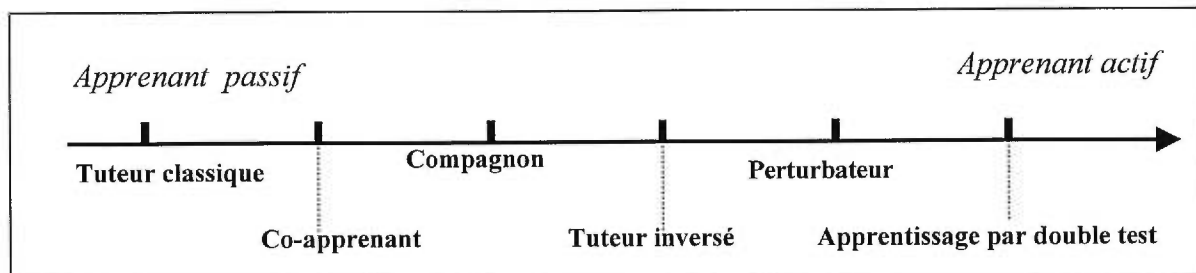


Figure 1. 3 : Évolution des stratégies tutorielles [Aïmeur et Frasson, 1996].

1.6 Typologie des STI

Plusieurs types de systèmes tutoriels intelligents sont développés, chaque type peut utiliser une ou plusieurs stratégies tutorielles. Les différents types de STI accessibles dans la littérature sont les suivants :

1.6.1 Les STI socratiques

Par la méthode Socratique, le système engage l'apprenant dans un dialogue qui lui permet de découvrir ses erreurs et ses contradictions. Une décision pédagogique est prise en fonction du résultat du dialogue. Ce type de STI est très approprié pour présenter les informations factuelles et aussi pour l'apprentissage de règles de principe ou de stratégies de résolution de problèmes [Alessi et Trollip, 1985]. WHY et SCHOLAR sont des systèmes qui utilisent un dialogue socratique pour assurer l'enseignement [Stevens et Collins, 1977].

1.6.2 Les STI de type guidage (Coach)

Dans ce cas, pour amener l'étudiant à s'apercevoir de ses propres erreurs, la présence d'un tuteur intelligent (coach) s'avère nécessaire afin d'évaluer constamment les décisions de l'étudiant pour le guider plus ou moins explicitement en lui faisant percevoir les

conséquences. Cette technique consiste à guider l'étudiant plutôt que lui dicter ce qu'il doit faire. Le système ne doit pas intervenir trop souvent et doit laisser l'étudiant se 'débrouiller' tout en lui donnant quelques informations utiles quand cela est nécessaire.

Le concept de 'coach' est important dans un environnement d'apprentissage dans lequel l'étudiant doit appliquer un ensemble de connaissances pour accomplir une certaine tâche. Entre autres, il est motivé autrement que s'il était en train d'appliquer des connaissances en dehors d'un contexte réel. Cette méthode se trouve dans les tuteurs WEST [Burton et Brown, 1979] ou STEAMER [Hollan *et al.*, 1987].

1.6.3 Les STI démonstrateurs

Le but de ces systèmes est de présenter des séquences de tâches à effectuer dans un contexte de formation [Alessi et Trollip, 1985]. Ils n'enseignent pas vraiment au sens des STI socratiques et des STI procéduraux, mais se contentent de montrer quelque chose à l'étudiant. Ils permettent à l'apprenant de voir et de comprendre le contexte réel étudié, en se basant sur une simulation (souvent interactive) du monde modélisé (on ne peut pas vraiment parler de STI pour ces systèmes, car l'intelligence est encapsulée dans la simulation). Nous trouvons ce principe dans certains systèmes experts, comme MYCIN [Shortliffe, 1976] par exemple.

1.6.4 Les systèmes critiques

Le but de ces systèmes est de critiquer les apprenants dans la résolution de problèmes (et d'amener l'utilisateur vers une solution correcte) [Silverman, 1992]. Le mode critique analyse les diverses étapes que l'étudiant a suivi pour résoudre le problème et critique ses actions pour 's'assurer' qu'il a vraiment acquis la connaissance du domaine. SOPHIE [Brown *et al.*, 1982] et STEAMER [Hollan *et al.*, 1987] sont des exemples des systèmes critiques.

1.6.5 Les systèmes sociaux

Les systèmes sociaux sont des STI qui expérimentent différentes combinaisons d'agents (par exemple : un ou plusieurs étudiants, un ou plusieurs enseignants, etc.). Ces agents

communiquent lors d'une session d'apprentissage. Chaque agent par exemple, peut donner son point de vue pendant la résolution d'un problème. Un exemple de ce type de système est le système compagnon (*LCS*) [Chan et Baskin, 1990].

1.6.6 Les systèmes planificateurs

Ces systèmes planifient les tâches à effectuer et interagissent avec l'apprenant sur la base de ces tâches [Lê, 1998]. Ce plan indique non seulement le contenu à enseigner mais aussi la manière d'enseigner ce contenu. De plus, lorsque l'apprenant éprouve les difficultés d'apprentissage, le système devra être capable de remédier à cette situation ou de générer un autre plan. BIP [Barr *et al.*, 1976] et WHY [Stevens et Collins, 1977] sont des systèmes planificateurs.

1.7 Quelques exemples de STI

Plusieurs systèmes d'EIAO ont été développés dans différents domaines d'enseignement. Certains sont restés au stade de prototypes, mais beaucoup d'autres ont été commercialisés et sont opérationnels.

1.7.1 Quelques systèmes représentatifs

Parmi les STI qui ont été développés dans les laboratoires de recherches, nous mentionnerons quelques systèmes représentatifs :

- ❖ **Le système SCHOLAR :** Le système SCHOLAR (c'est le premier système d'EIAO) concerne l'enseignement de la géographie d'Amérique du sud [Carbonell, 1970a]. SCHOLAR utilise un dialogue de type socratique.
- ❖ **Le système SOPHIE :** SOPHIE (SOPHisticated Instructional Environment) est l'un des systèmes les plus connus. C'est un système qui enseigne la matière sur la détection de pannes dans des circuits électroniques. Il a été développé dans les années 70 [Brown *et al.*, 1982].

- ❖ **Le système BIP** : Le système BIP ‘the Basic Instructional Program’ [Barr *et al.*, 1976] est présenté comme un atelier de résolution des problèmes. C’est un STI qui propose un cours de niveau secondaire ou collégial concernant la programmation BASIC.
- ❖ **Le système BUGGY** : BUGGY est un système pour l'apprentissage de règles arithmétiques élémentaires [Brown et Burton, 1978]. Ce système fournit un mécanisme pour expliquer pourquoi un étudiant fait une erreur, en plus de l’identifier.
- ❖ **Le système WHY** : SCHOLAR est étendu par le système Why. Il permet d'élucider les mystères de la météorologie (Pourquoi la pluie tombe-t-elle?). Comme SCHOLAR, ce système introduit également un dialogue de type socratique [Stevens et Collins, 1977].
- ❖ **Le système WEST** : WEST est un système d'EIAO pionnier dans l'application de l'enseignement basé sur la technique du guidage (‘coaching’) [Burton et Brown, 1979], pour le jeu éducatif sur ordinateur, populaire auprès des enfants utilisé dans les écoles primaires «how the WEST was won ».
- ❖ **Le système GEOMETRY Tutor** : ‘Geometry tutor’ est un STI utilisé pour enseigner la géométrie dans les écoles secondaires [Anderson *et al.*, 1985]. Ce système est l'exemple historique et typique d'un tuteur intelligent. C'est un produit commercial utilisé aujourd'hui aux États-Unis. La conception de ce système est basée sur le modèle cognitif fourni par la théorie d’ACT développée par Anderson (1984).
- ❖ **Le système LISP Tutor** : ‘LISP tutor’ [Reiser *et al.*, 1985] est un STI pour l'enseignement de la programmation LISP à des débutants. Comme ‘Geometry tutor’ ce système est également basé sur le modèle cognitif. Les deux systèmes incarnent les mêmes idées fondamentales.
- ❖ **Le système GUIDON** : GUIDON [Clancey, 1987] est un système qui enseigne la résolution de problèmes de diagnostics médicaux. Ce système a été développé à l'université de Stanford vers la fin des années 70 comme un projet de programmation de Ph.D. sous la direction de Bruce C. Buchanan.

- ❖ **Le système SHERLOCK :** SHERLOCK [Lesgold *et al.*, 1992] est un STI de type conseiller dont le but est de superviser et d'aider les apprenants pour mettre en pratique leurs connaissances dans la détection et la réparation des pannes d'avions militaires.

1.7.2 Autres systèmes

Au cours des dernières années, des dizaines de systèmes tutoriels intelligents ont été développés. Parmi ces systèmes on peut citer : le système MENTONIEZH [Py, 1990], le système CBMIP (A Case-Based Mathematics Instructional Planner) [Du et McCalla, 1991], le système STUDIA [Chevallier, 1992], le système TECHDOC-I [Peter et Rösner, 1994], le système ISIS-Tutor [Brusilovsky et Pesin, 1994], le système BIOTHECH [Woolf et Hall, 1995], le système MEDICUS [Folckers *et al.*, 1996], le système PSS-Tutor [Nkambou, 1996], etc.

1.8 Vers les STI intégrés

Les différents types de STI discutés précédemment ont chacun des aspects que d'autres n'ont pas, et qui sont pourtant des aspects nécessaires pour un enseignement ou un apprentissage de qualité; ce sont donc des systèmes complémentaires. Leur intégration dans un même système peut s'avérer très utile pour augmenter la puissance des STI. L'environnement modulaire proposé par Frasson *et al.* (1996a) va dans ce sens.

1.8.1 Architecture générale d'un STI

Plusieurs architectures de STI existent dans la littérature [Mengelle et Frasson, 1996] [Herzog, 1992] et [Frasson *et al.*, 1994]. Les modules similaires dans ces architectures peuvent être différents mais ils effectuent souvent des tâches similaires (souvent certains modules sont combinés).

Dans ce qui suit, nous représenterons l'architecture SAFARI qui est proposée par le groupe Héron [Frasson *et al.*, 1996a]. Cette architecture est décrite à la Figure 1.4. On y

trouve plusieurs composantes interdépendantes qui doivent être réunies. L'architecture Safari (du STI) est constituée des composantes suivantes [Frasson *et al.*, 1996a] :

❖ **Le sous-système Curriculum**

Le sous-système Curriculum [Nkambou, 1996] comporte deux parties :

1. la première partie (le curriculum) constitue une structuration d'une matière pour supporter l'enseignement; le curriculum comprend plusieurs structures de connaissances, comme par exemple : le réseau d'objectifs, le réseau des capacités, le réseau des ressources et le modèle pédagogique liant ces réseaux;
2. la deuxième partie est un objet de la classe Cours. Un cours est un programme d'enseignement qui vise à atteindre un ou plusieurs objectifs. À partir d'un curriculum on peut extraire plusieurs cours.

❖ **Le sous-système Modèle de l'apprenant**

Ce sous-système comporte également deux parties [Frasson *et al.*, 1996a]:

1. *le Modèle de l'apprenant* : selon Nkambou (1996), un modèle de l'apprenant comprend un recouvrement (overlay) sur les éléments correspondants du curriculum associé au cours suivi par l'apprenant.
2. *l'Information* : la classe information comporte les activités pédagogiques. Elle contient par exemple la connaissance concernée par la mise-à-jour, le niveau estimé de l'étudiant sur cette connaissance, etc.

Remarque

Actuellement dans l'architecture Safari (du STI), le modèle de l'apprenant contient un modèle de propagation et de contrôle permettant de gérer son évolution [Frasson *et al.*, 1996a]. L'initialisation, la consultation et la mise à jour du modèle de l'apprenant se font par le sous-système Gestionnaire de Session (GS) [Frasson *et al.*, 1996a].

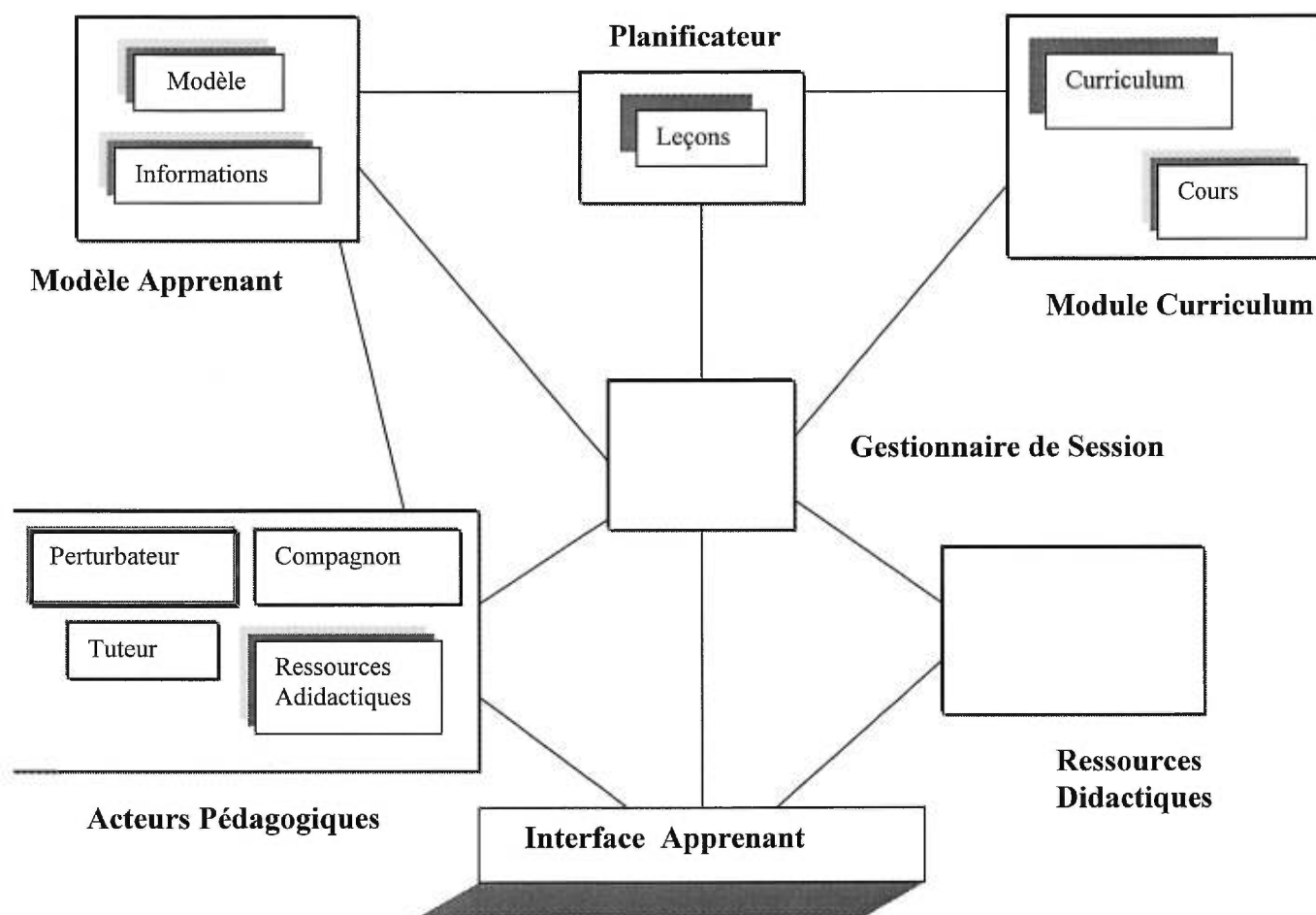


Figure 1. 4 : Architecture Safari [Frasson *et al.*, 1996a].

Évidemment, plus la modélisation de l'apprenant devient complexe (par exemple dans le cas où on veut développer une composante qui analyse le raisonnement de l'apprenant), plus le besoin de développer un sous-système qui peut effectuer lui même complètement ces processus devient nécessaire. Nous pensons que l'organisation de l'architecture d'un STI en sous-systèmes, de manière à ce que chacun puisse effectuer des processus spécifiques, nous permet d'augmenter l'efficacité, la facilité de construction, l'extensibilité et la flexibilité du système.

❖ Le sous-système Planificateur

Ce sous-système a pour objectifs [Frasson *et al.*, 1996a] :

- la construction (et la reconstruction) d'une leçon pour une session d'enseignement adaptée à un apprenant spécifique;
- la détermination de l'objectif qui doit être enseigné;
- la recommandation d'une ressource appropriée pour un objectif spécifique.

❖ **Le sous-système Gestionnaire de Session**

Le gestionnaire de session joue un rôle central dans le système; les objectifs de ce module peuvent se résumer comme suit (pour une description détaillée voir [Frasson *et al.*, 1996a]) :

- fournir l'enseignement le plus approprié possible à l'apprenant, pour lui permettre d'atteindre l'objectif en cause;
- décider du type de ressources qu'il utilisera durant une session d'apprentissage;
- déterminer comment on doit enseigner cet objectif;
- ordonner l'exécution d'une session d'apprentissage et contrôler des apprentissages effectués;
- faire le lien entre les divers éléments du système, communiquer avec la plupart des modules et interagir avec l'apprenant;
- se charger de faire la mise-à-jour du modèle de l'apprenant et construire les informations sur les activités pédagogiques;

❖ **Le sous-système Ressources Didactiques**

Selon Nkambou (1996), une ressource didactique est un moyen tactique utilisé par le système pour supporter les apprentissages. De façon générale, nous distinguons deux types de ressources didactiques : les ressources *intelligentes* et les ressources *passives*. Les ressources du premier type sont des ressources (exercices, problèmes, tests) pouvant évaluer les actions ou les réponses de l'apprenant. Les ressources du deuxième type sont des ressources hypermédia comme textes, images vidéo, sons, etc.

❖ **Le sous-système Acteurs Pédagogiques**

Le rôle principal de ce sous-système est de mettre en œuvre une stratégie pédagogique. Cette mise en œuvre se fait dans le but d'atteindre un objectif pédagogique précis par la coopération de plusieurs acteurs et à l'aide d'une ou plusieurs ressources adidactiques, par exemple, les connaissances utilisées pour choisir la stratégie tutorielle qui permet de tester la confiance de l'apprenant [Frasson *et al.*, 1996a].

❖ **Le sous-système Interface**

Ce sous-système a pour objectif de supporter l'interaction entre l'apprenant et les autres composants.

1.9 Problématique de développement des STI

La construction d'un STI implique la construction de chacun de ces modules en utilisant des techniques d'IA pour l'acquisition, la représentation et la construction des diverses bases de connaissances (les connaissances du domaine, les connaissances pédagogiques, les connaissances de l'apprenant, etc.). Ainsi que l'exploitation de ces connaissances, la détermination de tâches d'enseignement à accomplir par le système, la création de ressources didactiques pour réaliser ces tâches, etc. Nous allons décrire brièvement ici, les processus de développement, les difficultés liées au développement et les limites des systèmes tutoriels intelligents.

1.9.1 Les processus de développement des systèmes

Le développement de ces systèmes a pris un genre d'approche systématisée qui exige les processus suivants : analyse des tâches, conception, développement de prototypes, évaluation formative, implantation, évaluation sommative, et maintenance.

1.9.2 Les difficultés liées au développement des STI

Nous énumérons ici quelques difficultés liées au développement des STI :

1. Acquisition des connaissances d'un ou plusieurs enseignants dans un domaine donné.

2. Encodage de ces connaissances de manière à les communiquer aux apprenants.
3. La nécessité de prendre en compte les niveaux variés et les préférences de chaque apprenant.
4. Les difficultés d'appréhender dans de tels environnements certains types de raisonnements tenus par les apprenants.

1.9.3 Limites des systèmes tutoriels intelligents

On peut résumer les limites des systèmes tutoriels intelligents dans les points suivants :

- Le manque de moyens pour communiquer efficacement entre l'apprenant et l'ordinateur est un obstacle majeur. Le dialogue en langue naturelle n'est possible qu'à l'intérieur de situations structurées.
- La possibilité de saisir les connaissances d'un apprenant est très limitée car elle suppose que l'on soit capable de comprendre les différents procédés de son raisonnement.
- Les techniques IA sont plus difficilement applicables à des domaines moins bien structurés que la programmation, les mathématiques, etc.
- L'obligation de travailler en général sur des systèmes puissants n'allège pas vraiment le travail de développement.

1.10 Conclusion

Ce chapitre nous a permis d'analyser les systèmes tutoriels intelligents. Nous retiendrons par la suite, les points importants suivants :

- L'architecture classique d'un STI et de ses modules.
- La typologie des STI et les stratégies tutorielles.
- Les recherches sur les formes de systèmes tutoriels intelligents intégrant des variantes d'enseignants et d'apprenants ou des stratégies d'enseignement variées.
- L'architecture générale (l'architecture SAFARI) d'un STI et de ses modules.

Chapitre 2

Modèle de l'Apprenant

Intelligent tutoring systems research distinguishes itself from other educational software development in a number of ways. First, an ITS is concerned with **modelling** the knowledge of the learner in some computationally useful and inspectable way. Next, based on the cognitive model of the learner, the ITS **intervenes** in the interaction between the system and the learner with the goal of facilitating learning. Finally, the ITS **evaluates** the success of its intervention and adjusts its model of the learner; and the loop repeats.

- McCalla et Greer, 1991.

2.1 Introduction

Le modèle de l'apprenant est le composant du STI qui se charge des connaissances de l'apprenant. Ce dernier comprend *l'état courant des connaissances* de l'apprenant dans le domaine étudié. Il comprend aussi *un processus qui analyse*, puis évalue les performances de l'apprenant pour mettre à jour son modèle, appelé diagnostic [Burns et Capps, 1988] [VanLehn, 1988]. Ces deux composants devront être conçus ensemble pour constituer le modèle de l'apprenant.

Dans ce chapitre nous décrivons ce modèle, nous analysons ses problèmes, et nous indiquons les solutions proposées.

2.2 Utilité du modèle de l'apprenant

Si le système dispose de stratégies tutorielles, celles-ci seront mises en œuvre afin d'adapter l'enseignement à l'apprenant. C'est pour cela que le système dispose d'un modèle de l'apprenant plus ou moins fidèle, c'est-à-dire représentant plus ou moins bien l'apprenant humain (à la limite, dans un système sans modèle de l'apprenant, c'est l'interface qui gère les stratégies, sans connaître l'apprenant; donc l'enseignement n'est pas adapté).

Dans un STI, le modèle de l'apprenant est destiné à répondre aux questions concernant l'apprenant. Selon Self (1987) ces questions sont de quatre niveaux :

- Que peut faire l'étudiant ?
- Que sait l'étudiant ?
- À quel type d'étudiants le système s'adresse ?
- Qu'a fait l'étudiant ?

Selon VanLehn (1988) le modèle de l'apprenant est utilisé selon quatre buts :

- indiquer le prochain sujet enseigné par le STI;
- indiquer au tuteur le bon moment pour offrir de l'aide à l'étudiant;
- permettre au système de présenter à l'étudiant des problèmes adaptés à son niveau de connaissances;
- indiquer au tuteur le bon moment pour produire des explications personnalisées pour l'étudiant.

2.3 Problème de modélisation de l'apprenant

Les difficultés liées à la conception du modèle de l'apprenant sont décrites dans la littérature comme étant le problème de modélisation de l'apprenant.

Les modèles des apprenants ont pour objet de représenter toutes les facettes de ce dernier. Les informations qui doivent y être contenues sont liées à l'ensemble des tâches pédagogiques supportées par le système, ainsi qu'aux caractéristiques des matières enseignées.

Deux types de modèles sont proposés :

1. **Modèle de la matière** (Subject matter Model (SM)) : Des chercheurs proposent que les contenus du modèle de l'apprenant doivent tenter de représenter le plus fidèlement possible l'état de ces connaissances (réelles) dans le domaine étudié. Un SM est donc un modèle qui contient ces types de connaissances; il contient par exemple, les connaissances qu'il maîtrise, ses incompréhensions, etc. Un exemple de système avec un tel modèle est le 'Lisp Tutor' [Reiser *et al.*, 1985].
2. **Modèle des informations pédagogiques** (Pedagogic content Model (PM)) : Parfois les connaissances du domaine sont organisées selon les objectifs pédagogiques. Un PM est donc un modèle qui contient les informations sur l'évaluation des connaissances de l'apprenant par rapport aux connaissances pédagogiques stockées dans le système. Par exemple, WEST [Burton, 1982] modélise la force ou la faiblesse de l'étudiant dans les différentes habilités qui sont nécessaires pour le jeu.

Il y a un fossé entre les connaissances de l'apprenant dans le système et les connaissances du tuteur. La question qui se pose lorsque l'on veut modéliser les connaissances de l'apprenant est la suivante : 'comment peut-on établir un pont sur ce fossé ?'.

Cette question qui est identifiée par les chercheurs comme étant le problème de diagnostiquer les connaissances de l'apprenant, a attiré l'attention de beaucoup de chercheurs, par exemple on peut mentionner ici, les travaux de VanLehn (1988), Frasson et de La Passadière (1990), Holt *et al.* (1994), Verdejo (1994), et Ragnemalm (1996).

L'objet des sections qui suivent est de présenter en détails ce problème et d'examiner quelques approches présentées dans la littérature pour la modélisation de l'apprenant. Nous commençons tout d'abord par décrire les composants d'un modèle de l'apprenant.

2.4 La vue d'un modèle de l'apprenant par ses composants

De façon générale un STI maintient toutes les informations relatives à l'étudiant dans le modèle de l'apprenant. Il reflète la perception qu'a le STI de cet étudiant en modélisant

ses forces, ses faiblesses, et son niveau de connaissances du domaine. Nous détaillons ici les contenus de ce modèle en commençant par l'analyse de connaissances de l'apprenant.

2.4.1 La connaissance de l'apprenant

2.4.1.1 Types de connaissance de l'apprenant

VanLehn (1988) propose deux types de connaissances utilisées dans les systèmes : les connaissances *déclaratives* (les faits, les schémas, ...) et les connaissances *procédurales* (les règles de production). Une connaissance déclarative ne contient pas son mode d'emploi; par contraste, une connaissance procédurale indique explicitement comment elle est utilisée.

Définition de connaissances (déclaratives et procédurales) : les connaissances déclaratives sont représentées par des structures de données et sont liées aux savoirs, tandis que les connaissances procédurales sont représentées par des ensembles de règles qui infèrent des structures déclaratives et sont liées aux savoir-faire [Barr et Davidson, 1981] :

- Un savoir (ce que l'étudiant connaît à propos de la matière, par exemple la définition d'un objet, etc.) est une connaissance déclarative concernant un domaine donné représentée sous forme de faits.
- Un savoir-faire (ce que l'étudiant peut faire, par exemple, la réalisation d'une tâche ou la résolution d'un problème) est une connaissance procédurale, concernant un domaine donné, qui réfère à l'utilisation efficace de processus cognitifs et moteurs, relativement stables, dans la réalisation efficace d'une tâche.

Définition du modèle de l'apprenant : c'est une base de connaissances qui, par rapport au domaine enseigné, contient les informations relatives :

- aux savoirs acquis et aux savoir-faire maîtrisés par l'apprenant :
 - Un savoir acquis est une connaissance que possède à la fois le module expert du domaine et l'apprenant.
 - Un savoir-faire maîtrisé est une compétence que possèdent à la fois le module expert du domaine et l'apprenant.

- aux savoirs non acquis et aux savoir-faire non maîtrisés par l'apprenant :
 - Un savoir non acquis est une connaissance que possède le module expert du domaine, mais que l'apprenant ne possède pas.
 - Un savoir-faire non maîtrisé est une compétence que possède le module expert du domaine, mais que l'apprenant ne possède pas.
- aux incompréhensions de l'apprenant :
 - Une incompréhension est une connaissance que l'apprenant possède, mais que le module expert du domaine ne possède pas.

2.4.1.2 Synthèse de la connaissance de l'apprenant

Largeur de bande (bandwidth) : VanLehn (1988) précise une notion importante, celle de largeur de bande. Il s'agit de l'ampleur des données sur lesquelles se base le module apprenant pour appliquer les règles de diagnostic. Plus le « bandwidth » est élevé, plus les stratégies d'instruction peuvent être précises.

Selon VanLehn, le modèle de l'apprenant doit faire face à trois dimensions bien particulières de la modélisation de l'apprenant :

1. La première dimension (appelée «**La plage des informations**») vise la quantité d'informations connue par le système au sujet des agissements de l'apprenant face à un problème. Le minimum d'information consiste en la connaissance de la réponse d'un apprenant face à une question. Mais d'autres activités intermédiaires de l'apprenant peuvent être utiles, comme sa façon d'utiliser le clavier pour répondre. La plage des informations disponibles est répartie selon trois niveaux (Tableau 2.1) : les états *mentaux*, les états *intermédiaires* et les états *finaux*. Les états mentaux correspondent aux divers états cognitifs de l'apprenant. Les états intermédiaires correspondent aux diverses étapes de la résolution de problèmes par l'étudiant et les états finaux correspondent aux actions ou aux réponses fournies par lui.
2. La deuxième dimension, appelée **type de connaissances**, est représentée par une division des connaissances selon qu'elles soient déclaratives ou procédurales, avec une division supplémentaire au niveau de ces dernières (séquentielles ou hiérarchiques).

3. Enfin, la troisième dimension se rapporte à la capacité du système à établir les différences entre l'apprenant et l'expert.

2.4.2 Composants du modèle de l'apprenant

Dans cette section, nous étudions différentes vues d'un modèle de l'apprenant selon les modules qui le composent, par exemple :

- Un premier module représente les connaissances générales de l'apprenant qui ne sont pas directement reliées à la matière enseignée. Ces données peuvent par exemple représenter *l'âge, le nom et l'origine de l'apprenant*, etc. On peut aussi conserver ses aptitudes générales, son niveau de stress, etc.
- Un deuxième module est consacré exclusivement à la matière enseignée. C'est là que l'on retrouve le niveau de l'apprenant sur les différents points de la matière. Ce module s'emboîte généralement avec un curriculum représentant l'espace des connaissances à enseigner.
- Enfin, un troisième module, s'occupe de faire la mise à jour continue des deux modules précédents afin d'avoir à tout instant une représentation la plus fidèle possible de l'apprenant.

Les deux premiers modules forment la partie statique du modèle de l'apprenant, alors que le troisième module forme la partie dynamique.

En effet, on s'accorde généralement pour séparer le modèle de l'apprenant en différentes parties. Carr et Goldstein (1977) et Clancey (1979) citent par exemple, quatre informations nécessaires pour maintenir ce modèle : la difficulté du sujet de la matière à enseigner, les questions directement posées par l'apprenant, les performances et l'expérience d'apprentissage de l'apprenant.

Self (1987) définit le modèle de l'apprenant comme un 4-tuplet contenant les variables P (la connaissance procédurale), C (la connaissance conceptuelle), T (les caractéristiques individuelles), et H (l'historique).

Un modèle de l'apprenant comprend toujours des connaissances liées au domaine d'enseignement : ce que l'apprenant sait et ce qu'il sait faire. Dans son état le plus complet, ce modèle comprend aussi des connaissances indépendantes du domaine enseigné. Les unes concernent ses mécanismes d'apprentissage : comment l'apprenant fonctionne-t-il ? comment découvre-t-il de nouveaux concepts ? de nouvelles techniques ? etc. Les autres concernent les stratégies pédagogiques correspondantes : quels sont les types et les modalités d'intervention les plus efficaces ? etc.

Selon Nkambou (1996) le modèle de l'apprenant est identifié en trois parties : une partie cognitive (modèle cognitif), une partie inférentielle (modèle d'inférences), et une partie affective (modèle affectif).

Modèle cognitif : cette partie contient des informations sur l'état des connaissances de l'apprenant par rapport à la matière considérée. Ces informations portent sur :

- *les capacités* : l'information sur les capacités traduit le niveau de l'étudiant par rapport aux connaissances. Robert M. Gagné (1976) a classifié les capacités apprises en cinq catégories : les informations verbales, les habiletés intellectuelles, les attitudes, les habiletés motrices et les stratégies cognitives.
- *les objectifs* : l'information sur les objectifs traduit le fait que l'étudiant a déjà réalisé ou non un objectif. Par exemple, l'étudiant en médecine a acquis le savoir faire d'une technique de repérage de la plaque dentaire.
- *les ressources* : l'information sur les ressources (exercices, problèmes, tests, simulations, démonstration, etc.) traduit le fait qu'une ressource a déjà été utilisée par un apprenant, et le contexte dans lequel cette ressource a été utilisée.
- *et éventuellement les relations* : l'information sur les relations traduit le fait que l'étudiant a réussi ou pas à établir une relation (par exemple : de type analogie, abstraction, cas particulier, etc.) entre deux connaissances (et donc, la connaissance d'une relation entre deux connaissances est aussi une connaissance (méta-connaissance)).

Modèle d'inférences : cette partie est une sorte de moteur d'inférences qui tourne tout le temps (ou suite à un événement) pour ajuster le modèle de l'apprenant. Elle contient des

règles qui lui permettent de raisonner sur le modèle cognitif et sur le modèle psychologique (modèle affectif) pour inférer de nouvelles connaissances dans le modèle de l'apprenant.

Modèle affectif : ce modèle est un ensemble de données permettant de cerner la personnalité et les différentes facettes d'un étudiant. Il contient des connaissances relatives aux caractéristiques particulières permanentes ou momentanées de l'apprenant. Parmi celles-ci, nous avons :

- les connaissances relatives aux conditions mentales, par exemple l'apprenant est spatial ou verbal, il est réfléchi ou impulsif, etc.;
- les connaissances relatives aux sentiments et à la personnalité, par exemple l'apprenant est calme ou anxieux, il est attentif ou distrait, etc.

Enfin, notre choix s'est porté sur l'approche de Nkambou que nous avons détaillée; mais nous croyons que, pour que le modèle de l'apprenant soit efficace et plus complet, il faut séparer le moteur d'inférences du modèle d'inférences et consacrer cette partie pour les informations sur les modes d'inférences appliqués dans le raisonnement de l'apprenant. Le chapitre 4 offre plus de détails sur l'amélioration que nous avons faite par rapport à ce modèle.

2.5 Processus de Diagnostic

Le processus de diagnostic est divisé en trois phases [Ragnemalm, 1996] (Figure 2.1) :

1. *Acquisition des connaissances* : l'acquisition des connaissances est le processus qui effectue l'extraction des connaissances utiles pour la modélisation de l'apprenant.
2. *Transformation* : c'est le processus qui analyse les connaissances acquises dans la phase précédente (ou qui analyse les connaissances du tuteur) afin d'extraire les informations pertinentes et utiles dans le jugement d'habiletés de l'apprenant.
3. *Évaluation* : l'évaluation est le processus qui évalue les connaissances ou les conduites de l'apprenant. Cette évaluation se fait par rapport aux connaissances (celles du tuteur) stockées dans le système.

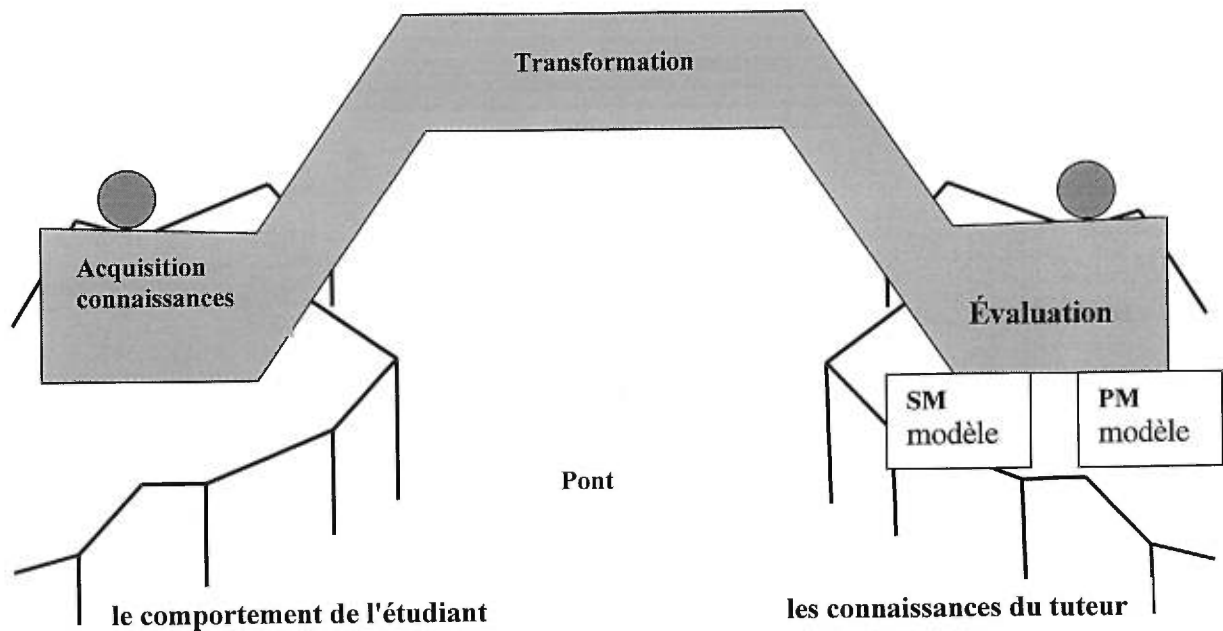


Figure 2. 1 : Les trois phases du processus de diagnostic [Ragnemalm, 1996].

En fait, la fonction de diagnostic permet au STI de mettre à jour les modèles de l'apprenant. Cette fonctionnalité est réalisée par le processus de diagnostic qui utilise différentes sources d'information pour exécuter son travail :

- *des informations implicites* : ces informations s'obtiennent directement en comparant les actions de l'étudiant avec celles de l'expert (exemple : le niveau de connaissances de l'apprenant inféré par le système);
- *des informations structurales* : ces informations proviennent du réseau des dépendances organisationnelles de la matière (exemple : liens entre les connaissances comme l'analogie, ou bien les liens entre une tâche et ses sous-tâches, etc.);
- *des informations explicites* : ces informations proviennent de l'interaction avec l'apprenant (exemple : demander directement à l'apprenant combien de fois il a exécuté une tâche);
- *des informations historiques* : ces informations permettent d'initialiser le modèle de l'apprenant (exemple : les informations statiques comme le nom de l'étudiant, ses aptitudes générales, etc.).

Pour évaluer les informations provenant de l'interaction avec l'étudiant, VanLehn (1988) analyse neuf stratégies différentes qui, historiquement, ont été appliquées dans les STI pour élaborer et mettre à jour le modèle de l'apprenant. Le tableau 2.1 suivant présente les techniques de diagnostic dont nous avons parlées.

Knowledge type Bandwidth	Procedural flat	Procedural hierarchical	Declarative
Mental States		Model tracing	
Intermediate States	Issue tracing	Plan recognition	Expert system
Final States	Path finding Condition induction	Decision tree Generate and test Interactive	Generate and test

Tableau 2. 1 : Techniques de diagnostic [VanLehn, 1988]

2.6 Les approches de modélisation de l'apprenant

Parmi les techniques de modélisation connues, nous pouvons citer :

2.6.1 Modèle de Recouvrement

La première approche consiste à construire le modèle de l'apprenant de façon comparative. Les performances de celui-ci sont comparées à un modèle idéal qui représente les connaissances d'un expert de la matière enseignée. Carr et Goldstein (1977) appellent cette technique, la *méthode de recouvrement* des connaissances de l'étudiant « overlay model ». Le modèle de l'apprenant est alors considéré comme un sous-ensemble du modèle idéal.

2.6.2 Modèle Correctionnel

D'autres STI représentent non seulement les concepts manquants, mais également certaines incompréhensions de l'étudiant à l'aide d'une librairie d'erreurs. Ce type de modèle est appelé *modèle correctionnel* [Burton et Brown, 1976]. Dans ce cas, le modèle de l'apprenant n'est pas un sous-ensemble du modèle idéal, mais plutôt un ensemble de règles qui représentent les connaissances (correctes et incorrectes) du domaine. Ainsi, les conceptions erronées de l'apprenant deviennent une variante de ces règles [Sleeman et Brown, 1982]. Du même coup, le cheminement du raisonnement de l'apprenant est modélisé par des règles de production qui prennent en compte les conceptions erronées.

2.6.3 Modèle de Reconstruction

Quelques STI se sont également intéressés à la détection des erreurs en utilisant une méthodologie basée sur la reconstruction des erreurs plutôt qu'à partir d'une liste d'erreurs prédéfinie dans une librairie. Dans ce cas, le module de diagnostic utilise deux librairies : une librairie de prédicats et une librairie d'actions. Dans la théorie de réparation, les erreurs sont générées en altérant un groupe de règles adéquates pour créer une situation où aucune règle ne s'applique (i.e. « impasse ») [Brown et VanLehn, 1980] [Burton et Brown, 1976]. Une réparation est une action qui consiste à contourner une impasse.

2.6.4 L'approche de Goldstein (Graphe Génétique)

Goldstein (1982) introduit l'idée de modélisation des connaissances de l'apprenant par l'intermédiaire d'un *graphe génétique* (Figure 2.2) représentant les relations d'analogie, de généralisation-spécialisation, de raffinement-simplification et de déviation-correction.

Le graphe génétique [Goldstein, 1982] offre une représentation des connaissances procédurales. Les règles d'une procédure sont représentées comme des nœuds, et les liens représentent les relations entre les règles. Ces liens peuvent être de plusieurs types (voir Figure 2.2) :

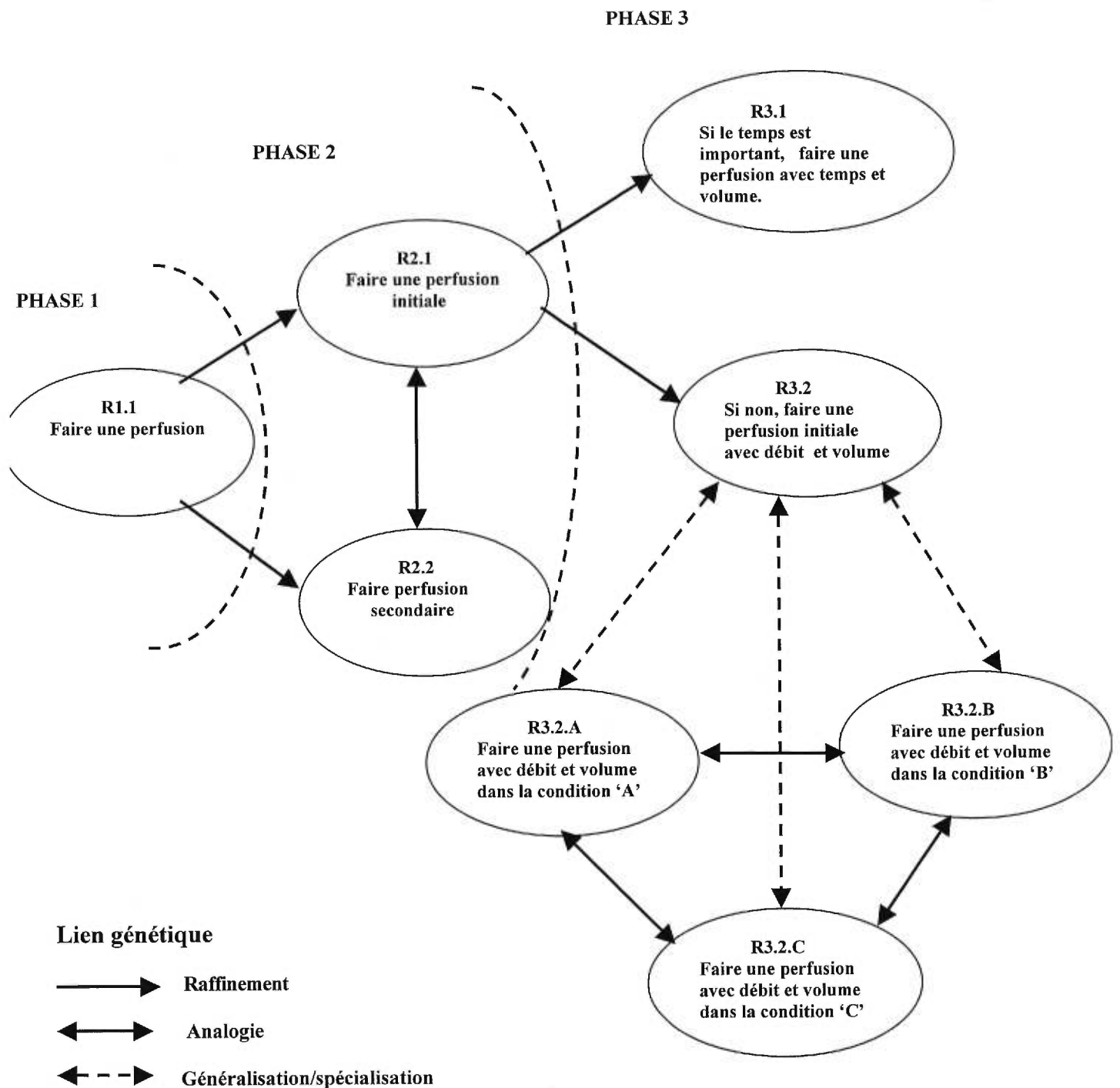


Figure 2. 2 : Exemple de graphe génétique [Goldstein, 1982].

- *les liens de généralisation (et les liens inverses de spécialisation)* : ces liens permettent de quantifier certaines connaissances. Dans la Figure 2.2 par exemple, le lien entre $R_{3.2}$ et $R_{3.2.A}$ est un lien de généralisation;

- *les liens d'analogie* : ces liens créent une correspondance entre deux connaissances (lien entre $R_{2,1}$ et $R_{2,2}$ par exemple);
- *les liens de raffinement (et les liens inverses de simplification)* : ces liens permettent de manipuler un sous-ensemble de connaissances;
- *les liens de déviation (et les liens inverses de correction)* : ces liens permettent de reconnaître les connaissances erronées.

Le problème sous-jacent à cette représentation est celui de la modélisation du raisonnement et de parcours correspondant dans le réseau. En effet, si l'on est capable de parcourir le réseau en fonction du raisonnement employé (analogie, induction, déduction, par l'exemple), on sera alors en mesure de mieux représenter l'apprenant et de mieux le comprendre.

Wasson (1985) a étendu les graphes génétiques en introduisant d'abord les nœuds composites, ensuite en utilisant ces nœuds pour représenter, en plus des règles, les faits, les concepts et les procédures. Chaque nœud est un graphe génétique qui peut être détaillé séparément. Ces graphes sont utilisés pour représenter les connaissances du professeur et celles de l'apprenant.

2.6.5 L'approche de John Anderson (La théorie ACT* (Adaptive Control of Thought))

John Anderson et son équipe ont proposé et expérimenté une méthodologie dit du traçage de modèle (*model tracing*) pour modéliser l'apprenant. L'objectif de leur recherche [Anderson et Pelletier, 1991] [Anderson *et al.*, 1995] est la conception et la validation d'une théorie (nommée à l'origine ACT (Adaptive Control of Thought)) cognitive d'apprentissage portant sur l'acquisition d'habiletés procédurales.

Anderson justifie la nécessité d'une rétroaction immédiate par des considérations pédagogiques à la base de sa théorie d'apprentissage. Essentiellement, selon lui, la rétroaction immédiate est nécessaire pour que l'apprenant puisse compiler de nouvelles connaissances.

La stratégie principale est la suivante :

- Il suppose que toutes les fonctions cognitives peuvent être représentées par une série de règles de production;
- Il est également stipulé que l'apprentissage se fait de façon déclarative initialement et qu'avec l'expérience et la pratique, ces connaissances sont assimilées par l'apprenant dans une forme analogue à la compilation de procédures;
- Finalement, une fois certaines règles de production acquises, elles ne peuvent être perdues.

Fonctionnement du traçage de modèle

Les systèmes dont la représentation de la matière est basée sur des règles de production effectuent leur analyse en essayant de reconnaître les règles appliquées par l'apprenant. Un bon exemple de cette démarche est donné par 'Lisp Tutor'. Ce dernier suit pas à pas le travail de l'apprenant et tente de déterminer quelle règle, bonne ou mauvaise est appliquée. Dès que 'Lisp Tutor' reconnaît que l'étudiant applique une règle erronée (« buggy rule »), il intervient pour corriger cette fausse conception. Le fonctionnement d'un tel système peut se résumer comme suit :

- Dans ce système, les activités cognitives de l'être humain sont représentées par un système de règles de production.
- La base de connaissances contient une série de règles montrant la suite d'étapes à exécuter pour atteindre un but donné, ainsi que les erreurs les plus courantes observées chez les apprenants. Ces informations sont utilisées pour construire le modèle de l'apprenant.
- Le système utilise les modèles « idéal » et « erroné » pour prédire le comportement de l'étudiant.
- À chaque nouveau symbole entré par l'apprenant, le système détermine quelle règle permet d'expliquer cette nouvelle entrée (la règle correspondant à l'entrée est considérée comme l'interprétation de l'action de l'apprenant).

- Avec toutes ces informations, le système peut détecter rapidement et facilement toutes les erreurs commises.
- De cette façon, le tuteur peut associer chaque action observable chez l'apprenant à un état cognitif supposé.

2.6.6 Autres approches

Plusieurs autres approches ont été employées pour mettre en application toutes ou certaines parties d'un modèle de l'apprenant : les réseaux de neurones par exemple ont été utilisés pour la classification des étudiants; des techniques d'apprentissage-machine [Sison et Shimura, 1998] peuvent être incorporées pour représenter comment la connaissance d'un étudiant change; et les approches numériques telles que l'approche « fuzzy » [Hawkes et Derry, 1990] ou le réseau de Bayes [Villano, 1992] sont utilisées pour la gestion d'incertitude.

2.7 Présentation du modèle de l'apprenant dans quelques STI

Dans ce qui suit, nous présentons quelques modèles d'apprenant dans des STI existants :

❖ Le système SCHOLAR

Le modèle de l'apprenant dans le système SCHOLAR [Carbonell, 1970b] ne contient que les connaissances que l'on suppose assimilées ainsi que les liens qui les relient. On utilise pour cela, la méthode de recouvrement appliquée sur le réseau sémantique représentant les connaissances du système. À un instant donné, le modèle de l'apprenant consiste en plusieurs régions distinctes de ce réseau que le tuteur tente de connecter selon une certaine stratégie d'enseignement afin d'unifier les connaissances du système et celles de l'apprenant.

❖ Le système BIP

BIP [Barr *et al.*, 1976] utilise un modèle de l'apprenant et un historique de l'étudiant pour la phase de diagnostic du travail de l'étudiant et pour la sélection de sa prochaine tâche. Le modèle de l'apprenant dans ce système est une forme de recouvrement.

❖ Le système WEST

Le système WEST [Burton, 1982] emploie un modèle de l'apprenant basé sur le modèle de recouvrement, pour analyser les réponses de l'étudiant. Dans WEST, lors de l'évaluation d'un savoir-faire, le processus de diagnostic est capable de découvrir les erreurs de l'apprenant en comparant ce que ce dernier est en train de faire avec ce qu'un expert ferait. Ils cherchent ainsi à pousser l'apprenant à optimiser ses actions et à maintenir son intérêt de participation durant le jeu.

❖ Les systèmes ALGEBRA et GREATERP

Les systèmes ALGEBRA [Lantz *et al.*, 1983] et GREATERP [Reiser *et al.*, 1985] sont également des systèmes qui utilisent :

- La représentation sous forme de règles de production pour représenter leurs bases de connaissances.
- La méthode de recouvrement pour représenter leur modèle de l'apprenant.
- Le modèle tuteur de ces systèmes renferme un certain nombre de stratégies sous forme de règles de production et guide constamment l'étudiant dans la résolution de ses problèmes.

❖ Le système SOPHIE

La philosophie adoptée dans le modèle de l'apprenant du système SOPHIE [Brown *et al.*, 1982] est un amalgame de la méthode de recouvrement et de détection des concepts erronés de l'étudiant. SOPHIE peut détecter toute inconsistance entre les actions de l'étudiant et la compréhension que le système suppose que celui-ci a du circuit (en utilisant le modèle de l'apprenant). Il décèle les conceptions erronées de l'étudiant au sujet du circuit et de son fonctionnement. Cette technique est appelée : identification des conceptions erronées.

❖ Le système BUGGY

BUGGY [Brown et Burton, 1978] [Burton, 1982] utilise un modèle de l'apprenant, basé sur le modèle correctionnel (avec une librairie d'erreurs). Ce système emploie une librairie contenant les mauvaises conceptions pour le domaine de l'arithmétique (soustraction).

❖ Le système WHY

Le système WHY [Stevens *et al.*, 1979] emploie un modèle de l'apprenant basé sur un concept qui consiste à identifier et corriger les conceptions erronées de l'apprenant (le modèle correctionnel). Il est à noter que lorsqu'il y a plus d'une erreur dans la réponse de l'étudiant, le système utilise des heuristiques pour décider quelle erreur il va corriger en premier.

❖ Le système ACM

Le système ACM [Langley et Ohlsson, 1984] utilise un modèle de reconstruction. Il reconstruit la séquence des opérateurs utilisés par l'apprenant à partir d'inférences sur les conditions d'application des opérateurs consistant en une conjonction de prédicats et d'une action élémentaire. En effet, le module de diagnostic utilise deux librairies : une librairie de prédicats et une librairie d'actions.

❖ Le système PIXIE

Le système PIXIE [Sleeman et Brown, 1982] utilise également un modèle de l'apprenant basé sur le modèle de reconstruction. Il génère des règles incorrectes en introduisant une action incorrecte de l'apprenant dans une procédure correcte.

❖ Le système Lisp Tutor

Le 'Lisp Tutor' [Reiser *et al.*, 1985] maintient un modèle idéal de l'étudiant qui représente les « bonnes » réponses aux divers problèmes que le système peut poser à l'étudiant, ainsi qu'un modèle d'erreurs contenant les différentes déviations de l'étudiant par rapport au modèle idéal. Ce système utilise la méthodologie du traçage de modèle

pour représenter le modèle de l'apprenant. Le module tuteur guide constamment l'étudiant dans une session d'apprentissage en lui donnant une rétroaction chaque fois que c'est nécessaire.

❖ **Le système Geometry Tutor**

« Geometry Tutor » [Anderson *et al.*, 1985] utilise également la méthodologie du traçage de modèle dans son approche. Ce système contient un modèle idéal indiquant la façon dont les apprenants devraient résoudre les problèmes soumis, et un modèle erroné représentant les erreurs typiques commises par des débutants. Le système guide l'étudiant dans la preuve des théorèmes en géométrie en lui donnant une rétroaction chaque fois qu'il est nécessaire. Les caractéristiques principales du style d'interaction de « Geometry Tutor » peuvent se résumer comme suit :

- À chaque étape de la construction d'une démonstration par l'apprenant, le tuteur génère toutes les possibilités d'application d'une règle dont il dispose.
- Il compare ensuite le pas de démonstration de l'apprenant avec celles-ci.
- S'il ne correspond à aucune possibilité, « Geometry Tutor » demande la modification du pas de démonstration.
- Au bout d'un certain nombre d'essais infructueux, il construit lui-même le pas par rapport à ce qu'il considère être la meilleure règle.

❖ **Le système STUDIA**

STUDIA [Chevallier, 1992] repose sur la mise en œuvre d'un modèle dynamique de dialogue structuré de façon hiérarchique autour des interventions, des échanges, des négociations et des plans (les plans sont des successions d'interventions du système). Le système conduit les dialogues, ouvre les négociations portant sur la résolution des problèmes. Les caractéristiques principales du style d'interaction de STUDIA peuvent se résumer comme suit :

- La communication entre le système et l'apprenant se déroule dans le cadre de négociations qui prennent en compte le modèle de l'apprenant.
- Le système dispose de plans d'analyse de la réponse de l'apprenant.

- Le choix du plan de questions se fait en fonction des hypothèses faites par le système sur l'historique de l'apprenant.
- Les interventions sont essentiellement de type question/réponse, appréciation d'une réponse, commentaires neutres d'une intervention de l'apprenant, conseils, aides, explications adressées à l'apprenant, ainsi que par des informations données sous forme d'extraits de cours.
- Le contenu de ces interventions évolue selon le déroulement de la négociation, selon les éléments significatifs survenus au cours des négociations précédentes et selon le niveau de l'apprenant.

❖ Le système SHERLOCK

Le système SHERLOCK [Lesgold *et al.*, 1992] est basé sur une solide analyse des tâches qui a permis d'identifier les aspects du travail de détection de pannes qui posent des problèmes en raison de leur difficulté. Une motivation de l'extension de ce système est le besoin de rendre le système plus intelligent, grâce à une modélisation plus fine du système physique et des performances de l'expert.

L'approche retenue dans ce travail consiste à séparer les activités d'apprentissage en deux parties :

1. Dans la première partie, l'élève reçoit des conseils quand il le désire, pendant qu'il résout un problème.
2. Dans la deuxième partie, l'élève peut rejouer sa solution en visionnant le diagramme d'un circuit réel, simulé par le système. En cliquant sur un composant du diagramme, il reçoit des explications, au regard de la solution prévue dans le module expert.

Ce système utilise une approche basée sur la théorie des ensembles flous (voir 2.6.6) à modéliser les capacités (imprécises) de l'apprenant.

2.8 Discussion

Les différentes approches de modélisation discutées précédemment ont chacune leurs avantages et leurs limites. Nous présentons ici certains avantages et limites de ces modèles.

- **Le modèle apprenant de type recouvrement**

1. un avantage de cette approche est qu'elle est simple à implémenter;
2. la représentation des connaissances de l'apprenant dans ce modèle est comme un sous-ensemble de celles du module expert du domaine. Cela restreint les STI aux domaines où les méthodes de résolution de problèmes sont limitées et toutes comprises dans le module expert;
3. ce modèle ne peut pas saisir la dimension procédurale liée aux incompréhensions de l'apprenant puisqu'il ne possède pas de représentation adéquate des processus impliqués dans l'application de savoir-faire qu'il modélise.

- **Le modèle apprenant de type correctionnel**

1. un grand avantage de cette approche est qu'elle permet de construire un modèle de l'apprenant comprenant aussi bien les connaissances correctes que les connaissances incorrectes relevées;
2. afin que la représentation du modèle de l'apprenant soit complète, la librairie d'erreurs doit être assez complète pour pouvoir représenter toutes les erreurs possibles. Une telle librairie n'est pas toujours disponible;
3. une erreur peut parfois être décomposée en d'autres erreurs plus élémentaires. Une combinaison de ces erreurs élémentaires doit être représentée dans la librairie. Il n'y a pas de méthodes pour déterminer la granularité nécessaire à la représentation de ces combinaisons [Wenger, 1987].

- **Le modèle apprenant de type reconstruction**

1. dans cette approche, on n'a pas besoin de fixer à l'avances la librairie d'erreurs, elle peut être reconstruite sur la base des erreurs observées;
2. cette approche ne peut pas représenter toutes les erreurs potentielles d'un apprenant.

- **Modèle de l'apprenant basé sur les graphes génétiques**

1. cette approche peut représenter la manière dont l'apprenant apprend à partir des connaissances déjà connues. Ceci est fait par la mise en évidence des processus d'apprentissage qui permettent de trouver les conditions dans lesquelles les liens génétiques (tels que l'analogie, la génération, ..) sont formés;
2. un des inconvénients important de cette approche est que le graphe génétique est statique, parce que les nœuds entiers doivent être prédéterminés. Le traçage du progrès de l'apprenant est limité donc au graphe initial. Il est donc peu réaliste pour un grand domaine [Brecht et Jones, 1988].

- **Traçage de modèle**

- a. l'avantage d'une telle approche est que la connaissance contenue dans le système est enregistrée sous une forme facilement communicable à l'apprenant. Puisque le tuteur possède en quelque sorte un instantané de l'activité cognitive de l'apprenant, il connaît le contexte dans lequel l'erreur (s'il y a lieu) s'est produite, et peut donc intervenir de façon efficace.
- b. les systèmes tutoriels intelligents basés sur cette approche sont efficaces pour l'enseignement de domaines très structurés mais perdent de leur efficacité lorsque le domaine est non structuré (comme l'histoire, la philosophie, etc.).
- c. une telle approche est en général très directive et ne laisse pas beaucoup de latitude à l'apprenant.

Nous aurons l'occasion dans les chapitres suivants de développer beaucoup plus le modèle de l'apprenant et de proposer un modèle basé sur le raisonnement à base de cas. Pour situer notre approche, nous abordons au chapitre suivant quelques aspects fondamentaux du raisonnement à base de cas et nous donnons un bref aperçu des applications concernées par ce mode de raisonnement.

2.9 Conclusion

Ce chapitre nous a permis d'analyser le modèle de l'apprenant. Nous retiendrons par la suite, les points importants suivants :

- Le rôle du modèle de l'apprenant est central au STI, puisque ce composant est la clé de la formation individualisée, dans l'interaction avec l'étudiant.
- La modélisation de l'apprenant a été identifiée comme une tâche complexe, difficile mais importante par les chercheurs.
- Plusieurs techniques de modélisation de l'apprenant existent, mais toutes ces approches ne permettent pas réellement de représenter fidèlement l'état des connaissances de l'apprenant en situation de formation.

Chapitre 3

Le Raisonnement à Base de Cas

Case-based reasoning can mean adapting old solutions to meet new demands, using old cases to explain new situations, using old cases to critique new solutions, or reasoning from precedents to interpret a new situation (much like lawyer do) or create an equitable solution to a new problem.

- Janet Kolodner, 1991.

3.1 Introduction

Le Raisonnement à Base de Cas (RBC ou Case-Based Reasoning (CBR)) [Kolodner, 1993] [Leake, 1996] [Watson, 1997], constitue une approche intéressante, parmi d'autres modes de raisonnements, pour la réalisation des systèmes à base de connaissances. Le raisonnement par cas (ou fondé sur des cas) est une forme de raisonnement analogique. Il consiste à raisonner à partir d'expériences et de cas déjà rencontrés pour résoudre de nouveaux problèmes. Ce raisonnement est utile pour différents types d'activités de résolution de problèmes, comme la planification de tâches, le diagnostic, la création d'objets, etc.

Ce chapitre donne une introduction conceptuelle aux techniques du raisonnement à base de cas. Ce que nous présentons ici sont : la conception du RBC, son origine, sa méthodologie, le cycle du RBC, ses applications, quelques exemples des systèmes représentatifs, et finalement l'intégration du RBC aux STI.

3.2 Conception du Raisonnement à Base de Cas

RBC est un type de raisonnement basé sur la réutilisation des expériences antérieures appelées « cas » pour résoudre de nouveaux problèmes.

Définition de Cas : Les cas sont des connaissances (statiques et (ou) dynamiques) expérimentales. Chaque cas représente une expérience, c'est-à-dire une situation rencontrée dans le monde réel, ou bien créée par le système au cours de son raisonnement.

Définition du Raisonnement à Base de Cas : le RBC est une méthodologie de l'intelligence artificielle pour effectuer un nouveau traitement en utilisant (éventuellement en les modifiant) un ou plusieurs cas mémorisés [Bichindaritz, 1994].

Le RBC signifie le raisonnement sur des exemples antérieurs qui ont été enregistrés dans ce que nous appelons une base de cas. Cette base d'expériences forme la mémoire du système de RBC.

Comprendre comment le RBC fonctionne est très simple et est une des raisons pour lesquelles il a été adopté aisément et de façon intensive. La Figure 3.1 décrit la démarche d'un système à base de cas. En général, un tel système comporte deux composants : un chercheur et un adaptateur de cas. Étant donné un nouveau problème en entrée, le chercheur identifie les cas les plus similaires dans la base de cas et les présente à l'adaptateur de cas. L'adaptateur de cas examine alors les cas trouvés et essaye de résoudre le nouveau problème en adaptant ces cas.

3.3 Les Origines du Raisonnement à Base de Cas

Le RBC a été initialement développé (années 80) comme méthodologie pour mettre en valeur la connaissance des ordinateurs. Son but était de permettre le développement de systèmes experts plus performants, ceux qui pourraient devenir des raisonneurs plus capables et plus efficaces par l'expérience [Schank, 1982].

Les origines de RBC sont situées à l'intersection de deux approches : la psychologie cognitive (dans la trame de l'étude de la mémoire) et le raisonnement par analogie.

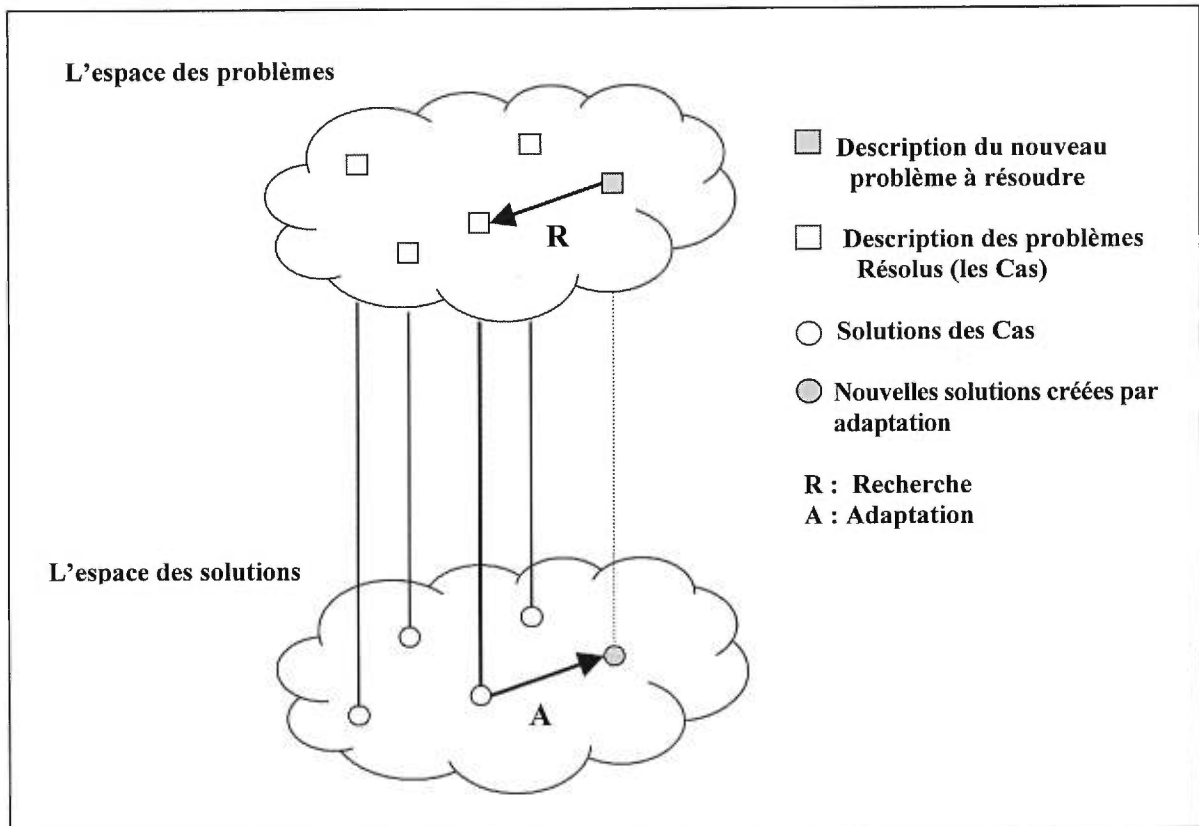


Figure 3. 1 : Conception du RBC [Leake, 1996]

Dans la suite, nous allons brièvement décrire la relation entre RBC et ces deux approches (c.-à-d., *la psychologie cognitive et le raisonnement par analogie*).

3.3.1 La mémoire dans les sciences cognitives

L'étude de la mémoire a fait l'objet de plusieurs travaux de recherche. Dans les années 70, Schank et ses collègues ont travaillé sur plusieurs structures de connaissances et leur influence sur la compréhension du texte [Schank, 1975] [Schank et Abelson, 1977] [Riesbeck et Schank, 1981]. Schank a trouvé que la compréhension du langage n'est possible que grâce à la mémoire. En 1982, il a proposé la théorie de la mémoire dynamique. Une mémoire dynamique peut être définie comme une mémoire qui peut

changer son organisation lorsque de nouvelles expériences le demandent; c'est donc une mémoire qui peut apprendre. Cette théorie qui est une extension de ses travaux précédents sur le phénomène de la compréhension, s'appuie sur les constatations personnelles et sur le phénomène de rappel. Selon Schank, le phénomène de rappel est une conséquence immédiate de la structure même de la mémoire : se rappeler d'une situation particulière alors que l'on est en train de traiter une autre résulte du fait que celle-ci partage des informations en commun avec la situation rappelée. Schank s'est inspiré des propriétés de la mémoire humaine pour proposer son modèle de la mémoire dynamique. Il propose que l'intégration d'une situation en mémoire, active d'autres situations qui ont certaines informations en commun. Plus le niveau d'intégration est profond plus la quantité d'informations en commun est grande. De plus, la situation est pertinente :

- elle permet plus de comprendre la situation à intégrer;
- de mieux orienter l'intégration;
- de mieux suggérer une solution si la situation à intégrer est un problème à résoudre.

Dans ses études, Schank (1982) a distingué deux grandes catégories de structures pour la mémoire :

1. *les structures dépendantes d'un domaine MOP (Memory Organization Paquets)* dont le rôle est de relier les différentes scènes, et qui contiennent des informations générales, dégagées d'un contexte. Les MOPs eux-mêmes sont organisés par des meta-MOPs. Un exemple de meta-MOP est celui du VOYAGE [Bichindaritz, 1994] :

meta-MOPs	MOPs
VOYAGE	AVION HOTEL CONGRES RECEPTION

2. *les structures indépendantes d'un domaine TOP (Thematic Organization Paquets)* dont le rôle est de renfermer des informations abstraites, assimilables à des types, et

permettant des remémorations entre des domaines différents (par exemple, le rapprochement des histoires de « Roméo et Juliette » et de « West Side Story ») [Bichindaritz, 1994].

3.3.2 RBC et raisonnement par analogie

Le raisonnement par analogie est une stratégie cognitive qui consiste en la mise en correspondance de deux éléments, dont l'un est connu (appelé source du raisonnement) et l'autre est partiellement connu (appelé cible du raisonnement). Son objet est de transférer des connaissances de la source vers la cible, afin que celle-ci soit ensuite mieux connue. Le raisonnement à base de cas peut donc être considéré comme un raisonnement par analogie dont la source et la cible appartiennent à un même domaine.

3.4 La méthodologie du raisonnement à base de cas

La réalité du RBC fait défaut à la flexibilité de la réutilisation que l'humain peut réaliser. Ceci, parce que les gens s'aperçoivent aisément que beaucoup de compétences humaines sont basées sur la réutilisation des solutions aux problèmes précédemment produits.

Il existe un grand nombre d'observations en psychologie cognitive qui montrent l'importance des expériences du passé pour résoudre un problème. Ross (1990), par exemple, a prouvé que les gens apprenant une nouvelle compétence renvoient souvent aux problèmes précédents pour régénérer leurs mémoires sur la façon dont a été réalisée la tâche.

En général, l'approche du RBC est basée sur deux principes au sujet de la nature du monde. Le premier principe est que le monde est régulier : les problèmes semblables ont des solutions semblables. En conséquence, les solutions pour les problèmes antérieurs sont un point de départ utile pour la résolution de nouveaux problèmes semblables.

Le deuxième principe est que les types de problèmes qu'un agent rencontre tentent de se reproduire. En conséquence, les futurs problèmes sont susceptibles d'être semblables aux problèmes actuels. Quand ces deux principes se tiennent, il est intéressant alors de se

rappeler et de réutiliser le raisonnement antérieur. De cette façon la méthodologie du RBC est proposée comme modèle du raisonnement humain

3.5 *Le cycle du RBC*

Le cycle du raisonnement à base de cas proposé par Aamodt et Plaza (1994) comporte un certain nombre d'étapes qui sont : *Recherche*, *Réutilisation*, *Révision*, et *Recyclage*.

La Figure 3.2 présente ce cycle : à la rencontre d'un nouveau problème, on cherche en premier dans la base de cas s'il y a des cas similaires. La recherche se base sur la comparaison de la nouvelle situation et des cas en mémoire. Une fois qu'un cas est retiré, sa solution est adaptée pour le présent problème. Le système essaiera ensuite de réparer et tester de nouveau si la solution qui en résulte ne s'applique pas à la situation en cours. Finalement, si la solution confirmée s'applique à la situation, alors un nouveau cas (le problème et sa solution) est ajouté à la base de cas.

Cependant, chaque étape de ce cycle est décomposée en plusieurs processus :

- **Recherche** est décomposée en trois étapes : *identifier des index* (identifier un ensemble approprié des descripteurs du problème), *rechercher* (retour d'un ensemble de cas qui sont semblables au nouveau problème) et *choisir* (choisir le cas le plus approprié).
- **Réutilisation** est décomposée en deux étapes : *copier* (prendre le cas choisi comme base) et *adapter* (transformer la solution du cas choisi en une solution pour le nouveau problème).
- **Révision** est décomposée en deux étapes : *évaluer la solution* (évaluer le succès de la solution construite dans la tâche de réutilisation) et *réparer le défaut* (détecter des défauts dans la solution actuelle et produire ou rechercher des explications pour elles).
- **Recyclage** est décomposé en trois étapes : *extraire* (identifier l'information qui doit être stockée), *indexer* (identifier le type d'incrémentaires nécessaires pour la recherche future aussi bien que la structure de l'espace de recherche), et *intégrer* (mettre à jour

la base de connaissances avec le nouveau cas (le problème courant + la solution trouvée) pour être utilisé dans la résolution future des problèmes).

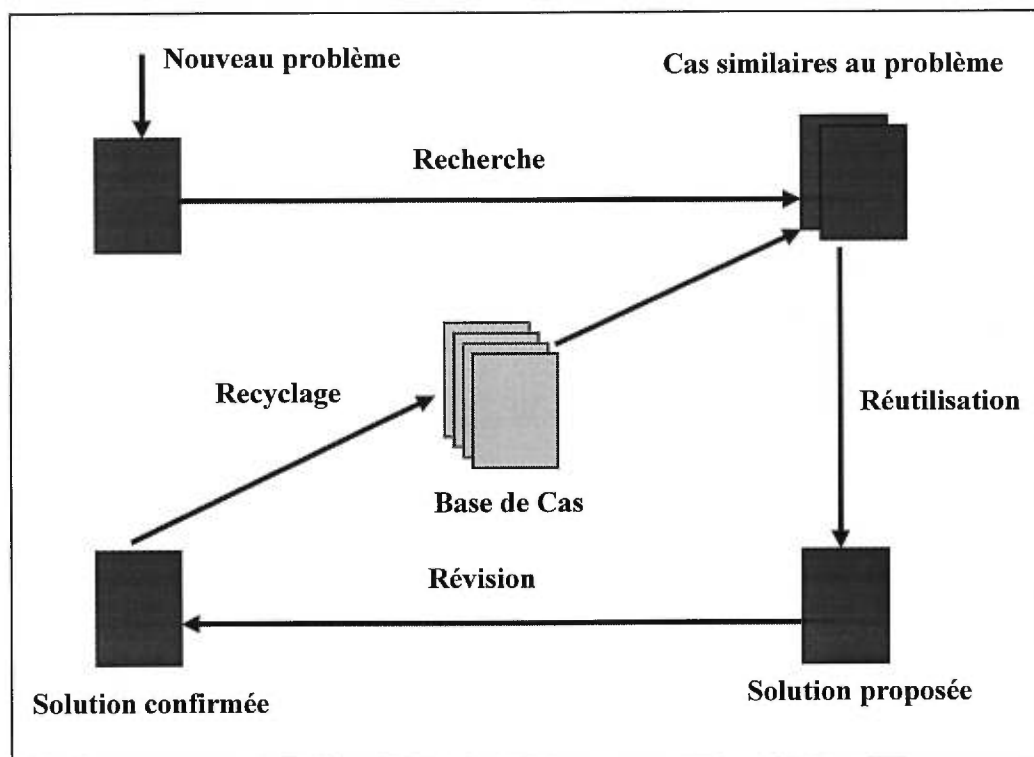


Figure 3. 2 : Le cycle du RBC

Nous utilisons le modèle proposé par Aamodt et Plaza, dans notre approche pour quatre raisons principales. D'abord, ce cycle ne diffère pas d'autres cycles de RBC décrits dans la littérature, en ce qui concerne son contenu de base. En second lieu, la combinaison de ce cycle et d'analyse au niveau de la connaissance est très utile pour notre problème. La troisième raison est que, ce modèle a été utilisé avec succès dans l'évaluation du RBC en tant que moyens d'analyse et de comparaison. Quatrièmement, ce cycle de RBC semble être largement cité dans la littérature (voyez par exemple [López de Mántaras et Plaza, 1997] et [Watson, 1997]).

3.6 *Éléments constitutifs du RBC*

Nous allons maintenant revoir en détail les concepts énoncés dans la section précédente :

3.6.1 Recherche

Le but de la recherche est de trouver les cas semblables que l'on pense plus utiles pour résoudre le problème actuel. Ceci est exécuté en recherchant des cas qui contiennent une définition qui est semblable à la définition du problème actuel. Il est évident que, plus la phase de recherche est efficace, plus les exécutions globales du système de RBC sont augmentées. Par conséquent, une attention particulière doit être prêtée à la conception et à la mise en place de cette phase. L'organisation et l'indexation de mémoire sont les issues principales pour la recherche efficace de cas.

3.6.2 L'organisation et l'indexation de la mémoire

L'organisation et l'indexation des cas dans la mémoire est une partie fondamentale du raisonnement à base de cas qui implique les processus 'apprendre' et 'raisonner'. Ce problème peut être divisé en deux éléments. Le premier est la sélection des index des cas qui peuvent être employés pour classer et rechercher les cas (indexation). Le second est l'organisation de la mémoire de sorte que le processus de recherche de cas soit efficace et précis.

3.6.2.1 *Indexation*

Le but de cette étape est d'indexer les cas pour qu'ils puissent être comparés aux cas déjà contenus dans la base de cas. Les index des cas agissent comme des index aux livres dans une bibliothèque. Le problème d'indexation tient au fait de s'assurer qu'un cas est consulté toutes les fois de manière appropriée. Ceci signifie que le cas devrait être classé afin d'être accessible toutes les fois que c'est nécessaire, il faut également que les algorithmes de cette recherche puissent employer ces index pour obtenir les cas appropriés en temps voulu.

En plus de devoir être appropriés à une tâche, les bons index pour le raisonnement à base de cas doivent satisfaire deux autres propriétés :

1. Les index devraient être assez abstraits pour rechercher un cas approprié dans une variété de situations futures.

2. Les index devraient être assez concrets pour être facilement reconnaissables dans les situations futures.

Nous présentons maintenant les notions d'index, d'indice et d'indexation.

Définition : L'indexation d'un cas consiste à sélectionner un certain nombre d'attributs qui deviennent ses indices [Kolodner, 1993].

Définition : Les index sont les structures en mémoire contenant les indices.

Définition : Un indice est un élément de description d'un cas permettant de le retrouver en mémoire. Les indices doivent permettre de différencier les cas lors de la phase de recherche et de comparaison des cas entre eux. Ils sont exprimés en fonction du vocabulaire de description du domaine.

Le choix des indices est un aspect essentiel du problème d'indexation, et conditionne l'accès à la mémoire dans l'avenir. Parmi les méthodes proposées pour le choix des indices, les plus importantes sont :

- **la méthode inductive pure** : elle consiste à extraire les différences ou les similitudes entre le nouveau cas et les autres cas dans la base de cas;
- **la méthode des listes à cocher** : elle utilise une liste de traits connus (d'après les connaissances des experts) pour leur capacité de prédiction.

3.6.2.2 *Organisation de la mémoire*

L'organisation de la mémoire, pour une recherche efficace, est un aspect important pour la mise en place d'un système de RBC. Le problème consiste à formuler les cas de manière à disposer de sous ensembles réutilisables. Pour ce faire, les questions qui se posent sont essentiellement les suivantes :

- Comment représenter un cas ?
- Comment structurer un cas en éléments qui puissent être réutilisables dans d'autres cas ?
- Comment archiver les cas dans une base de manière à pouvoir les retrouver ?

La plupart des systèmes actuels utilisent deux types d'organisations : une mémoire séquentielle (mémoire plate) ou une mémoire hiérarchisée.

Mémoire séquentielle : Dans le système de mémoire plate, des cas sont enregistrés séquentiellement dans une simple liste ou un fichier. L'avantage de cette approche est que : (1) toute la base de cas est recherchée, et son exactitude est donc une fonction de la qualité (de fonction) d'adaptation; et (2) ajouter de nouveaux cas (incrémentalement) est bon marché; cependant, ce dernier devient très cher quand la base de cas devient plutôt grande.

Mémoire hiérarchisée : les réseaux à traits partagés ont permis de grouper des cas de sorte que ceux qui partagent plusieurs indices soient groupés ensemble. Chaque nœud interne du réseau juge des indices partagés par des cas au-dessous de lui. Établir une telle hiérarchie prend du temps, mais elle peut être utilisée très efficacement pendant le processus d'inférences. Maintenant, ajouter des cas est une exécution complexe et il est difficile de maintenir le réseau de façon optimale. L'étude de certaines approches inductives comme ID3/C4 [Quinlan, 1986] ont prouvé que les coûts de modification des grandes bases de cas sont très élevés et peuvent mener parfois à une recompilation complète de la base de cas après chaque changement. Ceci montre que l'utilisation de la recherche inductive est plus efficace mais complique considérablement la phase d'étude en raison de l'architecture complexe de la structure utilisée [Utgoff, 1989].

En effet, l'organisation des cas et leur indexation sont donc importantes et le succès du système dépend en grande partie de la pertinence des critères servant à l'indexation. Nous discutons ci-dessous le rôle de la mesure de similarité dans la recherche de cas.

3.6.3 Mesure de similarité

Les index sont utilisés pour rechercher dans la mémoire de cas un cas similaire. Cette recherche nécessite une mesure de similarité permettant de choisir entre plusieurs cas candidats. Dans les systèmes à base de cas, la similarité de deux cas est calculée en comptabilisant les caractéristiques communes. Le calcul de similarité d'un problème se fait avec plusieurs cas en mémoire. Généralement, le cas ayant le plus grand score

numérique, sera pris comme le cas le plus similaire et sa solution sera adaptée pour tenir compte des différences avec le cas problème.

Il existe plusieurs méthodes de calculer le degré de similarité entre deux cas. Nous détaillons les méthodes suivantes : le *modèle logique*, le *réseau sémantique*, l'*échelle numérique ou qualitative* [Kolodner, 1980] [Aha, 1991].

1. Le modèle logique

Dans un modèle logique les objets sont représentés par une conjonction de prédicats. La similarité entre deux cas se fait en calculant le rapport entre le nombre d'attributs ayant la même valeur et le nombre total d'attributs. A partir de ce type de similarité, il est possible d'en dériver de plus complexes, prenant en compte par exemple le poids de chaque attribut ou la distance entre deux valeurs.

Exemple : La similarité entre deux oiseaux :

$E_1 : \text{vol}(\text{cou-tendu}) \wedge \text{forme-bec}(\text{cassé}) \wedge \text{couleur-bec}(\text{noir}) \wedge \text{taille-cm}(88)$

$E_2 : \text{vol}(\text{cou-tendu}) \wedge \text{forme-bec}(\text{cassé}) \wedge \text{couleur-bec}(\text{rose}) \wedge \text{taille-cm}(85)$

se fait en calculant le rapport entre le nombre d'attributs ayant la même valeur (vol et forme-bec) et le nombre total d'attributs (= 4). La similarité logique entre E_1 et E_2 est donc :

$$S(E_1, E_2) = 2 / 4 = 0.5.$$

2. Le réseau sémantique

Dans ce modèle les objets sont représentés par un graphe. Le degré de similarité est calculé en terme d'abstraction commune la plus spécifique pour deux valeurs.

Exemple : Si on considère la hiérarchie suivante (la Figure 3.3) :

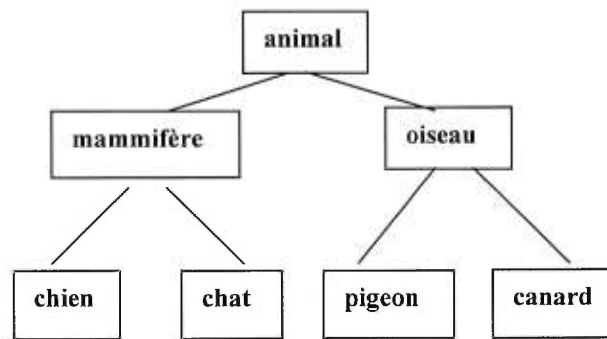


Figure 3. 3 : Une hiérarchie simple d'abstraction

On peut dire que la similarité entre chien et chat est supérieure à la similarité entre chien et canard, car l'abstraction commune pour chien et chat (mammifère) est plus spécifique que l'abstraction commune à chien et canard (animal).

3. L'échelle qualitative

L'échelle qualitative est utile seulement pour mesurer la similarité entre des valeurs qualitatives. Dans cette méthode si deux valeurs appartiennent à la même région qualitative, alors elles sont considérées comme égales, sinon la distance entre leurs régions qualitatives donne une mesure de leur similarité.

Exemple : CASEY utilise les intervalles suivants (tableau 3.1) pour comparer les âges des patients. Dans cet exemple l'âge 62 ans et l'âge 77 ans sont dans la même région qualitative. Les âges 40 et 62 ont un écart d'une région qualitative. L'usage d'une échelle qualitative pour mesurer la similarité entre deux valeurs numériques est acceptable quand l'écart entre elles est faible. Par contre aux extrémités des intervalles le degré de similarité est imprécis. Par exemple, 39 et 40 devrait avoir la même similarité comparé à 62 mais comme ils appartiennent à deux zones différentes, leurs degrés de similarité avec 62 sont différents.

intervalles	valeurs
60 ans et plus	Âge vieux
40 à 59	Âge moyen
20 à 39	Jeune adulte

Tableau 3. 1 : Intervalles des âges dans CASEY

3.6.4 Adaptation

L'adaptation de cas est un des aspects principaux du raisonnement à base de cas, c'est-à-dire, faisant des modifications appropriées aux cas passés afin d'utiliser leurs résultats en résolvant un problème actuel. Si ce dernier est très semblable au cas recherché, le processus d'adaptation devient simple, répétant le même ensemble d'actions que dans les cas passés. Mais, en général, le cas recherché n'est pas toujours près du problème donné; l'adaptation doit alors être modifiée en conséquence.

Malgré les progrès considérables réalisés dans les techniques utilisées lors de la recherche de cas, la partie adaptation de solution reste un problème ouvert [López de Mántaras et Plaza, 1997]. Elle est un problème complexe, difficile à définir et est l'étape principale du processus de RBC, parce que la qualité de la solution obtenue dépend largement de ce processus. Sa complexité dépend du genre de domaine d'application, et des différences entre le cas courant (problème) et des cas passés. Il est évident que quand la base de cas contient peu des cas, la probabilité que l'on ne trouve aucun cas adaptable est importante, et il sera alors nécessaire de faire un bon nombre de modifications pour réutiliser un cas. Les méthodes d'adaptation proposées par les chercheurs sont les suivantes [Aïmeur et Boudina, 1999] :

1. L'adaptation à base de règles

L'adaptation à base de règles peut se faire suivant plusieurs stratégies :

- ◆ **Adaptation par substitution** : Elle consiste à substituer les valeurs appropriées à la nouvelle situation par les valeurs dans l'ancienne solution. Quatre méthodes de substitution sont distinguées par Kolodner (1993) :
 - *Le remplacement*. Il consiste à remplacer dans une ancienne solution des objets par des nouveaux.
 - *L'ajustement de paramètres*. Il consiste à ajuster les valeurs numériques de paramètres d'une solution.
 - *La recherche locale*. Elle consiste à substituer une partie de l'ancienne solution par un nouvel objet.
 - *L'interrogation de la mémoire des cas*. C'est la recherche d'une nouvelle structure pour trouver un élément correspondant à une spécification précise.

- ◆ **Adaptation par transformation** : Elle consiste à ajouter, supprimer ou remplacer un attribut. Ceci peut être réalisé de deux façons [Kolodner, 1993] :
 - *transformation à base d'heuristiques*. Elle consiste à employer des heuristiques pour ajouter, supprimer ou remplacer des parties d'une solution.
 - *adaptation basée sur un modèle*. Un modèle causal permet de transformer une ancienne solution en l'adaptant à la nouvelle.

- ◆ **Adaptation par dérivation** : C'est une séquence d'inférences qui est appliquée au nouveau cas afin de dériver une nouvelle valeur pour un attribut.

2. L'adaptation à base de cas

Cette approche présente une base d'apprentissage pour améliorer les aptitudes d'adaptation au fil des expériences. Il est naturel d'appliquer le RBC au processus d'adaptation de cas lui-même, en remplaçant des règles d'adaptation prédéfinies par des adaptations de cas qui reflètent les expériences d'adaptation passées [Leake, 1994]. En effet, une trace de ce processus peut être stockée comme une adaptation de cas et utilisée ultérieurement dans un RBC relatif au processus d'adaptation lui-même. Cela reviendrait donc à faire du raisonnement à base de cas récursif.

Le processus d'adaptation est décomposé en deux parties :

1. la première est la sélection d'une transformation qui a été mémorisée et pour laquelle on a créé un index construit sur le type de problème.
2. la seconde est la recherche d'information nécessaire à l'application de ces transformations.

3. L'adaptation mixte

Cette approche combine les deux approches précédentes [Leake *et al.*, 1996]. Son objectif est de rendre les systèmes RBC capables d'assurer la transition de l'adaptation guidée par des règles vers l'adaptation guidée par un raisonnement à base de cas.

Le processus d'adaptation suit les étapes suivantes :

1. *Adaptation à base de cas* : On essaye d'abord de retrouver un cas d'adaptation qui a bien fonctionné dans le passé pour un problème similaire. Si un tel cas existe, il sera appliqué de nouveau et le contrôle passe à l'étape 3.
2. *Adaptation à base de règles* : Si aucun cas passé n'est trouvé, on sélectionne une règle de transformation associée au type de problème à traiter. On essaye, par la suite de remplir les conditions requises afin que la règle de transformation soit applicable.
3. *Évaluation* : On évalue ensuite la solution adaptée en vérifiant la compatibilité entre ses contraintes et celles de la solution de départ.
4. *Mémorisation* : Quand l'adaptation a réussi, la solution ainsi que le cas d'adaptation sont mémorisés pour une utilisation future.

3.6.5 Révision et Recyclage

Dans ces étapes, la solution proposée est testée de façon à vérifier si elle convient bien :

- si le test est satisfaisant :
 1. la solution est fournie à l'utilisateur;
 2. elle est également rangée dans la mémoire de cas.
- si le test est négatif :

1. il faut trouver la cause de l'échec;
2. la supprimer et tester la nouvelle solution.

Ceci nécessite un ensemble de connaissances permettant de mener à bien les modifications adéquates.

3.7 Les applications du RBC

Le RBC est une technique qui est couronnée de succès dans plusieurs domaines d'application. Les domaines qui bénéficient du RBC sont : planification, conception, analyse financière, maintenance technique, gestion du risque, contrôle de processus, contrôle de la qualité, contrôle du trafic téléphonique, classification d'objets, interprétation de photographies, génération d'explications, compréhension du langage naturel, apprentissage, jeu, etc. Dans la sous-section suivante, nous explorerons certaines de ces utilisations pour le RBC.

3.7.1 Les principaux domaines d'utilisation

Les principaux domaines d'utilisation du raisonnement à base de cas sont :

1. La résolution de problèmes

La résolution de problèmes est le plus souvent présente, même si ce n'est qu'implicitement, dans les systèmes utilisant le raisonnement à base de cas. En effet, mettre à profit une expérience passée pour résoudre un nouveau problème est une démarche très courante. Le tutoriel qui se propose d'enseigner comment résoudre les problèmes est constitué d'une base de connaissances et d'une base de faits qui contient les problèmes à résoudre. Le module de résolution aide l'apprenant dans son propre raisonnement en le suivant de près dans ses activités. Une des fonctions essentielles assignées à un tutoriel au cours de la résolution d'un problème, est donc de pouvoir aider, guider ou suivre l'apprenant en s'adaptant à lui. Le RBC a été appliqué à une grande variété de tâches de résolution des problèmes, y compris la planification, le diagnostic et la conception. Plusieurs systèmes ont été conçus pour accomplir ces tâches.

Les systèmes PROTOS [Bareiss, 1989] et CASEY [Koton, 1988] ont été par exemple, construits pour faire le diagnostic, le système CHEF [Hammond, 1986] [Hammond, 1989] pour la planification, CLAVIER [Hennessy et Hinkle, 1992], CADET [Sycara, 1992] et JULIA [Hinrichs et Kolodner, 1991] ont été construits pour faire la conception basée sur des cas.

2. L'interprétation de problèmes

Cette approche consiste à expliquer un phénomène en se rappelant un phénomène similaire en empruntant ses explications et en les adaptant. La méthode raisonne à partir d'expériences antérieures en tentant de chercher des éléments d'explication pour les suggérer à la situation nouvelle. Cette approche est très utilisée en droit et chez l'humain en général (exemple : le système HYPO [Ashley, 1991]). Elle est également utilisée pour des tâches telles que classifier une nouvelle situation dans le contexte. PROTOS, par exemple fait la classification en essayant de trouver le cas le plus assorti dans sa base de cas.

3. L'apprentissage

Le raisonnement à base de cas joue un rôle important dans l'apprentissage automatique des connaissances. En apprentissage, l'acquisition de nouveaux faits ayant une forte similitude avec des connaissances existantes permet de s'adapter à de nouvelles situations grâce à des situations déjà rencontrées.

En fait, de nombreux travaux relatifs au raisonnement à base de cas comportent un volet sur l'apprentissage. Il est à noter que le RBC est un cas particulier de l'apprentissage par analogie. Les systèmes CELIA [Redmond, 1991], CHEF [Hammond, 1989], HYPO [Ashley, 1991] et DECIDER [Farrel, 1987] utilisent cette approche.

Dans la section suivante, nous présentons quelques systèmes basés sur le raisonnement à base de cas.

3.7.2 Récapitulation de quelques systèmes de RBC existants

Plusieurs systèmes utilisant le RBC ont été développés, le Tableau 3.2 donne certains systèmes représentatifs de RBC et leurs domaines d'application.

Nom	Thème	Domaine	Références
CYRUS	diagnostic	système de recherche d'information	[Kolodner, 1983a] [Kolodner, 1983b]
MEDIATOR	planification	médiation	[Simpson, 1985]
CHEF	planification	recettes culinaires	[Hammond, 1986]
DECIDER	apprentissage	sécurité nationale	[Farrel, 1987]
CASEY	diagnostic	médecine	[Koton, 1988]
MEDIC	planification	diagnostic d'infections pulmonaires	[Turner, 1988]
PERSUADER	planification	résolution de conflits de travail	[Sycara, 1988]
PROTOS	diagnostic	audiologie	[Bareiss, 1989]
CELIA	diagnostic et apprentissage	diagnostic de voiture	[Redmond, 1991]
JULIA	conception	repas culinaires	[Hinrichs et Kolodner, 1991]
HYPO	interprétation de problèmes	droit	[Ashley, 1991]
ARCHIE	conception	immeubles de bureaux	[Pearce <i>et al.</i> , 1992]
CADET	conception	systèmes mécaniques	[Sycara, 1992]
CLAVIER	conception	la charge d'un autoclave	[Hennessy et Hinkle, 1992]
SQUAD	consultation	contrôle de qualité de logiciels	[Kitano <i>et al.</i> , 1992]
CASCADE	diagnostic	informatique	[Simoudis, 1992]
CADRE	conception	immeubles de bureaux	[Dave <i>et al.</i> , 1994]
Ocram-CBR	apprentissage	système de recherche d'informations	[Papagni <i>et al.</i> , 1997]

Tableau 3. 2 : Exemple de systèmes à base de cas.

3.8 L'intégration du RBC aux STI

Le raisonnement par cas est un type de raisonnement par analogie qui a donné lieu au plus grand nombre de développements pratiques, essentiellement dans le champ des

systèmes experts. Quelques exemples de tels systèmes ont été présentés dans la section précédente. Après des progrès considérables sur l'application du RBC afin de réaliser les systèmes experts, il y eut un nouvel engouement pour développer des STI en utilisant le RBC, par exemple, on peut mentionner ici les travaux de Gupta *et al.* (1994), McCalla *et al.* (1996), Elorriaga *et al.* (1995), Schank (1996), Fuji *et al.* (1996), et Zhang et Alem (1996). Voici quelques exemples des systèmes tutoriels intelligents qui utilisent le RBC dans ses approches.

3.8.1 Le système TECHDOC-I

TECHDOC-I [Peter et Rösner, 1994] est un système d'aide intelligent qui apporte du support à l'utilisateur dans le domaine de l'entretien de voitures. C'est un système interactif, facile à utiliser et ses conseils sont compréhensibles par l'usager. Le système génère des instructions et conseils en langage naturel. Il est basé sur des connaissances applicables à plusieurs domaines, c'est un système d'aide générique. Dans ce système, les activités sont représentées sous forme de tâches, qui sont subdivisées en un ensemble d'étapes ou actions. C'est donc une structure hiérarchique. À chaque tâche, on associe un niveau de difficulté et un niveau de connaissances appelé degré de familiarité. Le degré de similarité d'une tâche par rapport à une deuxième tâche est déterminé en calculant le rapport entre le nombre d'actions communes à deux tâches et la longueur de la tâche courante, où la longueur d'une tâche est le nombre d'actions qui constituent cette tâche.

Le modèle de l'apprenant dans TECHDOC-I contient aussi deux composantes : *l'historique de l'apprenant et l'administrateur du modèle de l'apprenant.*

L'historique de l'apprenant contient les informations sur : les tâches et les actions que l'usager a exécutées, les outils et les instruments techniques qu'il a utilisés, et le nombre de fois où ces outils ont été utilisés et dans quel contexte. L'administrateur s'occupe de mettre à jour le modèle de l'apprenant. Le système ne peut suivre ce que l'usager a fait, il suppose que l'apprenant a bien exécuté la tâche et par conséquent son niveau de connaissances est incrémental.

3.8.2 Le système Sarah

Sarah [Blin et Quiniou, 1996] est un système pour la modélisation de l'utilisateur à l'aide de techniques d'apprentissage automatique. Le système diagnostique les déficits des usagers afin de construire les modèles de l'utilisateur. Ce système est destiné à aider un patient aphasique (usager), au moyen d'exercices pertinents, à apprendre la programmation logique, et l'aide à compenser ses déficits en lui faisant prendre conscience de ses erreurs.

Sarah s'intéresse particulièrement à la programmation logique inductive (PLI) qui permet d'apprendre un programme logique à partir d'exemples de couples d'entrées/sorties de ce programme.

Les connaissances dans ce système sont sous forme de cas, et proviennent du programme d'apprentissage et non d'un expert humain. Ceci découle de l'expérience passée de l'apprenant sur le type de ses recherches dans l'espace de clauses. Le système sera alors, conduit à effectuer souvent, le même type de recherche dans l'espace de clauses.

3.8.3 Le système Ocram-CBR

Ocram-CBR [Papagni *et al.*, 1997] est un système pour les applications de formation et d'éducation. Ce système a été adapté pour la formation des cours en « systèmes d'information » et pour une deuxième application qui enseigne aux étudiants comment écrire des lettres commerciales de façon plus pertinente. L'architecture du système se compose du modèle de l'apprenant, du module tuteur, et de l'interface usager. Le sous-système modèle de l'apprenant utilise l'approche RBC pour réaliser le modèle d'étudiant qui est une représentation des objectifs et des caractéristiques des utilisateurs. Le modèle de l'apprenant est établi dans ce sous-système par un dialogue en utilisant une approche basée sur les 'stéréotypes'. Des 'stéréotypes' peuvent être visualisés comme descriptions d'un utilisateur représentatif.

Le sous-système tuteur planifie les leçons à enseigner en utilisant une bibliothèque de cas classés et successivement recherchés au moyen de *recherche par discrimination* et *recherche séquentielle*. Il permet à des étudiants de voir une série de diapositives

correspondant à leurs besoins spécifiques, déterminée pour chacun selon son modèle, son histoire, et les choix spécifiques qu'il fait pendant durant le cours.

3.9 Comparaison du RBC avec d'autres formes de raisonnement

Le RBC est connu pour être supérieur au raisonnement basé sur les règles et au raisonnement basé sur les modèles. Nous allons comparer dans cette section le RBC à trois formes de raisonnement.

3.9.1 Le RBC et le raisonnement à base de modèles (Model Based Reasoning (MBR))

Le RBC et le raisonnement à base de modèle (Model-Based Reasoning (MBR)) ont été développés pour éviter de raisonner à partir de zéro. La définition du raisonnement à base de modèles est comme suit.

Définition : Le raisonnement à partir de modèle est un raisonnement utilisant un modèle d'un système réel représenté au moyen d'une formalisation appropriée, de telle sorte que le comportement du modèle est similaire à celui du système sous les conditions attendues. Le raisonnement à partir de modèle doit donc reproduire le comportement du système qu'il représente [Bichindaritz, 1994].

Les différences fondamentales qui existent entre RBC et MBR sont :

- Le MBR est applicable quand un modèle causal existe.
- Un MBR utilise un modèle causal, alors que dans le RBC, les cas en mémoire décrivent comment les choses fonctionnent.
- Le MBR permet de vérifier les solutions, mais ne génère pas la solution; le RBC permet de générer des solutions.

3.9.2 Le RBC et le raisonnement à base de règles (Rule Based Reasoning (RBR))

Au lieu d'utiliser et d'interpréter à chaque fois un ensemble de règles pour trouver une solution, il est plus naturel de mémoriser les problèmes (avec leurs solutions, et (avec) éventuellement des informations sur les difficultés rencontrées ainsi que comment elles ont été résolues), et utiliser ces solutions pour de nouvelles situations.

Les différences fondamentales qui existent entre ces deux types de raisonnement sont :

- Dans le raisonnement à base de règles les connaissances de l'expert doivent être codées sous forme de règles, mais dans le raisonnement à base de cas, la plupart des connaissances sont sous forme de cas.
- Dans le RBR l'accent est mis sur le mécanisme de raisonnement, alors que dans le RBC l'intérêt majeur porte sur le contenu des cas.
- Le RBR explique son raisonnement par la chaîne de règles actives, alors que le RBC justifie son raisonnement en montrant le ou les cas similaires à partir desquels la solution a été dérivée.

3.9.3 Le RBC et le raisonnement par analogie

Nous avons déjà souligné que le raisonnement à base de cas est un cas particulier de raisonnement par analogie dont la source et la cible appartiennent à un même domaine.

Une description possible du rapport entre RBC et analogie est [Leake, 1996] :

$$\text{RBC} = \text{Recherche} + \text{Analogie} + \text{Adaptation} + \text{Apprentissage}.$$

Les différences fondamentales qui existent entre ces deux types de raisonnement sont :

- Le raisonnement par analogie s'intéresse aux inférences réalisées, mais le RBC s'intéresse généralement à la mémoire.
- Dans le raisonnement par analogie, la mémoire n'est pas structurée tandis que dans le RBC, elle est structurée.

Le RBC résout donc des difficultés du raisonnement par analogie. Il complète ce mode de raisonnement par un mécanisme de mémorisation et d'extraction d'expériences.

3.10 Avantages et Désavantages de RBC

Le raisonnement à base de cas fournit beaucoup d'avantages pour un raisonneur. Ce type de raisonnement permet au système de proposer la solution aux problèmes rapidement, évitant le temps nécessaire pour dériver ces réponses à partir de zéro. Il permet également de proposer des solutions dans les domaines qui ne sont pas complètement compris par le système.

Le raisonnement par cas est bien adapté aux domaines où il n'existe ni théorie ni modèle formalisé, et où le rôle de l'expérience est prédominant. Il donne aussi à un système des moyens d'évaluer des solutions quand aucune méthode algorithmique n'est disponible pour l'évaluation. Les cas sont utiles en interprétant des concepts ouverts et mal définis.

Les cas aident un système à concentrer son raisonnement sur les parties importantes d'un problème en précisant quels indices sont les plus importants.

Évidemment, il y a également des inconvénients en employant des cas pour raisonner :

- Un système pourrait être tenté d'utiliser de vieux cas aveuglément, comptant sur une expérience précédente sans la valider dans la nouvelle situation.
- Un système pourrait permettre à des anciens cas de biaiser la résolution de nouveaux problèmes.
- La recherche du meilleur cas n'est pas toujours évidente et l'utilisation de cas inappropriés peut engendrer des erreurs coûteuses.

3.11 Conclusion

Dans ce chapitre nous avons présenté le raisonnement à base de cas. Nous avons remarqué que la problématique du raisonnement à base de cas est double : d'une part, il est une méthodologie d'intelligence artificielle qui partage les problématiques du

raisonnement par analogie, et d'autre part, il propose un modèle cognitif des processus de raisonnement et d'apprentissage qui partage la problématique des mémoires dynamiques. Nous avons également décrit certains domaines d'applications. Plusieurs auteurs ont montré les avantages et les supériorités du raisonnement à base de cas par rapport aux autres modes de raisonnement.

Notre objectif est de développer et réaliser un système pour modéliser les connaissances et les raisonnements de l'apprenant, à l'aide du raisonnement à base de cas. Dans le prochain chapitre, nous allons présenter les principes de notre approche, et montrer comment il intègre des interactions avec les apprenants et réalise leur modélisation.

Chapitre 4

Modélisation de l'Apprenant par l'Analyse du Raisonnement : principes et conception

4.1 Introduction

Dans le chapitre 2, nous avons vu que le rôle du modèle de l'apprenant est central au STI, car ce composant est la clé de la formation individualisée dans l'interaction avec un étudiant. Nous avons aussi remarqué que ceci est une tâche complexe et difficile, mais importante. Cette dernière est réalisée par des méthodes différentes que nous avons exposées au chapitre 2.

Dans ce chapitre, nous présentons nos moyens et les principes que nous avons retenus pour traiter ce problème. Nous comptons appliquer le RBC pour réaliser cette tâche. Ce type d'application n'a pas (à notre connaissance) été exploité par les systèmes tutoriels intelligents.

Pour cela, nous étudions d'abord les caractéristiques du raisonnement et son rôle dans la modélisation des apprenants. Nous mettons ensuite, l'accent sur la présentation des principes sur lesquels notre modèle est fondé. Nous détaillons également la fonctionnalité de ce modèle en l'illustrant à l'aide d'exemples concrets. Cette fonctionnalité nous a conduit plus particulièrement à concevoir une architecture de modèle d'apprenant. Nous décrivons finalement la modélisation des connaissances et raisonnements de l'apprenant à l'aide de notre approche (voir chapitre 6 pour une description plus détaillée).

4.2 *Intégration du raisonnement dans les STI*

Le mot 'raisonnement' évoque habituellement une activité élaborée, que l'on retrouve par exemple dans les travaux des logiciens et des mathématiciens. En même temps, elle est couramment présente dans la vie de tous les jours et constitue la base de la plupart des activités mentales de l'être humain [Oléron, 1977].

Nous allons dans la suite de ce paragraphe donner quelques définitions du raisonnement.

4.2.1 Quelques définitions du raisonnement

Voici la définition du raisonnement du point de vue de certains auteurs :

Définition selon Kodratoff et Barès (1991) : En intelligence artificielle, on utilise le raisonnement comme un synonyme de l'ensemble d'inférences.

Définition selon Charles (1995) : En intelligence artificielle, on cherche à établir une nette distinction entre connaissance et raisonnement, même si ce dernier est, en dernier ressort, une forme de connaissances. Cependant, on peut considérer que le savoir comporte deux pôles :

1. Le savoir spécifique à un domaine (connaissance);
2. Le savoir sur l'utilisation des savoirs (raisonnement).

Définition selon Le Robert méthodique (Dictionnaire méthodique du français actuel) :

- *Le raisonnement* :
 1. l'activité de la raison, la manière dont elle s'exerce.
 2. le fait de raisonner en vue de parvenir à une conclusion.
- *Raison* : la faculté qui permet à l'homme de connaître, juger et agir conformément à des principes.
- *Raisonner* : faire usage de sa raison pour former des idées, des jugements.

Définition selon Haton (1991) : « Un raisonnement peut être défini comme un enchaînement d'énoncés ou de représentations symboliques conduit en fonction d'un but, ce but pouvant prendre des formes variées : démontrer, convaincre, élucider, interpréter, décider, justifier, expliquer, etc. Les énoncés relèvent eux-mêmes de divers formalismes : soit un symbolisme strictement défini, soit le plus souvent, des formes plus éclectiques : énoncés verbaux surtout, mais aussi informations sensorielles perçues directement, images mentales, souvenirs, expériences personnelles, hypothèses, etc. Un tel enchaînement est une caractéristique importante d'un raisonnement. Il est en général non linéaire et nécessite des retours arrières, présentés dans le quasi-totalité des systèmes d'IA tout comme dans la démarche humaine. Le raisonnement est une activité intentionnelle que l'on peut dans une certaine mesure opposer aux activités plus ou moins passives telles que la perception, l'association d'idées, etc. Il est néanmoins difficile de procéder à une partition stricte des diverses opérations mentales menées par un être humain ».

Définition psychologique : Le raisonnement se présente comme un enchaînement, une combinaison ou une confrontation d'énoncés ou de représentations, respectant des contraintes internes susceptibles d'être explicitées et conduits en fonction d'un but [Oléron, 1977].

4.2.2 Caractéristiques du raisonnement

L'analyse des définitions proposées nous amène à dégager quelques caractéristiques du raisonnement :

- le raisonnement est un enchaînement de jugements (la plupart des définitions s'accordent sur ce point) ;
- il est une activité intentionnelle, dans le raisonnement, le sujet poursuit un certain but et il combine les moyens en vue de ce but, comme par exemple :
 1. démontrer, convaincre, justifier une théorie, une résolution (d'un problème), une thèse, une interprétation, etc.;
 2. découvrir ou reconstituer un fait;
- il apparaît comme une opération complexe, par les éléments qu'il met en œuvre;
- il est une position et une conscience de raisons;

- le raisonnement lui même est un savoir (le savoir sur l'utilisation des savoirs).

Nous utiliserons ces caractéristiques pour justifier certains aspects de notre travail.

4.2.3 Rôle du raisonnement dans la modélisation de l'apprenant

Raisonner nécessite le plus souvent de parcourir un ensemble de solutions possibles, à l'aide des méthodes classiques de résolution de problèmes [Simon, 1983].

L'analyse du raisonnement d'un apprenant nous permet d'atteindre les objectifs suivants :

- mesurer l'état des connaissances de l'apprenant sur le sujet proposé;
- mesurer la capacité de l'apprenant à construire son raisonnement (en utilisant ses connaissances);
- produire des explications adaptées aux caractéristiques de l'apprenant afin de supporter différentes stratégies pédagogiques et différents modes d'interaction adaptables à cet apprenant.

4.3 Types de raisonnement en IA

Il existe une grande diversité de modes de raisonnement, Haton *et al.* (1991) mentionnent ceux que l'on rencontre le plus souvent :

- *Le raisonnement formel*, fondé sur la manipulation syntaxique de structures symboliques à l'aide de règles, dans le cadre d'une certaine sémantique. Le raisonnement logique est un exemple.
- *Le raisonnement procédural*, dans lequel toutes les connaissances, la façon de les utiliser et la conduite du raisonnement sont entièrement figées sous forme d'algorithmes ou d'automates finis.
- *Le raisonnement par analogie*, très naturel et efficace chez l'être humain car une caractéristique de l'intelligence humaine est sa capacité de repérer les analogies entre des situations différentes. On utilise souvent cette capacité pour prendre des décisions. Le raisonnement à base de cas est un cas particulier du raisonnement par analogie.
- *Le raisonnement par généralisation et abstraction*, largement répandu chez l'homme, mais encore assez mal connu. Il est lié à l'apprentissage par induction.

Parmi les modes de raisonnement de l'étudiant que nous détectons dans notre travail, nous nous intéressons plus particulièrement aux raisonnements *inductif* et *déductif* (types de raisonnement logique) car ils sont faciles à comprendre et applicables dans diverses situations d'apprentissage comme les stratégies cognitives :

1. *Raisonnement déductif* permettant de déduire des conclusions variées à partir d'un ensemble de prémisses (application de connaissances générales à un cas particulier).
2. *Raisonnement inductif* visant, à l'inverse du précédent, à généraliser des prémisses.

Nous examinons par la suite le rôle du raisonnement à base de cas dans la modélisation de l'apprenant dans un STI.

4.4 Modélisation de l'apprenant par raisonnement à base des cas : philosophie générale

4.4.1 Le rôle de la remémoration dans l'apprentissage

La théorie de la mémoire dynamique de Schank (1982) propose un modèle de mémoire douée d'apprentissage. Schank et Abelson (1977) reconnaissent que, lorsque des informations nouvelles sont traitées, elles sont comprises lorsque les expériences précédemment rencontrées sont remémorées. Ceci est possible parce que les structures de traitement d'une nouvelle expérience sont les mêmes que les structures utilisées pour conserver les informations en mémoire. Plus précisément, le but recherché par l'apprentissage est la mise-en-place de structures en mémoire les plus générales possibles pour faciliter la recherche en mémoire et pour la reconnaissance de données en entrée.

Le but de la remémoration est donc de rendre possible la modification automatique des structures de la mémoire, la rendant indissociable de la compréhension et de l'apprentissage. En fait, la remémoration permet de concevoir plusieurs modèles d'apprentissage dans la mémoire.

La Figure 4.1 résume les enchaînements des processus menant à l'apprentissage. Dans cette figure, les liens qui unissent remémoration, compréhension et apprentissage sont montrés. L'apprentissage intervient alors lorsque les prédictions générées suite à une remémoration

sont mises en échec, ce qui revient également à un échec de la compréhension recherchée. Les échecs des prédictions générées par les structures en mémoire doivent être interprétés, pour pouvoir apprendre à partir d'eux. L'explication des échecs est donc proposée comme le processus nécessaire à l'apprentissage. Ce processus, comme celui de compréhension, suit le même cycle de traitement (c.-à-d., par remémoration d'explications précédentes) pour aboutir à la compréhension de l'échec. D'autre part, lorsque la compréhension est un succès, l'apprentissage a également lieu à l'explication de la réussite des prédictions des informations remémorées. La génération d'explications est rendue possible par les mêmes structures que pour la compréhension [Bichindaritz, 1994].

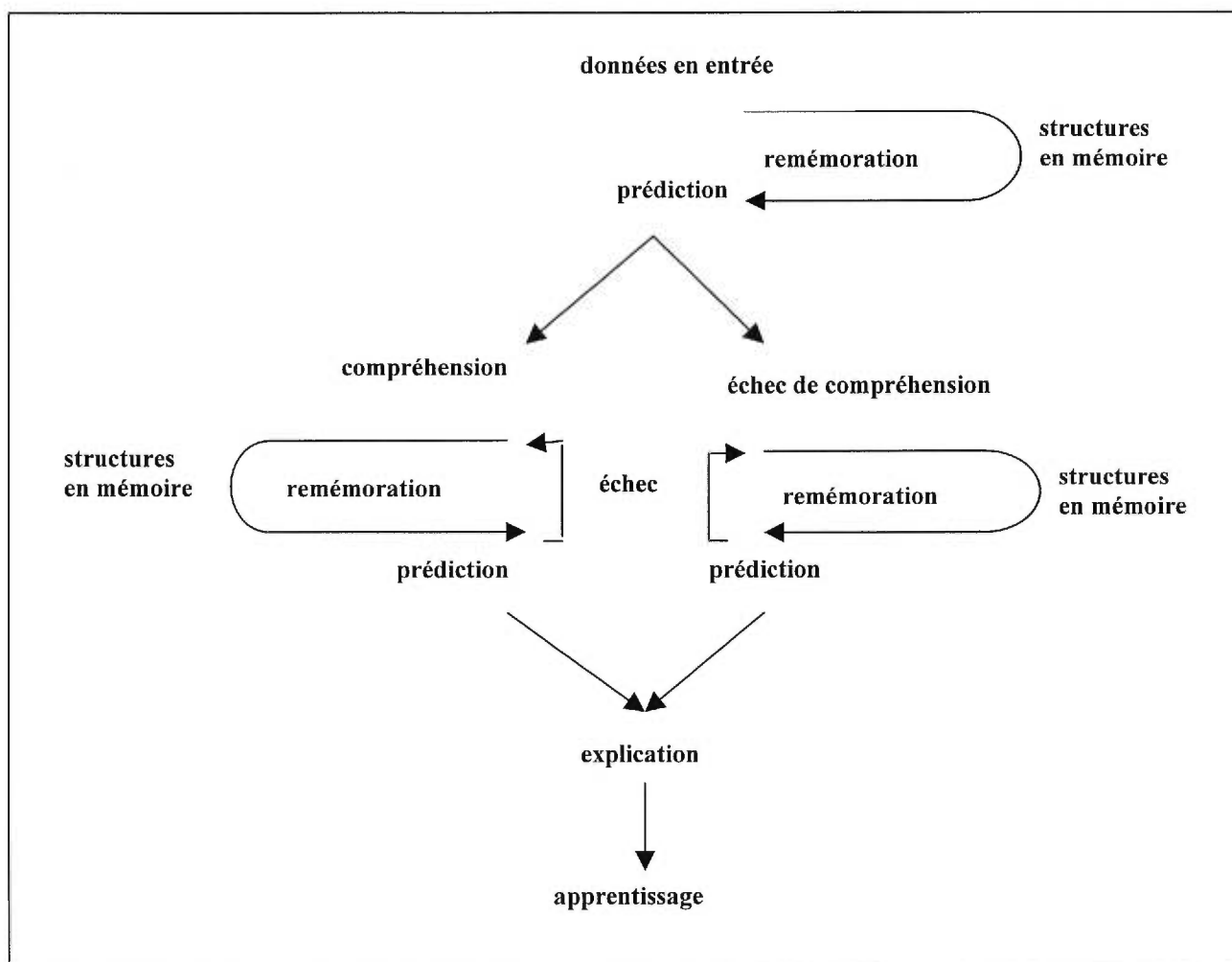


Figure 4. 1 : Relation entre remémoration et apprentissage [Bichindaritz, 1994].

CYRUS (Computerized Yale Retrieval and Updating System) [Kolodner, 1983a] est le premier système qui implante la théorie de la mémoire dynamique. C'est un système qui construit une mémoire à partir de données textuelles dans le but de répondre aux questions. C'est donc un système de recherche d'informations (ou IRS, Information Retrieval System). Il met en forme les informations en mémoire pour répondre à des questions du type « Quand as-tu rencontré Begin pour la dernière fois ? », « As-tu jamais rencontré la femme de Begin ? ». CYRUS analyse les questions et les décompose en sous-questions si nécessaire jusqu'à ce qu'une réponse soit retirée de la mémoire ou qu'aucune élaboration ne soit possible.

4.4.2 Philosophie d'expériences progressives

Nous savons par expérience pédagogique que beaucoup d'élèves éprouvent de grosses difficultés pour résoudre un problème; mais après des exercices d'entraînement multiples, une certaine compétence en résolution de problèmes est acquise progressivement.

Diverses recherches dans la pédagogie démontrent que l'humain exploite une expérience antérieure pour résoudre les problèmes semblables; par exemple la théorie de l'apprentissage de Bandura (1971) précise que le système humain est un système qui traite les informations selon ses processus de motivation. Dans cette théorie il a énoncé que l'on apprend à résoudre des problèmes en imitant un modèle.

Selon Polya (1957) résoudre des problèmes est une habileté pratique, tout comme nager. Selon lui, nous acquérons cette habileté par imitation et pratique : « En essayant de résoudre des problèmes, vous devez observer et imiter ce que d'autres font lorsqu'ils résolvent des problèmes » [Polya, 1957].

L'analogie, selon Polya, c'est une sorte de ressemblance, de similarité. Des objets similaires ont en commun certains aspects. Des objets analogues ont en commun certaines relations entre leurs parties respectives. Toutes les sortes d'analogies peuvent jouer un rôle dans la découverte d'une solution.

De même Ross (1989) a prouvé que les gens apprenant une nouvelle compétence renvoient souvent aux problèmes précédents pour régénérer leurs mémoires sur la façon dont ils font la tâche.

En résumé, nous avons du succès à résoudre des problèmes non pas par l'usage de la logique, mais en faisant correspondre le problème que nous désirons résoudre à des problèmes auxquels nous sommes très bons.

4.4.3 Résolution des problèmes et modélisation des apprenants

L'apprentissage est un ensemble d'activités qui permet à l'apprenant d'apprendre de manière efficace ce qu'on lui enseigne.

Selon Gagné *et al.* (1992), l'apprentissage est le résultat de l'interaction entre l'apprenant et son environnement. Ils ont cité neuf événements d'enseignement qui peuvent se produire dans un processus d'apprentissage [Tableau 4.1].

- | |
|---|
| <ol style="list-style-type: none">1. Attirer l'attention2. Informer l'apprenant des objectifs3. Rappeler les prérequis4. Présenter le matériel pédagogique5. Fournir les guides d'apprentissage6. Provoquer la performance7. Donner le « feedback »8. Évaluer la performance9. Accroître la rétention et le transfert |
|---|

Tableau 4. 1 : Les événements d'enseignement

Pour réaliser ces événements, un STI utilise des ressources de types différents. Il utilise selon le produit d'apprentissage des ressources didactiques différentes (par exemple, démonstration, présentation, vidéo, etc., voir 1.8) pour les événements 4 et 5 et les ressources didactiques du type exercices, problèmes, questions, etc. (ressources d'évaluation) pour les événements 6, 7 et 8.

La résolution de problèmes est donc un contexte idéal pour la modélisation de l'apprenant, dans le sens où elle permet à l'apprenant d'établir de nombreux liens entre ses connaissances. Le transfert des apprentissages exige que l'élève ait, dans sa mémoire

à long terme, c'est-à-dire dans sa base, les connaissances spécifiques nécessaires pour le problème en question. En effet, pour résoudre un problème, l'apprenant doit faire appel à toute connaissance déjà acquise pouvant l'aider dans sa démarche.

Nous pouvons récapituler cette section selon les points suivants :

1. la description des processus permettant le raisonnement est complétée par une description détaillée des structures de représentation des connaissances et d'organisation de la mémoire, désormais dirigée vers l'apprentissage;
2. la résolution de problème est un contexte idéal pour la modélisation de l'apprenant, dans le sens où elle permet à l'apprenant d'établir de nombreux liens entre ses connaissances;
3. la majeure partie de résolution des problèmes humains a lieu dans les espaces de problèmes qui sont bien connus ou changent légèrement des situations familières. Il est rare qu'une personne rencontre un problème qui ne lui rappelle pas les solutions potentiellement applicables aux problèmes semblables résolus ou observés dans l'expérience antérieure.

4.5 Conception de notre approche

Dans le but de concevoir notre modèle, nous présentons ici, d'abord les principes généraux, nous illustrons ensuite notre démarche par deux exemples simples.

4.5.1 Principes de conception et présentation générale

Les caractéristiques principales du style d'interaction de notre approche peuvent se résumer comme suit :

- Les problèmes et leurs solutions correspondantes (c'est-à-dire solutions des experts); ainsi que les cas et leurs solutions seront préétablis dans le système; un problème est un cas que l'on cherche à résoudre.
- Le système propose un problème à l'apprenant.

- L'apprenant résoudra le problème en adaptant la solution d'un (ou des) cas similaire(s) qu'il a déjà résolu(s) (et qui sont déjà contenus dans la base de cas); il proposera leurs solutions au système.
- Le système analysera les réponses de l'apprenant pour comprendre son raisonnement.
- Les résultats de cette analyse peuvent être utilisés de diverses manières :
 - pour vérifier la cohérence du travail de l'apprenant;
 - pour vérifier que la solution qu'il propose est correcte;
 - pour modéliser les connaissances et les raisonnements de l'apprenant.
 - pour modéliser les erreurs et les incompréhensions de l'apprenant.

La Figure 4.2 présente le déroulement du raisonnement de l'apprenant et de l'expert dans notre approche. La partie (a) de cette figure concerne le raisonnement proposé par l'apprenant pour un problème donné (P) : en effectuant une recherche de similarité entre P et les autres cas il trouve Cas-5 (C5), ensuite en adaptant la solution de ce cas (CS5), il extrait la solution SA.

Reliée à la phase précédente, la partie (b) de cette figure indique les raisonnements de l'expert pour le problème P : en effectuant une recherche de similarité entre P et les autres cas, l'expert a trouvé les Cas-1 (C1), Cas-4 (C4) et Cas-5 (C5), puis en adaptant les solutions de ces cas (CS1, CS4, CS5), il extrait les solutions SE1, SE4 et SE5.

Maintenant, la comparaison entre la solution de l'apprenant et les solutions de l'expert nous permet de connaître le raisonnement de l'apprenant.

Remarques

- chaque cas, peut avoir plusieurs solutions;
- à chaque solution, on fait correspondre le mode de raisonnement associé (par exemple : *inductif* ou *déductif*)
- l'expert peut avoir plusieurs solutions possibles;
- seule la base de cas est accessible par l'apprenant;
- la fonction de similarité utilisée par l'apprenant peut être différente des fonctions de similarité utilisées par l'expert.

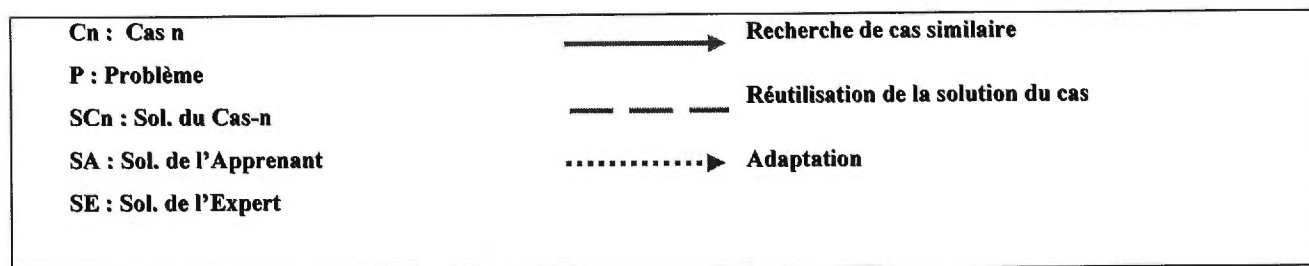
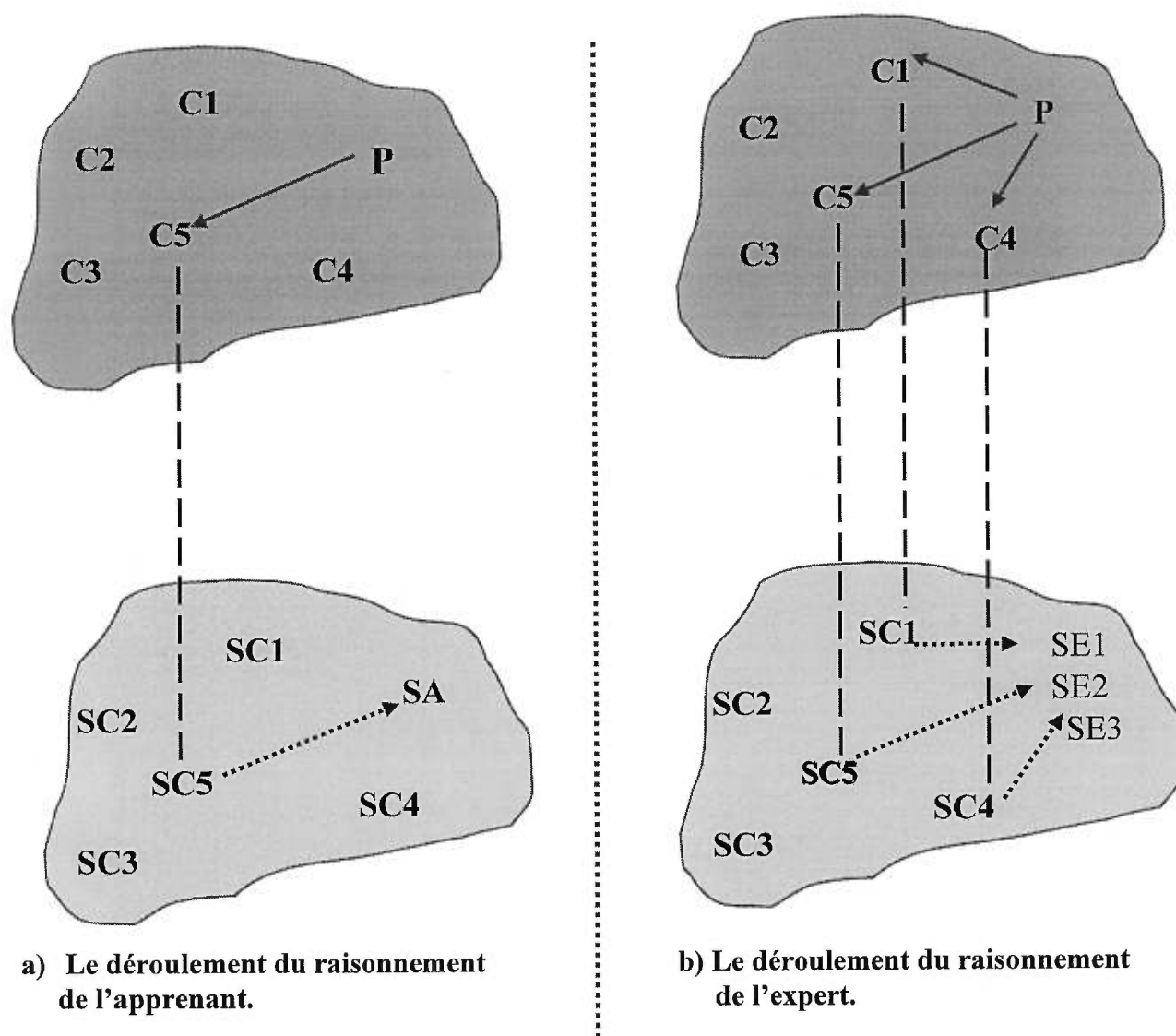


Figure 4. 2 : Le déroulement du raisonnement de l'apprenant et de l'expert dans notre approche.

4.5.2 Exemples

Nous illustrons maintenant notre démarche en prenant deux exemples simples, le premier dans le domaine du Calcul Intégral et le deuxième dans le domaine de la Théorie des Nombres.

4.5.2.1 Exemple 1

❖ **Problème 1** (soumis à l'apprenant) : Déterminer la fonction $Y = \int ((\ln x)^2 / x) dx$.

❖ **Cas 1** (disponible dans la base de cas) : Déterminer la fonction $Y = \int T^n dT$.

• **Solution (inductive) du Cas 1** : $Y = \frac{1}{n+1} T^{n+1} + C$.

❖ **Cas 2** (disponible dans la base de cas) : Déterminer la fonction $Y = \int U dV$.

• **Solution (inductive) du Cas 2** : $Y = UV - \int V dU + C$.

❖ **Solution de l'apprenant** (en utilisant la solution du Cas 1) :

$$\begin{aligned} Y &= \int ((\ln x)^2 / x) dx = \int T^2 dT + C. \quad (\text{Pour } T = \ln x, dT = \frac{1}{x} dx \text{ et } n=2) \\ &= \frac{1}{3} T^3 + C. \\ &= \frac{1}{3} (\ln x)^3 + C. \end{aligned}$$

❖ **Solutions de l'expert** :

1. **Solution 1** (en utilisant la solution du Cas 1) :

$$\begin{aligned} Y &= \int ((\ln x)^2 / x) dx \\ &= \int T^2 dT \quad (\text{Pour } T = \ln x, dT = \frac{1}{x} dx \text{ et } n=2) \\ &= \frac{1}{3} T^3 + C. \\ &= \frac{1}{3} (\ln x)^3 + C. \end{aligned}$$

2. **Solution 2** (en utilisant la solution du Cas 2) :

$$\begin{aligned} Y &= \int ((\ln x)^2 / x) dx \\ &= \int U dV \quad (\text{Pour } U = (\ln x)^2, dV = \frac{1}{x} dx) \\ &= UV - \int V dU + C. \\ &= (\ln x)^2 \cdot \ln x - 2 \int ((\ln x)^2 / x) dx + C. \\ &= \frac{1}{3} (\ln x)^3 + C. \end{aligned}$$

En conclusion, il semblerait que l'apprenant préfère l'intégration par changement de variables.

4.5.2.2 Exemple 2

❖ **Problème 2** (soumis à l'apprenant) : Soit $S_0 = a$, $S_{(n+1)} = b^* (S_n)$, ($n \geq 0$); montrez

que $S_5 = b^5 * a$.

❖ **Cas 3** (disponible dans la base de cas) : Soit $S_0 = k$, $S_{(n+1)} = (1/2) * (S_n)$, ($n \geq 0$); montrez que $S_5 = (1/2)^5 * k$.

- **Solution inductive du Cas 3 :**

$$S_1 = (1/2) * S_0 = (1/2) * k = (1/2)^1 * k;$$

$$S_{(n+1)} = (1/2) * S_n = (1/2) * ((1/2)^n) * k = (1/2)^{n+1} * k;$$

$$S_5 = (1/2)^5 * k.$$

- **Solution déductive du Cas 3 :**

$$S_5 = (1/2) * S_4 = (1/2)^2 * S_3 = (1/2)^3 * S_2 = (1/2)^4 * S_1 = (1/2)^5 * S_0 = (1/2)^5 * k.$$

❖ **Solution de l'apprenant** (en utilisant la solution inductive du Cas 3) :

$$S_1 = (b) * S_0 = (b) * a = (b)^1 * a;$$

$$S_{(n+1)} = (b) * S_n = (b) * ((b)^n) * a = (b)^{n+1} * a;$$

$$S_5 = (b)^5 * a.$$

❖ **Solutions de l'expert :**

1. **Solution 1** (en utilisant la solution inductive) :

$$S_1 = (b) * S_0 = (b) * k = (1/2)^1 * a;$$

$$S_{(n+1)} = (b) * S_n = (b) * ((b)^n) * k = (b)^{n+1} * a;$$

$$S_5 = (b)^5 * a.$$

2. **Solution 2** (en utilisant la solution déductive du Cas 3) :

$$S_5 = (b) * S_4 = (b)^2 * S_3 = (b)^3 * S_2 = (b)^4 * S_1 = (b)^5 * S_0 = (b)^5 * a.$$

En conclusion, l'apprenant semble plus à l'aise avec l'induction.

Nous désirons à l'aide de cette méthode pouvoir générer le modèle d'un apprenant en cours d'analyse de son raisonnement. L'objet des sections qui suivent est de présenter ce problème et la manière dont il est abordé dans notre approche. Nous commençons tout d'abord par identifier une architecture pour nos modèles d'apprenants.

4.6 Concevoir une architecture pour le modèle de l'apprenant

4.5.1 Les composants d'un modèle de l'apprenant

À la suite de Nkambou (1996), nous identifions aussi le modèle de l'apprenant en trois parties : partie cognitive (*modèle cognitif*), partie inférences (*modèle d'inférences*), et partie affective (*modèle affectif*). Mais selon nous, le modèle d'inférences n'est pas un

moteur d'inférences; ce modèle contient les informations sur les modes d'inférences que l'apprenant applique dans son raisonnement. Nous distinguons donc entre un modèle de l'apprenant et une partie qui fera la modélisation.

4.5.2 L'architecture du modèle de l'apprenant

Le sous-système modèle de l'apprenant sera un composant du STI qui contient les *modèles* (on parle plutôt de plusieurs instances d'un même modèle) de tous les apprenants, ainsi que les moyens pour évaluer, modéliser, et mettre à jour ces modèles (*modeleur*). Par la suite nous détaillons ces deux éléments.

4.5.2.1 *Modèle de l'apprenant*

Nous détaillons ci-dessous les trois parties de modèle de l'apprenant.

Modèle cognitif : cette partie contient des informations sur l'état des connaissances de l'étudiant par rapport à la matière considérée (par exemple, les problèmes qu'il a résolus, ses solutions, son niveau maîtrisé de la connaissance, etc., voir les chapitres 6 et 7).

Définition : Un modèle cognitif est une représentation approximative, probablement partielle, principalement qualitative de la connaissance de l'étudiant au sujet d'un domaine particulier, ou un sujet ou une compétence particulière dans ce domaine, qui peut entièrement ou partiellement expliquer des aspects spécifiques du comportement de l'étudiant.

Modèle d'inférences : Pour nous, ce modèle contient les informations sur les modes d'inférences que l'apprenant applique dans son raisonnement (par exemple les modes de raisonnement (*inductif* ou *déductif*) que l'étudiant utilise dans ses résolutions de problèmes) [Shiri, 1997].

Modèle affectif : ce modèle est un ensemble de données permettant de cerner les caractéristiques et les différentes facettes d'un étudiant. Il contient des connaissances relatives aux caractéristiques particulières permanentes ou momentanées de l'apprenant (ceci n'est pas disponible dans l'état actuel de notre modèle).

4.5.2.2 *Modeleur*

Le *Modeleur* a pour objectif de mettre à jour les connaissances de l'apprenant (gestion du modèle de l'apprenant); reconnaître ses bonnes performances; relever les connaissances erronées et suggérer des corrections. La Figure 4.3 illustre l'architecture du modèle de l'apprenant. Dans cette figure la notion « le comportement de l'apprenant » inclue ses réponses aux questions, ses actions (comme la tâche d'adaptation), et les résultats de ces actions comme les solutions obtenues, etc.

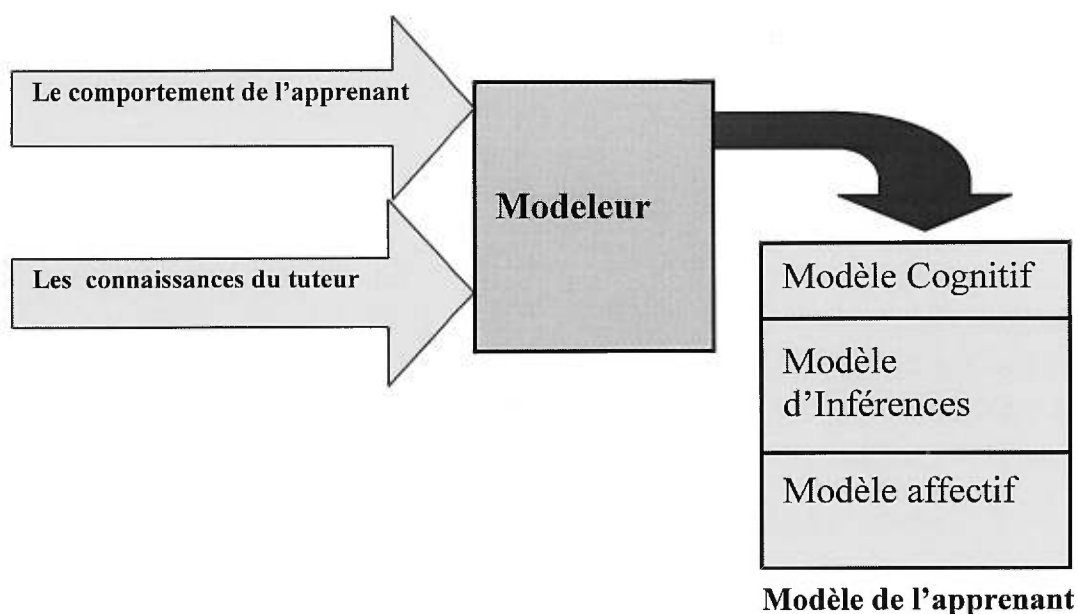


Figure 4. 3 : L'architecture du modèle de l'apprenant

Remarque 1

L'aspect important que l'on veut mettre en valeur tient au fait que contrairement à Nkambou (1996), le moteur d'inférences n'est pas dupliqué dans les différentes instances du modèle de l'apprenant; il est unique.

Remarque 2

Nous ne prenons pas en compte le modèle affectif dans notre tâche de modélisation de l'apprenant à cause de sa complication.

4.5.3 Caractéristiques d'un sous-système modèle de l'apprenant

Nous pensons que le sous-système modeleur est celui qui contient au moins les caractéristiques suivantes :

- la capacité de mettre à jour les connaissances de l'apprenant; il doit être capable de :
 - fixer l'ensemble des étapes à maîtriser par l'apprenant (par exemple; déterminer tous les problèmes résolus);
 - découvrir les étapes maîtrisées mais non exprimées par l'apprenant (par exemple, découvrir les relations entre un problème résolu et le cas qu'il a choisi à cette fin);
 - découvrir les étapes sous-jacentes non maîtrisées (lacunes de l'apprenant; par exemple déterminer tous les problèmes non résolus);
- la capacité de reconnaître les bonnes performances de l'apprenant (par exemple son niveau maîtrisé de la connaissance);
- la capacité de générer des problèmes (dans notre approche ceci se produit grâce au *Générateur des problèmes*, voir le chapitre 6);
- la capacité de relever les connaissances erronées;
- la capacité de les expliquer (par exemple, informer l'apprenant s'il a choisi un cas inadéquat pour la résolution du problème donné, etc.);
- la capacité de suggérer des corrections (par exemple montrer les relations entre le problème courant et les problèmes similaires que l'apprenant a déjà résolus, etc.).

4.5.4 Les connaissances utiles pour modéliser l'apprenant

Nous pensons que les connaissances suivantes ont un rôle important dans une modélisation exacte du raisonnement de l'apprenant :

- les connaissances prérequisées associées aux cas;
- la solution de l'expert pour les problèmes (aussi bien pour les cas);
- les cas qui seront sélectionnés par l'apprenant pour la résolution du problème à résoudre;
- la solution proposée par l'apprenant et les modes d'adaptation appliqués pour construire cette solution;

- les modes de raisonnements qui sont utilisés pour solutionner les cas;
- les degrés de similarités entre ces cas et le problème courant.

4.6 Modélisation des connaissances et des raisonnements de l'apprenant

Le problème est de déterminer une démarche permettant d'analyser le raisonnement de l'apprenant dans un contexte de résolution de problèmes. L'analyse du raisonnement consiste à concilier essentiellement les modèles mentaux de l'apprenant [VanLehn, 1988]. Nous nous proposons d'établir donc, un formalisme nous permettant de reconnaître et de regrouper les actions et les solutions de ce dernier en situation de résolution de problèmes. Ce formalisme nous permet également d'analyser la solution de l'apprenant plus en profondeur pour reconnaître son raisonnement. L'interprétation des résultats obtenus (lors de cette analyse) est une partie importante dans notre approche, puisque ces résultats sont précisément ceux qui sont recherchés et peuvent être utilisés pour modéliser les connaissances, les erreurs et les incompréhensions de l'apprenant; vérifier la cohérence de son travail et la solution qu'il propose.

4.6.1 Modes d'interaction

Nous identifions deux modes d'interaction pour notre approche : *AS mode (Accessible Solution mode)* et *US mode (Unaccessible Solution mode)*.

1. **AS mode (mode de solution accessible).** Dans ce mode, la base de cas est accessible par l'apprenant [Shiri *et al.*,1998b]. Pour chaque problème donné, l'apprenant cherche dans la base de cas ceux qui sont similaires au problème courant. Ensuite pour le cas trouvé, il effectue les processus d'adaptation afin d'atteindre la solution du problème.

La façon la plus simple pour illustrer cela, est de ramener les lecteurs aux cours de calculs ou de probabilité. Devant un nouveau problème, l'étudiant feuillette son livre et ses notes de cours afin de trouver un problème semblable dont il savait déjà la

solution. En répétant l'ensemble des étapes utilisées dans cette solution, l'étudiant pourrait résoudre le nouveau problème.

Cinq mécanismes sont particulièrement importants dans le processus de modélisation des connaissances de l'apprenant :

1. celui qui sélectionne le problème pour l'offrir à l'apprenant;
2. celui qui permet à l'apprenant de sélectionner le cas le plus proche au problème;
3. celui qui permet à l'apprenant d'adapter la solution du cas au problème;
4. celui qui analyse la solution de l'apprenant;
5. celui qui permet la mise à jour du modèle de l'apprenant.

Nous allons discuter ces mécanismes au chapitre 6.

Pour que notre système puisse mener à bien la modélisation de l'apprenant dans ce mode, il faut qu'il ait la possibilité de contrôler les actions de ce dernier. La reconnaissance des actions de l'apprenant pendant la résolution d'un problème est donc cruciale à la représentation correcte de l'état de ses connaissances.

2. **US mode** (*mode de solution inaccessible*). Dans ce mode (plus complexe), la base de cas n'est pas accessible. L'apprenant sans voir donc les solutions des cas, résoudra le problème en adaptant la (les) solution(s) d'un (ou des) cas similaire(s) qu'il a déjà résolu(s). Plus précisément, en se rappelant les étapes de la solution d'un cas similaire, ensuite en partant sur ces informations et en établissant une analogie avec la nouvelle situation, il peut résoudre ce problème. Il proposera ensuite, son raisonnement au système. Le système analysera ses réponses pour comprendre son raisonnement [Shiri *et al.*, 1998d].

La question qui se propose maintenant est : Pourquoi avons-nous besoin de deux modes d'interaction ? Dans ce qui suit, nous allons répondre à cette question.

4.6.2 Le besoin de différents modes d'interaction

Résolvant un nouveau problème en utilisant les techniques de RBC, il y a deux manières possibles de procéder :

1. *Réutilisation de la solution de vieux problèmes résolus.* De cette façon, l'apprenant a besoin de rechercher un problème semblable à la situation actuelle et puis d'adapter sa solution. Deux moyens donc sont nécessaires :
 - a. un outil qui aide l'étudiant à trouver un cas semblable au problème actuel;
 - b. un environnement qui peut faciliter la tâche d'adaptation.

Ce mode d'interaction est souvent utile pour des débutants. Car, comme nous savons de l'expérience en pédagogie, les débutants ont recours très souvent à l'exemple en situation de résolution de problèmes.

2. *Construction d'une solution (pour le problème) en se rappelant la solution de cas similaires.* Pour résoudre un problème par cette méthode, l'étudiant doit se rappeler d'abord comment est vu le problème comme variante de quelques problèmes familiers (c.-à-d. les cas). Il doit découvrir ensuite comment adapter la solution du cas pour dériver une solution pour le nouveau problème. Dans ce mode, l'étudiant est donc forcé de trouver la solution en utilisant des expériences antérieures.

Comme discuté dans [Frasson et Kaltenbach, 1991], l'idée est basée sur le fait que le niveau d'étude (et à notre avis, de même que la capacité de résolution de problèmes) est lié à l'état de saisie de connaissance éprouvées par l'étudiant.

Le résultat décrit dans [Chi *et al.*, 1989] [Frasson et Kaltenbach, 1991] prouve que les étudiants avancés relient le nouveau problème avec les problèmes précédemment résolus, en utilisant leur propre base de connaissance dans laquelle ils sont confiants et alors, ils essayent d'expliquer à eux-mêmes la solution d'un problème. Ce faisant, ils ramènent le raisonnement à leur propre approche de raisonnement. Cette méthode de résolution de problèmes est connue par des chercheurs comme l'effet de l'auto-explication (Self-Explanation Effect) [Chi *et al.*, 1989] [Frasson et Kaltenbach, 1991].

Enfin, les deux problèmes les plus importants qui se posent lorsqu'on veut modéliser les connaissances (et les raisonnements) de l'apprenant sont : le problème de choisir une représentation adéquate, et le problème de choisir un mécanisme pour analyser les réponses (la solution) de ce dernier. Nous allons présenter dans le chapitre suivant le premier problème, c.-à-d., notre base de connaissances et le formalisme que nous choisirons pour la représenter dans le système. Nous reviendrons ensuite, au chapitre 6 pour détailler notre approche sur le processus de modélisation de l'apprenant.

4.7 Conclusion du chapitre

Les anciennes solutions laissent une trace en mémoire qui s'active pour contribuer aux nouvelles solutions. On peut dire alors que l'apprentissage humain est basé sur des expériences progressives. Nous nous sommes basés sur cette philosophie (l'expérience progressive) pour développer une méthode de modélisation du raisonnement de l'apprenant. En effet, le raisonnement à base de cas qui retient l'expérience passée peut jouer un rôle important dans notre approche. Nous avons présenté les principes de la réalisation de ce dernier et nous avons discuté du besoin d'identifier deux modes d'interaction. Nous avons également présenté l'architecture que nous avons développée.

Chapitre 5

GCR : Un Formalisme pour la Modélisation et la Représentation des Connaissances dans les Systèmes à Base de Cas

Il faut noter que représentation sous-entend aussi raisonnement.

- Forbus, 1988.

5.1 Introduction

Typiquement, un certain nombre de tâches doivent être faites avant qu'un système à base de cas puisse proposer des réponses.

Il y a trois étapes de base dans n'importe quel système à base de cas; premièrement des données doivent être examinées et les index significatifs doivent être identifiés, deuxièmement des données doivent être saisies et représentées, et troisièmement des données doivent être classées pour une recherche efficace. La question qui se pose est comment représenter les connaissances pour pouvoir les traiter efficacement ?

Il est évident que, plusieurs formalismes de représentation de connaissances existent, comme les règles de production [Davis *et al.*, 1977], les réseaux sémantiques [Collins et Quillian, 1969], les schémas [Minsky, 1981], les graphes génétiques [Goldstein, 1982], les graphes conceptuels [Sowa, 1984], etc. Quel est alors le formalisme le plus approprié pour notre approche ?

Ce chapitre fournit nos réponses actuelles à plusieurs questions telles que : comment représenter les connaissances du domaine, comment classer les cas et les solutions (base de cas) pour l'accessibilité, comment mettre en application les processus de recherche pour l'efficacité, et ainsi de suite.

5.2 *Modélisation des connaissances du domaine*

L'objectif de cette section est de formaliser les connaissances du domaine. Ce formalisme nous permettra de mieux comprendre les caractéristiques que doit posséder notre système pour qu'il puisse diagnostiquer les solutions proposées par l'apprenant.

Les bases de connaissances que notre modèle exige sont : *la base de problèmes et la base de cas*. Il y a plusieurs questions que nous devons poser en établissant ces bases de connaissances :

- Qu'est ce qu'un cas ?
- Qu'est ce qu'un problème ?
- Quels éléments ont-ils et quels genres de connaissances doivent-ils encoder ?
- Quels sont les formalismes les plus utiles pour les représenter ?

D'autres enjeux concernent le choix des indices d'un cas (les indices d'un cas sont des combinaisons de ses descripteurs importants, ceux qui le distinguent d'autres cas), l'organisation de la base de cas et de la base de problèmes, etc. Nous allons maintenant répondre à ces questions.

5.2.1 **Base de cas**

La base de cas contient les cas stockés en mémoire qui seront utilisés par l'apprenant pour la résolution de problèmes (un cas est un problème résolu (description + solution)).

La construction d'une base de cas est un processus important, le choix des cas initiaux est une tâche qui doit être faite par un expert du domaine; il doit respecter deux principes :

- les cas doivent couvrir tout le champ de la tâche de raisonnement;
- les cas doivent couvrir les solutions les plus connues (ils doivent être représentatifs).

5.2.2 **Les cas**

Modéliser un cas nécessite de disposer de certaines connaissances :

- les informations pour identifier ou classifier les cas (comme par exemple, *l'indice, le nom du cas, etc.*);

- la description du cas et le but à atteindre;
- la solution de ce cas, le processus ayant abouti à la solution;
- les connaissances prérequisées associées à ce cas;
- le mode du raisonnement qui est utilisé dans la solution du cas.

Éléments de cas : Un cas est donc constitué de trois parties : *l'information, la description du cas, et les solutions* (à chaque solution, on fait aussi correspondre le mode de raisonnement associé).

5.2.3 Base de problèmes

Dans la base de problèmes sont enregistrés tous les problèmes connus. La construction de cette base et le choix des problèmes doivent aussi être faits par l'expert du domaine.

5.2.4 Les problèmes

Un problème est un cas que l'on cherche à résoudre en adaptant la solution d'un cas similaire déjà contenu dans la base de cas. Modéliser un problème nécessite aussi de disposer de certaines connaissances :

- les informations pour identifier ou classifier les problèmes;
- la description du problème et le but à atteindre;
- les connaissances prérequisées associées à ce problème;
- la solution de ce problème, le processus ayant abouti à la solution;
- le mode du raisonnement qui est utilisé dans cette solution;
- les liens entre le problème à résoudre et les cas similaires ainsi que les degrés de similarité entre eux.

Éléments de problème : Un problème est constitué de trois parties : *l'information, la description du problème, et tous les cas (problèmes résolus) similaires au problème*. Nous prouverons dans le prochain chapitre que les solutions des problèmes seront obtenues automatiquement à partir des solutions de cas semblables.

5.2.5 Notion d'espace de recherche

La résolution de problèmes chez l'humain, selon Newell et Simon (1972), prend la forme d'une recherche dans un espace de solutions qu'on peut représenter par un graphe. Dans ce graphe, les nœuds représentent les états des connaissances et les arcs représentent les opérateurs (les actions) qui indiquent qu'on peut passer d'un état à l'autre à la suite des actions.

Si résoudre un problème c'est chercher un cheminement dans un espace, il est fondamental de définir cet espace (Figure 5.1). La définition de cet espace est précisément le résultat de l'interprétation du problème.

Cette représentation a quatre composantes :

1. un *ensemble d'éléments* qui sont des structures de symboles; chaque structure représentant un état de la connaissance;
2. un *ensemble d'opérateurs* $\{OP_1, OP_2, \dots, OP_n\}$; chacun produisant de nouveaux états de connaissances à partir des états de connaissances existants;
3. un *état de connaissance initial* qui est la connaissance que le sujet avait au début du processus;
4. un *ensemble d'états finaux* qui sont destinés à être atteints en utilisant les opérateurs.

Définition : Une solution au problème est une séquence ordonnée d'opérateurs $(OP_1, OP_2, \dots, OP_n)$ nous permettant de passer de la situation initiale à la situation finale. Résoudre un problème revient à déterminer une telle séquence d'actions.

5.2.6 Exemple

Considérons le problème suivant :

Problème : montrez que, $\frac{\partial}{\partial x} [(x^n + k)(x^n - k)] = 2nx^{2n-1}$.

La Figure 5.1 présente un espace de solutions dont les séquences ordonnées d'opérateurs (OP_1, OP_2, OP_3, OP_4) , $(OP_1, OP_2, OP_3, OP_5, OP_6)$, $(OP_7, OP_8, OP_9, OP_{11}, OP_{13}, OP_{14})$, $(OP_7, OP_8, OP_{10}, OP_{12}, OP_{13}, OP_{14})$ donnent des solutions différentes pour ce problème.

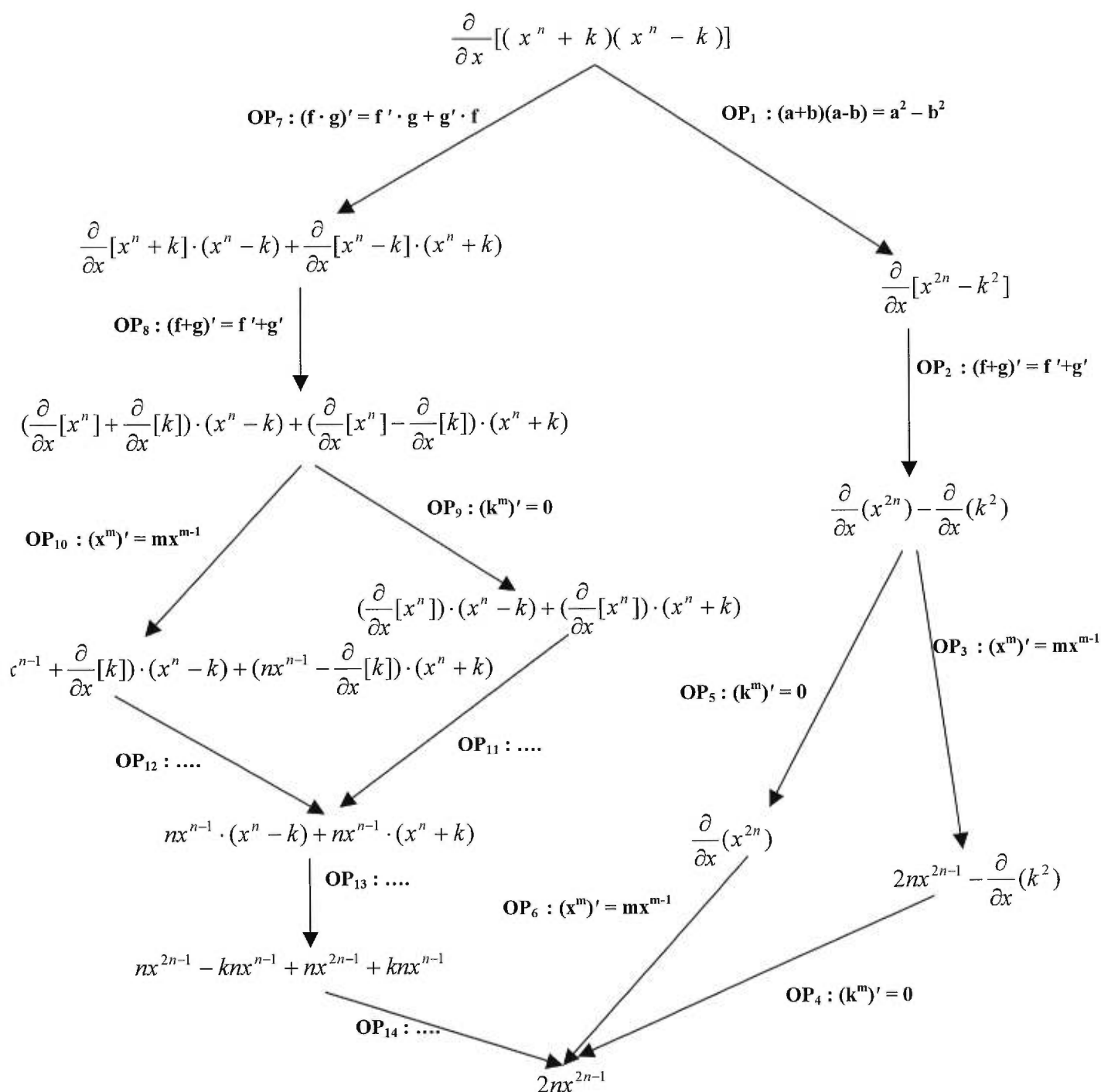


Figure 5. 1 : Un espace de solutions

5.3 L'approche de représentation des connaissances

Dans cette section, nous examinons quelques approches présentées dans la littérature pour représenter les connaissances.

5.3.1 Représentation des connaissances dans les STI

La représentation des connaissances constitue un aspect important de la conception d'un STI. Dans des domaines comme l'intelligence artificielle, le génie logiciel, les systèmes tutoriels intelligents etc., plusieurs types de connaissance sont manipulés et traités. Il n'existe pas de théorie générale sur la représentation des connaissances et il est difficile de trouver une représentation pour supporter toutes les différentes perspectives des connaissances manipulées dans la formation, du moins avec les techniques de représentation actuellement utilisées en intelligence artificielle

Or, face à la diversité des types de connaissance, différents formalismes tels les réseaux sémantiques, la logique des prédicats, les schémas, les règles de production, etc., ont été proposés dans la littérature. Les principales représentations utilisées dans les STI sont :

5.3.1.1 Règles de production

Dans un système basé sur les règles, les connaissances sont représentées par une série de règles de production ([Si <conditions> Alors <actions>]) [Davis *et al.*, 1977] comme par exemple : « **Si** un lien de déviation existe **Alors** présenter à l'apprenant la ressource de correction de déviation ». Les systèmes Lisp Tutor [Reiser *et al.*, 1985], Geometry Tutor [Anderson *et al.*, 1985] et Buggy [Brown et VanLehn, 1980] [Brown *et al.*, 1982] ont recours au formalisme des règles de production pour représenter ses connaissances.

5.3.1.2 La notation et/ou

La notation et/ou est aussi employée pour définir les concepts sous forme de nœuds dans un graphe. Cependant, les relations sont de type 'préalable' et permettent d'exprimer des conjonctions ou des disjonctions de 'préalables'. Elles décrivent les conditions nécessaires à l'enseignement d'un nouveau concept [McCalla *et al.*, 1982]. La Figure

5.2 montre un graphe et/ou représentant une partie de l'espace d'état pour l'intégration d'une fonction. Dans MYCIN [Shortliffe, 1976] par exemple, la connaissance se présente sous forme de graphe et/ou.

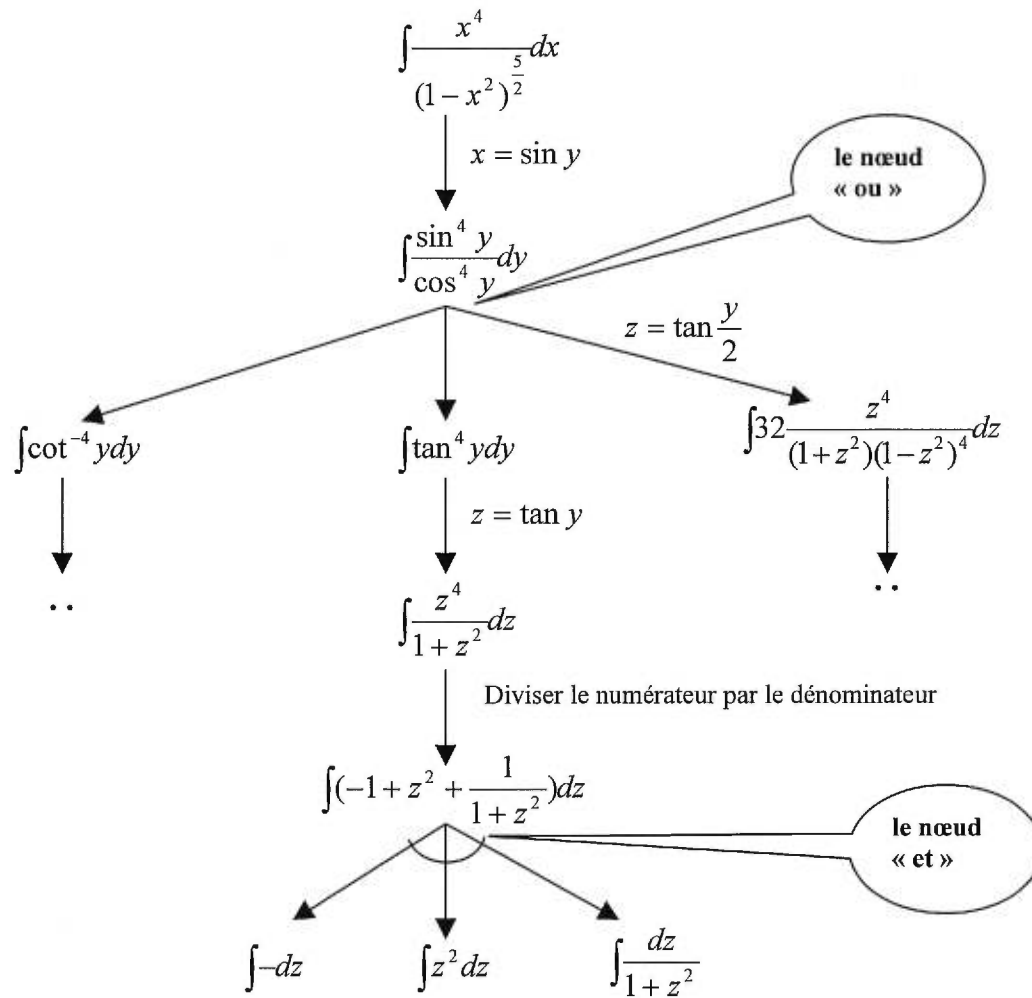


Figure 5. 2 : Exemple d'un graphe et/ou.

5.3.1.3 Les schémas

Dans certains systèmes, les connaissances sont encodées sous forme de schémas. Une définition d'un schéma est la suivante [Minsky, 1981] : un schéma est une structure de

données pour représenter une situation stéréotypée, telle que le fait de se trouver dans une certaine salle de séjour, ou d'aller à une fête d'enfants. Toutes sortes d'informations sont liées à chaque schéma. Une partie de cette information est liée à la façon d'utiliser le schéma, une partie est liée à ce que l'on croit concernant ce qui arrivera après, et une partie est liée aux actions à prendre si les prévisions ne se confirment pas. La Figure 5.3 représente un exemple de schéma pour la description d'un lit d'hôtel.

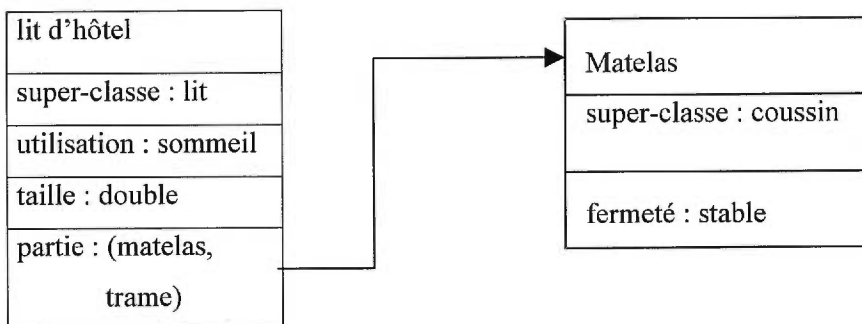


Figure 5. 3 : Un exemple de schéma (la description d'un lit d'hôtel)

Dans PROUST [Johnson et Soloway, 1985] par exemple, la connaissance représente l'aspect d'une hiérarchie de trois niveaux de schémas : des schémas servent à décrire des tâches, d'autres à spécifier des stratégies pour résoudre ces tâches et finalement, d'autres à désigner la façon d'implanter ces stratégies dans un langage de programmation donné.

5.3.1.4 Les réseaux sémantiques

Un Réseau Sémantique (RS) [Collins et Quillian, 1969] est un ensemble de nœuds, représentant les concepts (de l'apprenant ou du domaine), et un ensemble de relations (par exemple : de types 'partie-de' et 'généralisation') reliant les nœuds les uns aux autres. Ces relations servent à faire des inférences à partir du réseau [Anderson, 1988]. La Figure 5.4 représente une parcelle d'un réseau sémantique. Dans SCHOLAR [Carbonell, 1970b] et BUGGY [Brown et Burton, 1978] par exemple, la connaissance se présente sous la forme d'un réseau sémantique.

Il existe plusieurs types de réseaux sémantiques selon la signification des nœuds et des arcs. Par exemple, les nœuds peuvent représenter des schémas, des propositions, des

ensembles, ou des faits. La complexité du réseau est liée à la quantité d'informations contenues dans les nœuds, et au nombre d'arcs qui les relient. L'étiquette des arcs varie du simple caractère à l'utilisation des mots du langage naturel.

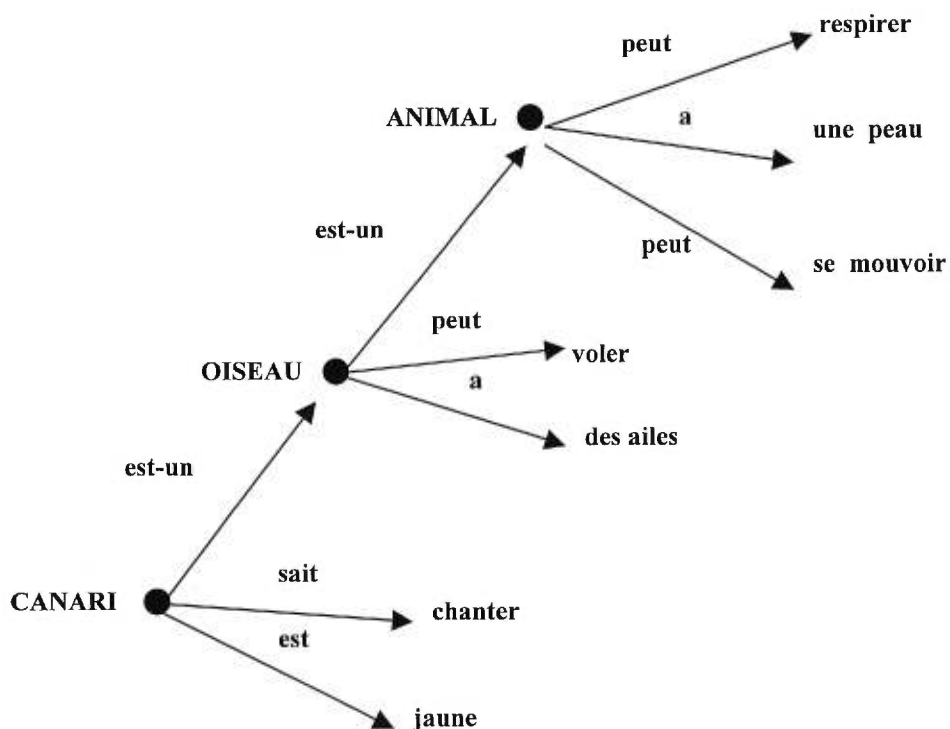


Figure 5. 4 : Un exemple de réseau sémantique [Collins et Quillian, 1969].

5.3.1.5 Graphes Conceptuels (GC)

Parmi ces différents formalismes de représentation de connaissances, les réseaux sémantiques ont été très souvent exploités dans les STI. Différentes formes de réseaux sémantiques ont été développées selon des besoins. John Sowa (1984) propose la théorie des Graphes Conceptuels (GC), comme modèle de représentation de connaissances.

Un GC est un graphe composé de nœuds (concepts) connectés par des relations. La Figure 5.5 montre un exemple de GC (le graphe conceptuel correspondant à la phrase « la femme Valérie aime l'homme André »). Un concept dans un GC est une spécification typée d'un référent ([Type : Référent]). Un référent peut être une instance particulière, une variable ou un GC.

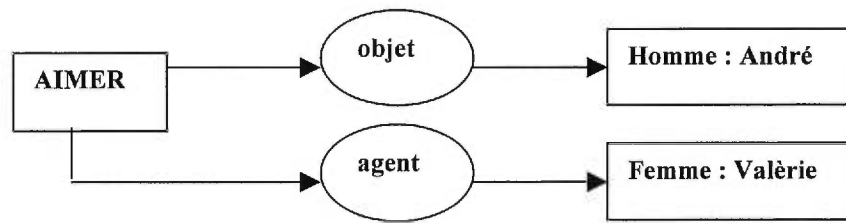


Figure 5. 5 : Exemple de GC

Le choix d'un modèle de représentation de connaissances basé sur le formalisme des graphes conceptuels a été motivé par un avantage certain [Aïmeur, 1994], par exemple :

1. ils constituent un modèle de représentation général très clair, c.-à-d., ils peuvent être utilisés pour représenter de façon uniforme toutes les connaissances utiles dans le système comme par exemple, les connaissances du domaine et les connaissances de l'apprenant;
2. ils permettent d'intégrer une culture générale très vaste et apportent enfin une rigueur logique de la représentation ainsi que des aspects sémantique riches;
3. ils offrent une gamme variée d'opérations, pouvant servir à la conception et l'implantation aisées d'un analyseur de raisonnement, etc.

Remarque :

Afin de pouvoir formuler des traitements plus complexes sur les graphes conceptuels, il est plus approprié d'employer la représentation linéaire de GC, laquelle est moins lisible mais plus pratique. La représentation linéaire correspondant au graphe représenté à la Figure 5.5 est par exemple :

[Femme : Valérie] ← (agent) ← [Aimer] → (objet) → [Homme : André].

5.3.2 Représentation des connaissances sous forme de graphes dans les systèmes à base de cas

La représentation du cas d'un système à base de cas doit supporter différents mécanismes : indexation, recherche, et comparaison ou adaptation. En fait, une représentation inadéquate de cas limite la capacité (du raisonnement) du système.

Plusieurs systèmes de RBC (ou systèmes de raisonnement analogique) tels que le PLEXUS [Alterman, 1998], GREBE [Branting, 1990], COOKIE [McCartney, 1993], ARCS [Thagard *et al.*, 1990], ACME [Holyoak et Thagard, 1989], SME [Falkenhainer *et al.*, 1990], CHIRON [Sanders *et al.*, 1997] et le CAPER [Sanders *et al.*, 1997] ont utilisé les représentations sous forme de graphes dans leurs approches.

Les représentations sous forme de graphes offrent des avantages significatifs tels que l'expressivité (car ils sont capables de représenter des relations entre les objets) et l'économie (capables de réduire le coût d'appariement et de saisie de cas) [Sanders *et al.*, 1997].

À la suite de cette section, nous allons décrire notre formalisme de représentation des connaissances.

5.4 Notre approche de représentation des connaissances

La représentation que nous proposons doit prendre en compte les quatre points suivants :

- nous permettre de modéliser les connaissances, les erreurs et les incompréhensions de l'apprenant;
- nous permettre d'exprimer les bases de connaissances et formuler les cas et les problèmes;
- nous permettre d'organiser les connaissances afin de pouvoir faciliter la résolution de problèmes;
- être facile à comprendre et à manipuler par l'apprenant et par la machine.

5.4.1 Graphe Classifié Concepts et Relations (GCR)

Dans la pratique, lors du développement de notre système, nous avons senti le besoin de moyens pour la classification des divers types de relation dans le GC. Pour répondre à ce besoin, nous avons proposé le *Graphe Classifié Concepts et Relations (GCR)* qui nous permet de classer aussi bien les relations que les concepts dans un graphe.

Par un GCR, nous entendons un graphe fini, orienté, connexe et bipartite. Les deux types de nœud de ce graphe représentent les deux notions de concept et de relation. Les arcs du graphe lient des nœuds d'un type donné à des nœuds de l'autre type. Dans ce graphe, les nœuds conceptuels (c'est-à-dire des nœuds de type concept) et aussi bien les nœuds relation sont représentés par un couple, comprenant un label de type et un référent (voir Figure 5.6).

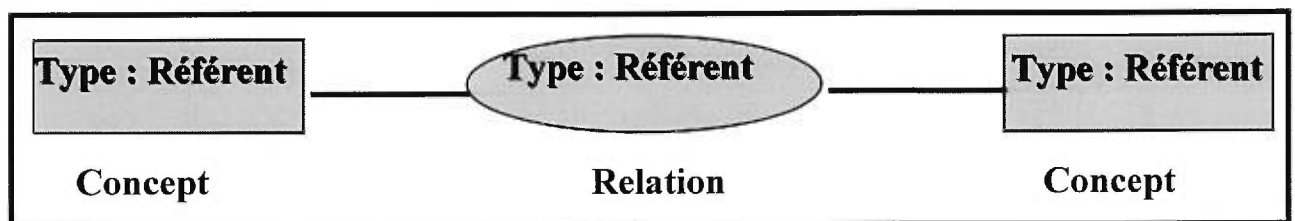


Figure 5. 6 : Les éléments de base d'un GCR.

Définition : nous définissons un *Graphe Classifié Concepts et Relations* (ou un GCR) comme étant un cas particulier de réseaux sémantiques où les concepts et les relations sont composés du type et du référent (Figure 5.6).

Remarque : dans les graphes conceptuels présentés par Sowa (1984), les relations n'ont pas de référent. Afin de montrer la nécessité d'avoir un tel référent (dans le GCR), nous présentons les deux exemples suivants :

Exemple 1 : Considérons par exemple le GC suivant qui indique « le problème P_1 est similaire au cas C_1 » : [Problème : P_1] \rightarrow (similaire) \rightarrow [Cas : C_1].

La question qui se pose maintenant est comment représenter la fonction de similarité (entre le problème et le cas) dans ce graphe ? Cette dernière est nécessaire pour produire

la solution de problème en adaptant la solution de cas similaire. GCR offre la possibilité de représenter ces fonctions comme les référents des relations. Ce dernier peut être décrit comme suit : $[\text{Problème} : P_1] \rightarrow (\text{similaire} : F(.)) \rightarrow [\text{Cas} : C_1]$ où $F(.)$ est la fonction de similarité entre le problème et le cas.

Exemple 2 : Considérons GC suivant qui indique « S_1 est une solution de cas C_1 » :

$[\text{Cas} : C_1] \leftarrow (\text{est une solution}) \leftarrow [\text{Solution} : S_1]$.

Comme pour l'exemple précédent, la question qui se pose est comment représenter le mode de solution dans ce graphe ? Le GCR permet de représenter ce dernier comme le référent d'une relation. Ceci peut être décrit comme suit :

$[\text{Cas} : C_1] \leftarrow (\text{est une solution} : \text{inductive}) \leftarrow [\text{Solution} : S_1]$.

Les exemples vus ci-dessus mettent en évidence l'importance du rôle des référents (de relations) dans le GCR.

5.4.2 Définition formelle

Un graphe classifié concepts et relations G est entièrement défini par la donnée d'un triplet $(C(G), R(G), L(G))$, où :

- $C(G)$ est l'ensemble des concepts figurant dans le graphe G . Un concept c est représenté par un couple, $c = [\text{Type}(c) : \text{Réfèrent}(c)]$, où $\text{Type}(c)$, est le type du concept c et $\text{Réfèrent}(c)$ est le référent du concept c . Un référent peut être une instance particulière, une variable, une fonction, un ensemble ou un GCR.
- $R(G)$ est l'ensemble des relations conceptuelles reliant les concepts dans G . Une relation est représentée par un couple, $r = [\text{Type}(r) : \text{Réfèrent}(r)]$, où $\text{Type}(r)$, est le type de la relation r et $\text{Réfèrent}(r)$ est le référent de la relation r . Un référent peut être une instance particulière, une variable ou une fonction.
- $L(G)$ est un sous-ensemble de $C(G) \times C(G) \times R(G)$ qui donne pour chaque couple de concepts les relations qui les relient dans le graphe G .

5.4.3 Les opérations sur les GCR

En utilisant les opérations définies sur les graphes conceptuels par Sowa (1984), on définit quatre opérations de base sur les GCR qui permettent de dériver un nouveau graphe à partir des GCR existants.

1. *La copie* : Si $G_1 = \text{copie}(G)$ alors $G = G_1$. Cette opération réalise une duplication du graphe original.
2. *La restriction* (d'un concept ou d'une relation) :
 - *La restriction d'un concept* : $\text{Restriction}(G) = G_1$, G_1 est obtenu en spécialisant dans G un de ses concepts. On spécialise un concept c_1 par deux transformations possibles :
 - a. Si c_1 est un concept générique alors on le spécialise en lui attribuant un référent qui soit conforme à son type.

Exemple : [Problème : *] peut être restreint à [Problème : P₁]

- b. Si c_1 est un concept spécifique on peut remplacer son type par un type t plus spécifique, à condition que le référent (c_1) soit conforme au nouveau type t .

Exemple : [Problème] peut être restreint à [Problème-résolu]

- *La restriction d'une relation* : obtenir un nouveau GCR en remplaçant le type d'une relation d'un GCR par un sous-type et/ou en remplaçant le référent générique d'une relation par un référent particulier.

Remarque : Cette opération n'existe pas chez Sowa.

Exemple : (Solution : *) peut être restreint à (Solution : Inductive).

3. *La jointure* : la jointure de deux graphes sur un concept identique produit un graphe obtenu en éliminant ce concept d'un des deux graphes et en raccrochant tous les arcs restants pendants à l'autre graphe.

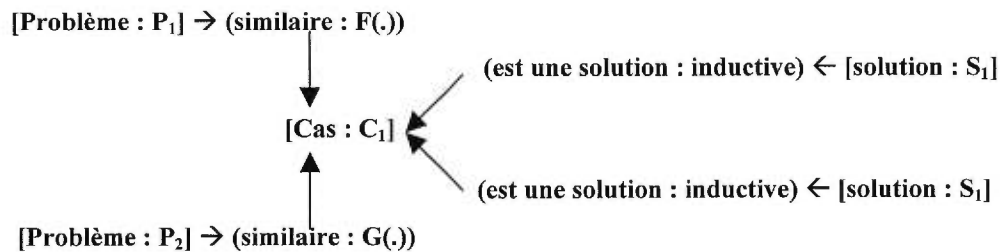
Exemple :

Soient les graphes G_1 et G_2 suivants :

G_1 : [Problème : P_1] \rightarrow (similaire : $F(.)$) \rightarrow [Cas : C_1] \leftarrow (est une solution : inductive) \leftarrow [solution : S_1]

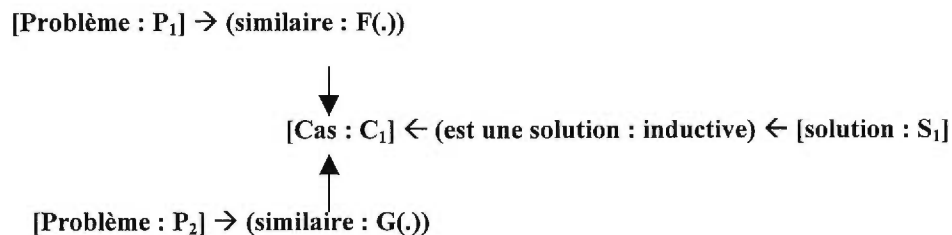
G_2 : [Problème : P_2] \rightarrow (similaire : $G(.)$) \rightarrow [Cas : C_1] \leftarrow (est une solution : inductive) \leftarrow [solution : S_1]

Le jointure de G_1 et G_2 sur le concept [Cas : C_1] nous donne le graphe G_3 suivant :



4. *La simplification* : L'opération de simplification permet d'éliminer dans un graphe toutes les relations dupliquées.

Exemple : Si on considère maintenant le graphe G_3 présenté ci-dessus, on peut éliminer l'une des deux relations (est une solution : inductive) \leftarrow [solution : S_1] dans ce graphe, d'où résulte le graphe suivant :



Nous présentons brièvement dans ce qui suit comment le GCR nous permet de représenter un cas et un problème.

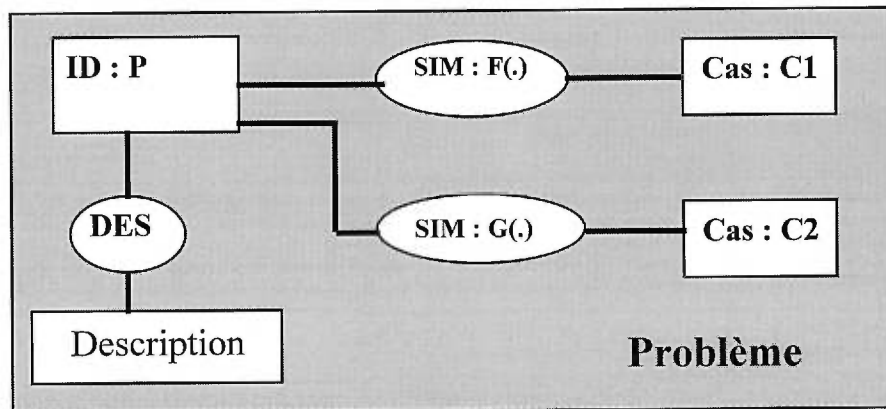
5.5 Représentation des connaissances dans notre système

Dans cette section, nous allons représenter les connaissances de notre système par le formalisme GCR.

5.5.1 Représentation d'un problème

La Figure 5.7 reflète la représentation d'un problème à l'aide d'un GCR. Dans cette représentation, deux types de relation sont utilisés :

1. relation de type **DE**scription (DES) : c'est une relation qui s'établit entre l'ID (les informations pour identification du cas, comme par exemple son nom) et la description du problème;
2. relation de type **SIM**ilarité (SIM) : c'est une relation qui relie les cas similaires au problème en question. Le référent de cette relation indique la fonction de similarité entre le problème et le cas correspondant.



ID = Information
DES = relation du type **DE**scription
SIM = relation du type **SIM**ilarité
F(.), **G(.)** = fonctions de similarité

Figure 5. 7 : La représentation d'un problème

- **Exemple 1** : $P = \text{Problème 1}$;
 Description : Déterminer $\int (\ln x)^2 dx/x$;
 $F_1(.) : T, n, dT \rightarrow (\ln x), 2, dx/x$; $C1 = \text{Cas 1}$;
 $F_2(.) : U, dV \rightarrow (\ln x)^2, dx/x$; $C2 = \text{Cas 2}$.
- **Exemple 2** : $P = \text{Problème 2}$;
 Description : soit $S_0 = a, S_{(n+1)} = b * (S_n), (n \geq 0)$; montrer que $S_5 = b^5 * a$.
 $G_1(.) : a, b \rightarrow c, 1/2$; $C1 = \text{Cas 3}$.

- **Exemple 3 :** P = Problème 3;

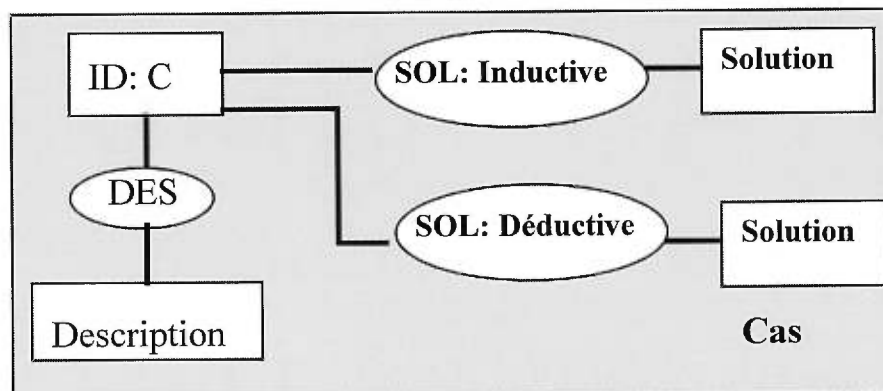
Description : montrez que, $\frac{\partial}{\partial x}[(x^n + k)(x^n - k)] = 2nx^{2n-1}$.

$H_1(\cdot) : n, k \rightarrow 2, b; C1 = \text{Cas 4}$.

5.5.2 Représentation d'un cas

La Figure 5.8 reflète la représentation d'un cas à l'aide d'un GCR. Dans cette représentation, deux types de relation sont utilisés :

1. relation de type **DE**scription (DES): c'est une relation qui s'établit entre l'ID et la description du cas;
2. relation de type **SOL**ution (SOL) : c'est une relation qui relie une solution à ID. Le référent de cette relation indique le mode de raisonnement (*Inductif* ou *Déductif*) appliqué à cette solution.



ID = Information
DES = relation du type DEscription
SOL = relation du type SOLution

Figure 5. 8 : La représentation d'un cas.

- **Exemple 1 :**

C = indice : (intégrale, par changement de variable); nom : Cas 1

Description = Déterminer $\int T^n dT$

SOL = Déductive;

Solution = $(1/(n+1))T^{n+1} + C$.

- **Exemple 2 :**

C = indice : (intégrale, par partie); nom : Cas 2

Description = Déterminer $\int U dV$

SOL = Dédutive;

Solution = $UV - \int V dU + C$.

- **Exemple 3 :**

C = indice : (série, géométrique); nom : Cas 3

Description : soit $S_0 = c$, $S_{(n+1)} = (1/2) * (S_n)$, ($n \geq 0$); montrer que $S_5 = (1/2)^5 * c$.

SOL = Inductive;

Solution : $S_1 = (1/2) * S_0 = (1/2) * k = (1/2)^1 * k$;

$$S_{(n+1)} = (1/2) * S_n = (1/2) * ((1/2)^n) * k = (1/2)^{n+1} * k;$$

$$S_5 = (1/2)^5 * k.$$

SOL = Dédutive;

Solution : $S_5 = (1/2) * S_4 = (1/2)^2 * S_3 = (1/2)^3 * S_2 = (1/2)^4 * S_1 = (1/2)^5 * S_0$
 $= (1/2)^5 * c$.

- **Exemple 4 :**

C = indice : (dérivée, ...); nom : Cas 4

Description = montrez que, $\frac{\partial}{\partial x} [(x^2 + b)(x^2 - b)] = 4x^3$

SOL = Dédutive;

Solution :

$$\frac{\partial}{\partial x} [(x^2 + b)(x^2 - b)] = \frac{\partial}{\partial x} [x^4 - b^2]$$

$$= \frac{\partial}{\partial x} (x^4) - \frac{\partial}{\partial x} (b^2) = \frac{\partial}{\partial x} (x^4) = 4x^3$$

Remarque : Nous devons mentionner ici que pour chaque cas, seulement les informations stockées dans les nœuds, c'est-à-dire les concepts (et non les relations) sont accessibles par l'apprenant.

5.5.3 Représentation de la base de connaissances

Une des questions qui se pose est celle de savoir comment organiser les bases de connaissances (la base de cas et la base de problèmes) afin de pouvoir faciliter la recherche des cas par l'apprenant et par le système. Cependant, un tel type de représentation doit respecter ces principes :

1. permettre à l'apprenant de retrouver facilement les cas qui se rapprochent le plus du problème à résoudre.
2. lui permettre ainsi de parcourir la base de cas pour retrouver les similarités entre le problème donné et d'autres cas.
3. permettre au système de générer facilement et efficacement des nouveaux problèmes au moment où celui-ci en a besoin.

Pour organiser nos bases de connaissances, nous utilisons une représentation basée sur la structure de « adressage dispersé ou hachage » dont la table principale (que nous appellerons ‘table d’index’) contient les indices.

La Figure 5.9 montre une représentation de la base de connaissances des problème 1, problème 2 et problème 3, et les cas 1, cas 2, cas 3 et cas 4, présentés dans les sections 5.5.1 et 5.5.2.

Ce type de représentation respecte les principes mentionnés au paragraphe précédent. Il s'agit du fait que : l'apprenant puisse d'une part parcourir les indices existant dans la table d'index lorsqu'il veut chercher un cas similaire au niveau du problème. Par la suite, il peut avoir accès au contenu des cas qui sont trouvés (et qui se rapprochent le plus du problème à résoudre), et sur lesquels il désire examiner la similarité avec le problème donné. D'autre part, pour consulter un cas connu, l'apprenant peut commencer par rechercher dans la table d'index (un fichier trié), ensuite il peut avoir accès au contenu de ce cas. De plus, c'est une organisation fondée sur une représentation hiérarchique, elle est donc efficace en temps de recherche des problèmes.

De cette façon, nous associons à chaque concept, des problèmes (et des cas) qui seront exploités au moment opportun pour les générer à l'évaluation du concept.

Définition : nous définissons les ressources didactiques de type problèmes ($R_P(C)$) associés au concept C , comme étant l'ensemble de tous problèmes (reliés à ce concept) qui sont destinés à l'évaluation du concept.

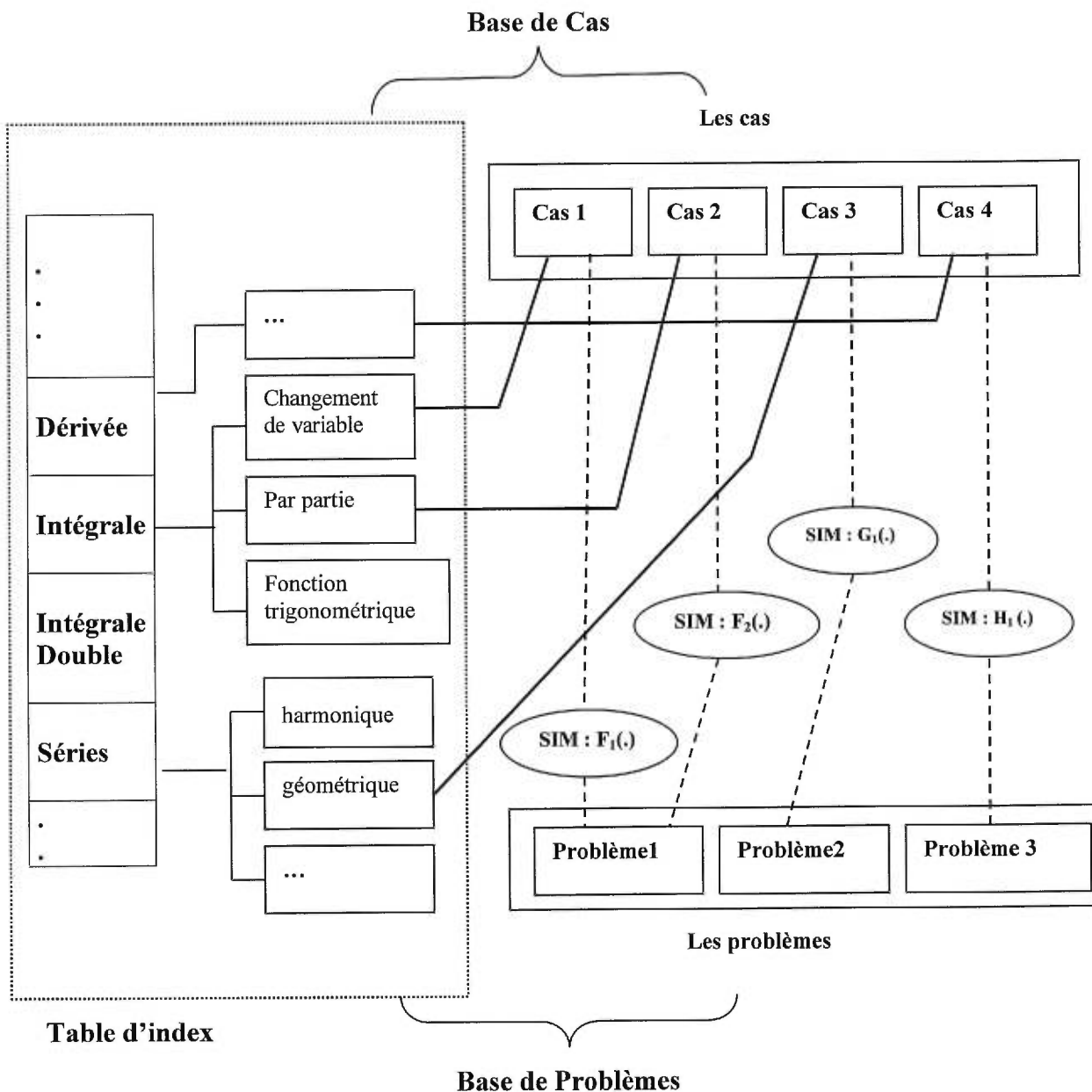


Figure 5. 9 : La représentation de la base de connaissances

Nous allons maintenant considérer certains problèmes qui se posent lorsque l'on développe la base de cas (ou la base de problèmes).

5.6 Quelques aspects fondamentaux

La construction d'un système de RBC est beaucoup plus sophistiquée que juste rechercher les cas semblables de la base de données et utiliser les solutions de ces cas. Nous présentons maintenant quelques tâches importantes qui sont nécessaires pour développer la base de connaissances.

Les tâches impliquent habituellement le choix des cas et des problèmes, la création de la base de cas et de la base de problèmes, l'interprétation des problèmes, l'identification des index qui devraient être utilisés pour rechercher les cas semblables, le calcul des similitudes entre les problèmes et les cas, la détermination des solutions des problèmes, et finalement le stockage de ces informations dans le système.

5.6.1 Le cycle du RBC pour l'expert

L'expert (avant même de déterminer les solutions des problèmes) construit la base de cas et la base de problèmes; pour ce faire, il réalise les étapes suivantes :

- *choix* des cas et des problèmes initiaux;
- *indexation* des cas et des problèmes;
- *stockage* des cas et des problèmes.

5.6.2 Le choix des indices

Le choix du meilleur ensemble d'indices pour classer les cas dans une base peut être difficile. La détermination des indices les plus appropriés peut donc exiger des expériences et doit être faite par l'expert du domaine.

5.6.3 La construction d'une base de cas

Les éléments les plus importants de tout système à base de cas est la mémoire de cas (base de cas) elle-même. C'est un dépôt des solutions passées du problème qui sert de base au processus entier de raisonnement. En tant que tel, la façon dont une mémoire de cas est organisée est importante. Cependant, la conception et l'entretien d'une base de cas

pour une application particulière peuvent être problématiques. Il est donc important d'adopter une méthode appropriée d'organisation et de recherche pour chaque application particulière.

Enfin, la construction d'une telle base est un processus interactif. On commence par collecter des cas et tester leur fiabilité et le degré de leur couverture du domaine de raisonnement.

5.6.4 Le problème de couverture des cas

Le succès de RBC dépend de la couverture de cas - combien de problèmes semblables peuvent être résolus par l'application de la solution d'un cas donné. La couverture de cas dépend des capacités d'adaptation de cas. Dans un domaine où l'espace de problèmes est continu, une division qualitative de cet espace de problèmes est nécessaire et la couverture de cas dépend de sa division qualitative. Il y a deux considérations contradictoires à prendre en compte en choisissant la division. Elle doit être assez bonne de sorte que tous les cas dont les solutions sont suffisamment semblables (que l'apprenant puisse transformer l'un à l'autre) devraient appartenir à la même classe. Mais la division doit être également assez brute de sorte que dans une grande base de cas, il existe une probabilité élevée de trouver au moins un cas qui apparie le problème donné.

5.6.5 Déterminer les solutions (de l'expert) des problèmes

Voici les étapes de raisonnement que fait l'expert pour déterminer les solutions de problèmes :

- *Recherche* : pour chaque problème il cherche dans la base de cas, tous les cas similaires au problème (tous les points de vue). Par exemple pour le problème 1, il trouve les cas 1 et cas 2 (voir 5.5 et la Figure 5.9).
- *Adaptation* : pour chaque cas similaire trouvé, il effectue les processus d'adaptation (en appliquant diverses fonctions de similarités) afin d'atteindre toutes les solutions possibles. Ces solutions seront stockées dans le système.

Remarque : Nous montrerons dans le prochain chapitre, la façon dont nous pouvons automatiser la génération des solutions pour les problèmes en utilisant les solutions de cas semblables et les fonctions de similarité.

Maintenant que nous avons déterminé un formalisme qui nous permet de représenter nos bases de connaissances, nous allons examiner plus en détails dans le chapitre suivant l'analyse du raisonnement de l'apprenant et le fonctionnement de notre approche lors de cette analyse.

5.7 Conclusion

Dans ce chapitre, nous avons vu d'abord quelques formalismes de représentation de connaissances. Ensuite nous avons proposé un formalisme (que on l'appelle GCR) pour représenter nos connaissances dans le système. Le formalisme que nous avons développé nous permet de classer aussi bien les relations que les concepts dans un graphe conceptuel. En particulier, il nous permet d'exprimer les relations entre les problèmes et les cas, d'exprimer les modifications qui sont nécessaires pour adapter le cas précédent à la situation actuelle et aux buts, et de représenter la relation entre un cas (ou un problème) et ses solutions. Il nous permet donc d'associer à chaque relation des informations (comme par exemple les modes des solutions, les fonctions d'adaptation, etc.) qui sont utiles pour une bonne modélisation de l'apprenant. Ceci sera montré dans le chapitre suivant.

Chapitre 6

Analyse du raisonnement de l'apprenant dans notre modèle

6.1 Introduction

Dans le chapitre 4, nous avons identifié deux modes d'interaction possibles : le mode de solution accessible (*AS mode*) et le mode de solution inaccessible (*US mode*).

- (1) *AS mode (Accessible Solution mode)*. Dans ce mode les solutions des cas sont accessibles par l'étudiant. Il adapte la solution d'un cas choisi, qu'il pense être semblable au problème donné, pour obtenir une solution pour le problème. Cette méthode est utile quand les STI s'adressent à un débutant.
- (2) *US mode (Unaccessible Solution mode)*. Dans ce mode les solutions des cas ne sont pas accessibles. Par conséquent, l'étudiant doit se rappeler les étapes de la solution du cas choisi, qu'il pense être semblable au problème donné, pour suggérer un moyen de résoudre le nouveau problème.

Dans ce chapitre, nous expliquerons les processus de modélisation de l'apprenant dans ces modes, nous donnerons ainsi l'algorithme de cette modélisation, nous montrerons comment la solution de l'apprenant est analysée, nous expliquerons les connaissances à modéliser, nous expliquerons comment ces connaissances sont acquises, nous montrerons comment le niveau de connaissances de l'apprenant est calculé, enfin, nous parlerons de la mise à jour du modèle de l'apprenant.

6.2 *Le processus de modélisation de l'apprenant en AS mode*

La Figure 6.1 montre le processus de construction du modèle de l'apprenant en *AS mode*. Ce processus implique plusieurs étapes [Shiri *et al.*, 1998b] :

- 1) *Sélection d'un problème*. D'abord, le STI choisit un problème pour le proposer à l'étudiant. La sélection du problème est basée sur des éléments du *modèle de l'apprenant* et des *concepts d'intérêt à enseigner*.
- 2) *Sélection des cas par l'étudiant*. Quand l'étudiant est confronté à un nouveau problème, il recherche dans la base un cas approprié qui convient le mieux au problème. Afin de permettre à l'étudiant de faire ceci, le STI est équipé d'un outil qui peut lui renvoyer la liste de cas pour les examiner.
- 3) *Sélection d'une solution (par l'étudiant)*. L'idée de cette étape est basée sur l'hypothèse que, normalement les humains suivent le chemin le plus connu pour eux, lorsqu'ils tentent de résoudre un problème ou de réaliser une tâche. Suivant cette hypothèse, on suppose que parmi les solutions existantes pour chaque cas, l'apprenant sélectionne la solution qui lui semble la plus vraisemblable. Nous rappelons que les solutions existantes dans la représentation de chaque cas utilisent différents modes de raisonnement et que ces modes de raisonnement ne sont pas accessibles par l'apprenant.

Remarque : Dans *AS mode*, nous utilisons normalement les solutions qui constituent le chemin le plus court et le plus efficace vers le but désiré pour les présenter à l'apprenant. Il est à noter que selon Newell et Simon (1972), concevoir la résolution d'un problème est comme le déplacement à l'intérieur d'un espace de recherche qu'on peut représenter par un graphe. Dans ce graphe, les nœuds représentent les états que peut prendre successivement la situation à la suite des actions du sujet et les arcs indiquent qu'on peut passer de l'un à l'autre des états qu'ils relient. L'un des nœuds de ce graphe représente la situation de départ et un autre, la situation à atteindre (le but). Chercher la solution revient à chercher un cheminement qui permet de relier le nœud départ et le nœud but.

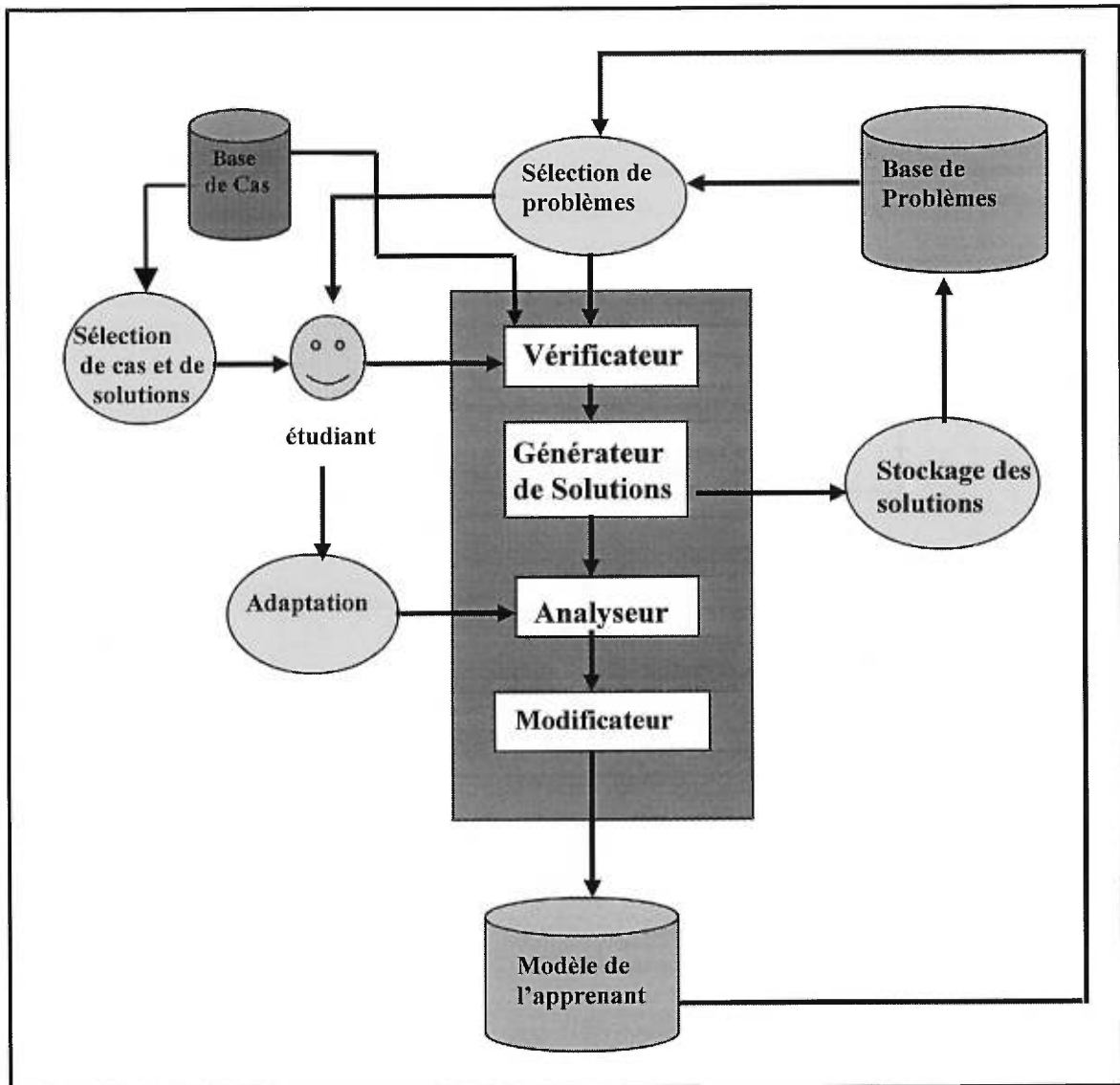


Figure 6. 1 : Le processus de modélisation de l'apprenant en *AS mode*.

Il y a en général plusieurs processus de solutions : certains peuvent être considérés comme meilleurs que d'autres en fonction de la longueur du chemin. Le processus optimal est celui qui correspond au chemin le plus court (voir 5.2.5 et 5.2.6).

- 4) *Vérification de la similitude entre le problème et le cas choisis par l'étudiant.* Le cas choisi par l'étudiant peut ne pas être semblable au problème. Dans cette condition, le *vérificateur* vérifie la sélection de l'étudiant en utilisant les liens entre le problème et

les cas similaires qui existent dans la représentation du problème (voir 5.5.1). Si le résultat n'est pas bon, le système informe l'étudiant qu'il n'a pas choisi un cas approprié.

- 5) *Adaptation de la solution du cas (par l'étudiant)*. Une fois que le cas (et la solution) a été choisi par l'étudiant, il crée une solution au problème actuel en adaptant la solution du cas choisi.

En général, il existe deux genres d'adaptation dans RBC [Watson, 1997] : *l'adaptation structurale* et *l'adaptation dérivative*.

- (a) L'adaptation structurale applique des règles ou des formules d'adaptation directement à la solution enregistrée dans les cas.
- (b) L'adaptation dérivative réutilise les règles ou les formules qui ont produit la solution initiale pour donner une nouvelle solution au problème actuel. Dans cette méthode, l'ordre de planification qui a construit cette solution initiale doit être enregistré comme attribut supplémentaire du cas. L'adaptation dérivative peut seulement être utilisée pour les domaines qui sont bien connus.

Enfin, l'adaptation structurale qui est utile pour *AS mode*, peut prendre plusieurs formes [Kolodner, 1993] : un élément de nouveau peut être inséré dans la solution de cas, un élément peut en être effacé, un certain élément peut être substitué à d'autres, ou une certaine partie de la solution de cas peut être transformée, etc.

- 6) *Génération de solutions*. Avant d'essayer d'analyser la solution de l'étudiant, le Générateur de Solutions devrait produire toutes les solutions possibles au problème en utilisant le même cas choisi par l'étudiant. Nous rappelons que les solutions produites par le système ont rivalisé avec la solution de l'étudiant. Afin d'obtenir les solutions pour le problème courant, le Générateur de Solutions adapte la (les) solution(s) du cas choisi par l'étudiant s'il est applicable, c.-à-d., si le cas choisi par lui est vraiment semblable au problème.

7) *Stockage des solutions*. Une fois les solutions d'un problème produites par le *Générateur de Solutions*, elles sont enregistrées dans le système pour la future référence. Les solutions produites peuvent être incluses dans la représentation du problème par la modification de la structure de cette représentation (voir la Figure 6.2). Ceci peut être fait : (1) en éliminant la relation entre le problème et le cas (ce qui est choisi par l'étudiant) de la représentation du problème; (2) et puis en reliant les solutions produites au problème. En général, afin d'identifier les solutions reliées, une relation du type SOLUTION (SOL) doit être utilisée. Le référent de cette relation devient un triplet $(f(\cdot), C_i, R_m)$ où C_i est le cas choisi par l'étudiant, $f(\cdot)$ est la fonction de similitude entre le problème et C_i , et R_m est le mode de raisonnement des solutions (par exemple : *inductif* ou *déductif*). La Figure 6.2 illustre comment ces éléments $(f(\cdot), C_i, R_m)$ ont été choisis pour un problème donné.

Remarque : La base de problèmes change dynamiquement chaque fois que des solutions sont produites par le système. Par conséquent, chaque fois que le système est face à la tâche de génération des solutions (pour un problème donné) il vérifie d'abord la base de problèmes. Si la solution de ce problème relativement au cas choisi par l'étudiant a déjà été enregistrée dans la base de problèmes, alors il n'a pas besoin de mettre en application les deux dernières tâches (c'est-à-dire, *Génération et Stockage de solutions*). Il est seulement suffisant de vérifier les similitudes entre le problème et le cas choisi par l'étudiant. La mise en place de cette tâche est nécessaire pour s'assurer que l'étudiant a choisi un cas approprié pour son raisonnement. En conséquence, le système produit et enregistre qu'une fois les solutions des cas semblable au problème courant.

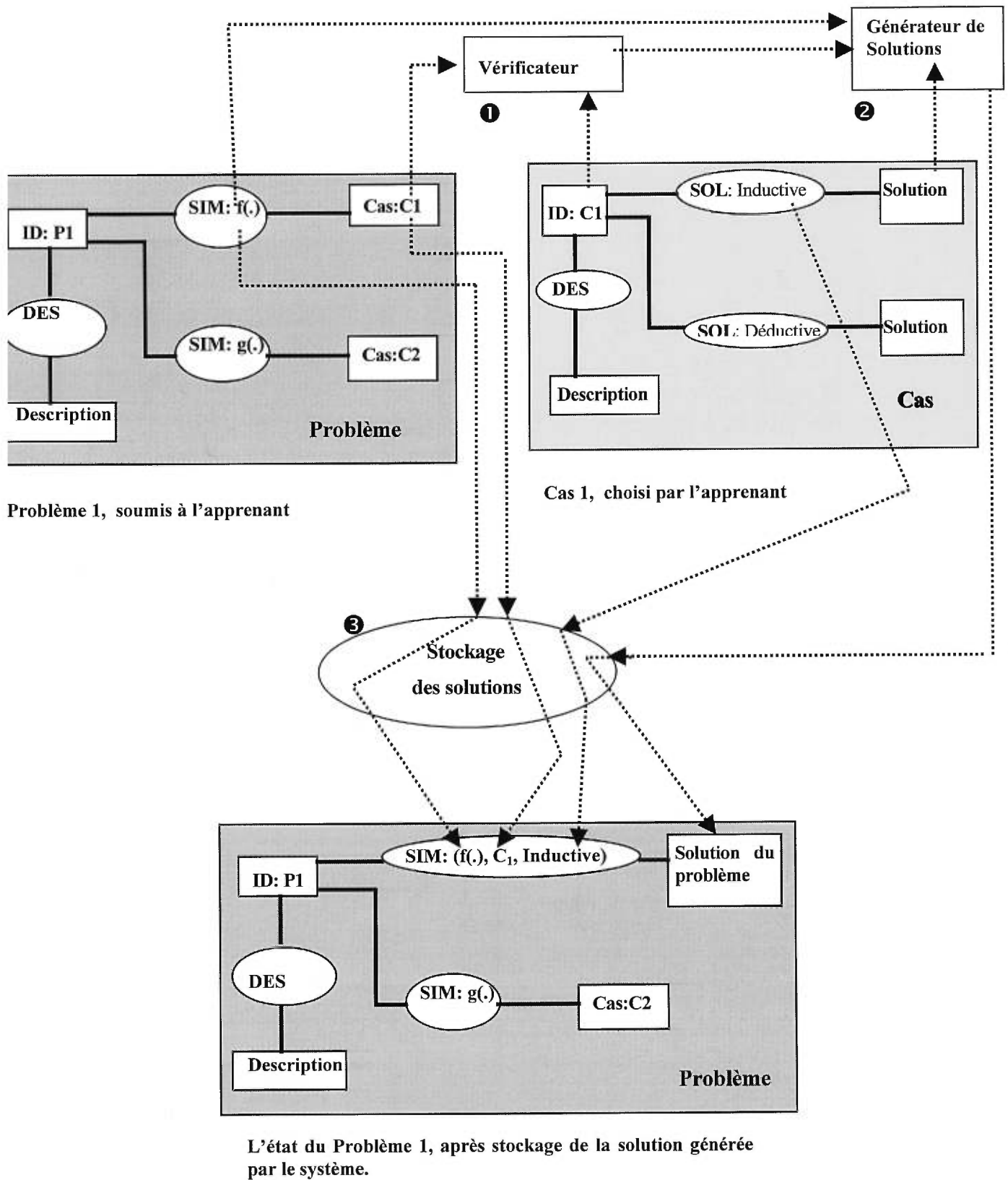


Figure 6. 2 : Génération et stockage de la solution du problème par le système.

- 8) *Analyse de la solution de l'étudiant*. L'analyse de la réponse de l'étudiant mène à la production du modèle de l'apprenant. À cette fin, *l'analyseur* compare cette solution à celles produites par le *Générateur de Solutions*, qui peut être considéré en tant que solutions d'experts. Nous détaillons cette étape dans la section suivante.
- 9) *Mise à jour du modèle de l'apprenant*. Enfin, le *modificateur* utilise les résultats des activités de l'analyseur pour mettre à jour le modèle d'étudiant (pour plus de détails voir 6.6).

6.3 L'analyse du raisonnement en AS mode

De manière générale, le rôle de l'analyseur est de gérer la réponse donnée par l'apprenant pour atteindre l'ensemble des concepts acquis et incompris par ce dernier, pour l'exercice en cours. De façon plus détaillée, voici les étapes suivies par le système lors de l'analyse du raisonnement de l'apprenant dans ce mode :

- L'étudiant(e) essaye d'abord de rechercher un cas semblable au problème actuel. Il est important d'expliquer ici que l'étudiant(e) ne peut pas rappeler un cas approprié à moins qu'il (ou qu'elle) comprenne la nouvelle situation. Ceci suggère que la compréhension ou l'interprétation d'une situation est une partie nécessaire du raisonnement pendant la résolution de problèmes. De cette façon, l'apprenant est prié d'établir un lien de similarité entre le problème et les cas. L'échec à trouver cette relation peut venir du fait qu'il ne connaît pas bien le problème.
- L'étudiant est alors invité à choisir la solution d'un cas pour sa tâche de résolution de problèmes. Une fois qu'il a sélectionné une solution, le système peut extraire le mode de raisonnement en utilisant les informations qui sont stockées dans la représentation des cas. En effet, les humains suivent normalement les modes de raisonnement qui leur sont les plus connus, quand ils essayent de réaliser une tâche ou de résoudre un problème [Shiri *et al.*, 1998a]. Basés sur cette hypothèse, nous supposons que l'étudiant choisit parmi les solutions existantes, celles qui sont les plus raisonnables ou qui lui sont les plus connues.
- Une recherche réussie de cas ne suffit pas pour juger la performance de l'apprenant. Pour évaluer son choix, son adaptation doit être aussi évaluée. En fait, dans tous les

cas, nous exigeons de nous assurer qu'il peut atteindre une solution. L'objectif de cette étape est donc que le système s'assure que l'apprenant est bien familier avec le domaine. Autrement dit, on veut s'assurer qu'il n'a pas choisi le cas et sa solution par hasard. Pour ce faire, à la suite de la sélection d'une solution par l'apprenant, le système lui demande de saisir les étapes de cette dernière. À chaque étape, il doit aussi extraire l'opérateur qui permet de transformer les états, puis il doit appliquer l'adaptation nécessaire afin d'atteindre l'étape correspondante de la solution associée au problème donné (voir la Figure 6.3). En cas d'échec dans les tâches demandées, on conclut que l'apprenant n'est pas bien familier avec la solution qu'il a choisie.

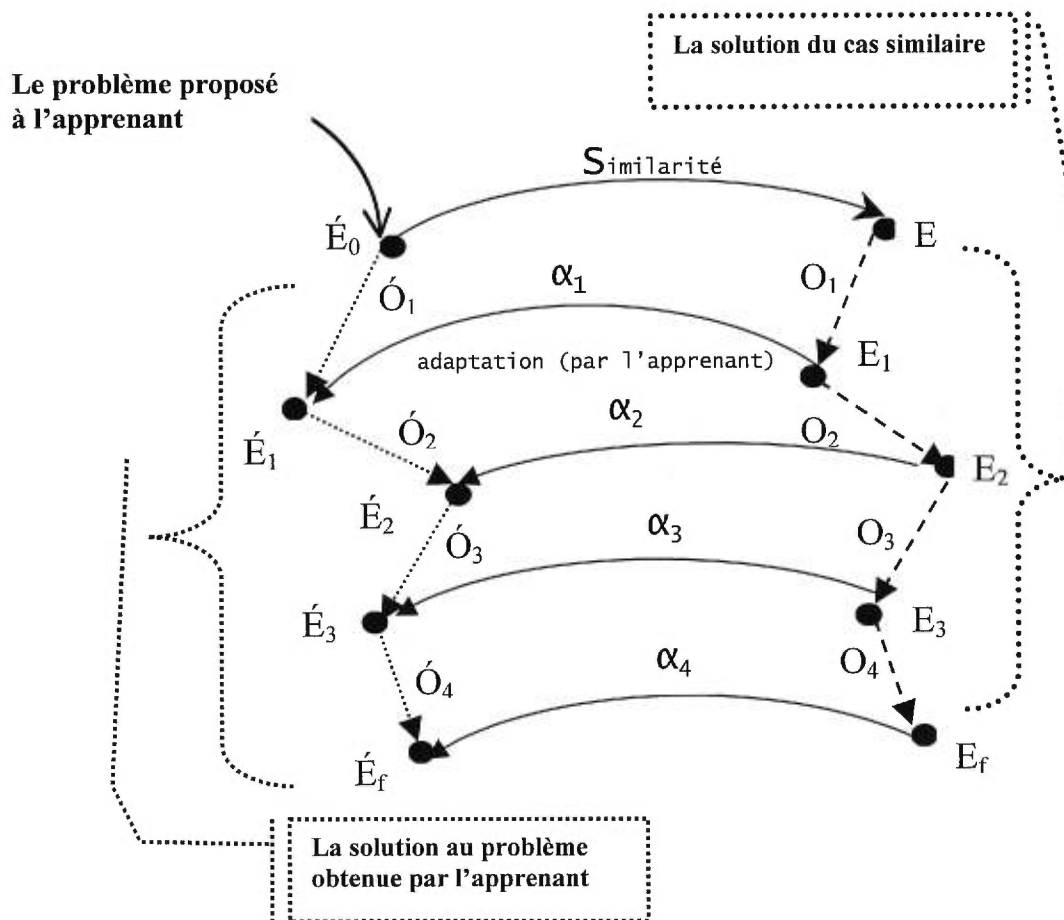


Figure 6. 3 : Le mécanisme d'adaptation.

Exemple : Afin de mieux comprendre le scénario de résolution de problèmes par l'apprenant en *AS mode*, voici un exemple d'exécution d'un exercice (Figure 6.4) : à

la rencontre d'un nouveau problème et une fois qu'un cas similaire et une solution de ce cas sont sélectionnés par l'apprenant, il utilise cette solution et l'adapte enfin de dériver une solution au problème. L'adaptation dans cet exemple se fait par remplacement le $(\ln x)$ par (T) et (2) par (n) dans la solution du cas.

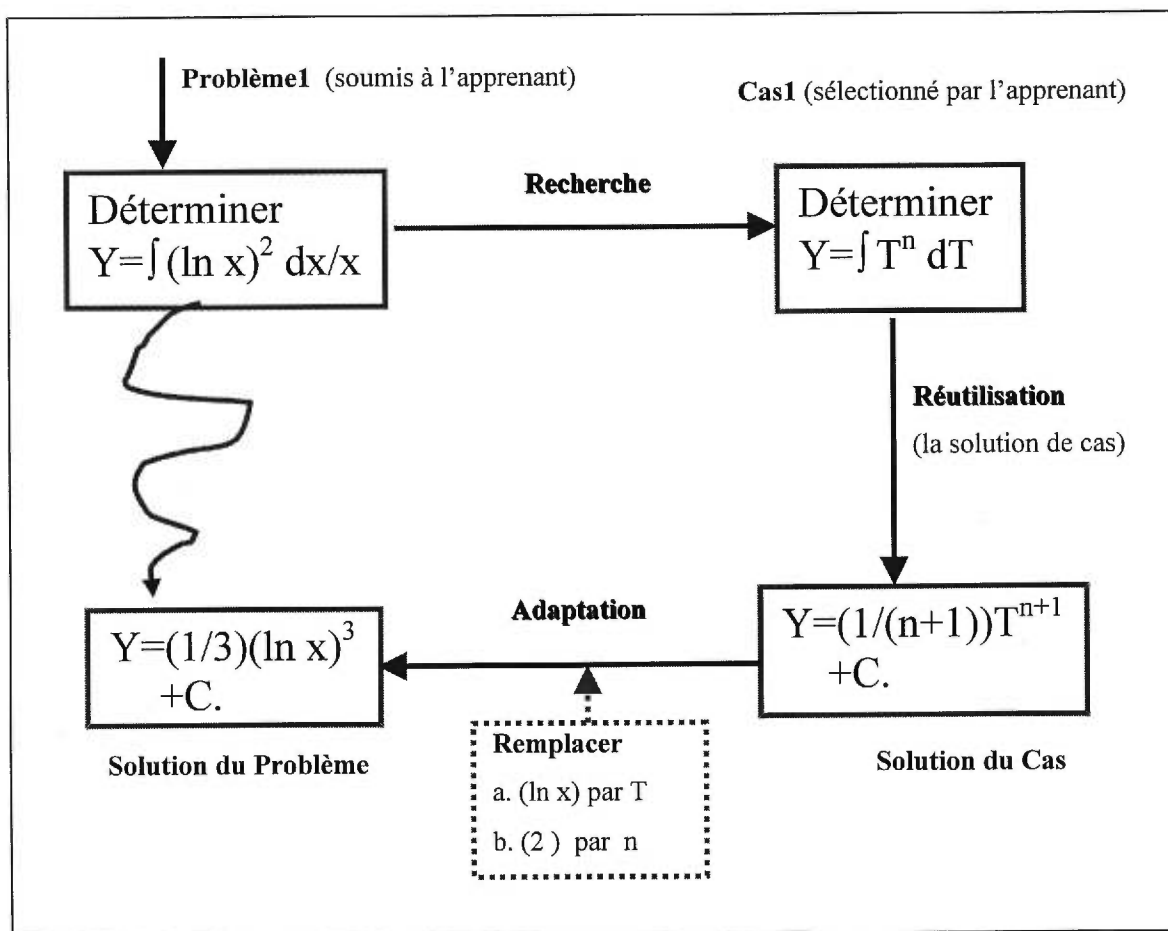


Figure 6. 4 : Un scénario de la résolution de problème par l'apprenant (en AS mode)

- La solution adaptée est évaluée par le système qui contrôle la compatibilité de la solution de l'étudiant avec celle de l'expert.
- Si l'adaptation est réussie, la solution résultante est enregistrée dans le modèle de l'apprenant.

- Quand aucun cas antérieur approprié n'est recherché, ou que la réponse de l'étudiant n'est pas bonne, ou que le problème n'est pas résolu par l'étudiant dans le temps disponible, le problème est enregistré dans le modèle de l'apprenant comme problème non résolu.

La Figure 6.5, identifie les informations d'entrée (et nécessaires) pour cette analyse ainsi que les informations résultant de cette analyse.

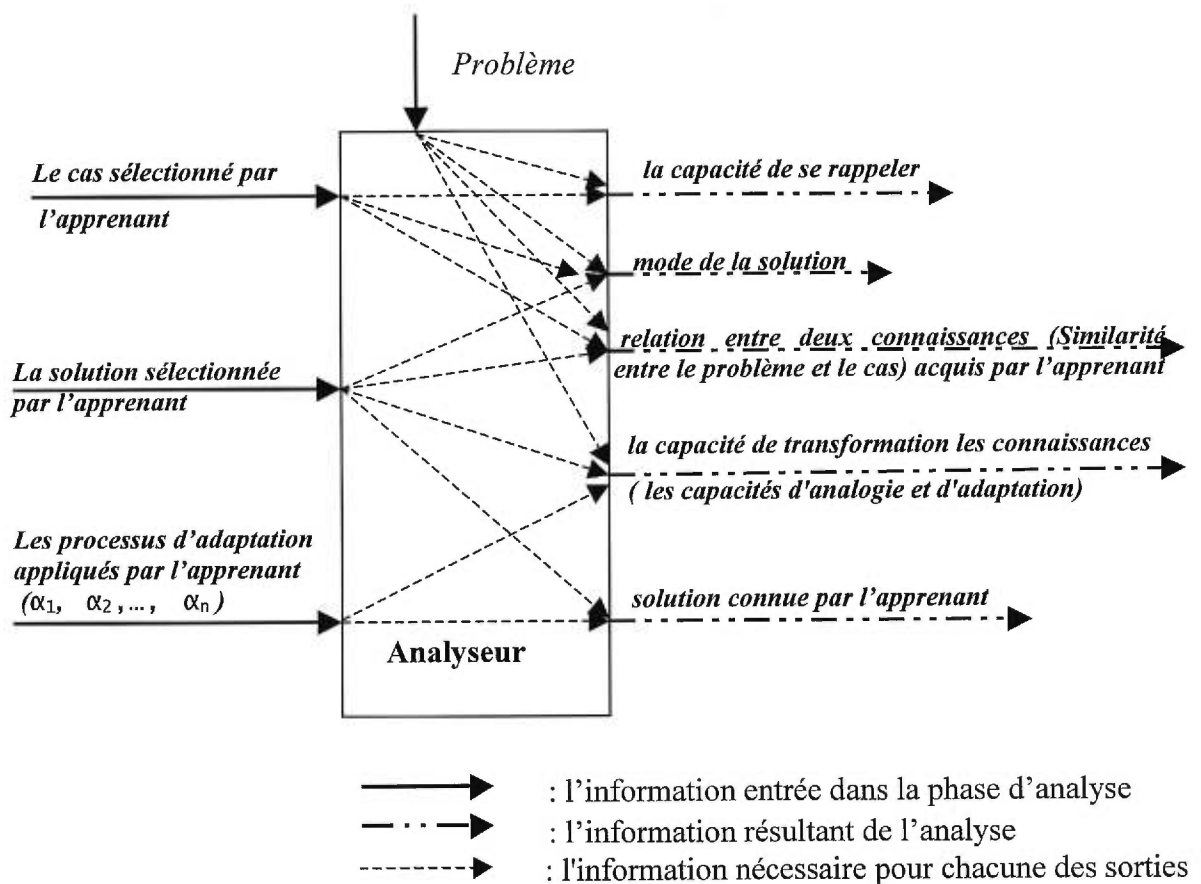


Figure 6.5 : L'information nécessaire pour l'analyse du raisonnement et l'information résultant de cette analyse (*AS mode*)

6.4 Modélisation des étudiants en *US mode*

Le modèle de l'interaction pour *US mode* est de beaucoup différent de ce qui est adressé en *AS mode*. Ici, le processus d'adaptation est exécuté en forme dérivative (voir 6.2). Dans l'adaptation de la forme dérivative nous réutilisons les règles ou les formules qui ont produit la solution initiale pour produire une nouvelle solution au problème actuel. L'aspect le plus important qui doit être considéré ici est que les débutants et les étudiants avancés raisonnent de manière différente.

Comme discuté dans [Frasson et Kaltenbach, 1991], pour résoudre un problème, les débutants essaient de l'associer à une liste de cas définis et identifiés afin d'adapter leurs solutions. Cependant, ils renvoient à l'exemple très souvent pour résoudre un problème donné, alors que les étudiants avancés procèdent en utilisant quelques références spécifiques aux exemples. En fait, les étudiants avancés relient le nouveau problème avec les problèmes précédemment résolus, en utilisant leur propre base de connaissances dans laquelle ils sont confiants et alors ils essaient de s'expliquer à eux-mêmes la solution d'un problème.

La Figure 6.6 met en évidence le processus de modélisation de l'apprenant en *US mode*. Nous décrivons maintenant les étapes suivies par le système lors de cette modélisation [Shiri *et al.*, 1998d] :

- 1) *Sélection d'un problème (par le système)* : tout d'abord le système choisit un problème pour le proposer à l'étudiant.
- 2) *Construction d'une solution (par l'apprenant)* : ensuite, l'apprenant est appelé à construire une solution. On s'attend à ce qu'il puisse se rappeler les scénarios de résolutions des problèmes qu'il a déjà résolus; puis en appliquant essentiellement les mêmes démarches de résolution, les mêmes stratégies, il peut construire une solution pour le problème actuel.

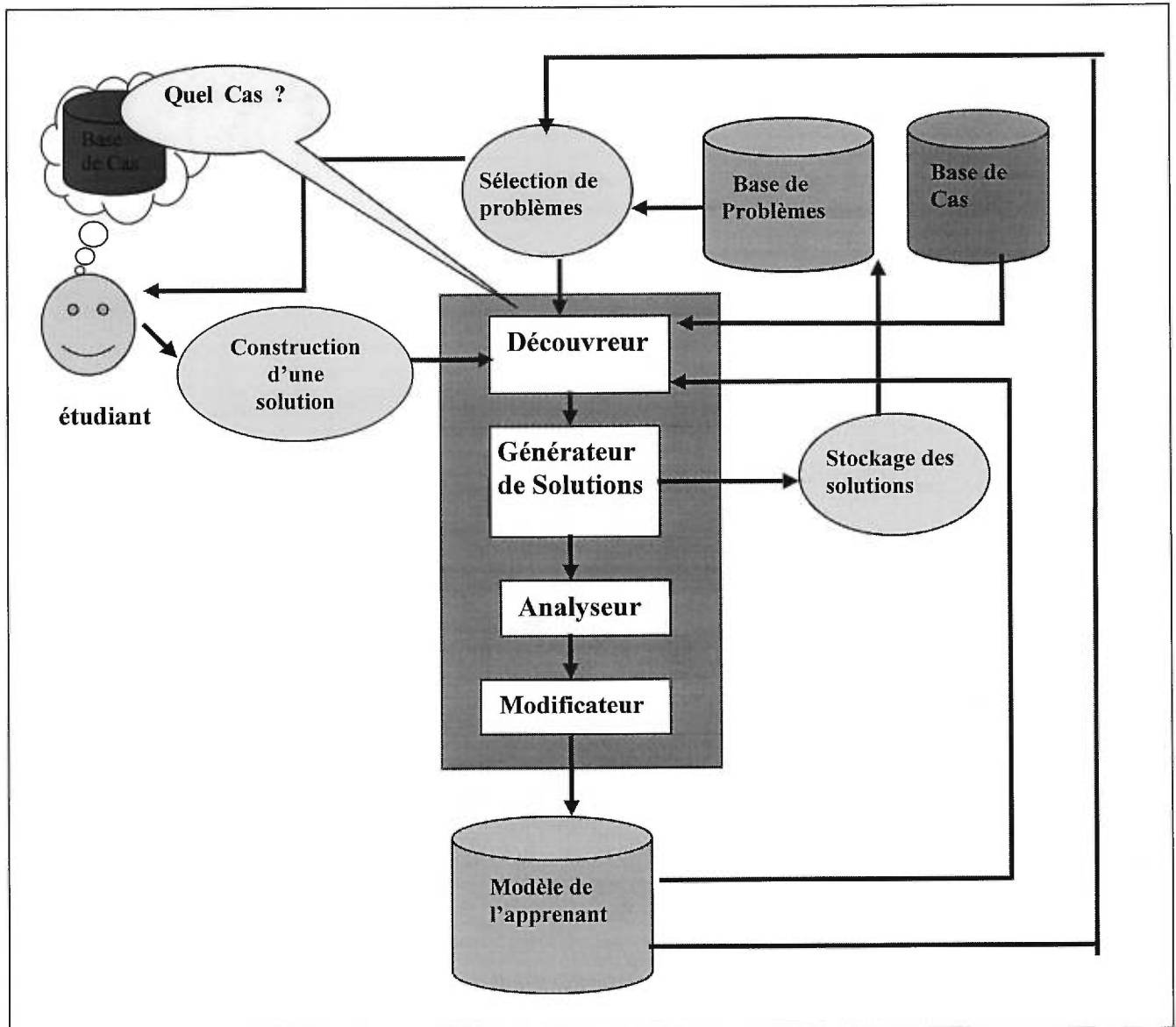


Figure 6. 6 : Le processus de modélisation de l'apprenant en US mode

- 3) *Extraire le cas (et la solution) choisi par l'apprenant* : le *découvreur* détermine ensuite le cas choisi par l'apprenant. Il utilise une heuristique basée sur le modèle de l'apprenant (discuté dans la section 6.5.1) pour déterminer ce cas.
- 4) *Génération et stockage des solutions*. Comme dans *AS mode* le *Générateur de Solutions* produit toutes les solutions possibles au problème (en utilisant le même cas

choisi par l'étudiant) afin d'être comparées à celle de l'apprenant. Ces solutions sont enregistrées dans le système pour une future référence.

- 5) *Analyse de la réponse de l'apprenant* : une fois que l'apprenant a construit une solution, l'*analyseur* détermine parmi les solutions existantes dans l'espace de solutions du cas choisi par l'apprenant, la solution qui est la plus proche de la solution de l'apprenant (pour plus de détails voir l'exemple 6.5.2). Cette solution est comparée avec celle de l'étudiant. De cette manière, il peut donc :
1. *extraire le mode de raisonnement* : en utilisant les informations existant dans la représentation des cas, le système peut extraire le mode de raisonnement de l'apprenant (voir la Figure 6.2).
 2. *détecter les erreurs de l'apprenant* : les erreurs de l'apprenant peuvent être détectées par comparaison de la solution de l'apprenant avec la solution obtenue (il s'agit de la solution la plus proche de celle de l'apprenant, déterminée par le système (voir 6.5.2)).
- 6) *Mise à jour du modèle de l'apprenant* : le système utilise enfin les résultats des activités de l'analyseur (voir 6.5.4) pour mettre à jour le modèle de l'apprenant.

6.5 Analyse du raisonnement en US mode

Un problème important est comment détecter maintenant la solution de l'étudiant ? Nous avons nécessairement besoin d'un modèle correct pour pouvoir tester la validité de la solution proposée par l'apprenant. L'analyse de la solution de l'apprenant se fait en général en fonction de celle de l'expert.

En fait, le principe de l'analyse est de pouvoir reconnaître, dans la réponse donnée par l'apprenant, ses modes de raisonnement, ses erreurs, etc.

Une voie commune pour accomplir cette tâche est donc de comparer la solution de l'étudiant à une liste de toutes les solutions possibles (de l'expert). Mais, étant donné que l'espace de recherche qui doit être traité est considérablement grand, une technique de recherche heuristique est normalement exigée pour le parcourir. Nous pouvons consulter le modèle de l'apprenant pour déterminer cette heuristique. La consultation consiste à

déterminer le nombre de cas semblables (au problème actuel) ayant été précédemment résolus par l'étudiant ainsi que les modes d'inférences utilisés dans ces solutions.

Nous allons maintenant présenter un exemple pour illustrer cette analyse et le genre de résultats que nous obtenons lors de l'analyse de la réponse de l'apprenant.

6.5.1 Exemple 1

La Figure 6.7 présente un problème que l'apprenant devra résoudre. Ce problème est similaire aux Cas 1, Cas 3, Cas 4 et Cas 8.

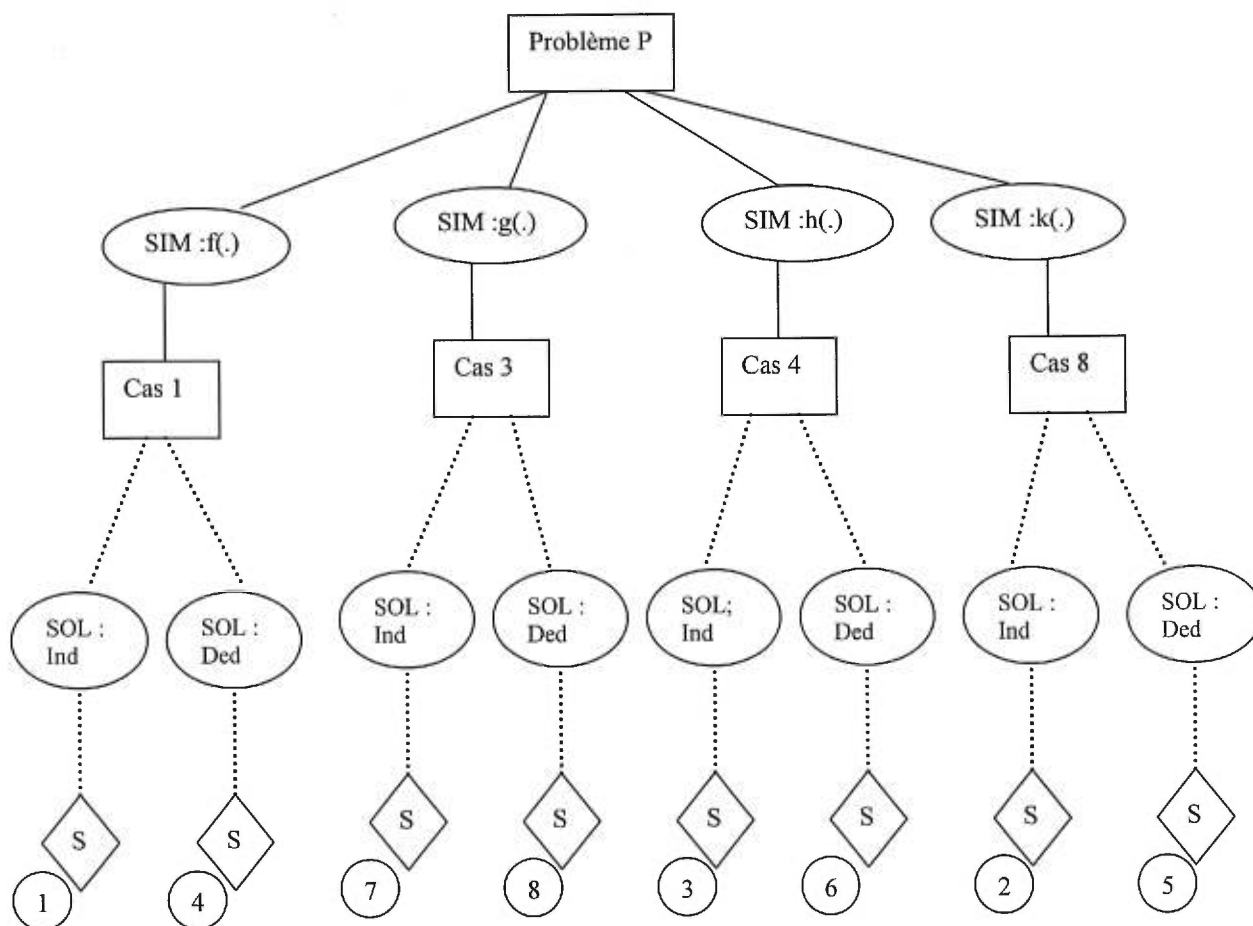


Figure 6. 7 : L'ordre à suivre en cours de recherche.

Le tableau suivant (Tableau 6.1) illustre les problèmes (similaires à ces cas, c.-à-d. aux cas 1, cas 3, cas 4, et cas 8) précédemment résolus par l'apprenant et le tableau 6.2 illustre le nombre total de ces problèmes et ainsi que le nombre de problèmes résolus de façon *inductive* ou *déductive*.

Les Cas similaires au problème P	Modes des solutions	Les problèmes précédemment résolus par l'apprenant en utilisant la solution (<i>inductive</i> ou <i>déductive</i>) du Cas i	Le nombre de problèmes similaires
Cas 1	Inductive	P ₁ , P ₃ , P ₄ , P ₇ , P ₈ , P ₁₄ , P ₁₅ , P ₂₁	8
	Déductive	P ₅ , P ₁₀ , P ₁₁ , P ₁₇	4
Cas 3	Inductive		0
	Déductive		0
Cas 4	Inductive	P ₁₂ , P ₁₆ , P ₂₀	3
	Déductive	P ₂	1
Cas 8	Inductive	P ₆ , P ₁₃ , P ₁₈ , P ₂₂	4
	Déductive	P ₉ , P ₁₉	2

Tableau 6. 1 : Les problèmes précédemment résolus par l'apprenant

Le nombre de problèmes résolus en utilisant la solution inductive	Le nombre de problèmes résolus en utilisant la solution déductive	Le nombre total de problèmes
15	7	22

Tableau 6. 2 : Les nombre de problèmes résolus par l'apprenant (en utilisant la solution *inductive* ou *déductive*)

Le Tableau 6.3, donne un exemple de calcul des valeurs heuristiques et la démarche à suivre dans le processus de recherche. Dans cet exemple : $I_m = N_{t_m} / N$, où N_{t_m} est le nombre de (tous) problèmes qui ont déjà été résolus par l'étudiant en utilisant la solution du mode m (inductif ou déductif) et N est le nombre de tous les problèmes résolus.

Les Cas similaires au problème P	Cas 1		Cas 3		Cas 4		Cas 8	
$N_m(C_i)$: Le nombre de problèmes similaires au (Cas i), précédemment résolus par l'apprenant en utilisant la solution du mode m (<i>inductive</i> ou <i>déductive</i>)	Sol.	Sol.	Sol.	Sol.	Sol.	Sol.	Sol.	Sol.
	Ind.	Ded.	Ind.	Ded.	Ind.	Ded.	Ind.	Ded.
	8	4	0	0	3	1	4	2
Les valeurs heuristiques associées aux solutions des cas : $H_m(C_i) := N_m(C_i) \times I_m$	5.45	1.3	0	0	2	0.3	2.7	0.6
L'ordre à suivre en cours de recherche	1	4	7	8	3	6	2	5

Tableau 6. 3 : Calcul des valeurs heuristiques

A titre d'exemple, la valeur heuristique associée à la solution *inductive* de Cas-1 est calculée comme suit :

$$H_m(\text{Cas-1}) = N_m(\text{Cas-1}) \times I_m = 8 \times (N_{t_m} / N) = 8 \times (15 / 22) = 5.45, \quad \text{ici } m = \text{inductive.}$$

Enfin, dans notre exemple, le premier espace de solutions choisi dans le processus de recherche est donc l'espace de solution inductive de Cas 1, puisque le nombre de problèmes résolus par l'étudiant semblables à ce cas et en utilisant la solution inductive est plus grand que pour les autres; le deuxième est l'espace de solution inductive de Cas 8, et ainsi de suite (voir le Tableau 6.3 et la Figure 6.7).

6.5.2 Exemple 2

Considérons le problème suivant proposé à l'apprenant :

$$\text{Problème : montrez que : } \frac{\partial}{\partial x} [(x^n + k)(x^n - k)] = 2nx^{2n-1}.$$

Supposons maintenant que la solution de l'apprenant soit la suivante :

Solution de l'apprenant :

$$\begin{aligned} & \frac{\partial}{\partial x} [(x^n + k)(x^n - k)] \\ &= \frac{\partial}{\partial x} [x^{2n} - k^2] \\ &= \frac{\partial}{\partial x} (x^{2n}) - \frac{\partial}{\partial x} (k^2) \\ &= \frac{\partial}{\partial x} (x^{2n}) \\ &= 2nx^{2n-1} \end{aligned}$$

En se basant sur la solution proposée par l'apprenant, le système peut déterminer dans l'espace de solutions du cas (voir 5.2.6), la solution la plus proche de celle de l'apprenant afin de l'utiliser pour évaluer et détecter cette dernière (voir la Figure 6.8).

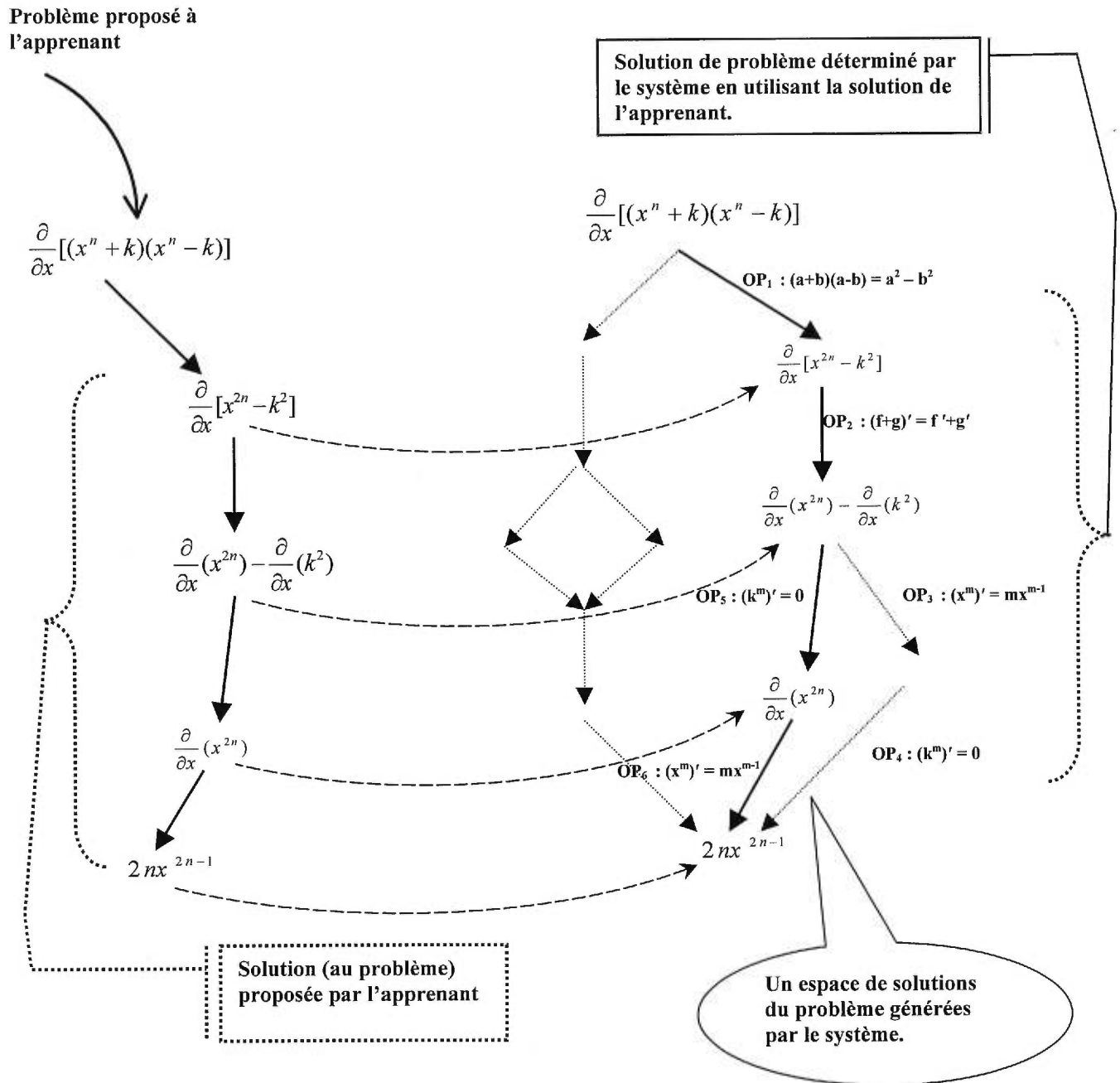


Figure 6. 8 : L'évaluation de solution de l'apprenant en *US mode*.

6.5.3 Algorithme de modélisation

L'algorithme de modélisation peut être décrit maintenant comme suit.

Algorithme :

1. ● Initialiser le modèle de l'apprenant.
2. ● Produire un problème à proposer à l'étudiant,
 - Soit $LC = \{ C_1, \dots, C_n \}$ la liste de tous les cas liés à ce problème.
3. ● En utilisant le modèle de l'apprenant, déterminer une heuristique pour le problème actuel (voir 6.5.1),
 - Appliquer la fonction heuristique sur les états de ce problème pour produire la liste des valeurs heuristiques.
4. ● Sélectionner la prochaine valeur la plus élevée V_i (commençant par le premier, pour $i=1,2,3, \dots, n$) de la fonction heuristique,
 - Soit $S^m_{C_i}$ ($C_i \in LC$ et $m = \textit{inductif}$ ou $\textit{déductif}$) une solution associée à cette valeur;
5. ● Soit S la solution de l'étudiant au problème actuel,
 - Si S apparie une des solutions de $S^m_{C_i}$, alors présumer que l'étudiant a résolu le problème actuel,
 - Mettre à jour le modèle de l'apprenant et allez à 8.
6. ● Si S n'apparie aucune solution du $S^m_{C_i}$, alors demander à l'étudiant si le cas qu'il a appliqué est similaire à celui découvert par le système,
 - S'il est différent, aller à 4.
7. ● Autrement, considère qu'il y a des erreurs dans la solution de l'étudiant,
 - Prendre la solution du $S^m_{C_i}$ qui convient le mieux à la solution de l'étudiant (voir 6.5.2),
 - Utiliser cette solution pour détecter la solution de l'apprenant,
 - Mettre à jour le modèle de l'apprenant.
8. ● Si la session est terminée ou l'étudiant demande l'arrêt, alors finir la session,
 - Autrement aller à 2.

6.5.4 Les connaissances utilisées pour l'analyse et les connaissances émises de cette analyse

Nous devons maintenant déterminer les connaissances utilisées pour l'analyse du raisonnement de l'apprenant. Comme nous le montrons à la Figure 6.9, les informations d'entrée et nécessaires pour cette analyse (en *US mode*) sont : le modèle actuel de l'apprenant, le problème qui lui a été soumis, et la solution qu'il propose pour ce problème. L'information émise lors de cette analyse peut être : le cas similaire et la solution utilisée par l'apprenant, le mode de la solution, la capacité de se rappeler, la relation entre deux connaissances (similarité entre problème et cas), la capacité de

transformation des connaissances (les capacités d'analogie et d'adaptation) et la solution connue par l'apprenant.

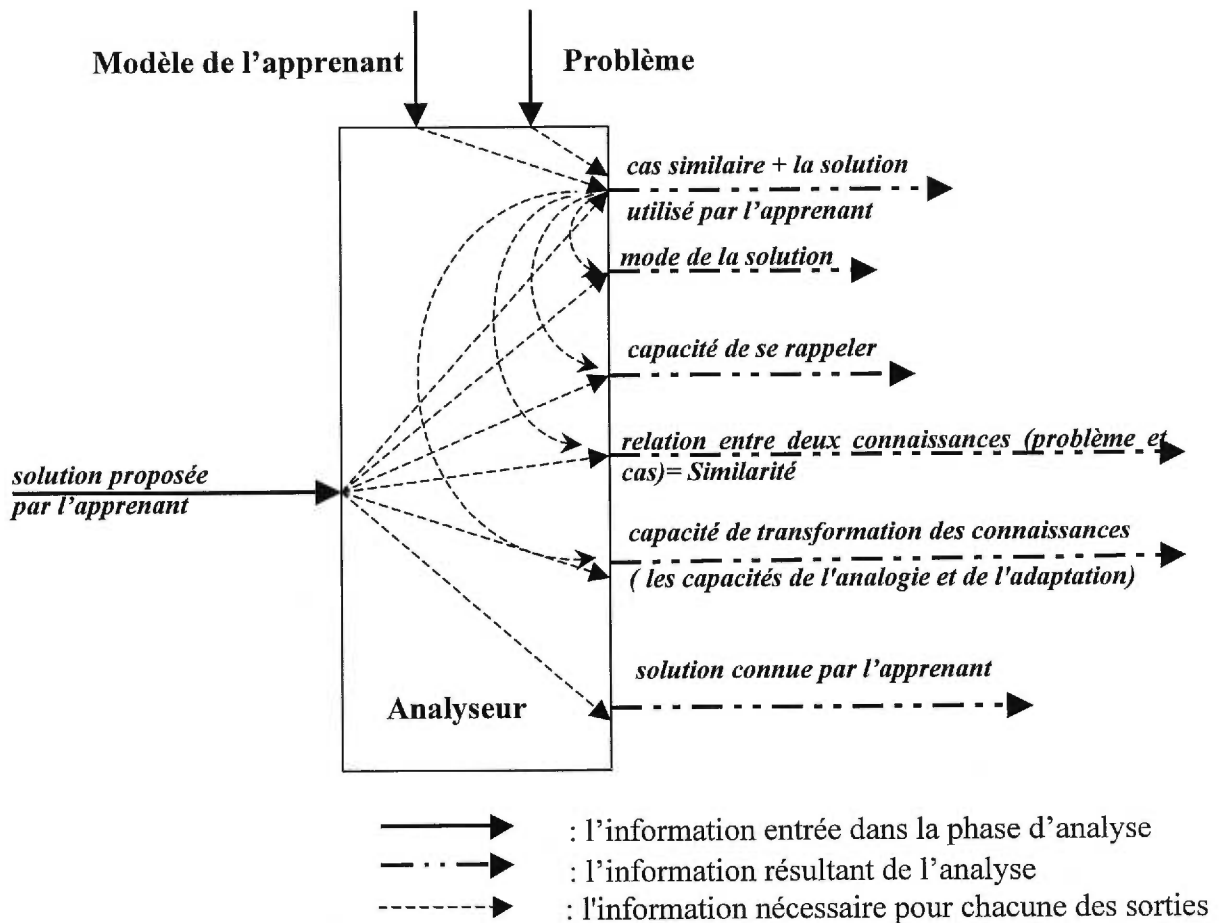


Figure 6. 9 : L'information nécessaire pour l'analyse du raisonnement et l'information émise lors de cette analyse (*US mode*)

6.6 Formation du modèle de l'apprenant

Une fois l'analyse de la réponse de l'apprenant complétée, le *Modificateur* utilise les résultats des activités de l'analyseur pour mettre à jour le modèle de l'apprenant (les modèles cognitif et inférentiel).

6.6.1 Mise à jour du modèle cognitif de l'apprenant

Le *Modificateur* utilise les informations suivantes pour mettre à jour le modèle cognitif de l'étudiant :

- 1) *le niveau de connaissances de l'étudiant en terme d'échelle de Gagné (1985).*
 Plusieurs travaux de recherche ont tenté de modéliser l'apprentissage, en particulier ceux de Lansky (1975) et de Gagné *et al.* (1992). Ce dernier paraît être l'un des plus explicites. La théorie de Gagné *et al.* (1992) nous offre des conditions d'apprentissage (pour chaque catégorie de connaissances) par lesquelles un individu acquiert de nouvelles connaissances. Selon lui, l'apprentissage est un ensemble d'événements externes à l'étudiant qui sont conçus pour supporter les processus internes d'apprentissage (correspondant à ce qui se passe dans la tête de l'apprenant). En utilisant ces conditions il est déterminé plusieurs niveaux d'apprentissage correspondant aux étapes principales du processus cognitif (Tableau 6.4). Les conditions d'acquisition caractérisent le type d'événements externes à activer afin d'atteindre le niveau correspondant d'apprentissage. Un étudiant atteint donc un certain niveau si les conditions correspondantes d'acquisition sont satisfaites. Supposons maintenant que l'apprenant est au niveau 6 avant la résolution de problèmes, nous l'augmenterons alors au niveau 7 quand le problème est résolu, et nous le ramènerons au niveau 5 quand le problème demeure non résolu;

Niveau d'apprentissage	Condition d'acquisition
0 indétermination	L'état inconnu
1 acceptation	Pré-requis disponibles
2 motivation	Motivation affectée et objectifs donnés
3 attention	Attention dirigée et rappel stimulé
4 présentation	Conseils d'étude fournis
5 déclenchement	Des exercices simples
6 intégration	Des exercices plus complexes
7 généralisation	Conservation et transfert de la connaissance

Tableau 6. 4 : L'échelle de Gagné

- 2) *les capacités d'analogie et d'adaptation* (de l'étudiant) pour la résolution de problèmes;
- 3) *et les solutions connues* par l'étudiant pour ces problèmes.

Nous appliquons le formalisme GCR pour représenter ce composant [Shiri *et al.*, 1998g] qui contient :

1. *tous les problèmes résolus par l'étudiant*. Nous devons noter que chaque problème est attaché à une solution qui est identifiée par l'étudiant. La solution est jointe au problème par une relation du type SOLUTION (SOL); le référent de cette relation sera un triplet $(f(\cdot), C_i, R_m)$ où C_i est le cas choisi par l'étudiant, $f(\cdot)$ est la fonction de similitude entre le problème et ce cas, et R_m est le mode de raisonnement de la solution (voir la Figure 6.10);
2. *tous les problèmes soumis et non résolus*. Il sera utile d'ajouter les problèmes non résolus à ce modèle, parce qu'ils indiquent les points de difficulté de l'étudiant.

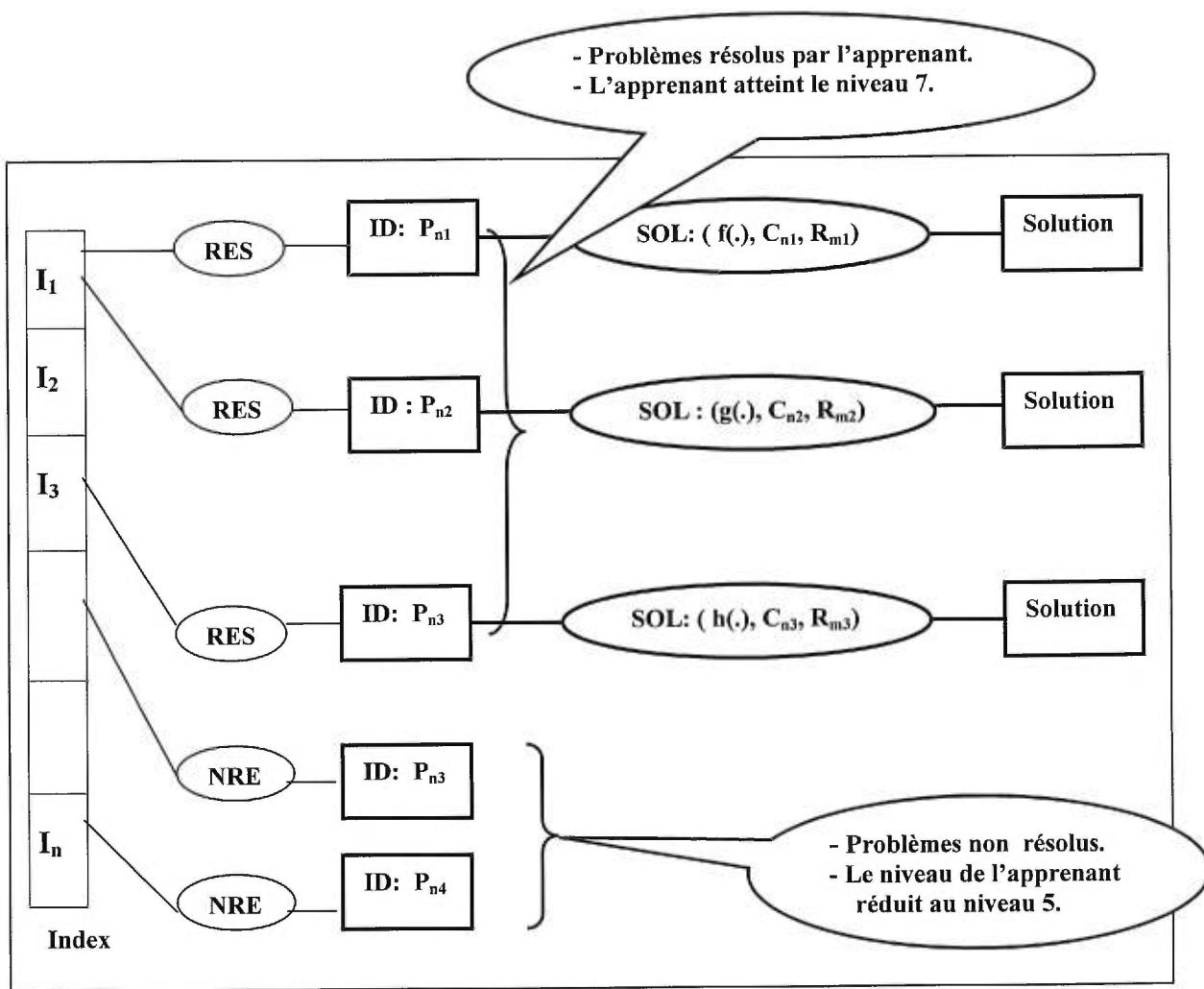
6.6.2 Mise à jour du modèle d'inférences

En se basant sur plusieurs sessions d'exercices, le système peut conclure que généralement, l'apprenant utilise tel type de raisonnement. Par exemple, les résultats issus de cette étape peuvent être les suivants :

- l'apprenant utilise tel type de raisonnement plutôt que tel autre; par exemple, il utilise le raisonnement *déductif* plutôt que le raisonnement *inductif*;
- il utilise parfois l'un et parfois l'autre;
- il utilise généralement un type de raisonnement et rarement l'autre; par exemple il utilise seulement le raisonnement *déductif* dans ses résolutions de problèmes.

Formellement, le modèle d'inférences peut être défini comme suit :

$$EP_C = \{ P : P \in R_P(C), P \text{ est un problème résolu par un étudiant} \}$$



RES : Relation du type RÉSolu
 NRE : Relation du type Non RÉsolu

Figure 6. 10 : Le modèle cognitif de l'apprenant.

soit l'ensemble de tous les problèmes résolus par l'étudiant, où $R_p(C)$ est l'ensemble de tous les problèmes destinés à l'évaluation du concept C (voir 5.5.3). Nous définissons le mode d'inférences pour (le concept) C comme :

$$I_m(C) = N_{t_m}(EP_C) / N(EP_C)$$

où m est le mode de la solution (*inductif* ou *déductif*); $Nt_m(EP_C)$ est le nombre de (tous) problèmes ($P \in EP_C$) déjà résolus par l'étudiant en utilisant la solution du mode m ; et $N(EP_C)$ est le nombre de tous problèmes ($P \in EP_C$) résolus. Puis, nous définissons :

Modèle d'inférences : = $\{ I_m(C) : \text{pour tous concepts } C, \text{ et tous } m = \textit{inductif} \text{ ou } \textit{déductif} \}$

Nous allons maintenant énumérer quelques avantages de notre approche.

6.7 *Les avantages*

L'approche que nous avons suivie présente certains points forts :

- Le raisonnement analogique (aussi bien le RBC) est une façon naturelle de penser pour l'être humain.
- Elle met en valeur l'état de connaissances de l'apprenant par rapport à toutes les solutions existantes du problème proposé.
- Elle nous permet de modéliser non seulement l'état de connaissances de l'apprenant, mais aussi bien les raisonnements de ce dernier.
- Elle est assez générale pour pouvoir être adaptée à plusieurs domaines d'applications.

Pour montrer la validité de notre approche nous allons présenter les prototypes que nous avons développés dans le chapitre suivant.

6.8 *Conclusion*

Notre approche nous permet de préciser non seulement l'état de connaissances de l'apprenant, mais aussi les connaissances du modèle d'inférences telles que les modes de ses raisonnements. La représentation de ces derniers est nécessaire afin qu'un modèle de l'apprenant puisse être considéré comme complet. C'est l'application du formalisme GCR pour la représentation de connaissances qui nous a permis d'extraire ce dernier.

À cause des variétés de niveaux des apprenants, nous avons considéré deux modes d'interaction, c.-à-d., *AS mode* et *US mode* pour réaliser notre approche. Le premier mode

prend en considération les actions que l'apprenant va entreprendre pour arriver à la solution et modélise l'état de ses connaissances en fonction de ses actions, tandis que le deuxième mode analyse sa solution par rapport à une solution déterminée en utilisant une heuristique basée sur le modèle de l'apprenant, pour accomplir cette tâche.

Chapitre 7

Implémentation

7.1 *Introduction*

Jusqu'à présent, nous avons présenté le modèle théorique de notre approche. Ce chapitre va présenter les prototypes des systèmes que nous avons développés et implantés pour concrétiser et tester nos idées théoriques. Nous avons développé deux prototypes de STI appelés SARA et SARA-II qui sont réalisés sur des stations de travail de type SUN. Nous avons choisi Java comme langage pour implanter ces systèmes. Ce dernier possède en effet un environnement de programmation facile et interactif et s'applique bien aux domaines de prototypage rapide d'application. De plus, il permet de construire des interfaces graphiques, et des applications de simulation.

Nous décrivons dans ce chapitre, d'abord l'état de réalisation du prototype SARA qui est développé pour tester le fonctionnement de *AS mode*. Nous décrivons ensuite le prototype SARA-II que nous avons réalisé pour tester le cœur de la théorie proposée dans les chapitres précédents. Puis, nous présentons certains aspects importants de ces systèmes comme par exemple, l'architecture, le fonctionnement, un exemple d'interaction entre le système et l'apprenant, les différences fondamentales entre ces deux systèmes, etc.

7.2 *Le Prototype SARA*

7.2.1 Introduction

Dans cette section nous présentons l'état actuel de développement de SARA (Système d'Analyse des Raisonnements de l'Apprenant), un STI pour le domaine de la théorie des Nombres. SARA [Shiri *et al.*, 1998c] [Shiri *et al.*, 1998f] utilise une approche basée sur les cas pour construire les modèles de l'apprenant. Notre but dans ce prototype de recherche est de fournir à l'étudiant l'accès aux cas et à leurs solutions associées (*mode de solution accessible*).

SARA est organisé autour de deux bases de connaissances principales : la base de problèmes et la base de cas. L'architecture du système se compose de plusieurs composants dont nous allons montrer les fonctionnalités. Nous présentons ensuite deux manières d'utiliser le système : (1) comme système pour la modélisation de l'apprenant, et (2) comme serveur fournissant des informations pour des applications. Nous étudierons également le moyen d'établir le modèle de l'apprenant avec ce système.

Le modèle de l'apprenant construit par SARA représente deux aspects importants d'un étudiant, à savoir le *modèle cognitif* et le *modèle d'inférences*.

7.2.2 Les idées principales de notre technique de modélisation des apprenants

Le cycle de RBC est exécuté en quatre phases. D'abord, le cas le plus semblable au problème soumis par le système est recherché dans la base de cas par l'étudiant; en second lieu, il réutilise la solution de ce cas; troisièmement, la solution proposée pour le cas est mise à jour ou adaptée par cet étudiant; et finalement, la solution de l'étudiant est évaluée par le système pour réaliser le modèle d'étudiant. Ainsi, notre modèle est organisé autour de deux bases de connaissance principales. La première est une **base de problèmes** qui est un ensemble de tous les problèmes enregistrés dans le système. Un problème se compose de trois éléments : l'information classée, la description du problème et les cas similaires.

La seconde est une **base de cas** qui est un ensemble de tous les cas enregistrés dans le système. Un cas est la représentation d'une expérience précédente (problème résolu); de même, il se compose de trois éléments : l'information classée, la description du cas et les solutions. Il est important de souligner ici que :

- Les cas et leurs solutions sont accessibles par les étudiants.
- Pour un cas donné, nous pouvons prendre deux modes de solutions : *inductif* ou *déductif*. Les modes de solutions ne doivent pas être présentés aux étudiants.
- Chaque problème est relié à tous ses cas semblables (ces relations ne doivent pas être présentées aux étudiants).
- Le formalisme GCR a été employé pour représenter les problèmes et les cas dans ce système.
- En comparant la solution de l'apprenant avec celle de l'expert le système évalue l'exécution de l'apprenant.

7.2.3 L'architecture de SARA

Cette section fournit une description de l'architecture du prototype SARA. Sur la Figure 7.1, on peut voir que cette architecture se compose de six composants, chacun encapsulant différents types de tâches. Nous décrivons brièvement la fonctionnalité de chaque composant et de leurs relations avec d'autres composants. Les composants principaux sont les suivants :

1. Interface

L'interface est responsable du support des interactions entre l'étudiant et le système. Dans cette interface, la sortie (c.-à-d. les explications, les problèmes, les cas, les solutions, etc.) apparaissent en tant que phrases en langage naturel. Cette interface se compose de quatre sections (Figure 7.2) :

- a) *L'information du système*. La section supérieure de l'écran contient deux boutons à utiliser pour l'information de système. Si l'étudiant a besoin de quelques informations sur le système, il peut cliquer le bouton « About.. »; les informations sur SARA apparaîtront alors. S'il a besoin d'informations sur la façon d'actionner le système lui-

même, en cliquant le bouton « Help » on atteindra le texte d'aide nécessaire dans une zone de dialogue.

- b) *L'environnement d'adaptation.* La section centrale de l'écran fournit un environnement où un étudiant peut résoudre les problèmes. Cette zone contient un ensemble de boutons qui facilitent l'adaptation et l'entrée dans le système.
- c) *Boutons de commande.* La section inférieure gauche de l'écran contient les boutons de commande du système.
- d) *Autre services.* La section inférieure droite de l'écran contient un bouton à employer par l'application pour quelques services (voir la sous-section 7.2.4.3).

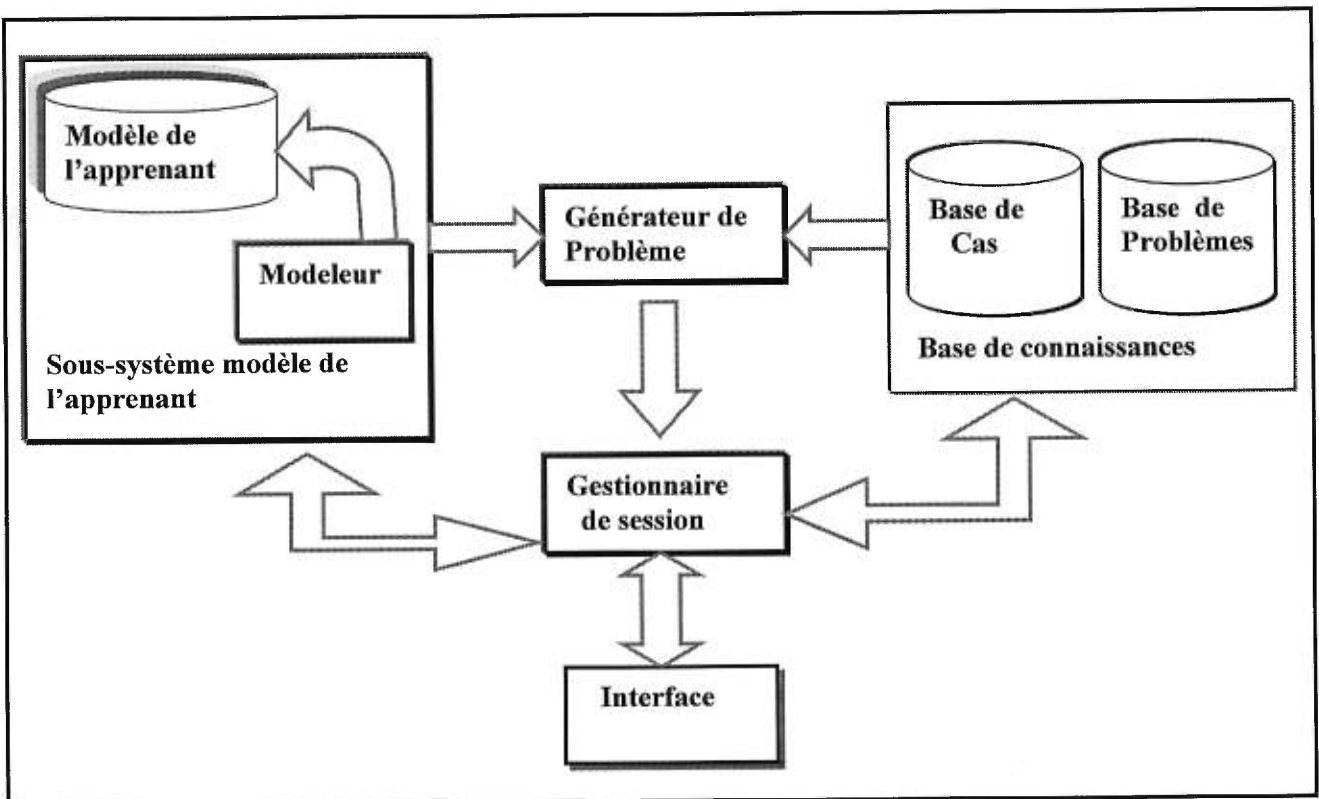


Figure 7. 1 : L'architecture de SARA.

2. Le sous-système modèle de l'apprenant

Il se compose de deux parties : un *modeleur* et le *modèle de l'apprenant*.

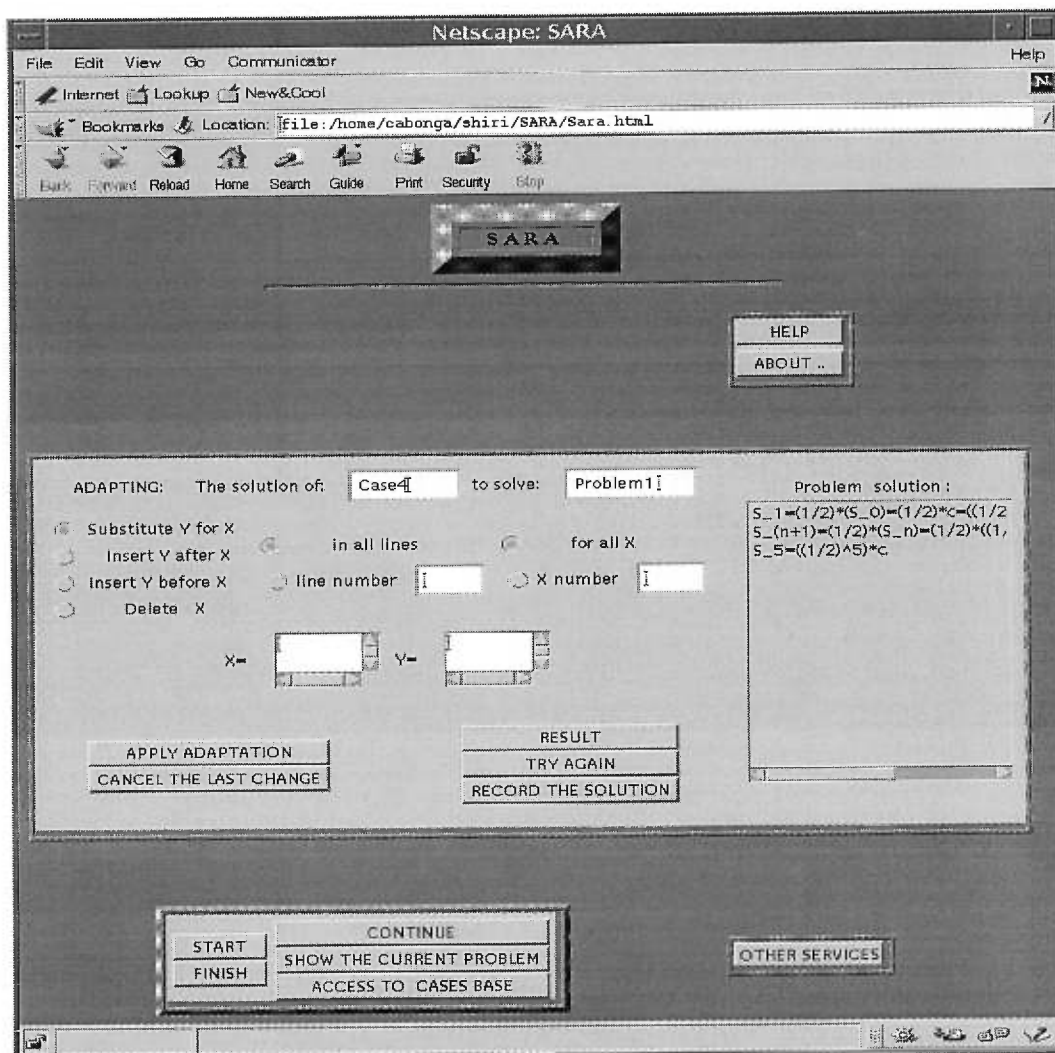
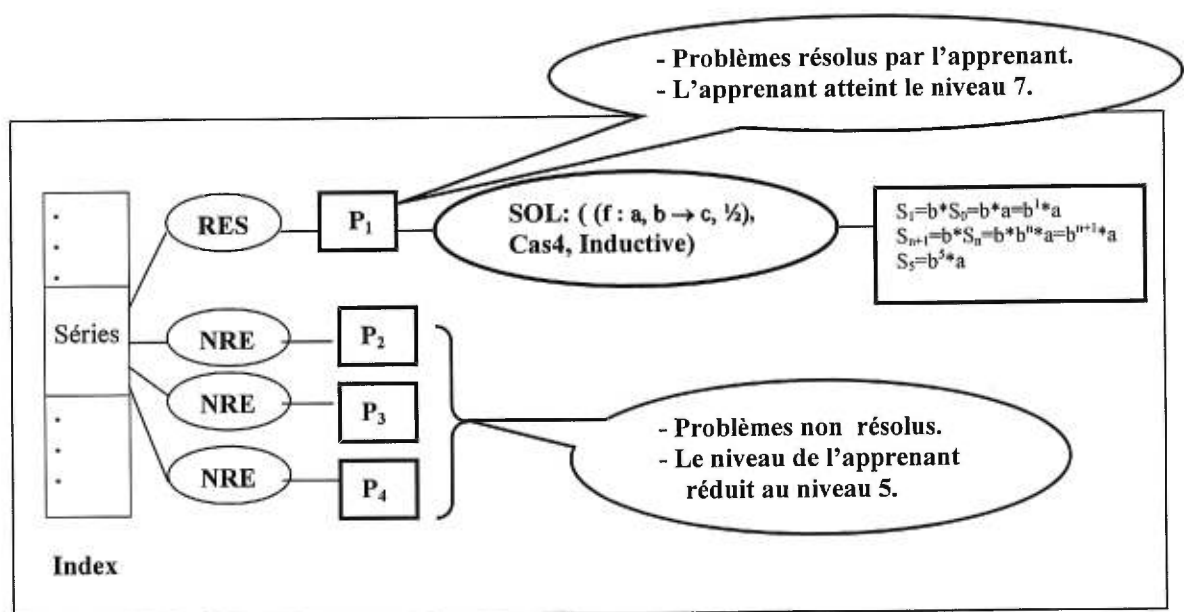


Figure 7. 2 : L'interface de SARA

- a. *Modeleur*. C'est un composant du système global qui est responsable de la modélisation des étudiants.

Les trois types d'information qui peuvent être identifiés comme entrées à ce composant sont le problème, le cas choisi par l'étudiant, et la solution de l'étudiant. L'information qui peut être identifiée comme sortie est le modèle de l'apprenant. Les deux composants principaux du *Modeleur* sont le *vérificateur* et l'*analyseur*. En général, ces deux composants sont développés pour aider le diagnostic de la solution de l'étudiant. En fait, leurs rôles sont de montrer, respectivement :

- L'étudiant a-t-il recherché un cas utile pour son raisonnement ?
 - L'adaptation est-elle réussie ?
- b. *Modèle de l'apprenant.* C'est le composant du système qui contient les informations sur un étudiant. Le modèle de l'apprenant dans SARA se compose de deux parties principales :
- *Modèle cognitif.* Ce composant contient divers types d'information liés à la connaissance d'un étudiant, telle que le niveau de la connaissance (en terme d'échelle de Gagné (1985), les capacités (d'analogie et d'adaptation), et les solutions connues par cet étudiant. La représentation de ce composant est basée sur le modèle décrit dans [Shiri *et al.*, 1998c] et dans le chapitre 6. La Figure 7.3 montre un modèle cognitif de l'apprenant dans SARA.



RES : Relation du type RÉSolU

NRE : Relation du type Non RÉsolU

Figure 7. 3 : Le modèle cognitif de l'apprenant dans SARA

- *Modèle d'inférences.* En général, ce modèle représente les préférences de l'étudiant pour quelques attributs, tels que des exemples, des notes, des modes de raisonnement (*inductif* ou *déductif*), etc. Cette information est employée par

le système afin de choisir le modèle de présentation et de fournir l'explication ou l'aide offerte à l'étudiant. Comme nous avons montré dans [Shiri *et al.*, 1998a], le système peut implicitement obtenir cette information lors de situations de résolution de problèmes. Présentement, ce modèle dans SARA représente seulement le mode de raisonnement de l'étudiant

3. Générateur de problèmes

Le générateur des problèmes est un composant très important de SARA. Il produit dynamiquement des problèmes (à partir de la *base de problèmes*) à proposer à l'étudiant. La génération des problèmes est basée sur les éléments du modèle d'étudiant, et sur la situation pédagogique, tels que : le but d'enseignement, les concepts à enseigner, et les concepts sur lesquels on veut évaluer les performances de l'étudiant.

Exemple

Supposons que le concept sur lequel on veut évaluer les performances de l'étudiant est la série géométrique, le système généra alors le problème suivant (voir 5.5.3) :

Problème 3 : If $S_0 = 1$, $S_n = 3^0 + 3^1 + \dots + 3^n$, for all $n \geq 0$; show that $S_4 = (3^5 - 1)/2$.

4. Base de connaissances

Elle inclut deux parties :

- a. *La base de cas*. Elle contient tous les cas (problèmes résolus). Ces cas sont accessibles à l'étudiant afin d'être utilisés pour résoudre les nouveaux problèmes.
- b. *La base de problèmes*. Elle contient tous les problèmes connus non résolus. La Figure 7.4 montre certains problèmes et cas du système.

5. Gestionnaire de session

Il est responsable du contrôle des sessions de travail. Il active des ressources, répond aux requêtes de l'apprenant, décide des changements d'activités, etc. Il communique par exemple avec l'apprenant, lui donne des tests ou lui présente les problèmes.

<p>ID: Problem 1 Statement : If $S_0 = a$, $S_{(n+1)} = b * (S_n)$, for $n \geq 0$; show that $S_5 = b^5 * a$. Similar to : Case 4 Similarity function: $F : a \rightarrow c, b \rightarrow \frac{1}{2}$</p> <p>ID: Problem 2 Statement : If $S_0 = 1$, $S_{(n+1)} = 3^n + (S_n)$, for $n \geq 0$; show that $S_5 = (3^5 - 1) / 2$. Similar to : Case 1 Similarity function: $F : 3^n \rightarrow 2^n$ for all n.</p> <p>ID: Problem 3 Statement : If $S_0 = 1$, $S_n = 3^0 + 3^1 + \dots + 3^n$, for $n \geq 0$; show that $S_4 = (3^5 - 1) / 2$. Similar to : Case 2 Similarity function: $F : \dots$</p> <p>----- ----- -----</p>	<p>ID: Case 1 Statement : If $S_0 = 1$, $S_{(n+1)} = 2^n + (S_n)$, for $n \geq 0$; show that $S_5 = (2^5 - 1) / (2 - 1)$. Solution (Ind) : $S_1 = S_0 + 2^0 = 1 + 2^0 = (2^1 - 1) / (2 - 1)$; $S_{n+1} = S_n + 2^n = ((2^n - 1) / 1) + 2^n = (2^{n+1} - 1) / 1$; $S_5 = (2^5 - 1) / (2 - 1)$. Solution(Ded) : $S_5 = S_4 + 2^4 = S_3 + 2^3 + 2^4 = S_2 + 2^2 + 2^3 + 2^4 =$ $1 + 2^2 + 2^3 + 2^4 = (2^5 - 1) / (2 - 1)$.</p> <p>----- ----- -----</p> <p>ID: Case 4 Statement : If $S_0 = c$, $S_{(n+1)} = (1/2) * (S_n)$, for $n \geq 0$; show that $S_5 = (1/2)^5 * c$. Solution(Ind) : $S_1 = (1/2) * S_0 = (1/2) * k = (1/2)^1 * k$; $S_{(n+1)} = (1/2) * S_n = (1/2) * ((1/2)^n) * k = (1/2)^{n+1} * k$; $S_5 = (1/2)^5 * k$. Solution(Ded) : $S_5 = (1/2) * S_4 = (1/2)^2 * S_3 = (1/2)^3 * S_2 = (1/2)^4 * S_1$ $= (1/2)^5 * S_0 = (1/2)^5 * c$.</p>
--	--

Figure 7. 4 : Les problèmes et les cas dans le système

6. Visualisateurs et outils

En plus de ces composants, SARA a quelques visualisateurs et outils qui sont utilisés pour différents buts.

7.2.4 L'utilisation de SARA

SARA peut être utilisée de différentes façons; nous illustrerons ici certaines de ces utilisations.

7.2.4.1 Utilisation de SARA pour la modélisation de l'apprenant

Le but principal de développer le système SARA était d'être utilisé pour créer, mettre à jour et évaluer les modèles de l'apprenant. Pour utiliser SARA de cette façon on dispose d'une interface (interface principale) et d'un outil d'extraction de cas «*Cases Retriever Tool*» :

- *Interface Principale.* Comme nous l'avons mentionné dans la section 7.2.3, le rôle de cette interface est de manipuler les entrées de l'étudiant et la sortie du système, pour fournir un environnement où un étudiant peut résoudre des problèmes en appliquant les opérateurs disponibles (par exemple : 'Substitute Y for X', 'Insert Y after X', etc.), et pour fournir les divers moyens par lesquels l'étudiant agit sur le système (voir la Figures 7.2).
- *L'outil d'extraction de cas (Cases Retriever Tool).* Quand l'étudiant est confronté à un nouveau problème, il doit rechercher dans la base un cas approprié qui convient le mieux au problème. Afin de permettre à l'étudiant de le faire, un outil qui peut renvoyer la liste des cas pour les examiner est nécessaire (voir la Figure 7.5).

7.2.4.2 Exemple d'interaction

En général, l'étudiant commence une session de travail en cliquant sur le bouton « Start ». Une fois que le système fonctionne, il doit construire un modèle individualisé de l'apprenant pour chaque nouvel étudiant. Pour cette raison, au début de la session, SARA demande à l'étudiant son nom et son mot de passe. Si l'étudiant n'a pas utilisé le système auparavant, alors un nouveau modèle de l'apprenant est produit. L'étudiant peut alors cliquer le bouton « Continue » qui mène le système à procéder à l'étape suivante, où un nouveau problème sera généré afin qu'il lui soit proposé.

Afin de résoudre ce problème, l'étudiant doit rechercher dans la base des cas (au moyen des index) afin de trouver un cas semblable au problème actuel. Cette recherche est faite à l'aide de « *Cases Retriever Tool* ». Une fois que le cas et une solution sont choisis, une copie de cette solution est automatiquement montrée par le système dans l'interface principale. Cela est nécessaire, puisque la solution du cas doit être employée par l'étudiant pour produire une nouvelle solution. En conséquence, l'étudiant agit comme un éditeur de solutions. Tout ce qu'il doit faire est alors d'adapter la solution du cas aux besoins du problème actuel.

L'adaptation de cas est fournie en créant les fonctions d'adaptation qui ajustent des valeurs basées sur la différence entre le cas recherché et le problème courant. Actuellement SARA permet ces formes d'adaptation : insérer un élément nouveau dans la

solution du cas, effacer un élément d'elle, substituer certains éléments à d'autres, ou transformer une certaine partie de la solution de cas. Les boutons disponibles ('Substitute Y for X', 'Insert Y after X', etc.) rendent l'adaptation facile.

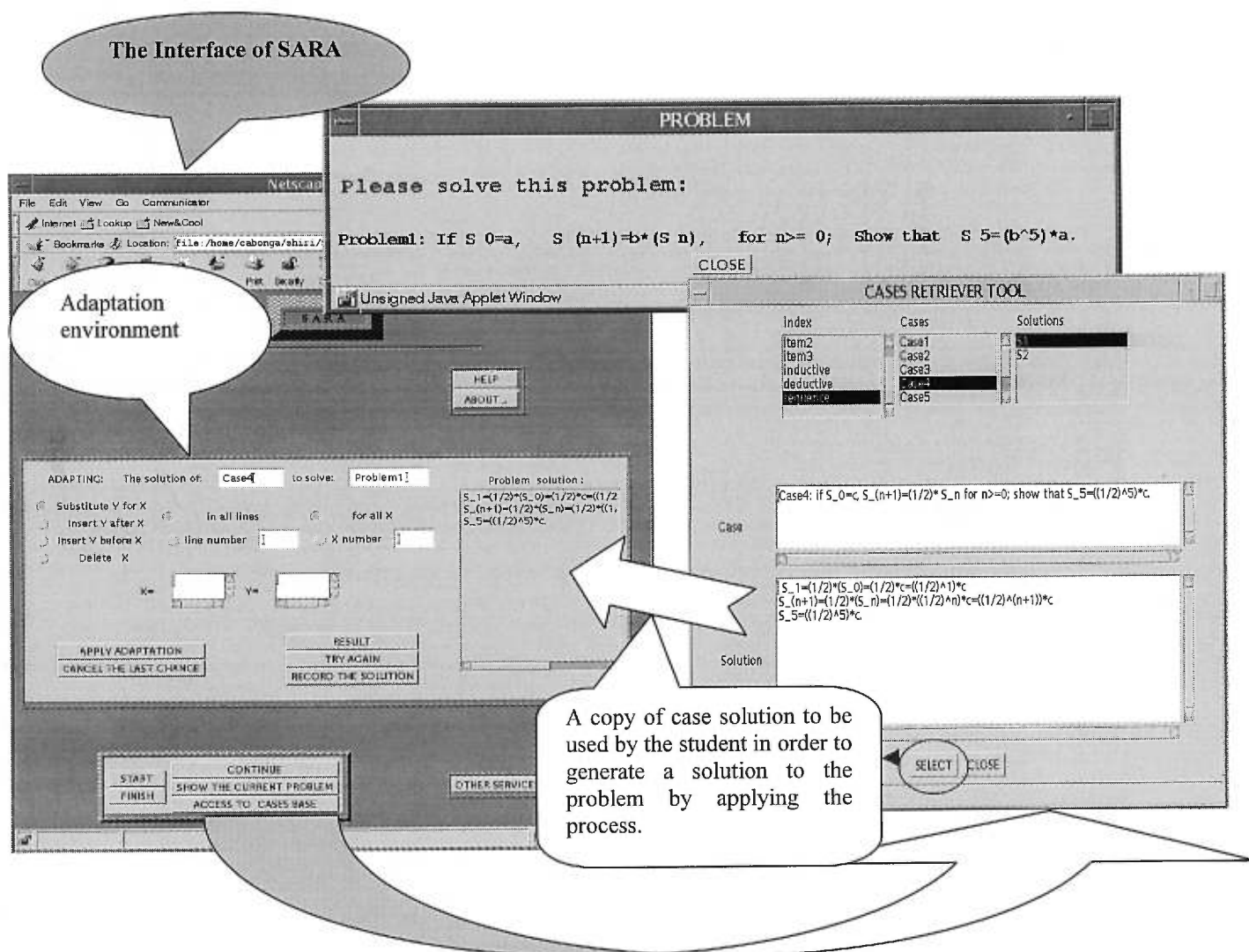


Figure 7.5 : L'Interface de SARA et le 'Cases Retriever Tool'

Il serait peut-être plus évident d'illustrer cette tâche par un exemple. Supposons que l'on propose le problème suivant à l'étudiant :

Problème 1 : If $S_0 = a$, $S_{n+1} = b \times S_n$ (for $n \geq 0$); Show that $S_5 = b^5 \times a$.

Supposons maintenant que, le cas et sa solution relative recherchée par cet étudiant sont :

Cas 4 : If $S_0 = k$, $S_{n+1} = \frac{1}{2} \times S_n$, (for $n \geq 0$); show that $S_5 = (\frac{1}{2})^5 \times k$.

Solution du cas :

$$S_1 = \frac{1}{2} \times S_0 = \frac{1}{2} \times k = (\frac{1}{2})^1 \times k$$

$$S_{n+1} = (\frac{1}{2}) \times S_n = (\frac{1}{2}) \times ((\frac{1}{2})^n) \times k = ((\frac{1}{2})^{n+1}) \times k;$$

$$S_5 = ((\frac{1}{2})^5) \times k.$$

En utilisant la solution du cas, l'étudiant pourrait dériver une solution pour le problème actuel, et pour cela substituer le 'a', 'b' et '4' dans la solution du cas par respectivement, 'k', '1/2' et '5'. En conclusion, la solution obtenue au problème est :

Solution du problème :

$$S_1 = b \times S_0 = b \times a = (b)^1 \times a$$

$$S_{n+1} = b \times S_n = b \times (b)^n \times a = ((b)^{n+1}) \times a;$$

$$S_5 = ((b)^5) \times a.$$

En comparant la solution de l'apprenant avec celle de l'expert le système peut évaluer l'exécution de l'apprenant; cette tâche sera faite par l'analyseur.

Il est important de noter ici que l'étudiant a toujours l'option d'annuler le dernier opérateur ou tout opérateur qu'il a appliqué avec retour aux étapes précédentes si nécessaire. Cette fonctionnalité est réalisée par l'intermédiaire des boutons « Cancel the last change » et « Try again ». Finalement, le bouton « Record the solution » mène le système à enregistrer la solution de l'apprenant et le bouton « Result » montre le résultat à l'étudiant. La session continue de cette façon jusqu'à que l'étudiant termine en choisissant

le bouton « Finish ». La Figure 7.6 montre un exemple d'interaction du système avec un étudiant.

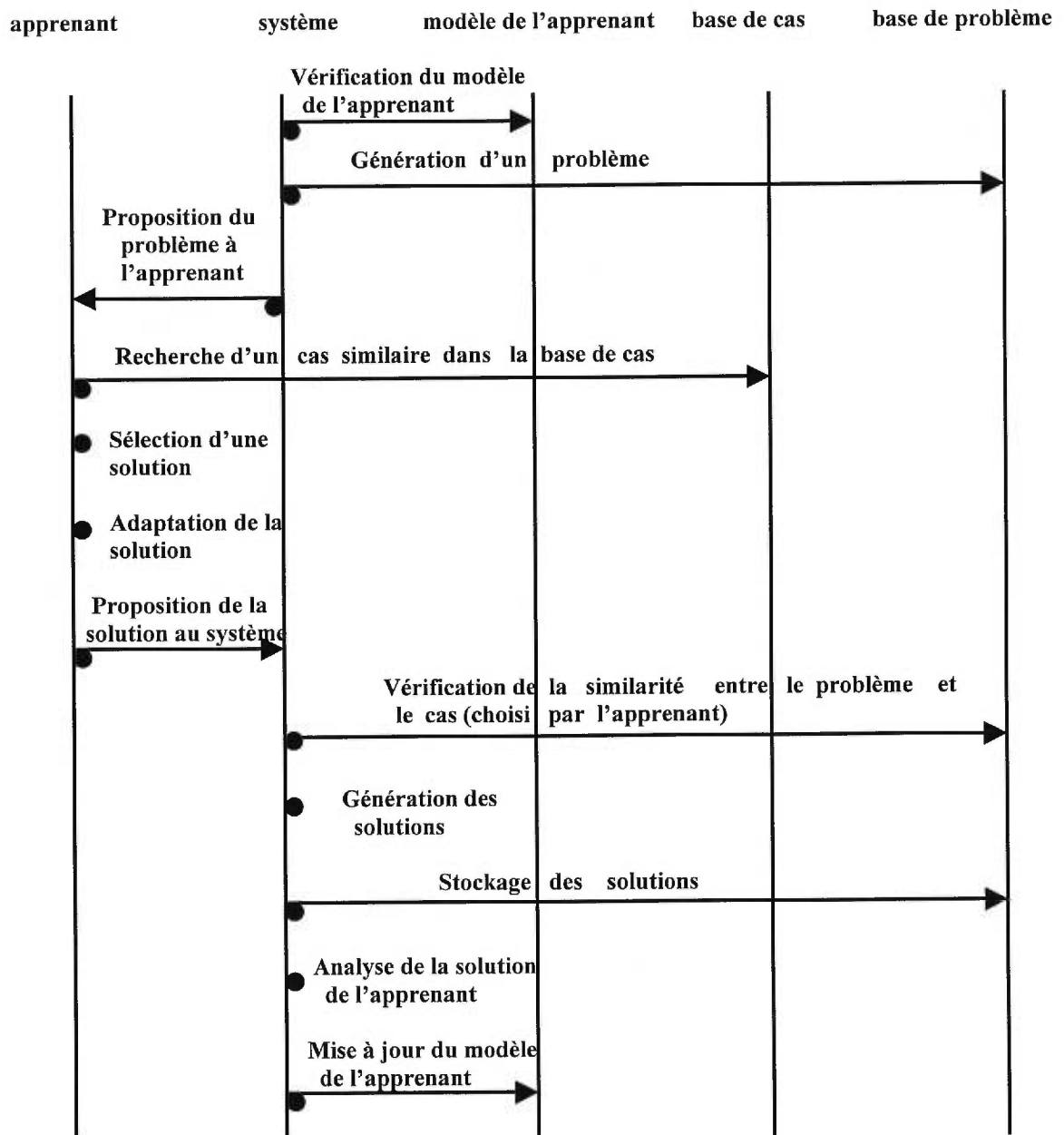


Figure 7.6 : Un exemple d'interaction

7.2.4.3 *Autres services fournis par SARA*

SARA fournit également un ensemble de services qui pourraient être employés par une application. Il fournit ces services au moyen de quatre visualisateurs de base :

- SL-Viewer (Visualisateur de la Liste de l'étudiant). SL-Viewer est contrôlé par une fenêtre qui peut être employée pour montrer la liste des étudiants utilisant le système. La fonctionnalité fournie par cette fenêtre permet à l'application de voir le nom des étudiants, d'ajouter un nouvel étudiant à la liste ou d'effacer des noms.
- CB-Viewer (Visualisateur de la Base des Cas). C'est une fenêtre qui fournit un accès facile aux cas et à leurs solutions associées. Cette fenêtre est divisée en 4 zones destinées à montrer la liste de cas et l'information liée à chaque cas, tels que la description et les solutions associées (*inductif* et *déductif*), etc.
- PB-Viewer (Visualisateur de la Base des Problèmes). PB-Viewer est conçu pour fournir l'accès facile aux problèmes et à leurs solutions associées (c-à-d, les solutions de l'expert). La fenêtre du PB-Viewer est semblable à celle du CB-Viewer avec l'écran, divisée en cinq sections pour montrer la liste de problèmes et leur information relative. Cette information comprend la description de chaque problème, des cas semblables à ce problème, les solutions et les détails de chaque solution.
- SM-Viewer (Visualisateur du modèle de l'apprenant). Un des services les plus importants assuré par SARA est d'informer l'application extérieure au sujet du contenu du modèle de l'apprenant. SM-Viewer peut être employé pour montrer le modèle de l'apprenant à l'étudiant lui-même ou à quelqu'un d'autre. Le SM-Viewer est contrôlé par une fenêtre qui peut visualiser l'état du modèle d'étudiant. La fonctionnalité fournie par cette fenêtre permet à l'application extérieure de charger un modèle particulier d'étudiant et elle le rend facile pour suivre le progrès et le contenu de modèle actuel de l'étudiant.

7.2.5 Mise à jour du modèle de l'apprenant

Une fois que le problème a été résolu par l'étudiant, le système doit mettre à jour le modèle actuel d'étudiant avec ce problème et sa solution. Le processus revient à :

- (1) mettre à jour le modèle cognitif du modèle d'étudiant en ajoutant la nouvelle information (le problème et la solution de l'étudiant au problème actuel) ou en modifiant certains éléments de ce modèle;
- (2) mettre à jour le modèle d'inférences en tenant compte du mode (inductif ou déductif) de la solution choisie par l'étudiant.

7.2.6 Avantages et limitations

Nous énumérons maintenant certains avantages et limites du système SARA :

- SARA est un système interactif et facile à utiliser.
- Il est développé pour éviter que l'apprenant raisonne à partir de zéro, chaque fois qu'un nouveau problème se présente à lui.
- Il est basé sur des techniques applicables à plusieurs domaines.
- Il permet d'évaluer les solutions générées par l'adaptation structurale, mais n'est pas bon pour évaluer les solutions formulées directement (c.-à-d. l'adaptation dérivative, voir 6.2). Nous allons voir dans la section suivante comment on peut parvenir à cette limitation dans SARA-II.

7.3 *Le prototype SARA-II*

7.3.1 Introduction

Pour pallier à certaines limites du système SARA nous avons développé le prototype SARA-II. Ce dernier est aussi organisé autour de deux bases de connaissance principales : la *base de problèmes* et la *base de cas*. Pour connaître les différences entre SARA et SARA-II, on souligne que SARA applique seulement *AS mode* dans le processus de modélisation de l'étudiant, et il peut donc traiter seulement avec des débutants; tandis que SARA-II (extension de SARA) peut appliquer deux modes d'interaction, le *AS mode* (le mode de solution accessible, pour les débutants) et le *US mode* (le mode de solution inaccessible, pour les étudiants intermédiaires ou avancés) qui permettent au système d'être plus adaptatif dans son interaction avec l'étudiant. Il considère donc trois niveaux débutant, intermédiaire et avancé pour chaque étudiant.

L'architecture de SARA-II est semblable à celle de SARA (voir la Figure 7.1). Les composants du système sont :

- *Interface*. L'interface fournit un environnement où un étudiant peut résoudre les problèmes. SARA-II emploie deux modes d'interface : l'interface du *mode AS* et l'interface du *mode US*.
- *Le sous-système modèle de l'apprenant*. Il se compose de deux parties : un *modeleur* et le *modèle de l'apprenant*.
 - a. *Le Modeleur*. C'est un composant qui est responsable de la modélisation des étudiants.
 - b. *Le Modèle de l'apprenant*. Chaque instance du modèle contient toutes les informations sur un étudiant donné.
- *Générateur de problèmes*. C'est un composant qui produit dynamiquement des problèmes pour les proposer à l'étudiant. Il choisit les problèmes appropriés à partir de la base de données selon le modèle d'étudiant.

- *Base de connaissances*. Comme mentionné ci-dessus, elle inclut deux parties :
 1. *Base de cas*. Elle contient tous les cas (problèmes résolus).
 2. *Base de problèmes*. Elle contient tous les problèmes connus non résolus.

7.3.2 Le sous-système modèle de l'apprenant

Ce sous-système contient la représentation de l'état de la connaissance des étudiants et des règles permettant de diagnostiquer ces états (que nous appelons *modeleur*).

7.3.2.1 Le modèle de l'apprenant

Comme SARA, même chose dans SARA-II contient aussi deux éléments : *modèle cognitif* et *modèle d'inférences*.

Remarque1

Nous considérons 3 niveaux (débutant, intermédiaire, avancé) pour les étudiants qui utilisent SARA-II.

Remarque2

Les informations sur l'étudiant, telles que les connaissances *affectives* pourraient être extraites et exploitées, mais elles n'ont pas été considérées dans nos prototypes (SARA et SARA-II) en raison de leurs complexités.

7.3.2.2 Le processus de modélisation de l'apprenant

Ce processus implique les actions suivantes :

- a. L'identification de l'étudiant actuel et la construction d'un modèle individualisé (de l'apprenant) pour chaque nouvel étudiant.
- b. La détermination de l'état des connaissances de l'apprenant et le modèle des inférences basées sur sa sélection de cas et des solutions pendant la résolution des problèmes.

- c. La mise à jour du modèle d'étudiant en insérant ou en enlevant des informations sur l'étudiant selon les résolutions faites.

Une partie des informations qui peuvent être employées par le modeleur pour réaliser sa tâche reposent sur le problème actuel, les cas et les solutions choisies par l'étudiant, la solution de l'étudiant, les solutions de l'expert (pour les problèmes, qui sont enregistrés dans la base de problèmes), etc.

7.3.3 L'analyse du raisonnement de l'apprenant (dans les deux modes d'interaction)

Nous discutons ici, les environnements et outils de résolution de problèmes qui ont été développés et explorés par SARA-II. Nous examinons également le processus de résolution de problèmes par l'étudiant dans cet environnement.

7.3.3.1 Modélisation de l'apprenant en AS mode

Plus particulièrement pour des débutants nous avons développé l'interface de mode *AS* (voir la Figure 7.7). Les activités de l'étudiant à travers cette interface reviennent à : étudier simplement les problèmes qui ont déjà été résolus (les cas), choisir parmi eux, ceux qui sont semblables au problème actuel, et adapter la solution choisie pour le besoin de la situation actuelle. Un ensemble de boutons permettent à cette interface de soulager la tâche d'adaptation. Par exemple, l'interface permet d'insérer quelque chose de nouveau dans la solution d'un cas, effacer un élément, ou substituer certains de ses éléments à d'autres.

Pour résoudre un problème, l'étudiant doit rechercher dans la base de cas au moyen des index afin de trouver un cas semblable au problème actuel. La recherche des cas est faite à l'aide de '*Cases Retriever Tool*'. À cette fin l'étudiant doit d'abord choisir un cas et ensuite choisir une solution associée (nous rappelons qu'à chaque cas est associé deux solutions, *inductive* et *déductive*, et que les modes des solutions ne sont pas accessibles par l'apprenant). Sur la Figure 7.7, l'écran montre la solution partielle d'un étudiant pour le problème-1. Ce problème et la solution de l'apprenant apparaissent dans le modèle de l'apprenant après que ce dernier l'ait résolu (voir 7.3.4.3).

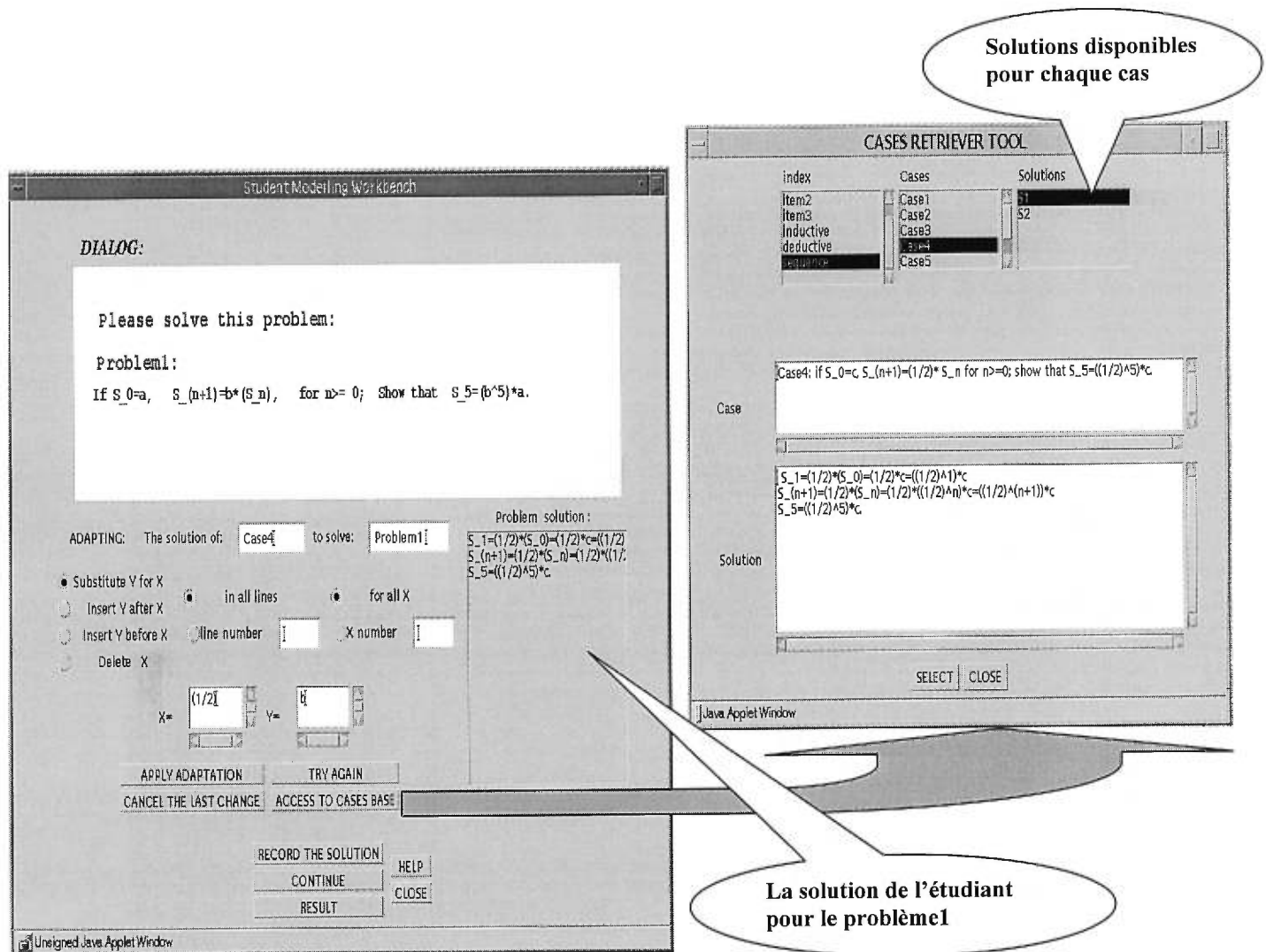


Figure 7.7 : L'interface du mode AS et le 'Cases Retriever Tool'.

Comme précédemment mentionné, en utilisant l'information du système (les solutions expertes ou l'information qui est nécessaires pour la résolution des problèmes, comme les cas semblables au problème, les processus d'adaptation, etc.) le système diagnostique le comportement de l'étudiant (voir la Figure 7.8) pour obtenir des informations nécessaire telles que : les capacités (d'analogie et d'adaptation) de l'étudiant, ses capacités de rappel, ses attributs de préférences (par exemple, mode du raisonnement), son niveau maîtrisé de la connaissance, etc.

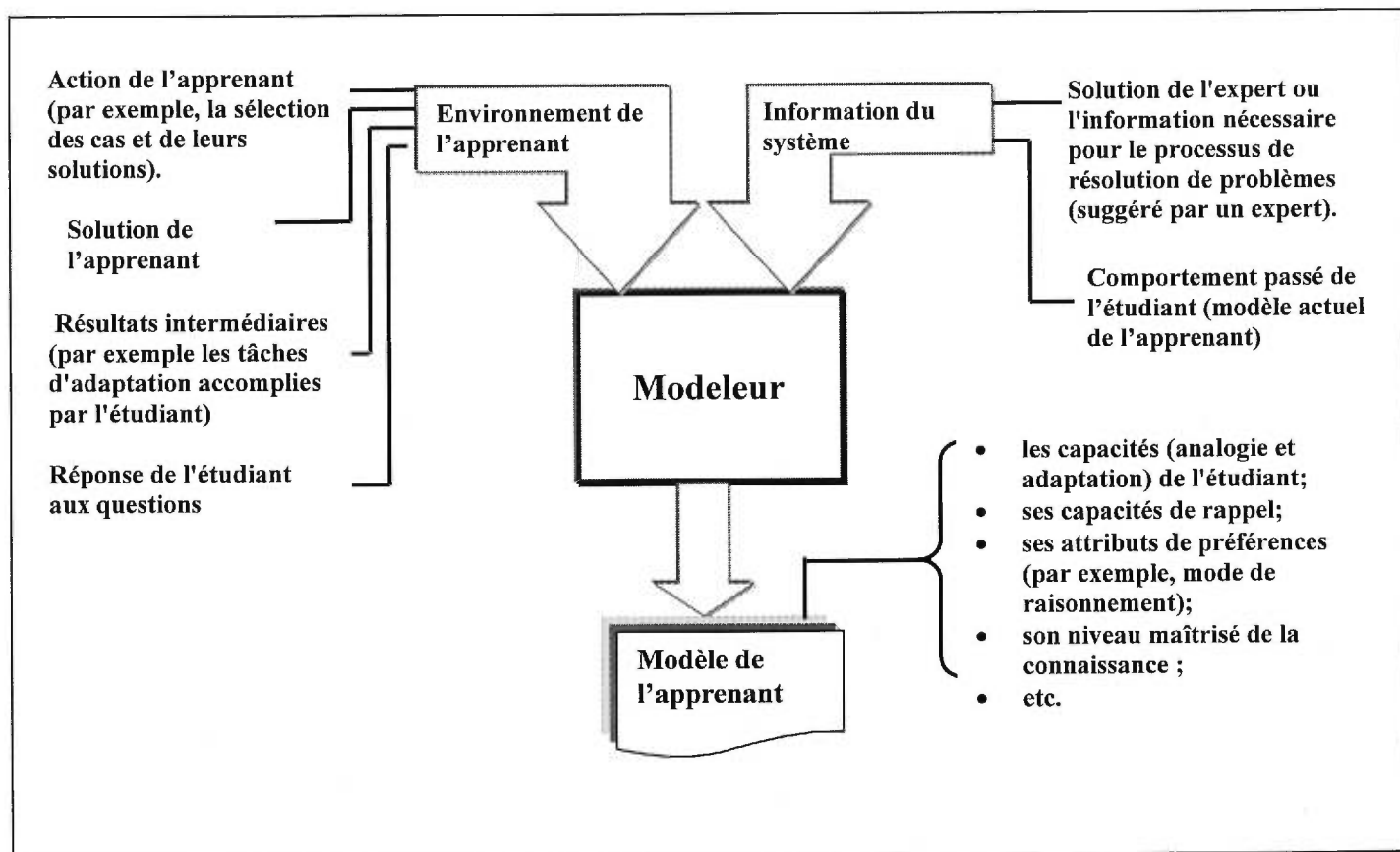


Figure 7. 8 : Un exemple d'analyse des actions de l'apprenant.

Dans [Shiri *et al.*, 1998g], nous avons discuté de la manière dont ces informations peuvent être obtenues et organisées dans le modèle d'étudiant.

7.3.3.2 Modélisation de l'étudiant en US mode

SARA-II utilise l'interface de mode *US* par laquelle un étudiant avancé (ou intermédiaire) peut former une solution. L'idée initiale derrière la conception de cette interface était de cacher la solution des cas des étudiants afin de les forcer à se concentrer sur leurs propres efforts. Pour résoudre un problème en utilisant cette interface, l'étudiant doit typiquement se rappeler les étapes de la solution du cas choisi, qu'il pense être semblable au problème donné et puis écrire sa solution étape par étape (voir la Figure 7.9). Pour détecter la solution de l'étudiant, le système la compare à une liste de toutes les solutions possibles (d'expert) existant dans le système. Mais, comme nous l'avons mentionné dans le chapitre

6, l'espace de recherche traité est considérablement grand, les techniques de recherche heuristiques sont normalement exigées pour réduire l'espace de recherche (voir 6.5.1).

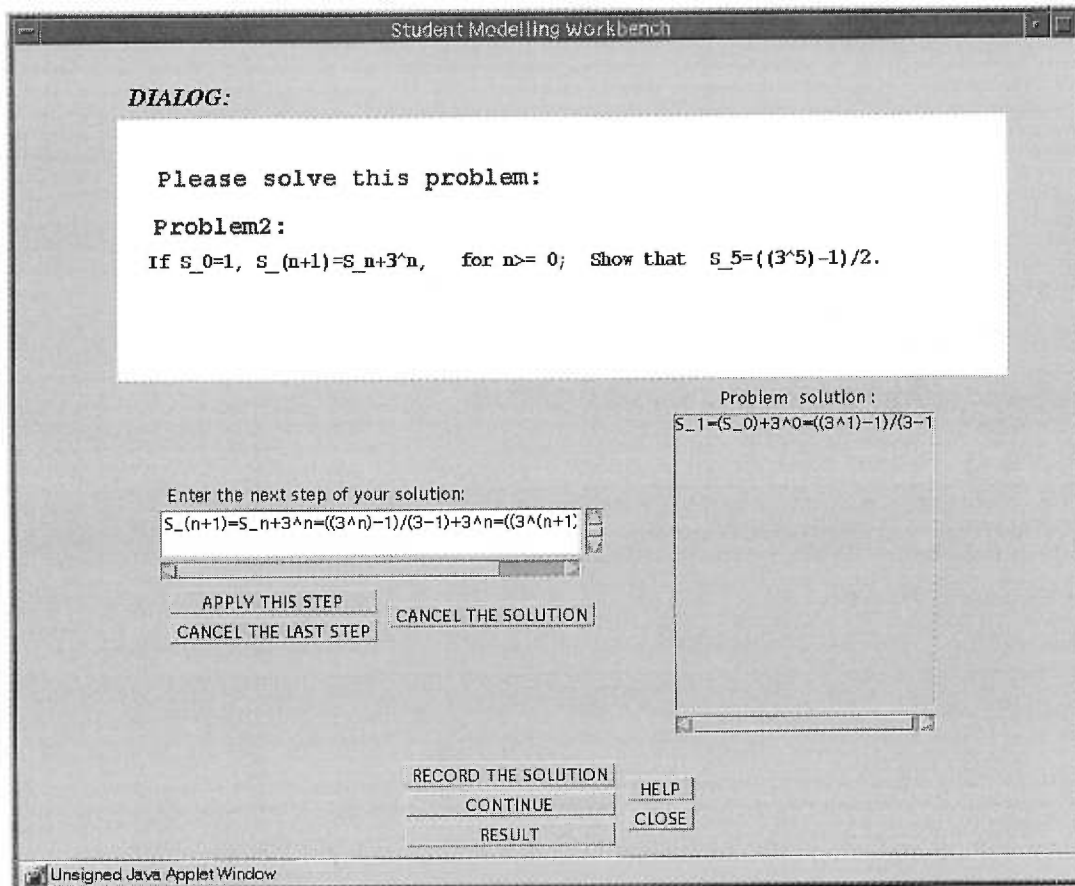


Figure 7. 9 : L'interface du mode US.

7.3.4 Discussion

De ce que nous avons vu ci-dessus, il est clair que beaucoup de questions importantes doivent être résolues. Celles-ci incluent :

- comment les débutant (ou les étudiants avancés) devraient-ils être traités ?
- pourquoi différents modes d'interface sont nécessaires ?
- comment le système peut-il choisir les modes d'interaction ?
- comment le système met-il à jour les modèles d'étudiant et comment utilise-t-il ces modèles ?

Nous allons examiner maintenant chacune de ces questions.

7.3.4.1 Le besoin de différents modes d'interface

Comme nous avons vu au chapitre 4, pour résoudre un nouveau problème en utilisant les techniques du RBC, on peut procéder de deux façons possibles :

- 1) Par réutilisation de la solution de vieux problèmes résolus. Deux moyens sont développés pour faciliter ces tâches :
 - a) un outil qui aide l'étudiant à trouver un cas semblable au problème actuel (*Cases Retriever Tool*).
 - b) un environnement qui peut faciliter la tâche d'adaptation (*l'interface du mode AS*).
- 2) Par construire une solution en se rappelant la solution de vieux problèmes (cas). Cela exige l'étudiant de se rappeler d'abord comment visualiser le problème comme une variante de quelques cas et comment adapter ensuite la solution de ces cas au nouveau problème. Nous avons développé donc *l'interface de mode US* pour lui permettre de construire sa solution.

7.3.4.2 Choisir les modes d'interaction et l'interfaces

Basé sur l'information de son modèle de l'apprenant, SARA-II fournit l'interface du *mode AS* pour des débutants et le *mode US* pour des étudiants intermédiaires ou avancés. Il est à noter que le système change d'environnement de résolution de problèmes si le niveau d'étudiant change en cours de travail. D'ailleurs, pour supporter une forme plus flexible de résolution de problèmes, le système donne aux étudiants qui ne peuvent pas trouver ou ne terminent pas la solution d'un problème en *US mode*, une chance de plus de la résoudre en utilisant le *AS mode*. Nous devons préciser que le '*Cases Retriever Tool*' sera alors automatiquement fermé, quand le système change son environnement du mode *AS* au mode *US*.

7.3.4.3 Mettre à jour le modèle de l'apprenant

Une fois que le problème a été résolu par l'étudiant, le système doit mettre à jour le modèle actuel de l'apprenant. Ce processus procède en :

- (1) mettant à jour le modèle cognitif du modèle de l'apprenant, en ajoutant la nouvelle information (le nouveau problème proposé à l'apprenant et sa solution) à ce modèle ou en y modifiant certains éléments (par exemple en changeant l'état d'un *problème non-résolu* à *problème résolu* si l'apprenant arrive à le solutionner);
- (2) mettant à jour le modèle d'inférences en tenant compte du mode (*inductif* ou *déductif*) choisi par l'étudiant lors de la résolution du problème.

La Figure 7.10 montre un modèle de l'apprenant dans SARA-II. Dans cette figure nous pouvons voir les problèmes qui n'ont pas été résolus par l'étudiant, les problèmes qu'il a résolus, ses solutions à ces problèmes, les cas qu'il avait utilisés pour obtenir ces solutions, les modes des solutions obtenues et son niveau à ce stade.

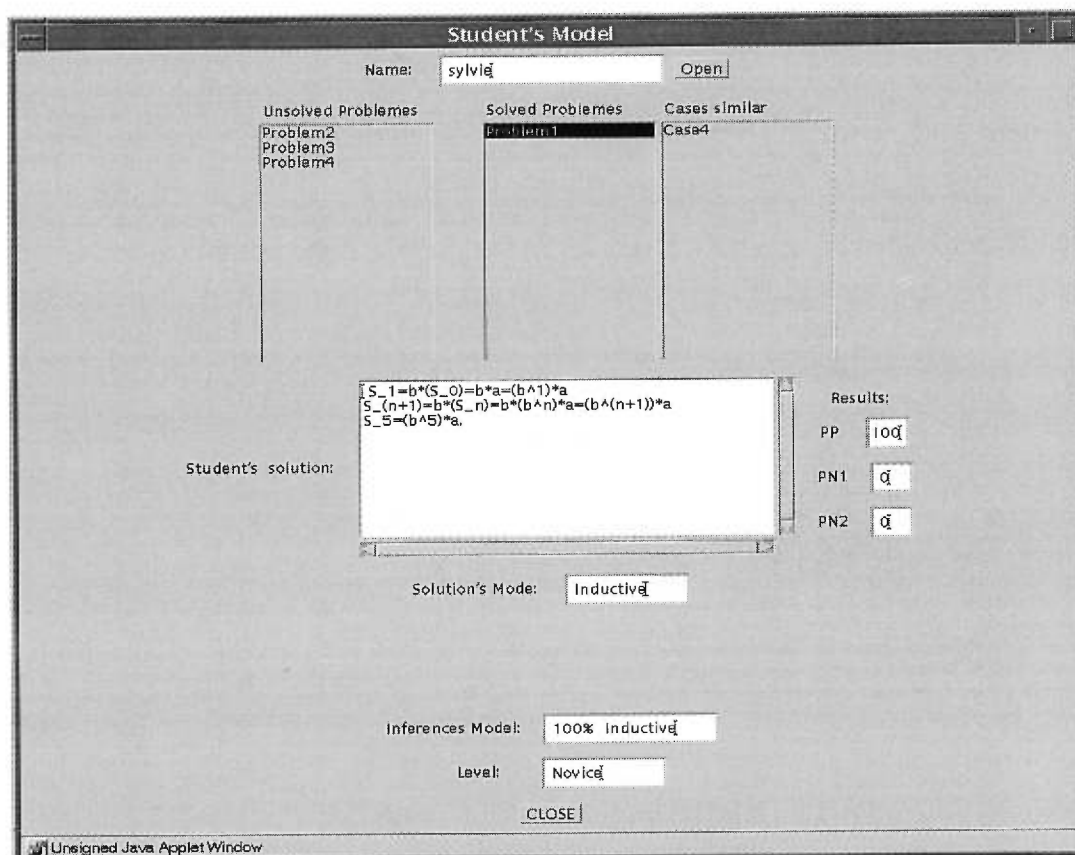


Figure 7. 10 : Le modèle d'un étudiant qui utilise le système

7.3.4.4 *Utilité du modèle de l'apprenant*

Le modèle de l'apprenant est utilisé pour examiner l'état des connaissances de l'étudiant, pour faire des hypothèses au sujet de ses conceptions et stratégies de raisonnement, afin de produire des exercices et des explications adaptées, etc. En appliquant des exercices à diverses sessions, SARA-II, par exemple, peut résumer quel genre de raisonnement est employé généralement par l'étudiant. Comme le montre le modèle de l'apprenant de la Figure 7.10, le raisonnement du type *inductif* est habituellement appliqué.

En effet, ces informations (les modes de raisonnement) qui peuvent être implicitement obtenues par le système lors de la résolution des problèmes, peuvent être employées pour adapter les explications ou l'aide offerte à l'étudiant selon ses modes préférés de raisonnement.

7.4 *Conclusion*

Dans ce chapitre, nous avons passé en revue l'état actuel du développement de SARA et SARA-II. Nous avons présenté plusieurs aspects de ces systèmes, y compris la représentation des connaissances, les modes d'interactions, etc. Nous avons aussi discuté de l'architecture des deux systèmes, de celle du sous-système modèle de l'apprenant, des éléments qui le constituent et enfin, des processus de modélisation (dans les deux modes).

Par SARA et SARA-II une tentative a été faite de créer un STI qui peut modéliser les étudiants en utilisant l'approche à base de cas. SARA est développé pour permettre le *mode accessible de solution*, et il peut fournir à l'étudiant l'accès aux cas et à leurs solutions associées. L'un des inconvénients majeur de SARA est qu'il ne permet pas d'évaluer les solutions formulées directement par l'apprenant. Pour parvenir à cette lacune, nous avons développé SARA-II. Afin d'accomplir la tâche de modélisation de l'apprenant, ce dernier, utilise un modèle d'interaction flexible. Il applique deux modes d'interactions : *AS mode* pour des débutants et *US mode* pour les étudiants intermédiaires ou avancés.

Nous allons examiner dans le chapitre suivant, l'amélioration de SARA et SARA-II, en nous basant sur l'idée que les agents intelligents pourraient nous offrir la possibilité d'interactions plus naturelles entre les apprenants et le système.

Chapitre 8

Méthodologie de construction d'un agent intelligent pour l'aide au raisonnement

Agent-based computing is likely to be the next significant breakthrough in software development.

- Sargent, 1992

8.1 Introduction

Le concept agent [Wooldridge et Jennings, 1995] est devenu important en intelligence artificielle et en informatique. La technologie des agents rendra possible une réduction importante du temps et de production d'un STI grâce à la modularisation et à la réutilisation des composantes. Notre objectif dans ce chapitre est d'essayer de développer une architecture générale pour la conception d'agent d'aide au raisonnement qui rapporte plus d'autonomie et de potentialités. L'intégration d'une telle technologie à nos systèmes introduit un aspect important de communication et d'interaction (particulièrement avec l'apprenant).

Ce chapitre décrit d'abord la technologie d'agent intelligent, et discute les avantages et les défis du développement d'un tel agent. Nous commençons par une brève introduction aux agents. Nous présentons le fond, les caractéristiques, les classifications, des approches de développement (y compris l'approche cognitive et réactive), et des domaines d'application d'agents intelligents. Nous passons en revue la technologie fondamentale en même temps que les composantes clés d'une architecture conceptuelle d'agent. Nous présentons ainsi, quelques exemples d'agents, y compris *l'agent BDI* (Belief, Desire, Intention) [Rao et Georgeff, 1993] et *l'agent délibératif* [Brenner et al., 1998].

Nous examinons ensuite, comment SARA et SARA-II peuvent être considérés comme des agents et comment nous pouvons les perfectionner selon cette approche.

Enfin, nous introduisons les systèmes multi-agents cognitifs [Frasson *et al.*, 1996b]. Notre effort sera concentré sur la discussion de la méthodologie de construction d'agents intelligents d'aide au raisonnement et sur la description de leurs caractéristiques. Nous verrons en outre, l'architecture, la coopération et la communication avec d'autres agents pour réaliser une tâche cognitive comme l'analyse du raisonnement, et les applications appropriées de tels agents. Enfin, nous fournissons en conclusion une description des futures directions de ce travail.

8.2 Qu'est-ce qu'un agent ?

La technologie d'agent est un domaine de recherche qui se développe rapidement. Les domaines d'application dans lesquels les solutions d'agents sont appliquées incluent la gestion de déroulement des opérations, gestion de réseau de télécommunication, exploitation de données, recherche/gestion documentaire, commerce électronique, éducation, robotique, bibliothèques digitales, base de données traditionnelles, etc. [Wooldridge et Jennings, 1995].

Afin de comprendre la notion d'agent, nous commençons par présenter certains concepts de base tels que la définition et les caractéristiques d'un agent.

8.2.1 Définition

Un agent (autonome) est un système situé dans un environnement complexe, qui perçoit son environnement et agit à l'intérieur de celui-ci de façon autonome, et de cette façon, réalise un ensemble de buts ou des tâches pour lesquels il est conçu [Maes, 1995].

8.2.2 Caractéristiques

Pour comprendre le comportement global et le fonctionnement d'un agent intelligent, il est nécessaire de discuter des caractéristiques et propriétés qui différencient un agent intelligent des programmes et des logiciels traditionnels. Nous décrivons ici en détail les

caractéristiques les plus importantes discutées dans la littérature. Nous détaillons ci-dessous ces caractéristiques (Figure 8.1) :

- ❖ *L'autonomie*. Se rapporte au principe qu'un agent peut agir de lui même sans intervention humaine.
- ❖ *L'intelligence*. Un agent doit agir avec son environnement pour réaliser ses buts. Il doit être capable de recueillir l'information de son environnement et de prendre la décision basée sur cette information, puis initier des actions spécifiques basées sur les décisions. Un agent exige donc une certaine quantité d'intelligence pour accomplir ses tâches. Les éléments principaux d'intelligence sont : *Apprentissage*, *Proactivité* et *Réactivité*.

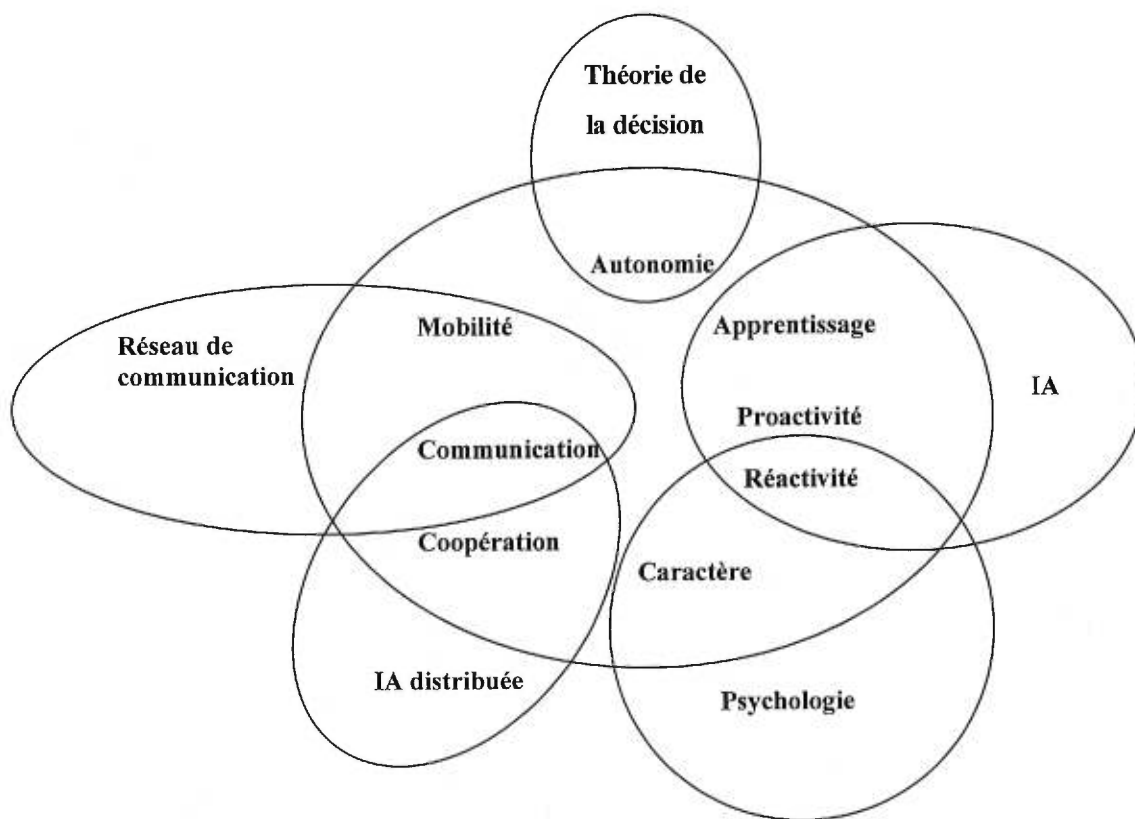


Figure 8. 1 : Domaines d'influence [Brenner *et al.*,1998].

- ❖ *Apprentissage*. Pour que l'agent soit vraiment intelligent il devrait apprendre pendant qu'il réagit et/ou agit avec son environnement externe.

- ❖ *Proactivité*. Un agent n'agit pas simplement en réponse à son environnement, il agit de façon à atteindre ses buts.
- ❖ *Réactivité*. Un agent perçoit son environnement et agit en temps opportun aux changements qui s'y produisent.
- ❖ *Les habilités sociales*. Ce sont les mécanismes concernant les activités de groupe. Un agent interagit avec les autres agents (et également avec des humains) à travers un langage de communication. Les éléments principaux d'habilités sociales sont : *Coopération* et *Communication*.
- ❖ *Coopération*. La coopération concerne fondamentalement le processus de distribution des buts, des plans, et de tâches à travers plusieurs agents. Un agent utilise des méthodes de coopération pour créer, mettre à jour et exécuter des plans multi-agents.
- ❖ *Communication*. Un agent peut employer la capacité de communication pour effectuer un contact avec son environnement.
- ❖ *La mobilité*. Un agent peut se transporter d'une machine à l'autre.

8.3 Classification

Comme indiqué sur la Figure 8.2, les systèmes d'agent peuvent être classifiés selon trois critères : *intelligence*, *mobilité* et *agence* [Gilbert *et al.*, 1995].

- *Intelligence* : l'intelligence est le degré de raisonnement de l'agent. Les préférences, le raisonnement, la planification, et l'apprentissage sont des propriétés qui déterminent de manière significative l'intelligence d'un agent.
- *Mobilité* : la mobilité est la possibilité qu'a l'agent de se déplacer dans un réseau d'ordinateurs. L'agent peut être statique, ou bien sous forme d'un script mobile ou d'un objet mobile.
- *Agence* : degré d'autonomie et d'autorité attribué à un agent mesuré selon la nature de l'interaction entre les agents (au minimum un agent peut fonctionner de façon asynchrone).

Les chercheurs ont employé ces trois caractéristiques pour dériver certains types d'agents,

par exemple : agents simples, agents intelligents, agents scripts mobiles, agents objets mobiles, systèmes multi-agents, etc.

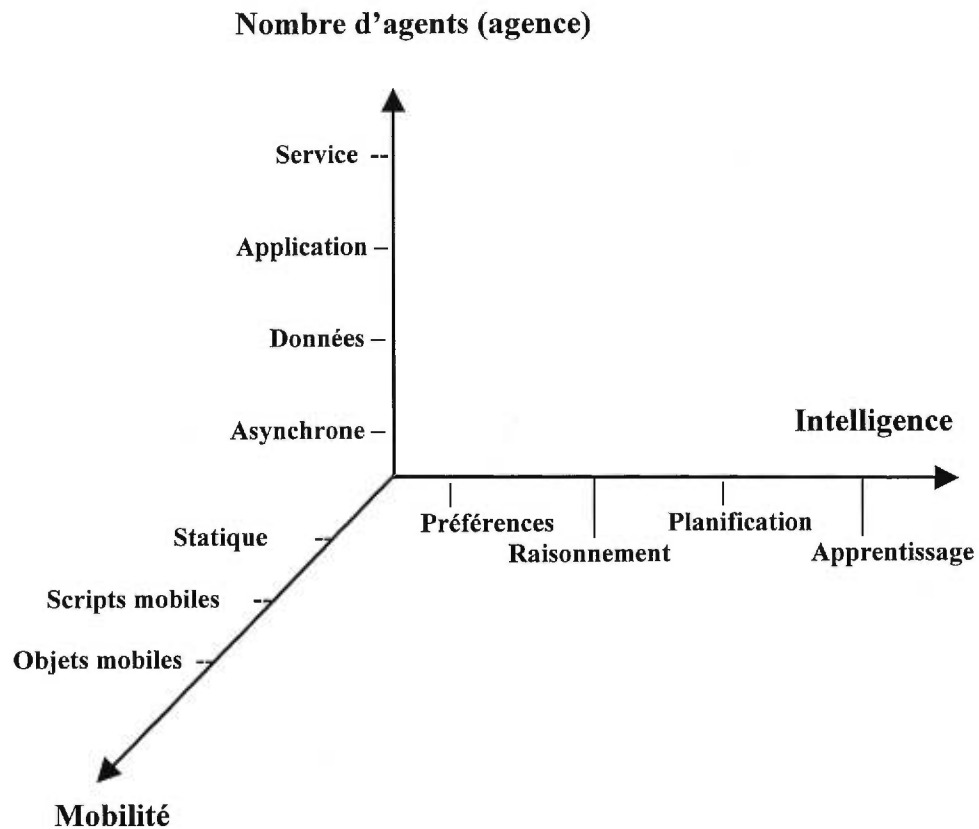


Figure 8. 2 : Classification des agents [Gilbert *et al.*, 1995]

- Les agents simples ont seulement une quantité limitée d'intelligence, tandis que les systèmes complexes (les agents intelligents) démontrent un comportement intelligent.
- Les agents simples se déplacent dans un environnement qui ne contient aucun autre agent. Ils contactent juste leur utilisateur et d'autres émetteurs d'informations tels qu'une base de données.
- Les systèmes multi-agents se composent d'un certain nombre d'agents qui peuvent communiquer ou même coopérer l'un avec l'autre.
- Des scripts mobiles sont des agents qui peuvent être envoyés à un autre ordinateur seulement avant leur exécution; tandis que, les objets mobiles peuvent changer leur position à tout moment pendant leur exécution.

8.4 Typologie d'agents

Les agents peuvent prendre plusieurs formes selon l'environnement dans lequel ils se trouvent. Quelques exemples d'agents seront décrits pour illustrer l'approche.

8.4.1 Les agents réactifs

Ces agents n'incarnent aucun type de raisonnement. Ils communiquent simplement l'information aux agents intéressés. Ils réagissent à des situations.

Exemple : Le système PENGI [Agre et Chapman, 1987] est un exemple d'agent réactif. C'est un jeu simulé dans lequel le caractère principal est contrôlé par un schéma et les activités du système sont routinières et ne requièrent aucun raisonnement.

8.4.2 Les agents délibératifs

Les agents fonctionnent dans les environnements très dynamiques, par conséquent, ils devraient être capables de prendre leurs décisions sur la base de la situation actuelle. Cependant, ceci est possible seulement jusqu'à un degré très limité pour les agents délibératifs.

Définition : Les agents délibératifs dérivent du paradigme délibératif de raisonnement qui soutient que les agents possèdent un modèle symbolique interne (de raisonnement), et ils s'engagent dans la planification et la négociation avec d'autres agents afin de réaliser leurs buts [Brenner *et al.*, 1998].

Les agents délibératifs s'appellent souvent les agents de BDI (Beliefs, Desires, Intentions) [Rao et Georgeff, 1998]. Nous présenterons, brièvement ces agents dans la section suivante.

8.4.3 Les agents coopératifs

La tâche principale de coopération des agents est de résoudre des problèmes complexes en utilisant le mécanisme de transmission et de coopération avec d'autres ressources d'objets, tels que des agents, humains ou des ressources externes. La coopération de

plusieurs agents permet une solution plus rapide et meilleure pour les tâches complexes qui excèdent les capacités d'un agent simple.

Définition : Un système multi-agents coopératif est un système où plusieurs modules informatiques dialoguent et agissent sur un même objet de travail dans le but d'atteindre un objectif commun [Tadié, 1998].

Exemple : Le système EDER_TC [Tadié, 1998] est un système multi-agents coopératif basé sur la distribution des connaissances représentant une tâche coopérative. C'est un système qui permet à des agents humains ou artificiels de simuler ou d'apprendre la réalisation d'une tâche coopérative.

8.4.4 Les agents cognitifs

Plusieurs théories traditionnelles d'agents intelligents se sont concentrées sur les aspects cognitifs d'agents. Elles sont donc appelées les architectures cognitives des agents.

Définition d'un agent cognitif : un agent cognitif est capable de raisonner sur ses intentions, ses pensées, afin de créer des plans d'actions qu'il exécute dans un environnement donné. C'est un agent réactif, adaptatif, instructif, et possède les capacités d'apprendre de l'expérience [Frasson *et al.*, 1997] [Frasson *et al.*, 1998].

Exemple : Un exemple d'un tel agent est l'agent nommé *acteur* par Frasson *et al.* (1996b). Un *acteur* est un agent cognitif et donc réactif, instructif, adaptatif, etc. [Frasson *et al.*, 1996b].

Un système cognitif comprend un petit nombre d'agents qui disposent d'une capacité de raisonnement sur une base de connaissances, d'une aptitude à traiter des informations diverses liées au domaine d'application, et d'informations relatives à la gestion des interactions avec les autres agents et l'environnement [Dieng, 1990].

Par la suite, nous introduisons les agents *BDI* et *délibératifs*, puis nous présentons leurs architectures.

8.5 Architecture d'un agent intelligent

Le but de cette section est de présenter les composants centraux de l'architecture d'un agent intelligent. Nous présentons ici l'architecture de type *BDI* et celle d'un agent délibératif.

8.5.1 L'architecture de BDI (Beliefs, Desires, Intentions)

Rao et Georgeff (1993) ont développé un cadre logique pour la théorie des agents basé sur trois modalités primitives : croyances, désirs, et intentions. Ils sont en particulier concernés par la notion du réalisme - la question de la façon dont la croyance d'un agent au sujet du futur affecte ses désirs et intentions. Dans un autre travail, ils considèrent également la possibilité d'ajouter un plan à leur formalisme [Rao et Georgeff, 1995]. Cette architecture est présentée sur la Figure 8.3. Nous détaillons maintenant chacune de ces notions.

- *Croyances*. contient la vue fondamentale d'un agent en ce qui concerne son environnement. Un agent les emploie, en particulier, pour exprimer ses espérances des états futurs possibles.
- *Désirs*. Ils contiennent les jugements de l'agent sur les situations futures. Ils sont dérivés directement de la croyance.
- *Buts*. Ils représentent un sous-ensemble des désirs de l'agent qui pourront être réalisés. Les buts forment le traitement potentiel de l'agent.

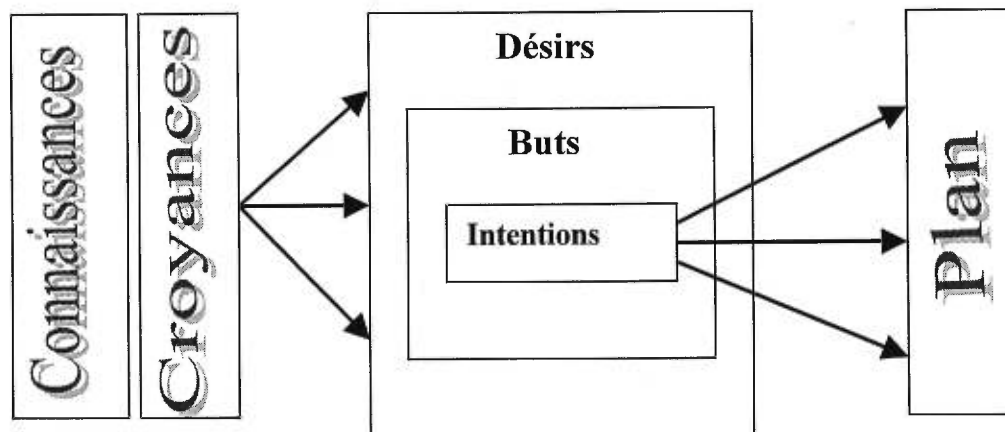


Figure 8. 3 : Structure de BDI [Rao et Gerorgeff, 1995]

- *Intentions* : Les intentions sont un sous-ensemble des buts. Si un agent décide de suivre un but spécifique, ce but devient une intention.
- *Plans*. Les plans combinent les intentions de l'agent dans des unités constantes. L'ensemble de tous les plans reflète les intentions de l'agent.

Exemple :

Le rapport entre la croyance, le but, et l'intention est illustré par l'exemple représenté par la Figure 8.4.

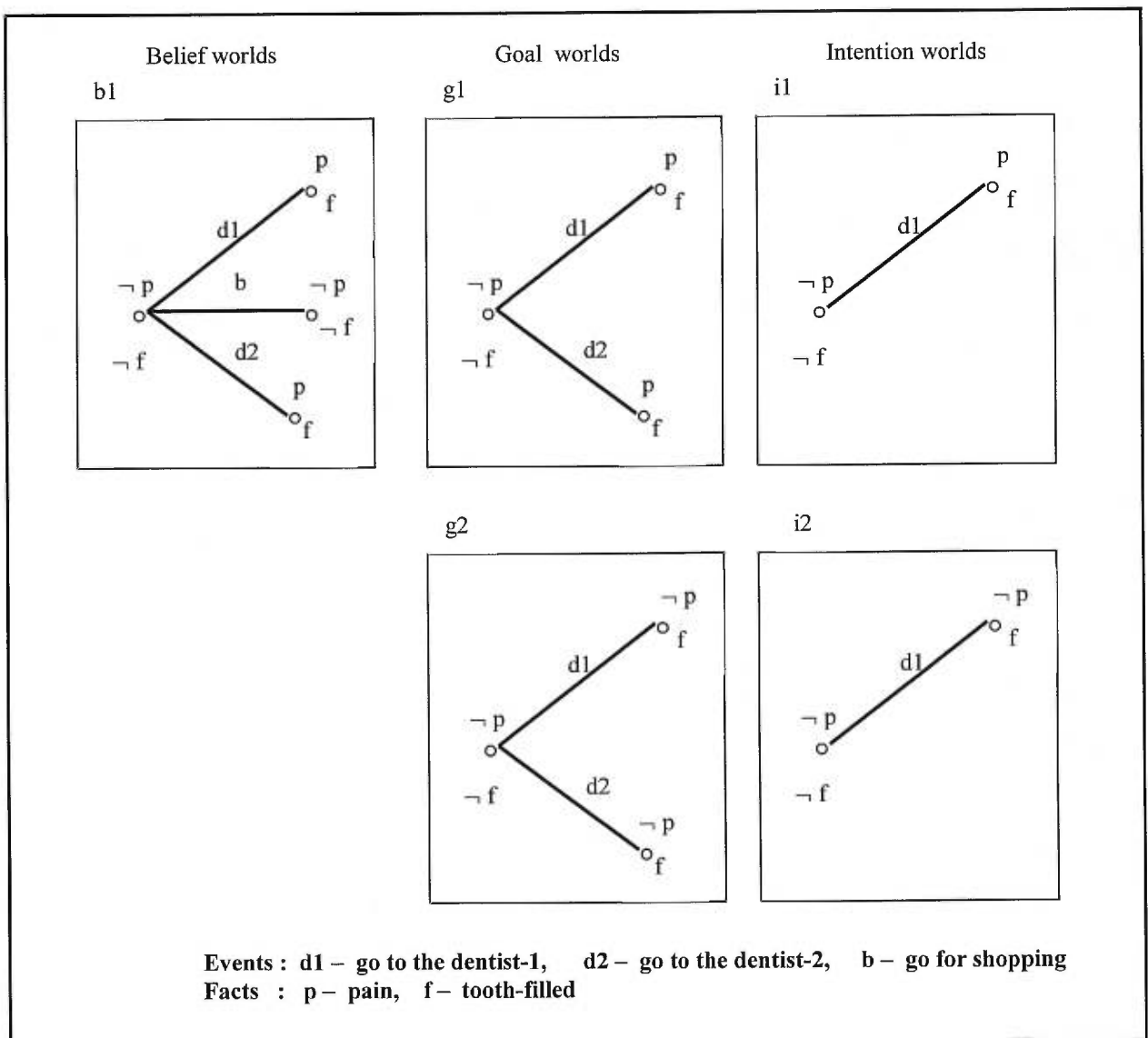


Figure 8.4 : Un exemple de *Croyances, Buts et Intentions* [Rao et Georgeff, 1998]

Dans cet exemple un agent croit qu'il est inévitable que la douleur (p) accompagne toujours 'avoir une dent à plomber (f)'. L'agent peut avoir encore le but (ou l'intention) d'avoir une dent à plomber sans également avoir le but de souffrir de la douleur. Bien que l'agent croie qu'inévitablement toujours le ($f \supset p$), il n'adopte pas 'p' comme but.

8.5.2 Architecture d'un agent délibératif

Les résultats du modèle de BDI ont un effet sur la conception de l'architecture d'un agent délibératif. La Figure 8.5 montre les composants centraux d'un tel agent.

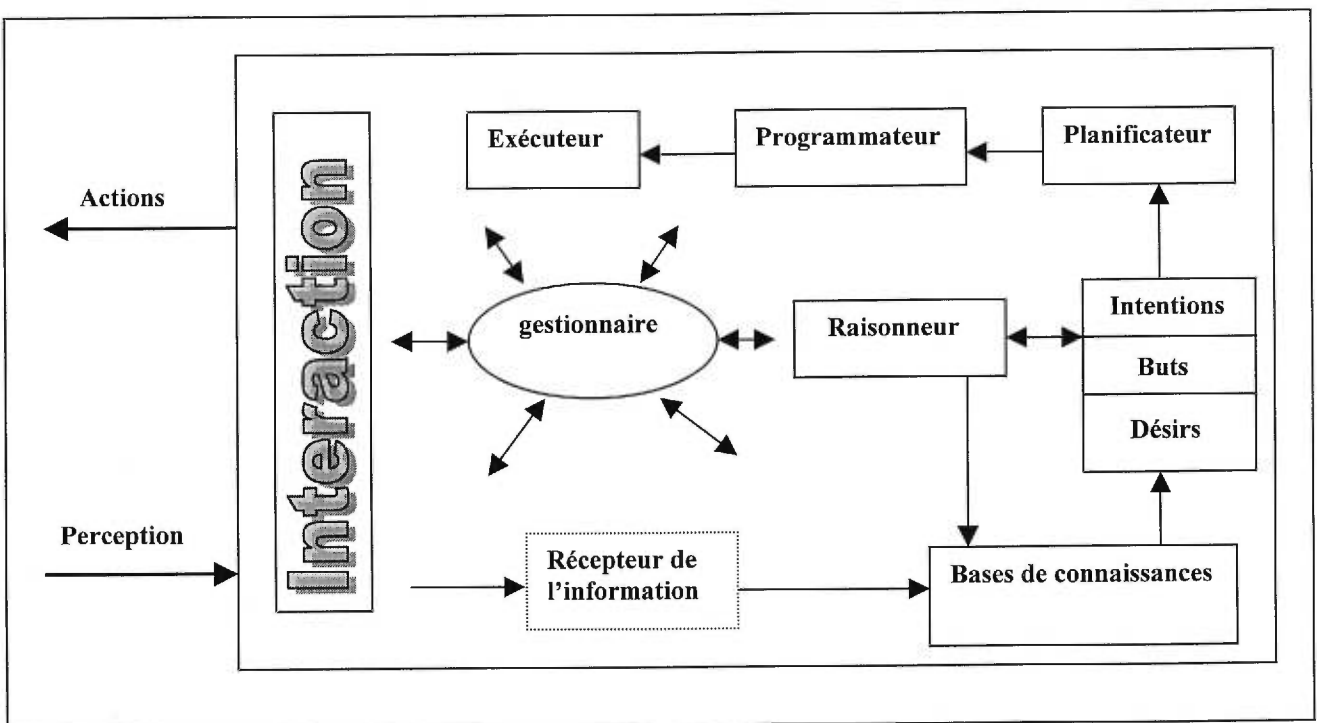


Figure 8. 5 : Architecture d'un agent délibératif [Brenner *et al.*,1998]

Le désir, les buts et les intentions d'un agent délibératif sont dérivés d'une base de connaissances suite à ses observations et interactions avec le monde (perception). Ceci constitue une tâche pour laquelle le *Raisonneur* accorde une importance centrale. Le *Planificateur* considère les intentions et les combine dans un plan global cohérent. Le *Programmateur* reçoit les plans actuels du *Planificateur*. Chaque plan se compose d'un certain nombre d'actions simples qui doivent être traitées séquentiellement ou en

parallèle. Le *Programmateur* doit décider quand des actions spécifiques doivent être rendues disponibles pour l'exécution. L'*Exécuteur* exécute la prochaine action, surveille son traitement, et termine son exécution.

Nous définissons maintenant, un agent intelligent comme étant un agent *cognitif* et *BDI*.

Nous allons voir maintenant la structure des systèmes SARA et SARA-II à travers les agents et nous discutons des attributs des agents utilisés dans ces systèmes.

8.6 Les caractéristiques des agents utilisés dans SARA et SARA-II

Comme nous l'avons montré dans la section 8.2, un agent intelligent pourrait posséder un certain nombre de caractéristiques importantes comme par exemple, l'autonomie, la réactivité, l'intelligence, etc. SARA et SARA-II possèdent trois caractéristiques de base (*l'autonomie, l'adaptabilité, la communication*), de plus, SARA-II possède la propriété de *perspicacité* que nous expliquons ci dessous :

- ❖ *l'autonomie* : SARA et SARA-II accomplissent déjà les tâches d'acquisition de connaissances de l'apprenant, l'analyse du raisonnement, la modélisation des apprenants, etc., sans interposition directe des humains;
- ❖ *l'adaptabilité* : l'adaptabilité du système est un aspect très important dans les agents pédagogiques. Du fait que, presque tous les systèmes tutoriels intelligents offrent des situations d'apprentissage personnalisées et individualisées. SARA-II par exemple, s'adapte au niveau de l'étudiant pour la sélection du mode d'interaction qu'il utilisera avec lui. Les deux systèmes (SARA et SARA-II) produisent également les problèmes en utilisant le modèle de l'apprenant;
- ❖ *la communication* : un des buts ultimes des agents pédagogiques est de communiquer des connaissances aux apprenants. SARA et SARA-II le font durant les sessions d'apprentissage;

❖ *la perspicacité* : nous pensons que la perspicacité est un autre attribut important pour les agents pédagogiques. Cet aspect prend place naturellement chez les êtres humains qui ont des capacités de prévoir certains événements ou certaines actions qui adviendront dans le futur en utilisant les connaissances qu'ils ont sur ces phénomènes. On parle alors de croyances d'un agent (les connaissances incertaines que l'agent possède sur les autres agents et sur lui-même). Ces connaissances ne sont pas nécessairement objectives, mais elles déterminent en grande partie le comportement 'intelligent' des agents et elles sont très utiles pour améliorer les performances du système.

Définition : nous définissons un *agent perspicace* comme étant un agent intelligent, qui a la capacité d'acquisition des connaissances (les croyances) sur d'autres agents en interagissant avec eux, et qui peut interpréter ces connaissances afin de prévoir les actions à entreprendre par ces agents dans les prochaines interactions. Ces prévisions peuvent être inexactes, mais nous espérons que l'agent puisse les améliorer pour les prochaines décisions.

Avoir une telle capacité est important pour certains agents (comme par exemple pour l'agent intelligent du jeu d'échecs). Dans notre cas (modélisation de l'apprenant), nos agents profiteront pleinement de cette capacité et s'en serviront pour mieux estimer l'action à entreprendre par l'apprenant lors de la résolution d'un futur problème; par la suite l'agent va mieux gérer ses actions. De plus, face à une telle situation, il peut :

1. générer des solutions appropriées pour être comparées avec la solution de l'apprenant;
2. gérer les interactions (avec l'apprenant) de façon flexible;
3. réagir plus rapidement pour préparer les réponses ou les « feedback » à l'étudiant.

Comme nous l'avons déjà remarqué dans le chapitre 7, SARA-II analyse les solutions exhibées par l'apprenant dans les étapes précédentes d'apprentissage (ce qui n'est possible qu'avec des agents perspicaces) pour prévoir les cas ou les solutions que l'étudiant va utiliser pour la résolution du problème actuel; tandis que SARA utilise les

réponses de l'apprenant afin de modéliser ses connaissances. Il n'a donc pas à être perspicace.

8.7 Possibilités de perfectionnement de nos agents intelligents

Du point de vue du développement de systèmes d'agents intelligents, certains perfectionnements sont également possibles dans notre architecture. Nous nous intéressons au développement d'une architecture multi-agents pour le STI et plus particulièrement pour un agent d'analyse du raisonnement dans un univers multi-agents. Il s'agit de faire coopérer un ensemble d'agents et de coordonner leurs buts et leur plans d'action pour la réalisation d'une tâche cognitive. Un STI décrit avec une approche multi-agents offre alors les avantages suivants :

1. *Premièrement*, dans un système multi-agents, tous les agents ont des caractères indépendants. Ceci limite la fonctionnalité fournie par chaque agent. La mise en place d'un tel agent peut donc être relativement simple.
2. *Deuxièmement*, en présentant la communication entre les agents dans le système, l'efficacité du système est également augmenté. Un bon exemple est qu'un agent peut partager l'information des autres agents sans nécessité d'accomplir réellement ces tâches.

8.8 Évolution vers les systèmes multi-agents cognitifs

Lorsque plusieurs agents se rassemblent pour travailler sur un même problème, on parle alors d'une société d'agents. La performance d'une communauté d'agents peut être évaluée par sa cohérence ou par le succès du chaque agent à atteindre leurs buts communs à travers la coopération [Byrne et Edwards, 1995].

Le développement d'une telle société d'agents implique les problèmes suivants :

1. Avec une communauté d'agents s'ajoute le problème de l'interprétation des actes des autres agents.
2. Un autre problème important concerne l'organisation sociale des agents.

Nous allons présenter en détail d'abord le deuxième problème (c.-à-d., l'organisation entre les agents); nous discutons ensuite le problème de l'interprétation des actes des autres agents afin de pouvoir décrire une architecture d'agents intelligents destinés à raisonner.

8.8.1 Organisation sociale d'agents cognitifs

L'organisation sociale d'un système multi-agents est la manière dont le groupe est constitué, à un instant donné, pour pouvoir fonctionner. Elle décrit l'ensemble des agents appartenant au groupe, leurs natures, leurs responsabilités, ainsi que les liens de communication entre eux [Labidi et Lejouad, 1993].

La Figure 8.6 présente une société d'agents (nous l'appelons SARA-III) qui coopèrent pour assurer une formation intelligente d'un apprenant. Par la suite, nous allons analyser cette architecture et les différents agents qui y participent :

- *Agent d'Interface (AI)*. Cet agent s'occupe de l'interaction et de la communication avec l'apprenant et de transmettre des informations entre l'apprenant et les autres agents. Il négocie avec un *Agent Tuteur (AT)* et un *Agent Intelligent pour l'Aide au Raisonnement (AIAR)* afin de décider du mode d'interface (par exemple, *interface du mode AS* ou *interface du mode US*, etc.) à appliquer.
- *Agent d'inFormation (AF)*. Un agent d'information est un agent qui a accès au moins à un, et potentiellement à plusieurs émetteurs d'informations, et peut assembler et manipuler de façon autonome l'information obtenue à partir de ces sources afin de répondre à des requêtes d'utilisateurs et d'autres agents.
- *Agent Intelligent pour l'Aide au Raisonnement (AIAR)*. Cet agent surveille les modèles d'apprenants, met à jour ses modèles, infère les niveaux d'acquisition des apprenants, etc.
- *Agent de Planification (AP)*. Cet agent détermine dynamiquement et de façon autonome les leçons ou les cours qui sont les plus adaptés à l'étudiant compte tenu du modèle de l'apprenant et suite à la demande de l'agent tuteur.

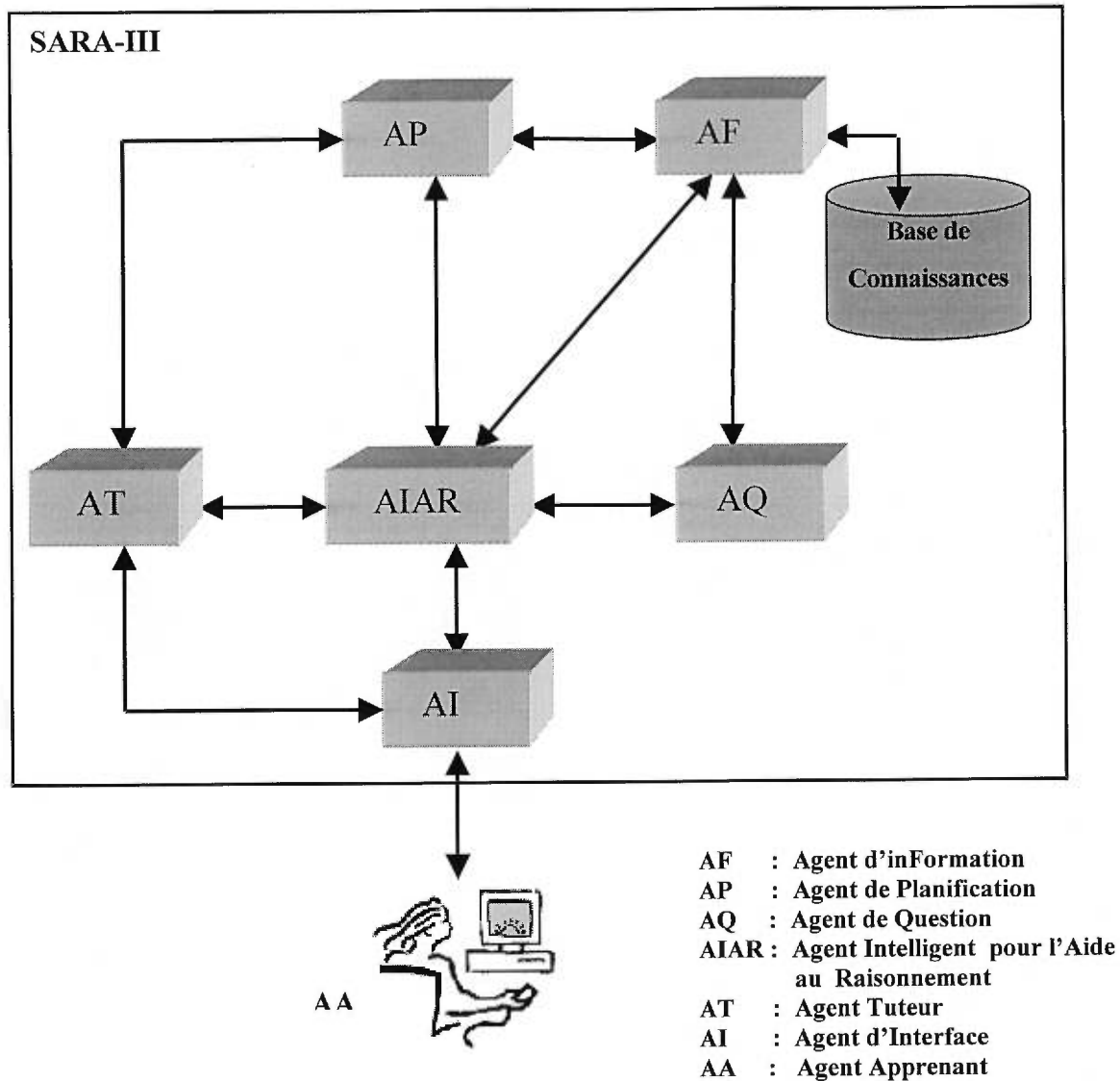


Figure 8. 6 : La coopération entre les agents pour un apprentissage dans SARA-III

- *Agent Tuteur (AT)*. C'est un instructeur programmé qui possède toutes les caractéristiques d'un agent intelligent et joue le rôle de l'enseignant. Il coopère avec les autres agents de la société pour contrôler une session d'enseignement, il négocie avec *AP* et *AIAR* pour décider du contenu à enseigner et de la stratégie à appliquer, etc.

- *Agent de Question (AQ)*. Cet agent produit des questions ou des problèmes adéquats en tenant compte du modèle de l'apprenant afin d'être utilisés par *AT* ou *AIAR* pour tester l'apprenant ou évaluer son état de connaissances.

Le STI (SARA-III) que nous venons de développer est un système multi-agents d'une façon générale dont différents type d'agents participent afin de réaliser une formation intelligente pour les apprenants.

8.8.2 Les différences entres SARA, SARA-II, et SARA-III

Comme nous l'avons vu dans section 8.6, l'architecture actuelle de nos systèmes SARA et SARA-II, nous offre certains aspects d'agents intelligents comme par exemple, *l'autonomie, l'adaptabilité, etc.*; mais ils ne possèdent pas des caractères ou des attributs propres associés aux systèmes multi-agents comme par exemple, *la communication et la coopération etc.* (que nous prévoyons de mettre dans SARA-III). Dans les tableaux 8.1 et 8.2, nous résumons les différences entre SARA, SARA-II, et SARA-III.

Composants du SARA (sous forme de modules)	Composants du SARA-II (sous forme de modules)	Composants du SARA-III (sous forme d'agents)
Base de connaissances	Base de connaissances	AF
Générateur de problèmes	Générateur de problèmes	AQ
Modèle de l'apprenant	Modèle de l'apprenant	AIAR
Interface (<i>AS mode</i>)	Interface (<i>AS mode + US mode</i>)	AI
		AP
		AT

Tableau 8. 1 : Rapport entre les composants de SARA, SARA-II et SARA-III.

caractéristiques	SARA	SARA-II	SARA-III
L'autonomie	√	√	√
L'adaptabilité	√	√	√
La réactivité	√	√	√
La perspicacité		√	√
La coopération			√
La communication	seulement avec l'apprenant	seulement avec l'apprenant	√
Nombre d'agents	1	1	multi-agents

Tableau 8. 2 : Rapport entre les caractéristiques de SARA, SARA-II et SARA-III.

Parmi les agents impliqués dans l'architecture de SARA-III, nous décrivons ci-dessous uniquement l'agent relié directement à notre recherche, c.-à-d. l'agent pour l'aide au raisonnement.

8.9 Agent Intelligent pour l'Aide au Raisonnement (AIAR)

Notre but dans cette section est d'examiner ce que nous percevons comme étant le plus important dans la conception et la construction d'agents intelligents pour l'aide au raisonnement. Nous distinguons les aspects suivants :

- la structure d'agent,
- les tâches à exécuter et les mécanismes de déduction propres au agent,
- les caractéristiques d'agent,
- le contrôle de la coopération entre agents (négociation, coordination, etc.)

Nous déterminons dans cette section d'abord les tâches à exécuter par un *agent intelligent pour l'aide au raisonnement* et nous donnons ainsi sa définition; nous présentons ensuite les propriétés principales de cet agent; et finalement son architecture.

8.9.1 Tâches à exécuter par AIAR

Afin de pouvoir spécifier un agent intelligent d'aide au raisonnement, nous devons maintenant déterminer les tâches susceptibles d'être accomplies. Ces tâches sont :

- ◆ *tâches d'acquisition et d'interprétation.* L'AIAR devrait pouvoir acquérir et interpréter des nouvelles informations de son environnement, en utilisant probablement certaines techniques de raisonnement, et selon l'information fournie par d'autres agents;
- ◆ *tâches d'analyse et modélisation.* L'AIAR devrait pouvoir analyser et modéliser les connaissances (solutions) reçues de l'apprenant;
- ◆ *tâches diagnostiques.* L'AIAR devrait pouvoir détecter les situations contradictoires ou les événements inattendus dans les solutions soumises par l'apprenant;
- ◆ *tâches de modification et d'entretien.* L'AIAR devrait pouvoir mettre à jour, manipuler et entretenir les connaissances représentant l'étudiant;
- ◆ *tâches informatives.* L'AIAR devrait pouvoir informer d'autres agents (par exemple, l'agent tuteur) de certains contenus des modèles d'apprenants;
- ◆ *tâches de réponse.* L'AIAR devrait être préparé à répondre à la question même si elles exigent l'effort supplémentaire d'inférence, requis par d'autres agents;
- ◆ *tâches de requête.* L'AIAR devrait pouvoir questionner les autres agents toutes les fois que cela est nécessaire pour exécuter ses tâches.

Après la discussion des tâches à exécuter par un agent intelligent pour l'aide au raisonnement sa définition est maintenant comme suit.

Définition : les agents intelligents pour l'aide au raisonnement effectuent continuellement les fonctions suivantes : (1)- percevoir les états dynamiques de leur environnement, réagir face à eux et enfin raisonner afin d'interpréter les perceptions, (2)- déterminer les actions à entreprendre, (3)- analyser et diagnostiquer les solutions des apprenants, (4)- effectuer des inférences sur les modèles de l'apprenant, (5)- élaborer des réponses à envoyer, et déterminer les états à communiquer à d'autres agents de la société (voir 8.8.1).

8.9.2 Les propriétés de l'AIAR

Il y a diverses propriétés d'agents qui devaient être exprimées sous forme de modèle d'agent intelligent pour l'aide au raisonnement. D'après la définition de l'AIAR et ses tâches à exécuter les propriétés suivantes sont les plus importantes pour cet agent :

- ⇒ *l'autonomie*. La capacité de contrôler ses propres actions (au moins dans certaines limites). AIAR doit être capable d'accomplir ses propres tâches sans intervention humaine;
- ⇒ *la réactivité*. La capacité de percevoir son environnement et de réagir en temps opportun aux changements qui se produisent;
- ⇒ *l'adaptabilité*. La capacité de contrôler ses aptitudes selon l'agent avec lequel il interagit. Ceci permet à notre agent de s'adapter et d'être plus flexible dans l'interaction avec les apprenants;
- ⇒ *la rationalité*. L'AIAR doit être capable d'agir de façon à atteindre ses buts;
- ⇒ *la perspicacité*. La capacité de prévoir les actions à entreprendre par d'autres agents qui interagissent avec lui. Elle n'est pas nécessaire mais elle très utile pour améliorer les performances de l'agent.
- ⇒ *la coopération*. La coopération est une caractéristique importante dans les systèmes multi-agents. L'AIAR doit être capable de participer à la réalisation d'une tâche coopérative (l'apprentissage) en exécutant les activités liées à son poste de travail;
- ⇒ *la communication*. Pour réaliser un tâche coopérative, les différents agents doivent être capables de se comprendre. Ils ont donc besoin de s'entendre sur un langage de communication accessible et compréhensible par tous. SARA et SARA-II possèdent un niveau bas de cette capacité. Ils communiquent seulement avec les apprenants; ceci se fait directement et pas à travers un langage de communication d'agents.

Pour être capable de respecter les capacités énoncées, nous proposons une architecture pour l'AIAR qui peut supporter les objectifs associés. Par la suite nous allons présenter cette architecture.

8.9.3 Architecture de l'AIAR

Dans cette section, nous présentons notre vision de l'architecture d'un agent intelligent d'aide au raisonnement et nous détaillons ensuite son fonctionnement.

Une architecture d'agent indique comment un agent peut être décomposé en ensembles de modules, et comment ces modules peuvent coopérer pour percevoir les environnements de l'agent et exécuter des actions pour arriver aux buts. L'architecture d'AIAR contient certains modules, par exemple, perception, réaction, contrôleur, analyseur, etc., distribué selon trois couches : couche réactive, couche de contrôle, et couche de raisonnement. Chaque couche permet d'implanter les fonctions particulières. L'architecture est illustrée à la Figure 8.7. Nous allons définir maintenant chaque module en détaillant son rôle.

- ❖ *Perception.* L'AIAR a des capacités d'interaction avec les autres agents comme par exemple *agent apprenant, agent tuteur, agent interface, etc.* (voir 8.8.1), il a donc la possibilité d'acquérir des connaissances sur l'environnement externe. Dans le cadre d'une telle activité, l'agent doit disposer de la connaissance permettant d'interpréter les messages reçus. Les systèmes SARA et SARA-II sont capables d'interpréter seulement les actions des apprenants.
- ❖ *Action.* Le module 'Action' regroupe toutes les actions qui permettent à l'agent de changer un environnement donné. Par exemple, donner une réponse, émettre une question, etc. L'agent doit donc être capable d'échanger les connaissances entre ses composants et les autres agents de la société. Si nécessaire, il doit traduire l'information dans le langage de ces agents (par exemple appliquer le langage naturel pour l'agent apprenant). S'il existe plus d'un agent, il est de plus responsable de diriger des messages aux agents appropriés. L'agent doit alors savoir les conditions de transmission des messages, certaines connaissances liées à la coopération et la communication sont donc nécessaires à cette fin. Cette capacité n'apparaît pas dans SARA et SARA-II.
- ❖ *Contrôle.* La connaissance de contrôle dans cet agent est représentée par les buts, les intentions, les plans et les tâches qu'il possède. En fonction des connaissances et

croyanances dont il dispose et des buts qu'il se fixe suite à une interaction avec le monde extérieur, l'agent élabore un plan d'action.

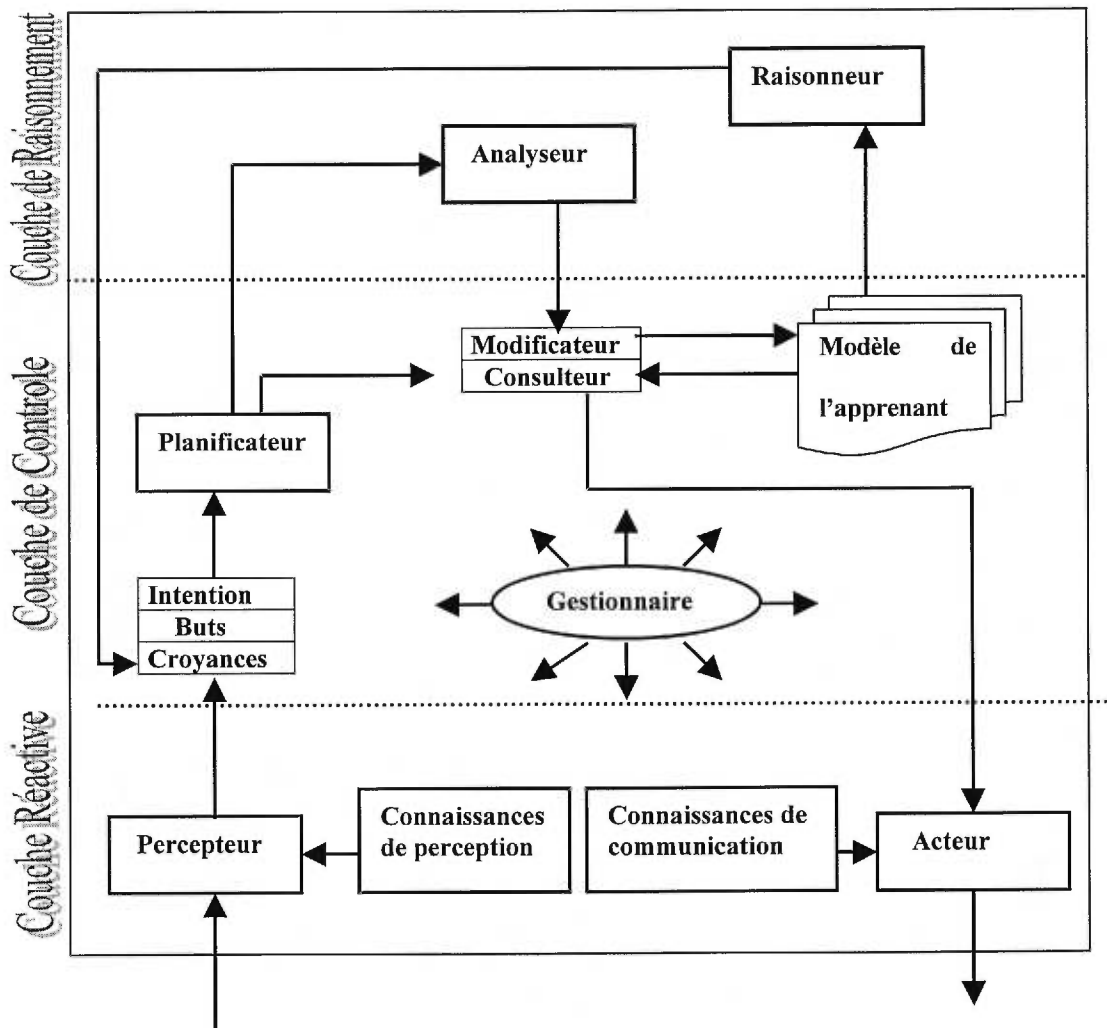


Figure 8. 7 : Architecture de l'AIAR

- ❖ *Planification.* L'AIAR doit être capable de décider quel est le but à retenir et à satisfaire en premier, pour ensuite planifier en fonction de ce but et passer à l'exécution. La Figure 8.8 montre un exemple du plan.
- ❖ *L'analyse du Raisonnement.* Afin de modéliser les apprenants, l'agent a besoin d'un mécanisme capable d'analyser les actions (les solutions) des apprenants. L'analyseur est alors employé pour accomplir cette tâche.

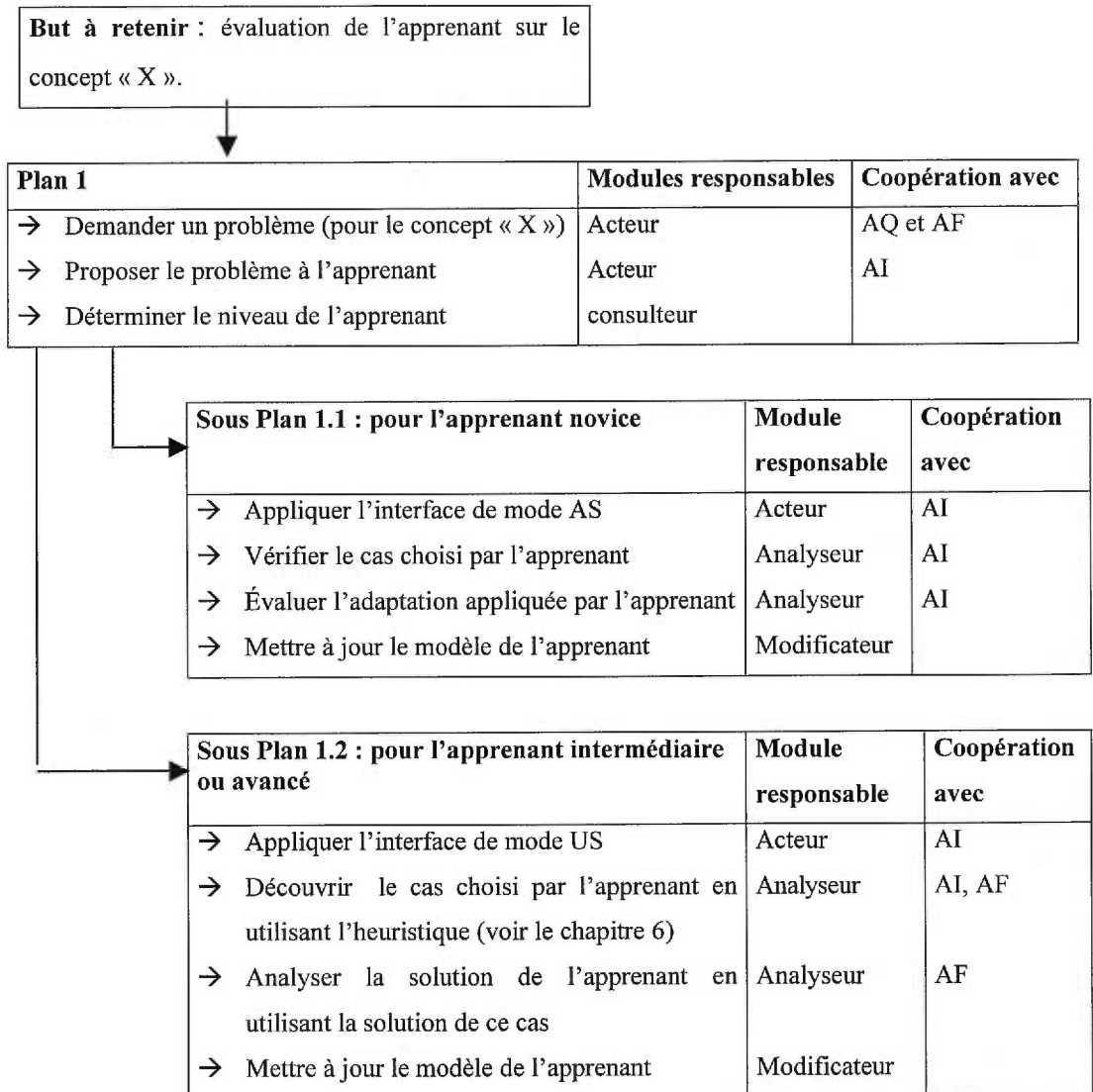


Figure 8. 8 : Exemple d'un plan

- ❖ *Prévision.* Le *Raisonneur* est responsable d'analyser et traiter les croyances actuelles afin que l'agent puisse améliorer ses prévisions futures. Par exemple si les croyances actuelles de l'agent est qu'un apprenant applique toujours le raisonnement *inductif* dans la résolution de problèmes, mais il a appliqué le raisonnement *déductif* pour le problème courant, l'agent changerait alors sa croyance pour « l'apprenant applique habituellement le raisonnement *inductif* et rarement le raisonnement *déductif* ».
- ❖ *Modification et Consultation.* La *modificateur* met à jour les modèles de l'apprenant en enregistrant les résultats émis par l'*analyseur* dans ces modèles. Le *consulteur*,

consulte les modèles de l'apprenant pour obtenir les informations sur un étudiant spécifique, demandées par d'autres agents.

- ❖ *Gestionnaire*. Le rôle du *Gestionnaire* est central dans notre architecture. Il gère les interactions entre les composants (éléments intérieurs) de l'*AIAR*.

Enfin, un point important considéré par cette architecture (*AIAR*) est qu'elle est plus orientée vers celle de l'agent intelligent et elle est plus adaptée à l'exploitation d'un système multi-agents. Dans la section suivante, nous examinons la coopération entre les agents (de cette société cognitive d'agents, c.-à-d. SARA-III (voir la Figure 8.7)) pour réaliser une tâche de résolution de problèmes.

8.10 La coopération entre agents pour l'enseignement et l'apprentissage

Selon Tadié (1998), l'enseignement (pour nous, aussi bien l'apprentissage) est un processus coopératif, il est mis en place par un certain nombre d'agents. Construire un système pour l'enseignement ou bien l'apprentissage, c'est donc construire une société d'agents, que l'on appelle la société d'agents cognitifs.

Nous nous intéressons ici à l'évaluation de la résolution de problèmes par les apprenants. La réalisation de cette tâche nécessite donc la coopération ou la mise en concurrence de plusieurs agents cognitifs (voir la Figure 8.6).

8.10.1 Supporter la résolution de problèmes

Nous croyons qu'il est important pour un système multi-agents cognitif de pouvoir aider un apprenant en situation de résolution de problèmes. Cette aide ne consiste pas à donner la solution à l'étudiant, mais à l'amener à trouver une solution appropriée.

8.10.2 Exemple

Ce qui suit (Figure 8.9) est un exemple de la façon dont les agents communiquent pour amener l'étudiant à trouver une solution pour un problème donné. Dans cet exemple nous

supposons que les bases de connaissances aussi bien que les modèles des apprenants sont représentées par des cas.

AT \rightarrow **AQ** : Veuillez me donner un problème pour évaluer la connaissance de l'étudiante (Sylvie) sur le sujet X.

AQ \rightarrow **AF** : Quels sont les problèmes par lesquels le sujet X peut être évalué?

AF \mapsto **AQ** : Voici les problèmes : $P_2=f(x)$, $P_5=g(x)$, $P_7=h(x)$.

AQ \rightarrow **AIAR** : L'étudiante (Sylvie) a-t-elle déjà résolu le problème $P_2=f(x)$, ou bien un problème semblable à ce problème?

AIAR \rightarrow **AF** : Quels sont les problèmes semblables à $P_2=f(x)$?

AF \mapsto **AIAR** : Les problèmes semblables à $P_2=f(x)$ sont : $P_9=k(x)$, $P_{15}=l(x)$.

AIAR \mapsto **AQ** : Oui, elle a déjà résolu un problème semblable : $P_9=k(x)$.

AQ \mapsto **AT** : Voilà un problème : $P_2=f(x)$.

AT \rightarrow **AA** : Veuillez résoudre ce problème : $P_2=f(x)$.

AA \mapsto **AT** : Je n'arrive pas à résoudre ce problème.

AT \rightarrow **AA** : Si vous ne pouvez résoudre le problème proposé, essayez de résoudre d'abord un problème relié.

AA \mapsto **AT** : Je ne trouve pas de problème semblable à celui-ci.

AT \rightarrow **AA** : Regardez les inconnues et essayez de penser à un problème familier que vous avez déjà résolu, qui possède les mêmes inconnues ou des semblables.

AA \mapsto **AT** : Je ne me rappelle pas de problème semblable que j'ai déjà résolu.

AT \rightarrow **AIAR** : L'étudiante (Sylvie) a-t-elle déjà résolu le problème similaire à $P_2=f(x)$?

AIAR \mapsto **AT** : Oui, elle a déjà résolu le problème : $P_9=k(x)$.

AT \rightarrow **AA** : Avez-vous déjà résolu ce problème ($P_9=k(x)$) ?

AA \mapsto **AT** : Bien sur, j'ai résolu ce problème.

AT \rightarrow **AA** : Bien, voilà un problème similaire; pouvez-vous l'utiliser ?

--

--

AF : Agent d'inFormation
AP : Agent de Planification
AQ : Agent de Question
AIAR : Agent Intelligent pour l'Aide au Raisonement
AT : Agent Tuteur
AI : Agent d'Interface
AA : Agent Apprenant

AF \mapsto **AP** : AF informe AP
AF \rightarrow **AP** : AF demande à AP

Figure 8. 9 : Exemple de dialogue entre agents

8.11 Conclusion

Ce chapitre a passé en revue les concepts et les aspects principaux associés à la théorie et à la pratique des agents intelligents. Il a fourni une perspective dans laquelle l'agent peut être formalisé, ainsi que la mise en œuvre d'une architecture appropriée.

Nous avons discuté ensuite, des attributs des agents utilisés dans les systèmes SARA et SARA-II et nous avons analysé la possibilité de perfectionnement de ces agents intelligents. Nous avons proposé par la suite une architecture pour le STI décrit avec une approche multi-agents (appelé SARA-III) ainsi qu'une architecture pour l'agent intelligent d'aide au raisonnement (*AIAR*) qui peut coopérer avec d'autres agents afin de réaliser une tâche cognitive. Nous avons montré que SARA-III présente des avantages supplémentaires par rapport aux SARA et SARA-II.

Nous envisageons de poursuivre nos travaux selon plusieurs axes complémentaires :

- études de systèmes d'agents cognitifs et le mécanisme de coopération entre ces agents;
- étude des rapports entre l'organisation de l'*AIAR* et son activité;
- étude du mécanisme de négociation et d'interaction entre les agents dans un environnement cognitif et en validant expérimentalement nos modèles;
- étude de l'acquisition et formalisation des connaissances dans un cadre multi-agents et la conception d'outils logiciels permettant d'automatiser ces opérations.

Chapitre 9

conclusion

Dans cette thèse, nous avons présenté une approche basée sur le raisonnement à base de cas pour modéliser le raisonnement de l'apprenant. Nous avons développé plusieurs aspects de ce problème, y compris la représentation des connaissances, les modes d'interactions, les processus de modélisation, une architecture pour le modèle de l'apprenant, etc. Nous avons également, développé et expérimenté deux prototypes appelés SARA et SARA-II.

Dans ce chapitre nous résumons la mise en œuvre de notre approche, puis nous la comparons avec d'autres approches, comme celles qui utilisent les techniques du RBC pour leur tâche de modélisation; nous énumérons ensuite ses avantages et ses limites, nous discutons également des avantages de notre formalisme de *Grphe Classifié Concept et Relation* (GCR), et finalement nous présentons les extensions possibles de notre recherche.

9.1 La réalisation de l'approche

Cette thèse avait comme objectif d'étudier et de modéliser les connaissances et le raisonnement de l'apprenant dans un STI, puis de développer et d'implanter des prototypes basés sur ce modèle pour le valider.

Nous avons appliqué des techniques de raisonnement à base de cas pour parvenir à cette problématique.

Nous avons proposé une architecture pour le modèle de l'apprenant qui contient deux composants : le *modeleur* et le *modèle de l'apprenant*, ce dernier contient deux types de connaissances (*cognitives* et *inférences*); rappelons que nous ne prenons pas en compte dans la tâche de modélisation, le modèle affectif à cause de sa complexité, que nous pensons néanmoins utile pour une représentation fidèle de l'apprenant.

Nous avons proposé un formalisme appelé *GCR* pour la représentation des connaissances du système. Ce formalisme est utilisé pour représenter les cas et les problèmes et pour organiser la base de cas et celle des problèmes.

Nous avons développé deux modes d'interactions, *AS mode* et *US mode*. Le premier mode, dans lequel les cas et solutions sont accessibles par les étudiants, est employé pour des débutants, le second pour lequel les cas et solutions ne sont pas accessibles, est employé pour des étudiants avancés ou intermédiaires.

Nous avons mis en œuvre deux algorithmes (un pour *AS mode* et l'autre pour *US mode*) permettant d'analyser les solutions de l'apprenant afin de créer son modèle.

Nous avons illustré notre démarche par deux séries d'exemples, le premier dans le domaine de la Théorie des Nombres et le deuxième dans celui du calcul intégral. Ces exemples ont été choisis en raison de leur simplicité qui permet une compréhension intuitive de l'approche.

Nous avons détaillé les processus de modélisation de l'apprenant dans les deux modes en soulignant que l'on considère uniquement les actions que l'apprenant entreprend lors de la résolution de problèmes dans le premier mode. Dans le second mode, on utilise une heuristique afin de déterminer la solution experte la plus appropriée qu'il faudra comparer à la solution de l'apprenant.

Pour valider notre approche nous avons développé deux prototypes de STI appelés SARA et SARA-II. Le premier est développé pour examiner le *AS mode*, et le deuxième est une extension du premier et peut employer de façon adaptative les deux modes (le *AS mode* et le *US mode*).

Nous avons aussi discuté de la possibilité de perfectionnement de SARA et SARA-II au moyen d'un agent intelligent dont nous avons développé une architecture globale pour un système multi-agents (appelé SARA-III), ainsi que l'architecture destinée à un agent intelligent d'aide au raisonnement (*AIAR*).

Enfin, notre approche peut aussi bien modéliser le raisonnement de l'apprenant (*modèle d'inférences*) que ses connaissances (*modèle cognitif*). Nous sommes convaincus qu'elle a des avantages significatifs par rapport à d'autres approches.

9.2 Comparaison avec d'autres approches

Pour montrer que l'approche de CBSM (Case-Based Student Modelling) a des avantages théoriques, nous la comparons à d'autres approches de modélisation de l'apprenant et nous discutons de ses forces et faiblesses.

En général, un modèle de l'apprenant basé sur des cas diffère d'autres types de modèles. Par exemple, une variété d'approches peuvent discerner la connaissance que l'étudiant a maîtrisé, mais seulement quelques-unes adressent la capacité et les inférences de l'étudiant comme nous le faisons. Nous croyons que notre approche peut en outre modéliser des aspects de l'étudiant dans lequel d'autres approches sont faibles. Par exemple, une difficulté avec la technique du traçage de modèle [Anderson *et al.*, 1990] est qu'elle ne reflète pas exactement les processus cognitifs de l'étudiant. Le modèle d'étudiant basé sur des cas peut refléter non seulement la compréhension de l'étudiant d'un problème (ou d'un concept) et la relation de ce problème avec d'autres problèmes mais il peut encore refléter comment la compréhension de l'étudiant d'un concept peut changer et évoluer avec le temps. D'une certaine manière, notre travail peut être considéré comme la construction d'un genre de graphe génétique [Goldstein, 1982]. Les relations génétiques du type proposées par Goldstein tendent vers l'évolution de la connaissance de l'étudiant [Brech et Jones, 1988]. Nous croyons que notre approche est supérieure à celle des graphes génétiques en raison de sa capacité à traiter différents modes de raisonnement de l'étudiant. Cela a nécessité d'avoir un mécanisme pour surveiller l'évolution (dans le temps) des stratégies et des processus cognitifs d'un étudiant ce qui permet finalement d'obtenir un modèle détaillé des stratégies qu'il emploie. Évidemment, cette connaissance est très importante pour le système, puisqu'elle peut être utilisée pour plusieurs buts, comme par exemple pour fournir les explications appropriées à l'étudiant. Ceci peut être réalisé en analysant le comportement de l'étudiant (alors qu'il fait de

nombreuses tentatives pour résoudre des problèmes semblables) et en prenant en compte les stratégies qu'il a apparemment maîtrisées ou celles avec lesquelles il a de la difficulté.

Un autre problème avec la technique du traçage de modèles est qu'elle utilise un modèle simple appelé le modèle idéal qui se compose d'un ensemble de règles de production pour interpréter les actions des étudiants. Mais nous savons de la recherche sur la théorie en science cognitive que les étudiants (au moins dans quelques domaines comme les mathématiques) n'appliquent pas toujours la même stratégie de résolution pour résoudre des problèmes semblables. Les débutants comptent souvent sur les analogies extérieures entre les problèmes, alors que les étudiants avancés tendent à utiliser des analogies fonctionnelles plus profondes. Dans notre modèle au lieu d'une solution générale simple, nous considérons toutes les solutions possibles qui peuvent être utilisées pour chaque cas afin de détecter les solutions des étudiants.

9.3 Comparaison avec des travaux similaires

Dans cette section nous allons comparer notre approche avec d'autres approches qui utilisent aussi les techniques du RBC (ou les techniques d'analogie) pour la modélisation des usagers ou des apprenants.

Certains chercheurs [Escott et McCalla, 1988] ont développé une approche de modélisation de l'apprenant en utilisant les techniques de résolution de problèmes basées sur l'analogie. Comme nous l'avons mentionné au chapitre 3, une description possible du rapport entre le RBC et l'analogie est indiqué dans [Leake, 1996] :

RBC = Recherche + Analogie + Adaptation + Apprentissage.
--

La modélisation d'apprenants par les techniques d'analogie souffre du problème de structuration de la mémoire, qui a une importance majeure dans notre approche.

Le CBSM résout des difficultés de modélisation des apprenants par l'analogie. De plus, il complète cette méthode de modélisation par un mécanisme de mémorisation et d'extraction d'expériences.

Nous utilisons beaucoup d'aspects d'organisation de la mémoire dans notre processus de modélisation. Cette organisation présente plusieurs avantages par exemple :

1. Elle nous permet d'offrir des cas à l'apprenant permettant d'utiliser leurs solutions pour résoudre un problème donné. La sélection des cas appropriés par l'apprenant est utilisée, pour pouvoir déterminer la capacité de mémorisation.
2. Elle nous permet d'exprimer les relations et la fonction de similarité entre les problèmes et les cas. Ces relations sont utilisées pour déterminer les solutions qui nous permettent également d'indiquer le traçage des connaissances qui doivent être comparées avec la solution de l'apprenant.
3. Elle permet à l'étudiant d'acquérir les relations (entre un problème donné et les cas similaires) qu'il a pu extraire.

Des chercheurs [Blin et Quiniou, 1996] [Papagni *et al.*, 1997] [Micarelli *et al.*, 1998] ont utilisé les techniques du raisonnement à base de cas pour développer des méthodes de modélisation des usagers. Ils ont mis l'accent sur la classification des usagers en 'stéréotypes', où chaque 'stéréotype' est la description d'un utilisateur représentatif. Étant donné un usager utilisant le système, le modèle le plus semblable à celui de l'étudiant peut être recherché dans la base de connaissances par ce système afin de personnaliser son interaction avec lui.

Cependant, la classification des usagers en 'stéréotypes' peut être difficile, et parfois inapplicable pour des STI. Car, chaque étudiant a les connaissances, les caractéristiques, et les préférences, qui lui sont propres.

Quant à nous, nous utilisons les techniques du RBC pour classifier la base de connaissances (des problèmes et des cas), et non les usagers (ou les apprenants). Nous construisons alors les modèles d'apprenants selon leurs états des connaissances, leurs caractéristiques et préférences, etc., détectés et évalués par le système au cours des interactions.

9.4 Avantages et limites de l'approche théorique

Nous voyons également comme avantages de notre approche, les aspects suivants :

- L'acquisition de connaissances est simplifiée. La base de connaissances en tant que cas modulaires, plutôt qu'en tant qu'ensemble de règles, peut considérablement réduire les efforts d'acquisition initiaux de la connaissance et même en améliorer la mise à jour [Vargas et Raj, 1993].
- L'approche de RBC (et en général l'analogie) est une façon naturelle de penser chez l'être humain.
- Cette approche est assez générale pour être adaptée à différentes applications.
- Elle n'a pas besoin d'algorithmes de calcul, en dehors de la fonction de similarité.
- Elle est également facile à mettre en application.

Les inconvénients de l'approche de CBSM sont :

- L'application de CBSM exige le développement de critères d'indexation et la réutilisation de cas antérieurs.
- La connaissance sous forme de cas peut ne pas être aisément disponible dans la plupart des domaines. Si elle l'est, elle doit être alors complètement analysée et reformulée afin de la transformer sous forme des cas.

9.5 Les avantages du formalisme de GCR

Nous devons souligner ici que le formalisme de GCR est également un principe fondamental pour nous. C'est parce qu'une des forces de notre approche se situe dans l'acquisition (de façon implicite) de la connaissance sur les modes de raisonnement de l'étudiant. L'efficacité de cette tâche dépend partiellement du GCR qui nous permet de représenter différents modes de solutions dans la représentation d'un cas.

Les avantages du formalisme de GCR peuvent être analysés selon deux points de vue. D'abord, du point de vue des réalisateurs de STI ; le GCR est un modèle étendu des graphes

conceptuels qui sont des représentations flexibles, flexibilité qui nous donne la capacité de l'utiliser pour des usages multiples. Par exemple, pour représenter les connaissances du domaine (des cas et des problèmes, voir le chapitre 5) et aussi bien pour représenter les modèles d'apprenants (voir chapitre 6). En second lieu, du point de vue des réalisateurs des systèmes à base de cas, le GCR a un certain nombre de capacités intéressantes; par exemple, il permet à un système de capturer une description raisonnablement complète (discuté ci-dessous) d'un cas.

La représentation raisonnablement complète est une représentation qui peut inclure toutes les informations sur un cas qui est susceptible d'être utile. Comme discuté dans [McCartney et Sanders, 1990], les représentations complètes augmentent la capacité de raisonnement. Ainsi, afin de maximiser leur utilité, les représentations de cas devraient être aussi complètes que possible. En particulier, notre représentation permet d'exprimer les relations entre les problèmes et les cas, d'exprimer les modifications qui sont nécessaires pour adapter les cas précédents à la situation actuelle, et de représenter la relation entre un cas (ou problème) et ses solutions.

9.6 *Expérimentation*

Afin de montrer que la modélisation de l'apprenant par cas offre des possibilités de travail intéressantes, nous avons conçu une expérience pour tester l'approche. À cette fin, deux systèmes ont été développés. Nous avons d'abord mis en application SARA [Shiri *et al.*, 1998c] pour examiner le *mode AS*. Ensuite nous avons employé nos expériences résultant de cette mise en place pour développer SARA-II : un système qui utilise *AS mode* pour des débutants et *US mode* pour les étudiants intermédiaires et avancés.

Ces systèmes ont été testés par un groupe d'étudiants (30) de l'Université de Montréal et ont été également présentés à divers ateliers et conférences [Shiri *et al.*, 1998e]. Les commentaires étaient généralement positifs, et les recommandations données ont motivé un certain nombre de choix présentés dans ce travail. Nous effectuons actuellement d'autres expérimentations afin de comprendre le potentiel final de l'approche du CBSM.

9.7 Recherches futures

Il existe un certain nombre d'extensions pour des travaux futurs qui pourraient être menés. Parmi ces extensions, nous pouvons citer :

9.7.1 Automatiser l'acquisition des connaissances

Le coût d'acquisition des cas et des problèmes peut être élevé, en particulier si un système de RBC utilise beaucoup de cas. L'acquisition des cas inclut les coûts de la représentation et de la classification des différents cas.

Dans la plupart des systèmes de RBC (aussi bien que dans des STI), les cas initiaux sont dérivés de l'expérience du créateur du système ou d'un autre expert, plutôt que d'un document. En conséquence, une traduction automatisée simple n'est pas possible. La nécessité d'encoder un cas à la main rend l'acquisition des cas plus coûteuse.

Dans cette optique (acquisition et formalisation de connaissances); nous nous intéressons dans un cadre de systèmes tutoriels à base de cas, à la conception d'outils logiciels permettant d'automatiser ces opérations, sinon complètement, au moins partiellement grâce aux techniques d'explication de connaissances.

9.7.2 Intégrer les résultats aux problèmes complexes

Nous nous intéressons également à la possibilité d'intégrer les résultats de notre travail à la résolution de problèmes complexes. Dans cette situation, l'étudiant doit remplacer la résolution du problème initial par la résolution d'une suite de sous-problèmes de même nature mais de taille et de complexité inférieure, et ensuite il doit combiner les solutions des sous-problèmes afin d'obtenir une solution globale.

Une des difficultés du raisonnement à base de cas est de savoir comment adapter un cas aux conditions de la nouvelle situation. C'est particulièrement difficile dans les domaines où il est nécessaire de combiner plusieurs cas pour trouver une nouvelle solution, comme les inconsistances locales parmi les cas secondaires qui empêchent souvent la synthèse efficace de la solution globale.

9.7.3 Améliorer notre prototype vers des systèmes utilisables dans plusieurs domaines

Les systèmes présentés dans cette thèse sont des prototypes utilisables pour des domaines spécifiques. Pour le passage du prototype à un système utilisable dans d'autres domaines, plusieurs axes doivent être développés. D'une part, les résultats doivent être intégrés dans d'autres domaines, d'autre part, les instruments doivent être conçus spécialement pour le système. Enfin, la validation générale doit être terminée.

9.7.4 Intégrer l'idée de l'agent intelligent aux STI

Dans ce cadre, nous nous intéressons à l'utilisation des techniques d'agents intelligents dans une architecture de STI, en particulier pour développer un agent intelligent d'aide au raisonnement. Dans cette optique, nous avons considéré l'idée d'un agent délibératif pour développer une architecture AIAR.

Bien que nous ayons commencé à appliquer l'idée des agents intelligents aux systèmes de modélisation de l'apprenant, il reste beaucoup de choses à faire. L'intégration d'une telle approche pour notre objectif (c.-à-d., pour le développement d'un agent intelligent pour l'aide au raisonnement) nécessite également le besoin : (1) de spécifications d'une architecture pour une société d'agents cognitifs qui permet à cette équipe d'agents de réaliser leurs tâches de façon coopérative, (2) de déterminer les tâches à exécuter par chaque agent de la société, (3) de spécifier des protocoles de communication et de coopération ainsi que les moyens de mise en œuvre des différentes lois sociales dans cette société d'agents, etc. Il est donc évident que l'évolution de la recherche sur un agent intelligent pour l'aide au raisonnement dépend de l'avancement des recherches sur les aspects énoncés ci-dessus.

Bibliographie

- [Aamodt et Plaza, 1994] A. Aamodt, E. Plaza.
Case based reasoning : Foundational issues, methodological variations and system approaches.
AI Communications, vol. 7, no. 1, pages 39-59, 1994.
- [Agre et Chapman, 1987] P. E. Agre, D. Chapman.
PENGI : an implementation of theory of activity.
In *Proceedings of the Sixth National Conference on Artificial Intelligence*, San Mateo, CA, pages 286-272. Morgan Kaufman, 1987.
- [Aha, 1991] D. W. Aha.
Case-based Learning Algorithms.
Proceedings of a Workshop on Case-based Reasoning, DARPA '91, Washington, pages 147-158.
Morgan Kaufmann, 1991.
- [Aïmeur, 1994] E. Aïmeur.
METIS : un système et une Méthode d'Explication de Taxinomies destinés à l'identification de Structures conceptuelles.
Thèse de doctorat, Université Paris VI, 1994.
- [Aïmeur et Frasson, 1995] E. Aïmeur, C. Frasson.
Eliciting the learning context in co-operative tutoring systems.
IJCAI-95 Workshop on Modelling Context in Knowledge Representation and Reasoning, Montréal, pages 1-11, 1995.
- [Aïmeur et Frasson, 1996] E. Aïmeur, C. Frasson.
Analyzing a new learning strategy according to different knowledge levels. *Computer and Education, An International Journal*, vol. 27, no. 2, pages 115-127, 1996.
- [Aïmeur et Fahmi, 1998] E. Aïmeur, M. Fahmi.
Pedagogical Agents in DTL : the Double Test Learning Strategy.
Workshop 2 on Pedagogical Agents in ITS'98, International Conference on Intelligent Tutoring Systems, San Antonio, Texas, pages 14-19, 1998.
- [Aïmeur et Boudina, 1999] E. Aïmeur, K. Boudina.
Financial Analysis by Case Based Reasoning.
Workshop on Exploring Synergies of Knowledge Management and Case-Based Reasoning, AAI'99, Sixteenth National Conference on Artificial Intelligence, Orlando, Florida, pages 1-5, 1999.
- [Alessi et Trollip, 1985] S. M. Alessi, S. R. Trollip.
Computer-Based Instruction: Methods and Development.
Prentice-Hall, Inc., Englewood Cliffs, NJ, 1985.
- [Alterman, 1988] R. Alterman.
Adaptive planning.
Cognitive Science, vol. 12, pages 393-421, 1988.

- [**Anderson, 1984**] J. R. Anderson.
Cognitive Psychology and intelligent tutoring.
Proceedings of the Cognitive Science Society Conference, Boulder, Colorado, pages 37-43, 1984.
- [**Anderson, 1988**] J. R. Anderson.
The expert module.
In *Polson, M.C. and Richardson J.J. (Eds.), Foundations of Intelligent Tutoring Systems*, Hillsdale, New Jersey, pages 21-53, Lawrence Erlbaum associates Publishers, 1988.
- [**Anderson et al., 1985**] J. R. Anderson, C. F. Boyle, G. Yost.
The geometry tutor.
In *Proceedings of the International Joint Conference of Artificial Intelligence*, Los Angeles, pages 1-7, 1985.
- [**Anderson et al., 1990**] J. R. Anderson, C. F. Boyle, A. T. Corbett, M. W. Lewis.
Cognitive modeling and intelligent tutoring.
Artificial Intelligence, vol. 42, pages 7-49, 1990.
- [**Anderson et Pelletier, 1991**] J. R. Anderson, R. Pelletier.
A development system for model-tracing tutors.
In *Proceedings of the International Conference of the Learning Sciences*, Evanston, IL, pages 1-8, 1991.
- [**Anderson et al., 1995**] J. R. Anderson, A. T. Corbett, K. Koedinger, R. Pelletier.
Cognitive tutors : Lessons learned.
The Journal of the Learning Sciences vol. 4, no. 2, pages 167-207, 1995
- [**Arsac, 1987**] J. Arsac.
Les machines à penser. Des ordinateurs et des hommes.
Éditions du Seuil, 1987.
- [**Ashley, 1991**] K. D. Ashley.
Reasoning with cases and hypothetical in Hypo.
International Journal of Man-Machine Studies, vol. 34, pages 753-796, 1991.
- [**Bandura, 1971**] A. Bandura.
Psychological Modeling.
Aldine Atherton, Chicago, 1971.
- [**Bareiss, 1989**] E. R. Bareiss.
Exemplar-based knowledge acquisition: A unified approach to concept representation, classification, and learning.
Academic Press, Boston, 1989.
- [**Barr et al., 1976**] A. Barr, M. Beard, R. C. Atkinson.
The computer as a tutorial laboratory: The standard BIP project.
International Journal of Man-Machine Studies, vol. 8, pages 567-596, 1976.
- [**Barr et Davidson, 1981**] A. Barr, J. Davidson.
Representation of knowledge.

In Barr, A, and Feigenbaum, E.A. (Eds.), *The handbook of Artificial Intelligence vol. I.*, pages 141-222, William Kaufmann, 1981.

[Barr et Feigenbaum, 1982] A. Barr, E. A. Feigenbaum.
The handbook of Artificial Intelligence vol. 2.
HeurisTech Press, William Kaufmann, Inc., 1982.

[Bestougeff et Fargette, 1982] H. Bestougeff, J. P. Fargette.
Enseignement et ordinateur.
Presses universitaires de France, Paris, 1982.

[Blin et Quiniou, 1996] J. M. Blin, R. Quiniou.
Raisonnement à partir de cas pour programmation logique inductive.
In *JAVA '96, Journées francophones sur l'apprentissage*, Sète, France, 1996.

[Bichindaritz, 1994] I. Bichindaritz.
Apprentissage de concepts dans une mémoire dynamique : raisonnement à partir de cas adaptable à la tâche cognitive.
Thèse de doctorat, Université Rene Descartes, Paris V, 1994.

[Booch et al., 1999] Grady Booch, James Rumbaugh, Ivar Jacobson.
The Unified modeling language user guide.
Reading Mass.; Don Mills, Ont., Addison-Wesley, 1999.

[Branting, 1990] K. L. Branting.
Integrating rules and precedents for classification and explanation: automating legal analysis.
Technical Reports AI90-146, Artificial Intelligence Laboratory, Department of Computer Sciences, University of Texas at Austin, 1990.

[Brecht et Jones, 1988] B. Brecht, M. Jones.
Student models: the genetic graph approach.
International Journal of Man-Machine Studies, vol. 28, pages 483-504, 1988.

[Brener et al., 1998] W. Brener, R. Zarnekow, H. Witing.
Intelligent Software Agents, Foundations and Applications.
Springer-Verlag, Berlin, Heidelberg, 1998.

[Brown et Burton, 1978] J. S. Brown, R. R. Burton.
Diagnostic models for procedural bugs in basic mathematical skill.
Cognitive Science, vol. 2, pages 155-192, 1978.

[Brown et VanLehn , 1980] J. S. Brown, K. VanLehn.
Repair theory: a generative theory of bugs in procedural skills.
Cognitive Science, vol. 4, pages 379-426, 1980.

[Brown et al., 1982] J. S. Brown, R. R. Burton, J. Kler.
Pedagogical, natural language and knowledge engineering in Sophie I, II and III, in D.
In *Sleeman and J.S. Brown (Eds.), Intelligent tutoring systems*, pages 227-282, Academic Press, New York, 1982.

[Brusilovsky et Pesin, 1994] P. Brusilovsky, L. Pesin.
An intelligent learning environment for CDS/ISIS users.
In *Proceedings of the international conference on complex learning in computer environments*, (CLCE94), EIC, Joensuu, Finland, pages 29-33, 1994.

[Burns et Capps, 1988] H. L. Burns, C. G. Capps.
Foundations of intelligent tutoring system: an introduction.
In *Polson, M.C. and Richardson, J.J. (Eds.) Foundation of Intelligent Tutoring Systems*, pages 21-53, Lawrence Erlbaum Associates Publishers, 1988.

[Burton et Brown, 1976] R. R. Burton, J. S. Brown.
Tutoring and student modelling paradigm for gaming environments.
SIGCSE Bulletin, vol. 8, pages 236-246, 1976.

[Burton et Brown, 1979] R. R. Burton, J. S. Brown.
An investigation of computer coaching for informal learning activities.
International Journal of Man-Machine Systems, vol. 11, pages 5-24, 1979.

[Burton, 1982] R. R. Burton.
Diagnosing bugs in a simple procedural skill.
In *Sleeman, D. H. and Brown, J.S. (Eds.) Intelligent Tutoring Systems*, pages 157-183. Academic Press, London, 1982.

[Byrne et Edward, 1995] C. Byrne, P. Edward.
Collaborating to Refine Knowledge.
Proceedings of Machine Learning 95, Workshop on Agents that learn from Other Agents, Tahoe City, California, 1995.

[Carbonell, 1970a] J.R. Carbonell.
AI in CAI : An artificial approach to computer-assisted instruction.
IEEE Transaction on Man-Machine systems, vol. 11, pages 190-202, 1970.

[Carbonell, 1970b] J. R. Carbonell.
Mixed-initiative Man-Computer instructional Dialogues.
Doctoral Dissertation. Massachusetts Institute of Technology, Cambridge, Massachusetts, 1970.

[Carr et Goldstein, 1977] B. Carr, I. P. Goldstein.
Overlay: A theory of modeling for computer aided instruction.
In *Artificial Intelligence Memo 406 (Logo Memo 40)*, Massachusetts Institute of technology, pages 27-33, 1977.

[Chan et Baskin, 1990] T. W. Chan, A. B. Baskin.
Learning companion systems.
In *Frasson, C. and Gauthier, G. (Eds.), Intelligent Tutoring Systems : At the Crossroads of Artificial Intelligence and Education*, pages 6-33. New Jersey : Ablex Publishing Corporation, 1990.

[Charles, 1995] S. Charles.
Le Langage de l'intelligence Artificielle.
Éditions Méthodes et Stratégies, 1995.

[Chevallier, 1992] R. Chevallier.

STUDIA : un système tutoriel intelligent coopératif fondée sur la négociation et sur un modèle dynamique de dialogue.

In C. Frasson, G. Gauthier and G.I. McCalla (Eds.) : *Intelligent Tutoring Systems, Second International Conference, ITS'92 Proceedings*, Montréal, Canada, pages 58-65, Springer-Verlag, 1992.

[Chi et al., 1989] M. Chi, M. Bassok, M. Lewis, P. Reimann, R. Glaser.

Self-explanation: how students study and use examples in learning to solve problems.

Cognitive Science, vol. 13, pages 145-182, 1989.

[Clancey, 1979] W. J. Clancey.

Tutoring rules for guiding a case method dialogue.

International Journal of Man-Machine Studies, vol. 11, pages 24-49, 1979.

[Clancey, 1987] W. J. Clancey.

Knowledge-based Tutoring : The GUIDON Program.

MIT Press, Cambridge, Massachusetts, 1987.

[Collins et Quillian, 1969] A. Collins, R. Quillian.

Retrieval time from semantic memory.

Journal of Verbal Learning and Verbal Behavior, vol. 4, pages 240-247, 1969.

[Dave et al., 1994] B. Dave, G. Schmitt, S.G. Shih, L. Bendel, B. Falting, I. Smith, K. Hua, S. Bailey, J.M. Ducret, K. Jent.

Case-based spatial design reasoning.

Proceedings Second European Workshop on Case-Based Reasoning, Paris, pages 115-124. Springer-Verlag Berlin Heidelberg, 1994.

[Davis et al., 1977] R. Davis, B. G. Buchanan, E. H. Shortliffe.

Production Rules as a Representation for a Knowledge Based Consultation System.

Artificial Intelligence, vol. 8, no. 1, pages 15-46, 1977.

[Dieng, 1990] R. Dieng.

Relation Linking Cooperating Agents.

In *Proceedings of the Second European Workshop MAAMAW'90*, Saint-Quentin en Yvelines-Frances, pages 185-202, 1990.

[Du et McCalla, 1991] Z. Du, G. McCalla.

CBMIP – A Case-Based Mathematics Instructional Planner.

In L. Birnbaum (ed.), *The Proceedings of the International Conference on the Learning Sciences*, pages 122-129. Association for the Advancement of Computing in Education. Evanston, Illinois, 1991.

[Elorriga et al., 1995] J. A. Elorriga, I. Fernandez-Castro, J. Gutierrez.

Case-based reasoning for self-improving intelligent tutoring systems.

In Jonassen, D. and McCalla, G. (Eds.), *International Conference on Computers in Education, Proceedings of ICCE'95*, Singapore, pages 259-266, Assoc. Advancement of Comput. Educ., Charlottesville, VA, USA, 1995.

[Escott et McCalla, 1988] J. A. Escott, G. I. McCalla.
 Problem Solving by Analogy : A Source of Errors in Novice Lisp Programming.
 In C. Frasson (Ed.) : *Intelligent Tutoring Systems, First International Conference, ITS'88 Proceedings*, Montréal, pages 312-320, 1988.

[Falkenhainer et al., 1990] B. Falkenhainer, K. D. Forbus, D. Gentner.
 The structure-mapping engine : Algorithm and examples.
Artificial Intelligence, v. 41, pages 1-63, 1990.

[Farrel, 1987] R. Farrel.
 Intelligent Case Selection and Presentation.
 In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence, IJCAI-87*, Milan, Italy, no. 1, pages 174-176, 1987.

[Folckers et al., 1996] J. Folckers, C. Möbus, O. Schröder.
 An Intelligent Problem Solving Environment for Designing Explanation Models and for Diagnostic Reasoning in Probabilistic Domains.
 In *Proceedings of the third International Conference on Intelligent Tutoring System (ITS'96)*, Montréal, pages 353-363. Berlin, Springer-Verlag, 1996.

[Forbus, 1988] K. D. Forbus.
 Intelligent computer-aided engineering.
AI magazin, Fall, pages 23-36, 1988.

[Frasson et de La Passadière, 1990] C. Frasson, B. de La Passadière.
 A student model based on learning context.
 In *Proceedings of the International Conference on Advanced Research on Computer in Education, IFIP*, Tokyo, Japan, pages 231-336, 1990.

[Frasson et Gauthier, 1989] C. Frasson, G. Gauthier.
 Intelligent Tutoring Systems: At the Crossroads of Artificial Intelligence and Education.
 Norwood, NJ : Ablex, 1989.

[Frasson et Kaltenbach, 1991] C. Frasson, M. Kaltenbach.
 Strengthening the Novice-Expert Shift Using the Self-Explanation Effect.
 In Greer, J.E. and McCalla, G.I. (Eds.) : *Student Modelling: The Key to Individualised Knowledge-Based Instruction*, pages 191-210, NATO-ASI Series F, Springer-Verlag, 1991.

[Frasson et Gauthier, 1994] C. Frasson, G. Gauthier.
 A gradual software environment for developing tutoring systems.
Advances in Artificial Intelligence, Theory and Application II, vol. 2, pages 73-79, 1994.

[Frasson et al., 1996a] C. Frasson et al.
 Rapport semestriel d'activités. Projet SAFARI.
 Université de Montréal, Laboratoire Héron, 1996.

[Frasson et al., 1996b] C. Frasson, T. Mengelle, E. Aïmeur, G. Gouardères.
 An Actor-based Architecture for Intelligent Tutoring Systems.
 In *Proceedings of the Third International Conference on Intelligent Tutoring Systems, ITS'96*, Montréal, pages 57-65, Springer-Verlag Berlin Heidelberg, 1996.

- [Frasson et al., 1997]** C. Frasson, T. Mengelle, E. Aïmeur.
Using Pedagogical Agents In a Multistrategic Intelligent Tutoring System.
Workshop on Pedagogical agents in AI-ED 97, World Conference on Artificial Intelligence and Education, Japan, pages 40-47, 1997.
- [Frasson et al., 1998]** C. Frasson, L. Martin, G. Gouardères, E. Aïmeur.
LANCA : A Distance Learning Architecture Based on Networked Cognitive Agents.
Fourth International Conference on Intelligent Tutoring Systems, ITS'98, San Antonio, Texas, pages 594-604. Springer Verlag, Berlin Heidelberg, 1998.
- [Fuji et al., 1996]** T. Fuji, T. Tanigawa, M. Kozeni, T. Inui, T. Saegusa.
A Case-Based Approach to Collaborative Learning for Systems Analyst Education.
In *Proceedings of the Third International Conference on Intelligent Tutoring Systems, ITS'96*, Montréal, pages 96-105, Springer-Verlag Berlin Heidelberg, 1996.
- [Gagné, 1976]** R. M. Gagné.
Les principes fondamentaux de l'apprentissage : application à l'enseignement.
Les Éditions HRW Ltée, Montréal, 4^e édition, 1976.
- [Gagné, 1985]** R. M. Gagné.
The condition of learning and the theory of instruction.
CBS College Publishing, Fourth ed., 1985.
- [Gagné et al., 1992]** R. M. Gagné, L. J. Briggs, W. W. Wager.
Principles of Instructional Design.
Harcourt Brace Jovanovich College, Publisher. Fort Worth, Texas, 1992.
- [Gilbert et al., 1995]** D. Gilbert, M. Aparicio, B. Atkinson.
Intelligent Agent Strategy.
Rapport technique, Research Triangle Park, IBM Corporation, 1995.
- [Goldstein, 1982]** L. P. Goldstein.
The genetic graph : a representation for the evolution of procedural knowledge.
In *Sleeman, D.H. and Brown, J.S. (Eds.), Intelligent Tutoring Systems*, pages 51-78, Academic Press, London, 1982.
- [Gupta et al., 1994]** U. G. Gupta, J. H. Bradley, K. R. MacLeod.
Using case based reasoning systems to teach quantitative model selection.
In *1994 Proceedings Decision Science Institute. 1994 Annual Meeting, vol. 1*, pages 366-368, Decision Sci. Inst, Atlanta, GA, USA, 1994.
- [Hall, 1989]** R. P. Hall.
Computational approaches to analogical reasoning : A comparative analysis.
Artificial Intelligence, vol. 39, pages 39-120, 1989.
- [Hammond, 1986]** K. J. Hammond.
CHEF: A Model of Case-Based Planning.
In *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, Pennsylvania, pages 267-271. Menlo Park, Calif.: American Association for Artificial Intelligence, 1986.

- [Hammond, 1989]** K. J. Hammond.
Case-based planning : Viewing planning as a memory task.
Academic Press, Boston, 1989.
- [Haton et al., 1991]** J. P. Haton et al.
Le raisonnement en intelligence artificielle.
InterEdition, Paris, 1991.
- [Hawkes et Derry, 1990]** L. W. Hawkes, S. J. Derry.
Error diagnosis and fuzzy reasoning techniques for intelligent tutoring systems.
Journal of Artificial Intelligence in Education, vol. 1, pages 43-56, 1990.
- [Hennessy et Hinkle, 1992]** D. H. Hennessy, D. Hinkle.
Applying case-based reasoning to autoclave loading.
IEEE Expert, vol.7, no. 5, pages 21-26, 1992.
- [Herzog, 1992]** Ch. Herzog.
From elementary knowledge schemes towards heuristic expertise-Designing an ITS in the field of parallel programming.
In C. Frasson, G. Gauthier and G.I. McCalla (Eds): *Intelligent Tutoring Systems, Second International Conference, ITS'92 Proceedings*, pages 96-105, Springer-Verlag Berlin Heidelberg, 1992.
- [Hinrichs et Kolodner, 1991]** T. Hinrichs, J. Kolodner.
The roles of adaptation in case-based design.
In *Proceedings of AAAI-91*, Cambridge, MA: AAAI Press / MIT Press, 1991.
- [Hollan et al., 1987]** J.D. Hollan, E. L. Hutchins, L. Weitzman.
STEAMER : an interactive inspectable simulation-based training system.
In G. Kearsley (ed.), *Artificial intelligence and Instruction*, Addison Wesley, pages 113-134, 1987.
- [Holt et Wood, 1990]** P. Holt, P. Wood.
Intelligent tutoring systems : a review for beginners.
Canadian Journal of Educational Communication, vol. 19, no. 2, pages 107-123, 1990.
- [Holt et al., 1994]** P. Holt, S. Dubs, M. Jones, J. Greer.
The state of student modelling.
In Greer, J.E. and McCalla, G.I. (Eds.) : *Student Modelling: The Key to Individualized Knowledge-Based Instruction*, pages 3-35, NATO-ASI Series F, Springer-Verlag, 1994.
- [Holyoak et Thagard, 1989]** K. J. Holyoak, P. R. Thagard.
Analogical mapping by constraint satisfaction.
Cognitive Sciences, pages 295-355, 1989.
- [Johnson et Soloway, 1985]** W. L. Johnson, E. M. Soloway.
Proust : an automatic debugger for Pascal programs.
Byte, vol. 10, no. 4, pages 179-190, 1985.
- [Kearsley, 1982]** G. Kearsley.
Authoring systems in computer based education.

Communications of the ACM, vol. 25, no. 7, pages 429-437, 1982.

[Kitano et al., 1992] H. Kitano, A. Shibata, H. Shimazu, J. Kajihara, A. Sato.
Building Large Scale and Corporate-Wide Case-Based Systems.
In *Proceedings of the Tenth National Conference on Artificial Intelligence, San Jose, California*,
pages 843-849. Menlo Park, Calif. : AAAI Press, 1992.

[Kodratoff et Barès, 1991] Y. Kodratoff, M. Barès.
Base Terminologique de l'intelligence Artificielle.
Technique et documentation, Lavoisier, Paris, 1991.

[Kolodner, 1980] J. L. Kolodner.
Retrieval and Organizational Strategies in Conceptual Memory.
Ph.D. Thesis, Yale University, 1980.

[Kolodner, 1983a] J. L. Kolodner.
Maintaining Organization in a Dynamic Long-T Memory.
Cognitive Science, vol. 7, no. 4, pages 243-280, 1983.

[Kolodner, 1983b] J. L. Kolodner.
Reconstructive Memory : A Computer Model.
Cognitive Science, vol.7, no. 4, pages 281-328, 1983.

[Kolodner, 1991] J. L. Kolodner.
Improving Human Decision Making Through Case-Based Decision.
AI Magazine, vol. 2, no. 2, pages 52-68, 1991.

[Kolodner, 1992] J. L. Kolodner.
An Introduction to Case-Based Reasoning.
Artificial Intelligence Review, vol. 6, pages 3-34, 1992.

[Kolodner, 1993] J. L. Kolodner.
Case-Based Reasoning.
Morgan Kaufmann Publisher, Inc., San Mateo CA, 1993.

[Kolodner, 1997] J. L. Kolodner.
Educational Implication of Analogy, A view From Case-Based Reasoning.
American Psychologist, vol. 32, no. 1, pages 57-66, 1997.

[Koton, 1988] P. Koton.
Reasoning about Evidence in Causal Explanation.
In *Proceedings of the Seventh National Conference on Artificial Intelligence*, Saint Paul,
Minnesota, pages 256-261. Menlo Park, Calif. : American Association of Artificial Intelligence,
1988.

[Labidi et Lejouad, 1993] S. Labidi, W. Lejouad.
De l'intelligence Artificielle Distribuée aux systèmes Multi-Agents.
Rapport de recherche no. 2004, Institut national de recherche en informatique et en automatique,
Août 1993.

[Langley et Ohlsson, 1984] P. Langley, S. Ohlsson.

Automated cognitive modelling.

In *Proceedings of the Fourth National Conference on Artificial Intelligence, AAAI-84*, Austin, Texas, pages 193-197. MA: AAAI Press / MIT Press, 1984.

[Lansky, 1975] P. Lansky.

L'informatique éducationnelle.

In *École d'été UCODI*, Genève, 1975.

[Lantz et al., 1983] B. S. Lantz, W. S. Bergar, A. M. Farley.

An intelligent CAI system for teaching equation solving.

Journal of Computer-Based Instruction, vol. 10, pages 32-42, 1983.

[Lê, 1998] Tang-Ho Lê.

Planification de l'enseignement individualisé dans un système tutoriel intelligent à grand échelle.

Thèse de doctorat, Université de Montréal, 1998.

[Leake, 1994] D. B. Leake.

Toward a Computer Model of Memory Search Strategy Learning.

Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society, Hillsdale, N.J., pages 549-554, 1994.

[Leake, 1996] D. B. Leake.

CBR in Context : The Present and Future.

In *D. B. Leake (Ed.). Case-Based Reasoning : Experiences, Lessons and Future Directions*. Pages 3-31. AAAI Press, Menlo Park, California, 1996.

[Leake et al., 1996] D. B. Leake, A. Kinley, D. Wilson.

Learning to Improve Case Adaptation by Introspective Reasoning and CBR.

In *D. B. Leake (Ed.). Case-Based Reasoning : Experiences, Lessons and Future Directions*. Pages 3-31. AAAI Press, Menlo Park, California, 1996.

[Lefevre, 1984] J. M. Lefevre.

Guide pratique de l'EAO.

Édition Cedic Nathan, 1984.

[Lesgold et al., 1992] A. Lesgold, S. Lajoie, M. Bunzo, G. Eggan.

A coached practice environment for an electronics troubleshooting job.

In *J.L. Larkin and R.W. Chabay (Eds), Computer-assisted instruction and intelligent tutoring system : shared goals and complementary approaches*, pages 201-238, Lawrence Erlbaum Associates, Hillsdale, NJ, 1992.

[López de Mántaras et Plaza, 1997] R. López de Mántaras, E. Plaza.

Case-Based Reasoning : An Overview.

AI Communication Journal, vol. 10, no. 1, pages 21-29, 1997.

[Mandl et Lesgold, 1988] H. Mandl, A. M. Lesgold.

Learning issues for intelligent tutoring systems.

Springer Verlag, New York, 1988.

[Marcenac et Herin-Aime, 1988] P. Marcena, D. Herin-Aime.

Une approche orientée objet pour les systèmes tuteurs intelligents.

In *Proceedings of the first International Conference on Intelligent Tutoring Systems, ITS-88*, Montréal, pages 342-350, 1988.

[Maes, 1995] P. Maes.

Artificial Life meets Entertainment : Lifelike Autonomous Agents.

Communications of the ACM, vol. 38, no. 11, pages 108-114, ACM Press, 1995.

[McCalla et al., 1982] G. I. McCalla, L. Reiser, P. F. Schneider.

Plan creation, plan execution and knowledge acquisition in a dynamic microworld.

International Journal of Man-Machine Studies, vol. 16, pages 82-112, 1982.

[McCalla et Greer, 1991] G. I. McCalla, J. E. Greer.

Granularity-Based Reasoning and Belief revision in Student Models.

In *Greer, J. E. and McCalla, G. I. (Eds.), Student Modelling : The Key to Individualized Knowledge-Based Instruction*, pages 39-63, NATO-ASI Series F, Springer-Verlag, 1991.

[McCalla et al., 1996] G. I. McCalla, F. Searwar, J. Thomson, J. Collins, Y. Sun, B. Zhou.

Analogical user modelling : A case study in individualized information filtering.

In *UM-96, Proceedings of Fifth international Conference on User Modeling*, Kailua-Kona, Hawaii, pages 13-21, Published by User Modeling, Inc., 1996.

[McCartney et Sanders, 1990] R. McCartney, K. E. Sanders.

The case for cases : a call for purity in case-based reasoning.

In *proceedings of the AAAI Symposium on Case-based Reasoning*, pages 12-16, 1990.

[McCartney , 1993] R. McCartney.

Episodic cases and real-time performance in a case-based planning system.

Expert Systems with Applications, v. 6, pages 9-22, 1993.

[Mengelle et Frasson, 1996] T. Mengelle, C. Frasson.

A multi-Agent architecture for an ITS with multiple strategies.

In *A. Diaz de Ilarraza Snchez and I. fernandez de Castro (Eds.) : Computer Aided Learning and Instruction in Science and Engineering, Third International Conference "CALISCE'96" Proceedings*, San Sebastián, Spain, pages 96-105, Springer-Verlag Berlin Heidelberg, 1996.

[Meyer, 1979] M. Meyer.

Découverte et justification en science.

Editions klincksieck, Paris, 1979.

[Micarelli et al., 1998] A. Micarelli, F. Sciarrone, L. Ambrosini and V. Cirillo.

A Case-Based Approach to User Modelling.

Fourth European Workshop on Case-Based Reasoning, EWCBR'98, Dublin, Ireland, pages 310-322. Springer Verlag, Berlin Heidelberg, 1998.

[Miller et Dollard, 1941] N. E. Miller, J. Dollard.

Social learning and imitation.

New Haven, CT : Yale University Press, 1941.

[Miller, 1988] J. R. Miller.

The role of human-computer interaction in intelligent tutoring systems.

In *Polson, M.C. and Richardson, J. J. (Eds.), Foundation of Intelligent Tutoring Systems*, pages 143-191. Lawrence Erlbaum Associates, 1988.

[Minsky, 1981] M. Minsky.
A Framework for Representing Knowledge.
Mind Design, Edited by J. Haugeland. pages 95-128. Cambridge, MA: The MIT Press, 1981.

[Newell et Simon, 1972] A. Newell, H. A. Simon.
Human Problem Solving.
Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1972.

[Nkambou, 1996] R. Nkambou.
Modélisation des connaissances de la matière dans un système tutoriel intelligent : modèles, outils et applications.
Thèse de doctorat, Université de Montréal, 1996.

[Ohlsson, 1986] S. Ohlsson.
Some principles of intelligent tutoring.
Instructional Science, vol. 14, pages 293-326, 1986.

[Oléron, 1977] P. Oléron.
Que sais-je Le Raisonnement.
Presses Universitaires de France, 1977.

[Papagni et al., 1997] M. Papagni, V. Cirillo, A. Micarelli.
Ocram-CBR : A Shell for Case-Based Educational Systems.
In *Proceedings of Second International Conferences on Case-Based Reasoning, ICCBR-97*. RI, USA, pages 104-114. Springer Verlag, Berlin Heidelberg, 1997.

[Pearce et al., 1992] M. Pearce, A. Goel, J. L. Kolodner, C. Zimring, L. Sentosa, R. Billington.
Case-based design support : A case Study in architectural design.
IEEE Expert, vol. 7, no. 5, pages 14-20, 1992.

[Péninou, 1993] A. Péninou.
M. A. C. T. : Un Modèle d'Agents Centrés Tâches pour la production des systèmes tuteurs intelligents par l'Atelier de Génie Didacticiel Intégré.
Doctorat de l'Université de Toulouse III, laboratoire A.P.I., 1993.

[Perelman et Olbrechts-Tyteca, 1970] C. Perelman, Olbrechts-Tyteca.
Traité de l'Argumentation.
Éditions de l'Uni. de Bruxelles, 2^e ed., 1970.

[Peter et Rösner, 1994] G. Peter, D. Rösner.
User-Model-Driven generation of instructions.
User Modeling and user-Adapted Interaction, vol. 3, no. 4, pages 289-319, 1994.

[Polson et Richardson, 1988] M. C. Polson, J. J. Richardson.
Foundation of Intelligent Tutoring Systems.
Lawrence Erlbaum Associates Publishers, Hillsdale, New Jersey, 1988.

[Polya, 1957] G. Polya.

How to solve it.
Doubleday, New York, 1957.

[Py, 1990] D. Py.
MENTONIEZH, a geometry I.T.S. for figure drawing and presetting.
Journal of Artificial Intelligence in Education, vol. 1, no. 3, pages 41-56, 1990.

[Quinlan, 1986] J. R. Quinlan.
Induction of Decision Trees.
Machine Learning, vol. 1., no. 1, pages 81-106, 1986.

[Ragnemalm, 1996] E. L. Ragnemalm.
Student diagnosis in practice; bridging a gap.
User Modeling and User-Adapted Interaction vol. 5, no. 2, pages 93-116, 1996.

[Rao et Georgeff, 1993] A. S. Rao, M. P. Georgeff.
A model-theoretic approach to the verification of situated reasoning systems.
In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, Chambéry, France, pages 318-324, 1993.

[Rao et Georgeff, 1995] A. S. Rao, M. P. Georgeff.
BDI agents : From Theory to Practice.
In *Proceedings of the first International Conference on Multi-Agent-Systems (ICMAS-95)*, San Francisco, pages 312-319, 1995.

[Rao et Georgeff, 1998] A. S. Rao, M. P. Georgeff.
Modeling Rational Agents within a BDI-Architecture.
In *M. N. Huhns and M. P. Singh (Eds.), Readings in Agents*, pages 317-328, Morgan Kaufmann Publishers Inc., San Francisco, California, 1998.

[Redmond, 1991] M. A. Redmond.
Improving Case Retrieval through Observing Expert Problem Solving.
In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pages 516-521. Ann Arbor, Mich.: Cognitive Science Society, 1991.

[Reiser et al., 1985] B. Reiser, J. A. Anderson, R. Farrel.
Dynamic student modelling in an intelligent tutor for lisp programming.
In *Proceedings of the Ninth International Joint Conference of Artificial Intelligence*, Los Angeles, pages 8-14, 1985.

[Riesbeck et Schank, 1981] C. K. Riesbeck, R. C. Schank.
Inside computer understanding.
Lawrence Erlbaum associates, 1981.

[Ross, 1989] B. H. Ross.
Some psychological results on case-based reasoning.
In *Hammond, K.J. (Ed.) : Second workshop on case-based reasoning (DARPA'89)*. Pensacola Beach, Florida, pages 144-147, San Mateo, CA: Morgan Kaufmann, 1989.

[Ross, 1990] B. H. Ross.
The access and use of relevant information: a specific case and general issues.

In R. Freedle (Ed.), *Artificial intelligence and the future of testing*, pages 173-213. Hillsdale, N.J.: Erlbaum, 1990.

[Sargent, 1992] P. Sargent.
Back to school for a brand new ABC.
In *The Guardian*, 12 March 1992, page 28, 1992.

[Sanders et al., 1997] K. E. Sanders, B. P. Kettler, J. A. Hendler.
The Case for Graph-Structured Representations.
Proceedings of Second International Conference on Case-Based Reasoning, ICCBR-97. RI, USA, pages 245-255. Springer Verlag, Berlin Heidelberg, 1997.

[Schank, 1975] R. C. Schank.
Conceptual Information Processing.
Amsterdam : North Holland, 1975.

[Schank et Abelson, 1977] R. C. Schank, R. Abelson.
Scripts, Plans, Goals and Understanding: An Inquiry into Human knowledge Structures.
Hillsdale, N.J : Lawrence Erlbaum associates, 1977.

[Schank, 1982] R. C. Schank.
Dynamic Memory : A theory of Learning in Computers and People.
New York : Cambridge University Press, 1982.

[Schank, 1996] R. C. Schank.
Goal-Based Scenarios : Case-Based Reasoning Meets Learning by Doing.
In *Case-Based Reasoning : Experiences, Lessons, and Future Directions*, edited by D.B Leake, pages 295-349, Cambridge, MA : AAAI Press / MIT Press, 1996.

[Self, 1987] J. Self.
Student Models : What use are they ?
Center for Research on Computers and Learning, University of Lancaster, 1987.

[Self, 1988] J. A. Self.
Bypassing the intractable problem of student modelling.
In *Proceedings of the first International Conference on Intelligent Tutoring Systems, ITS-88*, Montréal, pages 18-24, 1988.

[Shiri, 1997] M. E Shiri A.
Étude et modélisation des connaissances et raisonnement de l'apprenant dans un STI.
Rapport de l'examen prédoctoral oral. Département d'Informatique et de Recherche Opérationnel. Université de Montréal, 1997.

[Shiri et al., 1998a] M. E Shiri A., E. Aïmeur, C. Frasson.
Detecting the Student's Reasoning in an Intelligent Tutoring System.
Technical Report no. 1118, Dep. d'informatique et de recherche opérationnelle, Uni. Montréal. 1988.

[Shiri et al., 1998b] M. E Shiri A., E. Aïmeur, C. Frasson.
Student Modelling by Case-Based Reasoning.

Fourth International Conference on Intelligent Tutoring Systems, ITS'98, San Antonio, Texas, pages 394-404. Springer Verlag, Berlin Heidelberg, 1998.

[Shiri et al., 1998c] M. E Shiri A., E. Aïmeur, C. Frasson.
SARA : A Case-Based Student Modelling System.
Fourth European Workshop on Case-Based Reasoning, EWCBR'98, Dublin, Ireland, pages 425-436. Springer Verlag, Berlin Heidelberg, 1998.

[Shiri et al., 1998d] M. E Shiri A., E. Aïmeur, C. Frasson.
Case-Based Student Modelling : unaccessible solution mode.
Colloque International NTICF'98, INSA de Rouen, France, pages 271-279, 1998.

[Shiri et al., 1998 e] M. E Shiri A., E. Aïmeur, C. Frasson.
"SARA-II : A Case-Based Intelligent Tutoring Systems for Student Modelling",
Second Telelearning Conference, Vancouver, BC, November 14-17, 1998.

[Shiri et al., 1998f] M. E Shiri A., E. Aïmeur, C. Frasson.
Towards a Case-Based Intelligent Tutoring System for Student Modelling.
The Sixth International Conference on Computers in Education, ICCE'98, Beijing, China, pages 528-535, 1998.

[Shiri et al., 1998g] M. E Shiri A., E. Aïmeur, C. Frasson.
Interactive Student Modelling in the Case-Based Tutoring Systems.
Fourth Annual International CSI Computer Conference CSICC'98, Tehran, Iran, pages 111-117, 1998.

[Shortliffe, 1976] E. H. Shortliffe.
Computer-based medical Consultations: MYCIN.
New York: Elsevier, 1976.

[Silverman, 1992] B. G. Silverman.
Building expert critiquing systems: A situated tutoring alternative.
In C. Frasson, G. Gauthier and G. I. McCalla (Eds) : *Intelligent Tutoring Systems, Second International Conference ITS'92, Tutorial Proceedings*, Montréal, 1992.

[Simon, 1983] H. A. Simon.
Search and reasoning in problem solving.
Artificial Intelligence, Vol. 21, pages 7-29, 1983.

[Simoudis, 1992] E. Simoudis.
Using Case-Based Retrieval for Customer Technical Support.
IEEE Expert, Vol. 7, No. 5, Pages 7-13, 1992.

[Simpson, 1985] R. L. Simpson.
A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute Mediation.
Georgia Institute of Technology, School of Information and Computer Science, technical Report No. GIT-ICS-85/18, 1985.

[Sison et Shimura, 1998] R. Sison, M. Shimura.
Student Modeling and Machine Learning.

International Journal of Artificial Intelligence in Education, v. 9, pages 128-158, 1998.

[Sleeman et Brown, 1982] D. Sleeman, J. S. Brown.
Intelligent tutoring systems.
Academic Press, New York, 1982.

[Sowa, 1984] J. F. Sowa.
Conceptual structures : Information Processing in Mind and Machine.
Addison-Wesley Publication Co., 1984.

[Stevens et Collins, 1977] A. L. Stevens, A. Collins.
The goal structure of a Socratic tutor.
In *Proceedings of the National ACM Conference, Seattle, Washington*, pages 256-263,
Association for Computing Machinery, New York, 1977.

[Stevens et al., 1979] A. Stevens, A. Collins, S. E. Goldin.
Misconceptions in student's understanding.
IJMM, 11, pages 145-156, 1979.

[Sycara, 1988] K. Sycara.
Using Case-Based Reasoning for Plan Adaptation, Repair.
In *Proceedings of the Case-Based Reasoning Workshop (DARPA '88)*, Clearwater Beach, Florida,
pages 425-434, 1988.

[Sycara, 1992] K. Sycara.
CADET : A Case-Based Synthesis Tool for Engineering Design.
International Journal for Expert Systems, vol. 4., no. 2, pages 157-188, 1992.

[Tadié, 1998] S. Tadié G.
Un système multi-agents pour l'enseignement et la simulation de tâches coopératives.
Thèse de doctorat, Université de Montréal, 1998.

[Thagard et al., 1990] P. R. Thagard, K. J. Holyoak, C. Nelson, D. Gochfeld.
Analog retrieval by constraint satisfaction.
Artificial Intelligence, vol. 46, pages 259-310, 1990.

[Thayse et al., 1988] A. Thayse et co-auteurs.
Approche Logique de l'intelligence Artificielle, vol. 1 : de la logique classique à la
programmation logique.
Dunod, Bordas, Paris, 1988.

[Turner, 1988] R. Turner.
Organizing and Using Schematic Knowledge for Medical Diagnosis.
In *Proceedings of the Case-Based Reasoning Workshop (DARPA '88)*, Holiday Inn, Clearwater
Beach, Florida, pages 435-447. Morgan Kaufmann, 1988.

[Utgoff, 1989] P. E. Utgoff.
Incremental induction of decision trees.
Machine Learning, vol. 4, pages 161-186, 1989.

[VanLehn, 1988] K. VanLehn.

Student modeling.

In *Polson, M. C. and Richardson, J.J. (Eds.), Foundation of Intelligent Tutoring Systems*, pages 55-78, Lawrence Erlbaum Associates, Inc, New Jersey, 1988.

[VanLehn et al., 1994] K. VanLehn, S. Ohlsson, R. Nason.

Application of simulated students: an exploration.

Journal of artificial intelligence in education, vol. 5, no. 2, pages 135-175, 1994.

[Vargas et Raj, 1993] J. E. Vargas, S. Raj, S.

Developing Maintainable Expert System Using Case-Based Reasoning.

Expert Systems, vol. 10, no. 4, pages 219-225, 1993.

[Verdejo, 1994] M. F. Verdejo.

Building a student model for an intelligent tutoring system.

In *Greer, J.E. and McCalla, G.I. (Eds.) : Student Modelling: The Key to Individualized Knowledge-Based Instruction*, pages 147-163, NATO-ASI Series F, Springer-Verlag, 1994.

[Villano, 1992] M. Villano.

Probabilistic student models: Bayesian belief networks and knowledge space theory.

Proceedings of the second International Conference on Intelligent Tutoring Systems'92, Montréal, pages 491-498, Springer-Verlag Berlin Heidelberg, 1992.

[Wasson, 1985] B. J. Wasson.

Student models: the genetic graph approach.

Master's thesis, University of Waterloo, Ontario, 1985.

[Watson, 1997] I. Watson.

Applying Case-Based Reasoning: Techniques for Enterprise Systems.

Morgan Kaufmann Publishers Inc., San Francisco, California, 1997.

[Wenger, 1987] E. Wenger.

Artificial Intelligence and Tutoring Systems.

Morgan Kaufmann Publishers, Inc, Los Altos, 1987.

[Wooldridge et Jennings, 1995] M. J. Wooldridge, N. R. Jennings.

Intelligent Agents : Theory and Practice.

Knowledge Engineering Review, vol. 10, no. 2, pages 115-152, 1995.

[Woolf et Hall, 1995] B. P. Woolf, W. Hall.

Multimedia pedagogues : interactive systems for teaching and learning.

IEEE computer, pages 74-80, 1995.

[Zhang et Alem, 1996] D. M. Zhang, L. Alem.

Using Case-Based Reasoning for Exercise Design in Simulation-Based training.

In *Proceedings of the Third International Conference on Intelligent Tutoring Systems, ITS'96*, Montréal, pages 96-105, Springer-Verlag Berlin Heidelberg, 1996.

Annexe A

Types des raisonnements en IA

Haton *et al.* (1991) regroupent les modes de raisonnement dans les types suivants : le *raisonnement formel*, le *raisonnement procédural*, le *raisonnement par analogie*, le *raisonnement par généralisation et abstraction* et le *raisonnement géométrique*.

A.1 Le raisonnement formel

Ce raisonnement est fondé sur la manipulation syntaxique de structures symboliques à l'aide de règles, dans le cadre d'une certaine sémantique. Le raisonnement *logique*, le raisonnement *approximatif*, le raisonnement *temporel et temps réel*, et le raisonnement *hypothétique et maintien de vérité* sont des raisonnements formels que nous allons examiner.

A.1.1 Le raisonnement logique

Les raisonnements logiques peuvent être classés en trois grandes catégories qui sont : le *raisonnement déductif*, le *raisonnement inductif* et le *raisonnement abductif*.

A.1.1.1 Le raisonnement déductif

La déduction est le mode de raisonnement le plus courant, le raisonnement déductif permet de déduire des conclusions valides à partir d'un ensemble de prémisses (application de connaissances générales à un cas particulier). Il est présent dans pratiquement tous les systèmes et joue de ce fait un rôle particulier en IA. Une déduction peut être décrite par le schéma d'inférences suivant :

$$\frac{P(\alpha) \forall x (P(x) \Rightarrow Q(x))}{Q(\alpha)}$$

Il s'agit alors, d'aller du général au particulier.

➤ **Le raisonnement valide**

Un système formel de déduction de la logique classique est composé d'un ensemble de schémas d'axiomes et de règles d'inférence. Un tel système permet d'inférer des conclusions à partir de prémisses et définit donc une relation de déduction entre formules, notée (\vdash). Cette relation possède les propriétés suivantes :

- propriété de réflexibilité :
 $\{p_1, \dots, p_n, q\} \vdash q$;
- propriété de monotonie :
 si $\{p_1, \dots, p_n\} \vdash q$ alors $\{p_1, \dots, p_n, r\} \vdash q$;
- propriété de transitivité :
 si $\{p_1, \dots, p_n\} \vdash r$ et $\{p_1, \dots, p_n, r\} \vdash q$ alors $\{p_1, \dots, p_n\} \vdash q$;

Ces propriétés fondamentales traduisent formellement certaines exigences du raisonnement valide.

- Inférer une conclusion identique à l'une des prémisses est une opération valide.
- Un résultat acquis n'est pas remis en cause par des résultats ultérieurs.
- Des résultats intermédiaires peuvent être utilisés pour établir la validité d'une conclusion.

➤ **Le raisonnement révisable :**

Face à une information incomplète, incertaine ou évolutive, nos raisonnements sont souvent conjecturaux, simplement plausibles et sujets à révision. Il est clair a priori qu'un système logique permettant de formaliser pareil raisonnement révisable doit être non monotone.

Tous les raisonnements révisables ne sont pas de la même nature. Il existe des causes très diverses qui rendent nos raisonnements sujettes à révision. Il est nécessaire de comprendre les phénomènes et hypothèses qui entrent en jeu dans nos raisonnements pour définir les techniques de modélisation les plus adéquates [Thayse,1988].

Il y a deux classes de raisonnement révisable.

- Le raisonnement révisable car incertain et conjectural :

☞ Généralement, les objets de type X ont la propriété P. Si A est un objet de type X, alors je déduis que A possède (vraisemblablement) la propriété P.

La connaissance qui me permet d'inférer pareille conclusion est du type « la plupart des oiseaux volent », ou encore « un oiseau typique vole ». Ce raisonnement est tout à fait incertain et un complément d'information peut m'amener à devoir le réviser.

- Un raisonnement révisable car de nature introspective (de nombreux raisonnements révisables car incertains peuvent être modélisés par un raisonnement introspectif) :

☞ Sur la base de mon état de connaissance, je peux déduire que ...

➤ **Le raisonnement par défaut :**

Un système a rarement à sa disposition toute l'information qui lui serait utile. Mais souvent lorsqu'une telle information manque, certaines estimations sensées peuvent être faites, en l'absence de preuve contradictoire. La construction de ces estimations est connue sous le nom de raisonnement par défaut.

A.1.1.2 Le raisonnement inductif

Le raisonnement inductif vise, à l'inverse du raisonnement déductif, à généraliser des prémisses. Il consiste à aller des effets à la cause, à généraliser des faits particuliers. Il est plus difficile de raisonner de manière inductive que déductive, ce type de raisonnement est utilisé en apprentissage à partir d'exemples, en vue de définir les concepts. Une induction peut être décrite par le schéma d'inférence suivant :

$$\frac{P(a)Q(a)}{\forall x(P(x) \Rightarrow Q(x))}$$

Il s'agit alors, d'aller du particulier au général.

A.1.1.3 *Le raisonnement abductif*

Le raisonnement abductif cherche à attacher des causes plausibles à un ensemble de prémisses. Ce type de raisonnement est souvent utilisé dans la résolution de problèmes de diagnostic, de compréhension du langage naturel ou pour des problèmes liés à la reconnaissance de formes. Il intervient aussi dans les systèmes de maintien de la vérité fondés sur les hypothèses.

Une abduction peut être décrite par le schéma d'inférence suivant :

$$\frac{\forall x(P(x) \Rightarrow Q(x))Q(\alpha)}{P(\alpha)}$$

A.1.2 **Le raisonnement approximatif**

Dans la réalité, le raisonnement humain s'appuie fréquemment sur des connaissances et des données inexactes, incertaines ou encore dont l'expression verbale est elle-même entachée d'imprécision. Il y a divers aspects de cette imperfection (imprécision, incertitude, etc.) dans les activités de la vie courante ou professionnelle de chacun, comme par exemple :

- la connaissances experte : cette connaissance est souvent imprécise et incertaine, car elle représente la plupart du temps une longue pratique chez le spécialiste d'un domaine ou d'une tâche;
- l'expression de connaissances humain : l'humain utilise la plupart du temps un langage de nature qualitative pour exprimer ses connaissances;
- les sources d'information sont la plupart du temps non fiables, et alors l'information manipulée peut être partielle et non sure.

Modéliser ce type de connaissances et automatiser le raisonnement qui peut être capable de le prendre en compte est une activité importante en IA, et fait l'objet de raisonnement approximatif.

Ce type de raisonnement est ainsi lié au raisonnement par classification (pour la comparaison floue d'objets), au raisonnement par analogie et au raisonnement hypothétique (une hypothèse étant souvent assortie d'une mesure de confiance).

Les problèmes abordés : les grands types de problèmes concernés sont essentiellement le diagnostic, l'interprétation de signaux ou d'images, la prise de décision ou encore l'interrogation de bases de données.

A.1.3 Le raisonnement temporel et temps réel

Le temps représente une dimension importante et originale des systèmes à base de connaissances, raisonner sur le temps permet d'assurer un ensemble de fonctions de grande importance dans une application réelle :

- rendre compte de liens temporels entre des événements;
- raisonner sur des événements à venir;
- maintenir un ensemble de faits et de connaissances temporaires;
- planifier des actions et gérer leurs effets futurs;
- maintenir des objectifs contraints dans le temps.

Il y a deux types de raisonnement, dans lesquels le temps joue un rôle fondamental :

1. Le raisonnement temporel, permettant de raisonner sur ce qui s'est passé, sur ce qui se passera ou pourrait se passer dans l'avenir.
2. Le raisonnement en temps réel, qui utilise des contraintes de fonctionnement spécifiques au système. En *IA*, le raisonnement en temps réel implique presque toujours un raisonnement temporel.

Les problèmes abordés : les grands types de problèmes concernés sont les problèmes de planification, de conduite de procédés industriels, de gestion d'installations et plus généralement de situations dynamiques.

A.1.4 Le raisonnement hypothétique et maintien de vérité

Un raisonnement fondé sur des hypothèses permet de compléter une base de connaissances lorsque celle-ci est incomplète et inapte à produire le résultat recherché.

Une hypothèse peut se définir comme une proposition ayant une valeur de vérité indéterminée que l'on veut pouvoir fixer, elle est par nature révisable et peut donc également représenter des informations évolutives ou incertaines.

Les systèmes de maintien de vérité constituent une approche qui permet de gérer la dimension hypothétique d'un raisonnement.

Plusieurs approches de tels systèmes existent, dont TMS et ATMS qui sont les deux approches principales.

- L'approche TMS (Truth Maintenance Systems) a pour objectif de construire un état (appelé contexte) unique et cohérent de l'ensemble des déductions produit par le MR (Module de Raisonnement).
- L'approche ATMS (Assumption-based Truth Maintenance Systems) a pour objectif de gérer plusieurs contextes à la fois (et donc pour améliorer l'approche TMS). Cette méthode explore les hypothèses en parallèle.

A.2 Le raisonnement procédural

Pour ce mode de raisonnement nous allons examiner seulement le raisonnement par classification.

Le raisonnement par classification

Le sens accordé au terme «classification» en IA est celui de classement (c'est-à-dire étant donné un objet, trouver sa classe d'appartenance). Ce mode de raisonnement permet de rendre explicites les dépendances qui existent entre un nouvel objet et les objets présents dans la base de connaissances.

La mise en œuvre du raisonnement par classification peut être définie comme un cycle avec trois étapes : *instanciation* (création d'un nouvel objet), *classification* (mise en place de cet objet dans une hiérarchie H), et *propagation des modifications*.

Les problèmes abordés : en médecine, les classifications jouent un rôle fondamental. Il y a de nombreux travaux dans ce domaine pour classifier des maladies, des actes médicaux, des médicaments, etc.

Ce mode de raisonnement joue également un rôle primordial dans la recherche, la gestion et la maintenance des informations.

A.3 Le raisonnement par analogie

L'analogie est une forme de raisonnement très courante et efficace chez l'être humain. Elle consiste à inférer que ce qui est vrai pour une situation particulière, peut encore l'être pour une autre situation. C'est-à-dire deux choses qui se ressemblent sur un certain nombre de points peuvent se ressembler sur un autre.

Définition : une analogie est une ressemblance R_α entre A et B qui incite à supposer que A et B se ressemblent en tous points, et à inférer que la propriété qui caractérise A doit aussi caractériser B, et dénoter un ensemble α possédant la propriété R [Meyer, 1979].

On doit mentionner ici l'idée de Perelman et Olbrechts-Tyteca (1970), en l'occurrence que l'analogie serait une ressemblance de rapports, et non une identité de rapports de ressemblance. Ils ne disent pas :

$$\frac{A}{B} = \frac{C}{D}$$

mais ils disent A est à B comme C est à D.

Plusieurs études ont montré que les mécanismes mis en œuvre dans l'analogie se retrouvent dans d'autres types de raisonnement, en particulier le raisonnement par classification [Haton, 1991] [Hall, 1989].

Les problèmes abordés : L'analogie est une composante clé de l'intelligence, elle constitue également un outil puissant d'apprentissage symbolique, elle permet en effet la découverte de nouvelles connaissances grâce à un processus de généralisation de type

inductif. Elle est également utile dans la compréhension du langage naturel, ou l'analyse de scènes visuelles. Elle joue un rôle important dans les processus cognitifs humains.

A.4 Le raisonnement par généralisation et abstraction

Ce mode de raisonnement est lié à l'apprentissage par induction et est d'une importance majeure. Il peut en particulier mettre en œuvre le mécanisme d'inférences par héritage (il est alors lié au raisonnement par classification, voir A.2).

A.5 Le raisonnement géométrique

Ce mode de raisonnement a pour but de concevoir des systèmes recevant des informations visuelles sur le monde qui les entoure. Les domaines qui sont généralement utilisés dans ce mode de raisonnement sont : géométrie algébrique, géométrie différentielle, géométrie algorithmique, et autres types de géométrie.

Les problèmes abordés : les problèmes abordés par le raisonnement géométrique sont ceux de :

- la manipulation d'objets physiques et des décisions à prendre concernant ces objets;
- la planification de mouvements de robots;
- la synthèse d'images;
- la reconnaissance de formes;
- l'interprétation d'images;
- et la vision par ordinateur.

Annexe B

Base de connaissances

B.1 Base de cas

➤ **ID:** Case 1

Statement : If $S_0 = 1$, $S_{(n+1)} = 2^n + (S_n)$, for $n \geq 0$; show that $S_5 = (2^5 - 1)/(2 - 1)$.

Solution (Ind) : $S_1 = S_0 + 2^0 = 1 + 2^0 = (2^1 - 1)/(2 - 1)$;

$$S_{n+1} = S_n + 2^n = ((2^n - 1)/1 + 2^n) = (2^{n+1} - 1)/1;$$

$$S_5 = (2^5 - 1)/(2 - 1).$$

Solution(Ded) : $S_5 = S_4 + 2^4 = S_3 + 2^3 + 2^4 = S_2 + 2^2 + 2^3 + 2^4 = 1 + 2^2 + 2^3 + 2^4 = (2^5 - 1)/(2 - 1)$.

➤ **ID:** Case 2

Statement : If $S_0 = 1$, $S_n = 2^0 + 2^1 + \dots + 2^n$, show that $S_4 = 2^5 - 1$.

Solution(Ind) : $S_1 = 2^0 + 2^1 = (2^2 - 1)/(2 - 1)$

$$S_{n+1} = S_n + 2^{n+1} = (2^{n+1} - 1)/(2 - 1) + 2^{n+1} = (2^{n+2} - 1)/(2 - 1)$$

$$S_4 = (2^5 - 1)/(2 - 1).$$

Solution(Ded) : $S_4 = 2^0 + 2^1 + 2^2 + 2^3 + 2^4 = (2^5 - 1)/(2 - 1)$.

➤ **ID:** Case 3

Statement : If $S_0 = 1$, $S_{n+1} = 2 * (S_0 + S_1 + \dots + S_n)$, for $n \geq 0$; show that $S_4 = 2 * 3^3$.

Solution (Ind) : $S_1 = 2 * S_0 = 2 * 1 = 2 * (2 + 1)^0$

$$S_{n+1} = S_n + 2 * S_n = 2 * 2 * 3^{n-1} * (2 + 1) = 2 * 3^n$$

$$S_4 = 2 * 3^3.$$

Solution(Ded) : $S_4 = 2 * (S_0 + S_1 + S_2 + S_3) = 2 * (S_0 + 3 * 2 * S_0 + 3 * 2^2 * S_0 + 2^3 * S_0) = 2 * (2 + 1)^3 * 1 = 2 * 3^3$.

➤ **ID:** Case 4

Statement : If $S_0 = c$, $S_{(n+1)} = (1/2) * (S_n)$, for $n \geq 0$; show that $S_5 = (1/2)^5 * c$.

Solution(Ind) : $S_1 = (1/2) * S_0 = (1/2) * k = (1/2)^1 * k$;

$$S_{(n+1)} = (1/2) * S_n = (1/2) * ((1/2)^n) * k = (1/2)^{n+1} * k;$$

$$S_5 = (1/2)^5 * k.$$

Solution(Ded) : $S_5 = (1/2) * S_4 = (1/2)^2 * S_3 = (1/2)^3 * S_2 = (1/2)^4 * S_1 = (1/2)^5 * S_0 = (1/2)^5 * c$.

B.2 Base de problèmes

➤ **ID:** Problem 1

Statement : If $S_0 = a$, $S_{n+1} = b \times S_n$, for $n \geq 0$; show that $S_5 = b^5 \times a$.

Similar to : Case 4

Similarity function : $F : a \rightarrow c$, $b \rightarrow \frac{1}{2}$

➤ **ID:** Problem 2

Statement : If $S_0 = 1$, $S_{n+1} = 3^n + S_n$, for $n \geq 0$; show that $S_5 = (3^5 - 1)/2$.

Similar to : Case 1

Similarity function : $F : 3^n \rightarrow 2^n$ for all n

➤ **ID:** Problem 3

Statement : If $S_0 = 1$, $S_n = 3^0 + 3^1 + \dots + 3^n$, for $n \geq 0$; show that $S_4 = (3^5 - 1)/2$.

Similar to : Case 2

Similarity function : $F : 3^n \rightarrow 2^n$ for all n

➤ **ID:** Problem 4

Statement : If $S_0 = 1$, $S_{n+1} = a \times (S_0 + \dots + S_n)$; Show that $S_4 = a \times (a+1)^3$.

Similar to : Case 3

Similarity function : $F : a \rightarrow 2$

➤ **ID:** Problem 5

Statement : If $S_0 = b$, $S_{n+1} = k \times (S_0 + \dots + S_n)$, for $n \geq 0$; Show that $S_4 = k \times (k+1)^3 \times b$.

Similar to : Case 3

Similarity function : $F : a \rightarrow k$, $b \rightarrow 1$

➤ **ID:** Problem 6

Statement : If $S_0 = a$, $S_{n+1} = 1 \times (S_0 + \dots + S_n)$, for $n \geq 0$; Show that $S_4 = a \times 2^3$.

Similar to : Case 3

Similarity function : $F : a \rightarrow 1$, $1 \rightarrow 2$

Annexe C

Modélisation les bases de connaissances

Nous avons utilisé une approche orientée-objet (Java) pour l'implémentation de nos systèmes. Chaque élément du langage est représenté comme un objet ayant des attributs et répondant à des messages. Les objets constituent les entités d'exécution de base dans un système orienté-objet. Ils regroupent des données et des procédures appelées opérations ou méthodes.

Une classe se définit comme étant le modèle ou la description d'une famille d'objets semblables. Il s'agit aussi d'un mécanisme pour définir de nouveaux types. Ainsi, la classe sert de modèle pour créer ses représentants, c.-à-d., les instances.

La modélisation par objet saisit la structure statique d'un système, en montrant les objets dans le système, leurs relations, ainsi que les attributs et les opérations qui caractérisent chaque classe. Ce mode de conception de système présente un intérêt plus important par rapport à la programmation traditionnelle puisque le modèle objet correspond étroitement au monde réel. Il est par conséquent plus souple pour les modifications.

Nous allons maintenant proposer des concepts du modèle d'UML dans le but de pouvoir modéliser nos bases de connaissances.

C.1 Le langage de modélisation UML

Le langage de modélisation UML (Unified Modeling Language) a été défini sous l'égide de Rational Software (éditeur d'outils logiciels) par Grady Booch, James Rumbaugh et Ivar Jacobson (1999). UML est un langage pour spécifier, visualiser, construire, et documenter les artefacts des systèmes informatiques, ainsi que pour la modélisation des entreprises et autres systèmes non-informatiques.

Le standard UML permet aux informaticiens, à l'instar des architectes en bâtiments, de concevoir des plans de systèmes afin de gérer et développer efficacement les solutions d'affaires.

Avant d'aborder le problème de modélisation, il serait bon de présenter certains concepts généraux utiles à la compréhension du travail.

Le vocabulaire et les notations de base

Nous présentons par la suite les notations principales de cette approche :

Relation : Une connexion sémantique entre les éléments de modélisation. Les relations comprennent les associations et les généralisations.

Association : Une relation qui décrit un ensemble de liens.

Association binaire : Une association entre deux classes. Un cas particulier d'une association n-aire.

Agrégation : Une forme spéciale d'association qui spécifie une relation "tout-partie" entre l'agrégat (le tout) et une partie.

Composition : Une forme d'agrégation qui exprime une forte propriété entre le tout et les parties, ainsi qu'une subordination entre l'existence des parties et du tout.

Généralisation : Une relation taxonomique entre un élément plus général et un élément plus spécifique. La généralisation est représentée par une flèche qui pointe de la classe la plus spécialisée vers la classe la plus générale.

Dépendance : Une relation entre deux éléments de modélisation, par laquelle un changement dans un élément de modélisation (l'élément indépendant) va également provoquer un changement dans l'autre élément de modélisation (l'élément dépendant).

En UML, les éléments graphiques suivants (Tableaux C.1) symbolisent les relations entre objets :






Notation	Symbole
agrégation	
composition	
dépendance	
association	
Généralisation	

Tableau C. 1 : Notation UML

Multiplicité : Une spécification des valeurs de cardinalité admissibles pour un ensemble. Elle précise par exemple, combien d'instances de classe peuvent se rattacher à une seule instance de classe associée. UML définit la syntaxe suivante :

1	un et un seul
0..1	zéro ou un
m .. n	de m à n
*	plusieurs
0..*	de zéro à plusieurs
1..*	de un à plusieurs

Classe : Une description d'un ensemble d'objets qui partagent les mêmes attributs, opérations, méthodes, relations et contraintes. Une classe réalise un type.

Diagramme : Une représentation graphique d'une collection d'éléments de modélisation, le plus souvent visualisée comme un graphe d'arcs (relations) et de sommets (autres éléments de modélisation).

Diagramme de classes : Un diagramme qui montre un ensemble d'éléments de modélisation déclaratifs (statiques), comme les classes, les types, avec leurs contenus et relations.

Une classe est dessinée dans UML comme rectangle avec 3 compartiments séparés par les traits horizontaux. Le compartiment supérieur décrit le nom de classe; le compartiment moyen décrit une liste d'attributs et le compartiment inférieur décrit une liste d'opérations (voir la Figure C.1)

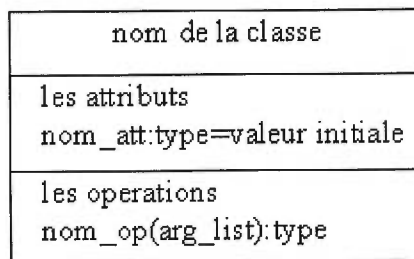


Figure C. 1: Diagramme de classe

Les interfaces des classes : UML permet de représenter une interface des classes (qui décrit le comportement visible de la classe) au moyen d'un cercle qui leur est associé ou par une classe avec le stéréotype <<interface>>.

C.2 La modélisation des bases de connaissances

Comme nous l'avons montré au chapitre 5, nous avons structuré le plus possible les connaissances nécessaires à la résolution de problèmes. Nous avons distingué deux types essentiels de connaissances : les problèmes et les cas. En effet, nous définissons différentes classes pour ces connaissances. Dans leur forme actuelle, nos bases de connaissances sont un ensemble organisé de classes dont les principales sont : la classe `ProblemBase`, la classe `CaseBase`, la classe `Problem`, la classe `SolvedProblem`, etc. Chacune des classes possède des attributs, des messages (ayant été spécifiés dans la représentation) et des comportements propres. La Figure C.2 illustre la base de connaissances intégrée sous forme d'objets.

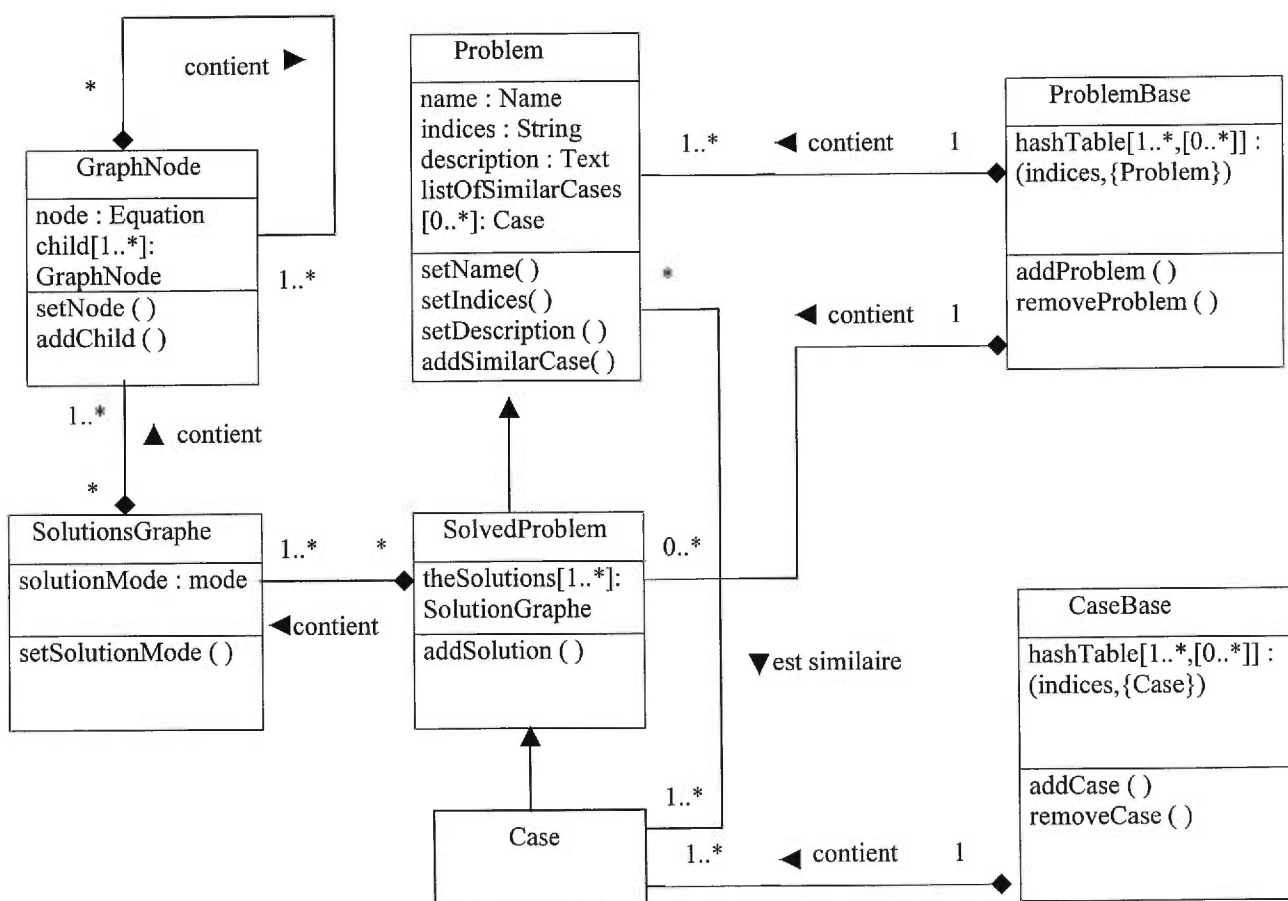


Figure C. 2: La base de connaissances intégrée sous forme d'objets

C.3 La modélisation du modèle de l'apprenant

Pour modéliser et structurer toutes les connaissances relatives au modèle de l'apprenant, nous avons introduit des classes qui regroupent différents attributs concernant ces informations. Certaines de ces classes sont présentées à la section précédente. La classe principale contient trois catégories de fonctions : les fonctions d'accès aux informations du modèle de l'apprenant, les fonctions de mise-à-jour du modèle et les fonctions dédiées à l'initialisation du modèle. La Figure C.3 illustre le modèle objet de ce dernier ainsi que les liens qui existent entre les différentes classes.

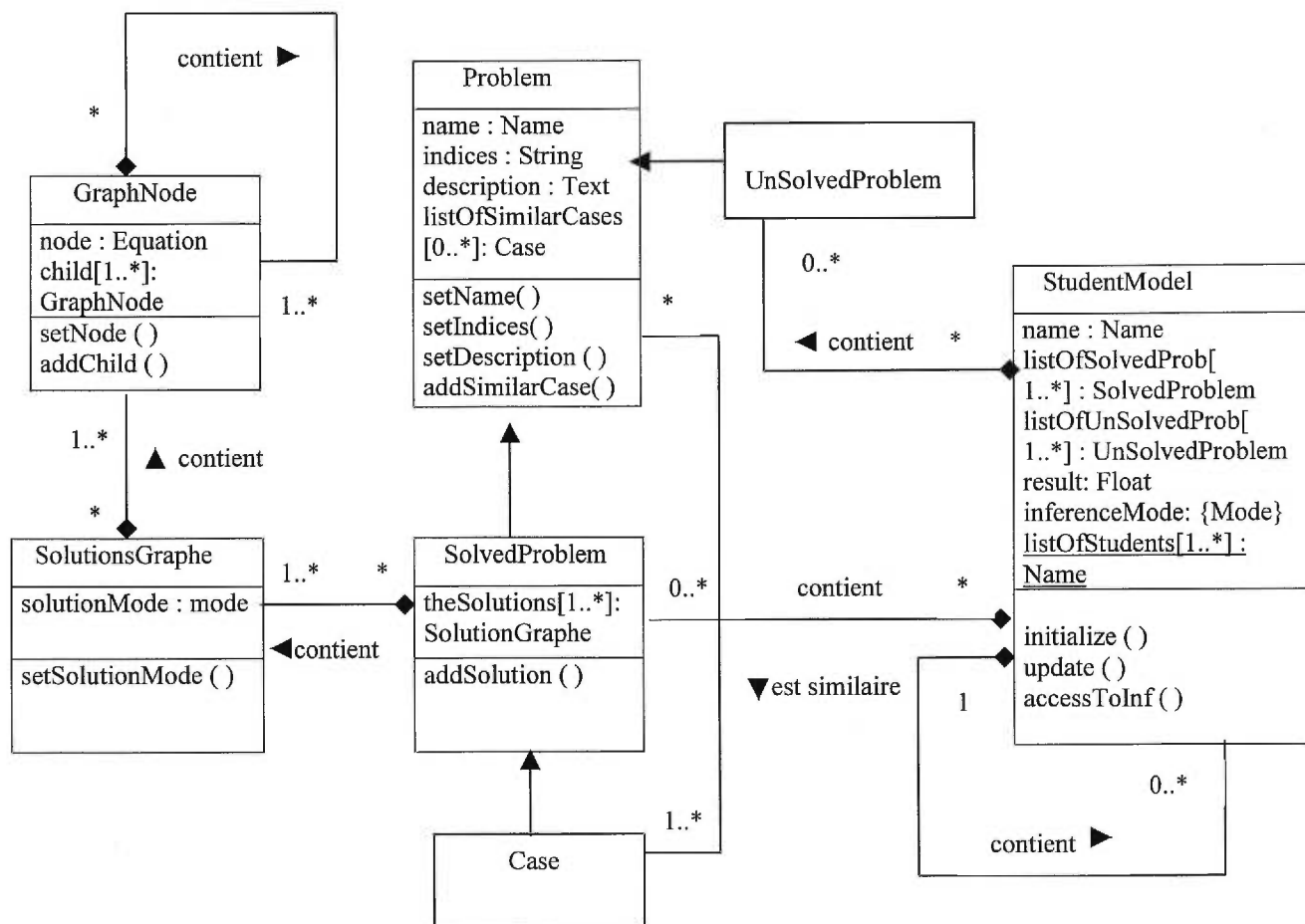


Figure C. 3: Digramme du modèle de l'apprenant

C.4 Le digramme d'état global de SARA-II

Nous illustrons maintenant par la Figure C.4, la vue du modèle statique du système SARA-II. Cette Figure montre les interactions entre les objets et les liens entre eux.

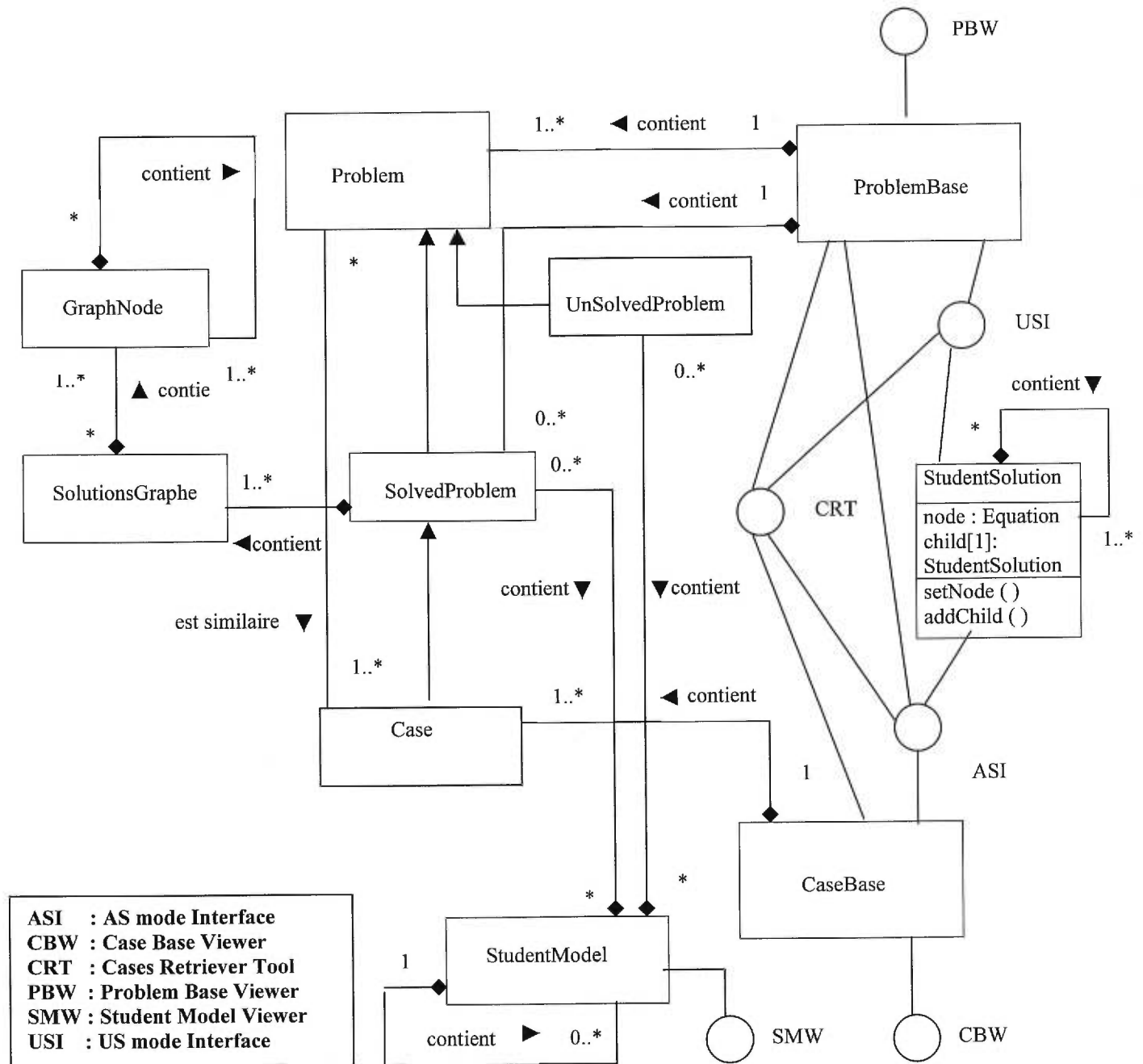


Figure C. 4: Diagramme d'état global (SARA-II)