



HAL
open science

Une approche pro-pédagogique du diagnostic cognitif dans les STI: conception, formalisation, et implémentation

Josephine Tchetagni

► To cite this version:

Josephine Tchetagni. Une approche pro-pédagogique du diagnostic cognitif dans les STI: conception, formalisation, et implémentation. 2005. hal-00257165

HAL Id: hal-00257165

<https://telearn.hal.science/hal-00257165>

Submitted on 18 Feb 2008

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

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

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

UNE APPROCHE PRO-PÉDAGOGIQUE DU DIAGNOSTIC COGNITIF DANS LES STI:
CONCEPTION, FORMALISATION ET IMPLÉMENTATION

THÈSE
PRÉSENTÉE
COMME EXIGENCE PARTIELLE DU
DOCTORAT EN INFORMATIQUE COGNITIVE

PAR
JOSÉPHINE-MURIEL PELLE-TCHÉTAGNI

OCTOBRE 2005

REMERCIEMENTS

Au terme de ce travail, j'aimerais exprimer ma profonde reconnaissance à toutes les personnes qui m'ont aidée, orientée et soutenue durant les années de doctorat.

Je remercie tout particulièrement mon directeur de thèse, Monsieur Roger Nkambou, professeur au département d'informatique à l'UQÀM, et ma codirectrice de thèse, Madame Jacqueline Bourdeau, professeure à la Télé-Université. Leur confiance, leur patience et leur générosité ont permis la réalisation de ce travail. Je leur suis particulièrement reconnaissante pour leur disponibilité constante, leurs conseils avisés, leur rigueur académique et leurs encouragements répétés.

Mes très sincères remerciements vont également à Monsieur Ghislain Lévesque, directeur du programme de doctorat en informatique cognitive à l'UQÀM, pour sa disponibilité et ses précieux conseils académiques.

Cette thèse s'inscrit dans un projet mené en partenariat avec le groupe *Infotel-Inc., Canada*, l'*Agence spatiale canadienne* et le *CRSNG-Canada*. Ces organismes ont fourni un financement qui a permis de mener à terme ce travail et mis à notre disposition des professeurs et des chercheurs dont les suggestions et les commentaires ont positivement influencé ma recherche. Je pense particulièrement au Professeur Froduald Kabanza, au Dr Jean Roy et au Dr Rolland Yatchou.

Je tiens à remercier les professeurs France Henri, Rokia Missoui et Gilbert Paquette qui ont accepté de porter leur jugement d'expert sur mon examen de synthèse et mon projet de thèse. Leurs suggestions ont largement contribué aux idées développées dans ce travail.

Mes remerciements s'adressent également à tous mes collègues du laboratoire de gestion et d'acquisition des connaissances de l'UQÀM (GDAC) et du laboratoire en informatique cognitive et environnements de formation de la Télé-Université (LICEF). Je pense particulièrement à: Khaled Belglith pour sa contribution à l'intégration de la librairie *JavaBayes* dans *Prolog-Tutor*; Jessica Faivre pour les discussions enrichissantes sur les sciences cognitives, ainsi que pour les suggestions avisées sur des questions importantes tout au long des années de doctorat; Valéry Psyché et Delia Drogozan pour leurs critiques constructives sur les articles et les prototypes qui ont été développés dans ce travail. Je tiens aussi à mentionner la présence chaleureuse et encourageante des membres du GDAC: Fatiha, Hervé, Khaled, Lilia, Mayi, Patrick, Vincent, etc. et du LICEF: Claire, Delia, Ileana, Juliana, Marcelo, Michel, Mihai, Roselyne, Valery, etc.

Pour leurs encouragements, leur soutien psychologique et leur confiance, je remercie ma famille: Joseph-Désiré, Élina et Manewa; mon père et ma mère; ma sœur Honorine et Éric; mes frères Steeve et Patrick; tous mes amis.

Je remercie enfin tous les membres du jury de me faire l'honneur de contribuer à l'évaluation de ce travail.

TABLE DES MATIÈRES

INTRODUCTION	1
Contexte de recherche.....	1
Sujet de recherche: le DC dans les STI.....	1
Problématique: la relation DC, contexte, et exploitation pédagogique du DC	2
Buts et objectifs de la thèse.....	6
Originalité de la contribution	10
Organisation de la thèse	11
CHAPITRE I	13
DIAGNOSTIC COGNITIF DANS LES STI: UNE PERSPECTIVE PRO-PÉDAGOGIQUE	13
1.1 Introduction.....	13
1.2 Une vision pro-pédagogique du DC.....	14
1.3 Paradigmes de cognition et courants de théories d'apprentissage	19
1.4 Intégration DC-exploitation pédagogique: la boucle diagnostic-remédiation	33
1.5 Conclusion	36
CHAPITRE II	38
ÉTAT DE L'ART	38
2.1 Introduction.....	38
2.2 Conception formelle du DC dans les STI	39
2.3 Conception formelle du DC et contexte pédagogique	42
2.4 . La boucle diagnostic-remédiation: stimulation de la pensée réflexive comme approche de remédiation.....	45
2.5 Conclusion	50
CHAPITRE III	52
CADRE DE SPÉCIFICATION POUR LE DIAGNOSTIC COGNITIF DANS UN STI	52
3.1 Introduction.....	52
3.2 Cadre de spécification: signification et utilités	53
3.3 CSDC de base	55
3.4 Intégration de la pensée réflexive dans la spécification du DC	72
3.5 Considération explicite du contexte pédagogique du DC	88
3.6 Exploitation du CSDC: CD-SPECIES ou un outil de support à la conception du DC	94
3.7 Conclusion	111

CHAPITRE IV	114
FORMALISATION DE LA BOUCLE DIAGNOSTIC-REMÉDIATION: UNE APPROCHE BAYESIENNE	114
4.1 Introduction.....	114
4.2 Les inférences bayésiennes.....	115
4.3 Choix d'une approche bayésienne: justification.....	121
4.4 Implication de l'approche bayésienne sur l'implémentation du DC dans un STI: le modèle causal	122
4.5 Représentation formelle du DC selon un raisonnement bayésien.....	125
4.6 Conclusion.....	132
CHAPITRE V	133
INTÉGRATION D'UNE PERSPECTIVE PRO-PÉDAGOGIQUE DU DC DANS UN STI : PROLOG-TUTOR	133
5.1 Introduction.....	133
5.2 Implémentation générale de Prolog-Tutor.....	134
5.3 Le module pédagogique de Prolog-Tutor: le diagnostic cognitif.....	146
5.4 Conclusion.....	161
CHAPITRE VI	163
ILLUSTRATIONS D'UNE PERSPECTIVE PÉDAGOGIQUE DU DC DANS UN STI: PROLOG-TUTOR	163
6.1 Introduction.....	163
6.2 Prolog-Tutor: domaine étudié et capacités visées.....	164
6.3 DC avec réflexion basée sur un dialogue tutoriel générique.....	168
6.4 . DC avec réflexion basée sur un dialogue tutoriel à structure réflexive.....	181
6.5 DC avec réflexion basée sur l'édition du modèle de l'apprenant.....	188
6.6 Discussion.....	194
6.7 Conclusion.....	196

CHAPITRE VII	198
ÉVALUATION QUALITATIVE DU CSDC ET DE LA PENSÉE RÉFLEXIVE DANS PROLOG-TUTOR	198
7.1 Introduction.....	198
7.2 Choix de l'approche qualitative d'évaluation	199
7.3 Buts de l'évaluation	200
7.4 Évaluation qualitative du CSDC	201
7.5 Évaluation qualitative de Prolog-Tutor.....	218
7.6 Conclusion	233
CONCLUSION	235
Contribution à la recherche AIED	237
Fondements cognitifs de la contribution.....	240
Leçons tirées et pistes de recherche	241
BIBLIOGRAPHIE	304

LISTE DES APPENDICES

APPENDICE. A	244
Illustrations de CD-SPECIES	244
APPENDICE. B	247
Illustration de la formalisation de la boucle diagnostic-remédiation	247
APPENDICE. C	251
Illustration de l'implémentation de prolog-tutor.....	251
APPENDICE. D	262
Caractéristiques des participants à l'évaluation et codes d'analyse des données qualitatives d'évaluation	262
APPENDICE. E	274
Transcription des enregistrements effectués lors des évaluations.....	274

LISTE DES FIGURES

Figure I.1. Démarche du Chapitre 2.....	15
Figure I.2. Perspective «pro-pédagogique» du Diagnostic Cognitif.....	37
Figure III.1 Relation entre le DC et les modules d'un STI.....	62
Figure III.2 La pensée réflexive selon Dewey	74
Figure III.3. CD-SPECIES: Interface de spécification d'une instance de DC.....	101
Figure III.4. CD-SPECIES: Interface du programmeur de STI.....	104
Figure III.5. CD-SPECIES: Interface de vérification de la cohérence.....	107
Figure III.6. CD-SPECIES : Sauvegarde et Indexation d'une spécification de DC	108
Figure IV.1. Causalité sémantique entre deux capacités dans un domaine.....	115
Figure IV.2. Déduction bayésienne Probabilité conditionnelle de A sachant B	116
Figure IV.3. Abduction bayésienne: vraisemblance de H sachant e.....	117
Figure IV.4. Probabilité de l'état global d'un réseau bayésiens de n variables.....	118
Figure IV.5. Déduction bayésienne pour la mise à jour du modèle de l'apprenant...	120
Figure IV.6. Abduction bayésienne pour le diagnostic des erreurs de l'apprenant ...	121
Figure IV.7. Formalisation de la boucle diagnostic-remédiation.....	126
Figure IV.8. Diagnostic d'une incapacité à <i>appliquer</i> une <i>unification</i> en programmation logique	129
Figure V.1. Représentation bayésienne d'une partie des capacités de Prolog-Tutor ..	135
Figure V.2. Modèle de l'apprenant dans Prolog-Tutor.....	139
Figure V.3. Prolog-Tutor: État des connaissances de l'apprenant à court terme.....	140
Figure V.4. Prolog-Tutor: Interface Principale	143
Figure V.5. Dynamique de communication dans Prolog-Tutor	144
Figure V.6. Structure des dialogues tutoriels et modèle causal associé.....	148
Figure V.7. Inférence des causes les plus probables d'une réponse incorrecte	152
Figure V.8. Adaptation et Révision du flux d'un dialogue tutoriel	157
Figure V.9. Trace des actions dans le module pédagogique	158
Figure V.10. Dynamique du module pédagogique dans Prolog-Tutor	159
Figure VI.1. Éléments de connaissances dans Prolog-Tutor.....	166

Figure VI.2. Construction dialogique d'une solution à l'interface de Prolog-Tutor...	171
Figure VI.3. Construction libre d'une solution à l'interface de Prolog-Tutor	172
Figure VI.4. Raffinement et confirmation d'un DC dans Prolog-Tutor	177
Figure VI.5. Prolog-Tutor: Dialogue dont la structure favorise la pensée réflexive...	182
Figure VI.6. Explicitation de la démarche de résolution d'un sous-but	185
Figure VI.7. Implication de l'apprenant dans sa modélisation:exercice de performance	194

LISTE DES FIGURES DE L'APPENDICE A

Figure A. 1. Dynamique de CD-SPECIES pour la visualisation d'une spécification .245

Figure A. 2. Processus opérationnels de CD-SPECIES et données associées246

LISTE DES FIGURES DE L'APPENDICE C

Figure C. 1. Prolog-Tutor: Représentation des objectifs et des ressources.....	252
Figure C. 2. Visualisation des capacités nécessaires pour répondre à une question	253
Figure C. 3. Réponse directe de l'apprenant à une question du module pédagogique	254
Figure C. 4. Interface de formulation des résultats d'une unification	255
Figure C. 5. Prolog-Tutor : Representation graphique de la résolution d'un but	256
Figure C. 6. Interface secondaire pour les questions à choix multiples	257
Figure C. 7. Interface secondaire pour prouver l'Échec d'une unification	258
Figure C. 8. Interface secondaire pour la résolution d'un sous-but.....	259
Figure C. 9. Classes définissant les patrons de questions selon le type de capacité dans la taxonomie de Gagné.....	260
Figure C. 10. Questions prédéfinies dans une base de données.....	261

LISTE TABLEAUX

Tableau I.1. Situation d'apprentissage: rationaliste versus pragmatiste	30
Tableau III.1. Un cadre de spécification de base pour le DC dans un STI	58
Tableau III.2. Illustrations concrètes des types de diagnostic cognitif	67
Tableau III.3. Deux situations de stimulation de la pensée réflexive	75
Tableau III.4. OLM et pensée réflexive d'après Dewey	78
Tableau III.5. Intégration de l'édition libre du modèle de l'apprenant dans le DC	81
Tableau III.6. Structure d'un dialogue tutoriel explicite pour la pensée réflexive.....	85
Tableau III.7. CSDC: intégration explicite de la stimulation de la pensée réflexive	87
Tableau III.8. CSDC: Paradigmes de cognition et inférence du DC.....	89
Tableau III.9. CSDC: Paradigmes de cognition et pensée réflexive.....	89
Tableau III.10. CSDC: Paradigmes de cognition et acquisition d'indice dans le DC..	93
Tableau III.11. Utilisation de CD-SPECIES.....	98
Tableau III.12. Vocabulaire dans la base de connaissances de CD-SPECIES	105
Tableau III.13. Illustration: règles de la base de connaissances de CD-SPECIES	106
Tableau IV.1. Un algorithme générique de DC basé sur l'inférence bayésienne	130
Tableau VI.1. Un sous-ensemble des objectifs d'apprentissage dans Prolog-Tutor...	167
Tableau VI.2. Les capacités reliées à l'exécution de la résolution d'un but	168
Tableau VI.3. Spécification d'une instance de DC épistémique dans Prolog-Tutor...	169
Tableau VI.4. Exemple de dialogue tutoriel tel que conçu dans Prolog-Tutor	175
Tableau VI.5. Illustration de l'usage pédagogique de l'élaboration du DC	180
Tableau VI.6. Analyse d'un dialogue pour favoriser la réflexion explicite	186
Tableau VI.7. Buts de la pensée réflexive selon Drake (1976).....	187
Tableau VI.8. Réflexion explicite pour l'application d'un principe.....	187
Tableau VI.9. Spécification d'un processus de DC: contexte empirique/béavioriste	188
Tableau VII.1. Procédure de collecte des données pour l'évaluation du CSDC	204

Tableau VII.2. Instrument de collecte de données pour l'évaluation du CSDC.....	205
Tableau VII.3. Croisement des arguments et des avis des experts concepteurs	208
Tableau VII.4. Résumés des entrevues avec les experts en conception pédagogique	209
Tableau VII.5. Résumé des données de l'évaluation de Prolog-Tutor	219
Tableau VII.6. Pensée extériorisée dans les sous dialogues de Prolog-Tutor.....	223
Tableau VII.7. Fréquences d'occurrence des omposantes de la pensée réflexive	227
Tableau VII.8. Conscience de la démarche de pensée réflexive.....	228
Tableau VII.9. Étapes auxquelles la démarche de pensée réflexive est conscientisée	229
Tableau VII.10. Fréquence de demande d'explication sur les questions des sous dialogues	230
Tableau VII.11. Fréquences d'occurrences de la pensée réflexive à des moments inattendus	232

LISTE DES TABLEAUX DE L'APPENDICE B

Tableau B.1. Approche bayésienne générique: attributs et fonctions de l'objet «modèle du domaine».....	248
Tableau B.2. Approche bayésienne générique: attributs et fonctions de l'objet «modèle de l'apprenant».....	248
Tableau B.3. Approche bayésienne générique: attributs et fonctions de l'objet «modèle causal»	249
Tableau B.4. Approche bayésienne générique: attributs et fonctions de l'objet «diagnostic cognitif».....	250

LISTE DES TABLEAUX DE L'APPENDICE D

Tableau D. 1. Caractéristiques des participants à l'évaluation du CSDC.....	263
Tableau D. 2. Codes de convergence des experts sur les questions d'entrevue.....	263
Tableau D. 3. Codes de convergence des experts les questions d'évaluation.....	263
Tableau D. 4. Expérience des participants à l'évaluation de Prolog-Tutor avec Prolog.....	264
Tableau D. 5. Échelle de maîtrise de Prolog considérée.....	264
Tableau D. 6. Codes associées aux sous dialogues et aux questions du dialogue étudié.....	265
Tableau D. 7. Codes associés aux composantes de la pensée réflexive.....	271
Tableau D. 8. Codes associés aux participants de l'expérimentation avec Prolog-Tutor	272
Tableau D. 9. Durée des enregistrements par participant	273

LISTE DES TABLEAUX DE L'APPENDICE E

Tableau E. 1. Transcription des données recueillies auprès de l'expert E2	275
Tableau E. 2. Transcription des données recueillies auprès de l'expert E3	279
Tableau E. 3. Transcription de l'enregistrement audio_1 avec Prolog-Tutor	281
Tableau E. 4. Transcription de l'enregistrement audio_2 avec Prolog-Tutor	285
Tableau E. 5. Transcription de l'enregistrement audio_3 avec Prolog-Tutor	289
Tableau E. 6. Transcription de l'enregistrement audio_4 dans Prolog-Tutor	293
Tableau E. 7. Transcription de l'enregistrement audio_5 avec Prolog-Tutor	297
Tableau E. 8. Transcription de l'enregistrement audio_6 avec Prolog-Tutor	300

LISTE DES ABBRÉVIATIONS

AIED	: <i>Artificial Intelligence in Education</i>
CD-SPECIES	: <i>Cognitive Diagnosis Specification Tools</i>
CSDC	: Cadre de spécification pour le diagnostic cognitif dans un STI
DC	: Diagnostic cognitif
IA	: Intelligence Artificielle
STI	: Système tuteur intelligent

RÉSUMÉ

Dans un STI, la question de la conception et de l'implémentation du diagnostic cognitif (DC) du comportement de l'apprenant est abordée de manière intuitive et au "cas par cas", selon les objectifs spécifiques de ce STI. D'un point de vue purement opérationnel, cette approche n'est pas vraiment néfaste puisque les STI qui en résultent "fonctionnent". Cependant lorsqu'il s'agit de rendre au DC dans un STI, la dimension pédagogique qui lui est naturellement siéenne dans le tutorat humain, un regard en perspective sur des considérations plus fondamentales favorise la réflexion dans ce sens.

Cette thèse formule, formalise et implémente explicitement *une dimension «pro-pédagogique» du DC* dans les STI. Cette dimension pro-pédagogique s'exprime principalement à travers deux relations qui n'ont jamais été explicitement étudiées et mises en pratique dans la recherche sur le DC dans les STI: la relation entre le DC et les paradigmes de cognition; la relation entre le DC et son exploitation pédagogique, en l'occurrence la remédiation des difficultés de l'apprenant, ce à travers une boucle diagnostic-remédiation.

Outre leur originalité conceptuelle, ces relations sont articulées dans un cadre de spécification pour le DC lorsqu'il s'agit de l'appliquer dans un STI. L'originalité de l'idée d'un tel cadre de référence est qu'il favorise une préservation de la fidélité pédagogique du DC au cours de son implémentation dans un système informatique.

Sur le plan informatique, l'originalité de la contribution de cette thèse est qu'elle *formalise* cette perspective du processus de DC dans un STI, notamment en systématisant la dynamique de la boucle diagnostic-remédiation à travers: (1) un algorithme générique de type «*générer et tester*», (2) l'intégration à cet algorithme de mécanismes formels de raisonnement incertain par le biais des inférences bayésiennes (pour tenir compte des facteurs d'incertitude reliés au DC dans un STI) et (3) l'implémentation de cet algorithme dans une librairie de programmes génériques et donc réutilisables par tout membre de la communauté AIED désireux d'adopter cette philosophie de diagnostic cognitif dans un STI.

INTRODUCTION

Contexte de recherche

L'usage des machines intelligentes à des fins d'enseignement remonte à 1926 (par exemple, la machine de Pressey basée sur les conditions de Thorndike (1912)). Un demi-siècle plus tard, le développement de l'intelligence artificielle (IA) a favorisé l'émergence de l'enseignement assisté par ordinateur (Computer Assisted Instruction ou CAI) suivi des systèmes tuteurs intelligents (STI) (Carbonell, 1970; Livergood, 1991; Sleeman et Brown, 1982; Solloway et Bielaczyc, 1995). Les débuts des STI coïncident avec l'hégémonisme de l'explication rationaliste de la cognition humaine (Pylyshyn, 1989) et leur conception s'en est trouvée fortement influencée. Plus tard, le pragmatisme avec des théories d'apprentissage telles que la théorie de l'activité (Leontiev, 1984) et la théorie "constructionniste" (Papert, 1999) a aussi influencé les STI. La priorité pédagogique résidait désormais dans la "manière" et l'historique du processus d'apprentissage. Dans tous les cas, l'objectif commun des STI est la capacité de fournir une assistance intelligente aux apprenants qui les utilisent. Il s'agit d'adapter les priorités pédagogiques en fonction des besoins de l'apprenant, établis sur la base d'un diagnostic cognitif (DC) de ses actions.

Sujet de recherche: le DC dans les STI

L'intelligence d'un STI tient surtout d'une capacité d'accommodation de la pédagogie aux besoins spécifiques de chaque apprenant qui l'utilise. Le défi à relever porte sur l'inférence des besoins de chaque apprenant. La fonction pédagogique dénommée "diagnostic cognitif" est le moteur de l'adaptabilité d'un STI. Au cours d'un processus de DC, les actions de l'apprenant doivent être interprétées afin d'inférer, "ce qu'il fait", "ce qu'il maîtrise" et surtout "ce sur quoi il éprouve des difficultés" (Ohlsson, 1987). Vu sous cet angle, le résultat

du processus de DC inspire les autres fonctions pédagogiques: planification de l'apprentissage, explications du tuteur, focalisation des activités d'apprentissage, etc.

Problématique: la relation DC, contexte, et exploitation pédagogique du DC

L'effectivité du DC dans le processus d'apprentissage à travers un STI dépend d'un certain nombre de considérations. Bien qu'il s'agisse d'une fonction pédagogique naturelle pour le tuteur humain (GrandBastien, 1999; Shute, Lajoie, et Gluck, 2000), le DC dans un STI se réduit à un algorithme. Mais à l'instar des tuteurs humains, cet algorithme doit toujours être considéré comme un moyen au service d'une fin pédagogique et non l'inverse. Dans ce sens, il ne suffit donc pas simplement de définir un algorithme "computationnel" associé au DC, et interprétable par une machine; une caractérisation formelle du lien entre le moyen pédagogique qu'il constitue et la fin pédagogique qu'il sert devient aussi une considération importante pour la valeur pédagogique de ce DC dans un STI. L'hypothèse principale de cette thèse souligne le fait que le contexte pédagogique associé à un STI et l'exploitation (ou la fin) pédagogique qui justifie la mise en oeuvre du DC dans ce STI en influencent la nature, la forme et l'implémentation.

Une instance du contexte pédagogique est le paradigme de cognition adopté lors de la conception du DC et dans ce sens, il exerce une influence conceptuelle sur le DC. Nous pensons que ce paramètre influence la nature du résultat ou des inférences à cibler lors d'un DC dans un STI.

Une instance d'exploitation pédagogique du DC est la remédiation des difficultés diagnostiquées chez l'apprenant. Nous pensons que dans un STI, la remédiation est plus qu'une nécessité conséquente au DC, elle doit y être intégrée à travers une *boucle diagnostic-remédiation*. Ici, la remédiation est considérée dans un sens large et non formel, comme toute action pédagogique visant à améliorer les capacités d'un apprenant par rapport à des objectifs d'apprentissage d'un STI. L'idée de "boucle" doit se comprendre dans le sens suivant: (1) les résultats du DC permettent au système sous-jacent à un STI de déterminer les points sur lesquels porteront la remédiation (influence du diagnostic sur la remédiation); (2) le comportement et les réactions de l'apprenant dans le cours de toute forme de remédiation permettent au système sous-jacent à un STI d'évaluer la qualité ou d'élaborer la substance du DC précédemment posé, d'où la boucle.

La problématique de cette thèse porte sur une considération explicite de la nature du DC (1), des relations DC-paradigmes de cognition (2) et des relations DC-exploitation pédagogique du DC (3), lors de la conception et lors de l'implémentation de cette fonction dans les STI. Il s'agit de favoriser la mise en œuvre d'une *perspective « pro-pédagogique »*. Au *niveau général*, une perspective pédagogique du DC dans les STI doit s'entendre comme une approche de conception et d'implémentation où le souci est de *préserver la fidélité pédagogique*: l'algorithme informatique associé au processus de DC dans un STI doit effectivement refléter les objectifs pédagogiques pour lesquels il a été construit. Au *niveau spécifique* une perspective pédagogique du DC dans les STI doit s'entendre comme une approche où la mise en œuvre du processus de DC dans un STI reflète: une cohérence conceptuelle eu égard au paradigme de cognition qui sous-tend ce système; une continuité pédagogique avec la remédiation que ce DC occasionne auprès de l'apprenant pour améliorer encore et toujours l'apprentissage de ce dernier. Dans ce sens, notre réflexion s'articule à travers deux hypothèses principales:

- 1) Hypothèse sur la conception du DC pour les STI. Pour être réalisée en général, la perspective pédagogique du DC doit être *explicitée*: toute instance de DC à implémenter dans un STI se réclamant de la perspective pédagogique du DC préconisée dans cette thèse, doit être *spécifiée à part entière sur la base d'un cadre de référence* reflétant cette perspective.
- 2) Hypothèse sur l'implémentation du DC pour les STI: Pour être réalisée sur le plan informatique, la perspective pédagogique doit être *formalisée*: un algorithme générique de DC reflétant cette perspective pédagogique est une manière d'en capturer formellement les caractéristiques clés.

Préserver la fidélité pédagogique du DC en général et la cohérence avec le paradigme de cognition en particulier

Étant donné la nature cognitive du sujet auquel il s'applique, l'algorithme de DC dans un STI s'appuie sur un ensemble de représentations de connaissances qui en cautionnent l'efficacité. Ces représentations concernent les différents aspects du traitement de l'information qui a lieu pendant le DC: représentation des connaissances à propos du domaine d'apprentissage, représentation des connaissances à propos de l'état cognitif de l'apprenant, représentation des connaissances pédagogiques – en l'occurrence, les connaissances à propos du DC -.

Les approches de représentation des connaissances sont variées du fait des derniers développements en IA. Le choix de l'approche adéquate de représentation est une condition indispensable au succès du DC. Nous pensons que les caractéristiques conceptuelles du DC influencent réellement ce choix: par exemple, la nature du résultat de ce DC, le contexte d'apprentissage dans lequel il se déroule, etc. En effet, elles déterminent quelles informations sont nécessaires pour la mise en œuvre adéquate d'un DC, et elles déterminent la manière dont ces informations doivent être représentées pour favoriser leur usage computationnel. On peut considérer que ces caractéristiques constituent un corpus de connaissances conceptuelles à propos du DC.

Il est donc nécessaire de formuler le lien qui existe entre chaque caractéristique considérée comme faisant partie du concept de DC, et les approches adéquates de représentation des connaissances. Dans ce sens, deux intervenants deviennent concernés: le concepteur pédagogique de STI et le programmeur de STI. Le concepteur pédagogique fournit au programmeur une spécification formelle des interactions pédagogiques – entre un apprenant et un système tuteur – qu'il désire faire émerger de la mise en œuvre d'un STI, probablement parce qu'elles favorisent la réalisation des objectifs d'apprentissage ciblés dans ce STI. Le programmeur construit un ensemble d'artéfacts informatiques dont l'exécution devrait refléter ces interactions pédagogiques. Dans la réalité, deux facteurs peuvent constituer des obstacles à cette collaboration.

D'une part, les connaissances conceptuelles sur le DC sont généralement implicites pour le concepteur pédagogique. Non seulement il doit connaître les caractéristiques du DC, mais aussi les relations d'interdépendance et de cohérence entre les différentes instances de ces caractéristiques. Par exemple, la nature du contexte pédagogique (considération béhavioriste de la cognition par exemple) influence le type de résultat du DC qui doit être établi (volume des connaissances retenues par l'apprenant).

D'autre part, même si les caractéristiques conceptuelles d'une instance de DC sont formellement spécifiées par un concepteur pédagogique, le programmeur ne maîtrise pas nécessairement la signification de ce vocabulaire pédagogique. Les mécanismes computationnels qui permettent à une machine de réaliser cette instance DC demeurent à déterminer.

La première problématique examinée dans cette thèse s'intéresse à l'écart sémantique que l'on doit combler entre le concepteur pédagogique et le programmeur de STI, pour préserver la fidélité pédagogique lors de l'implémentation d'un module de DC dans un STI.

Favoriser une continuité entre le DC et la remédiation

"L'un des aspects les plus significatifs du DC sur le plan pédagogique est sa capacité de rattacher l'application correcte ou incorrecte d'une connaissance diagnostiquée à des interactions entre les composantes conceptuelles de cette connaissance" (Wenger, 1987b). Il s'agit là d'un exemple révélateur de l'acuité qui peut caractériser l'exploitation des résultats d'un DC. Dans ce cas d'espèce, on constate qu'au-delà de déterminer quelle connaissance est utilisée et appliquée par l'apprenant (correctement ou incorrectement), ce processus doit aller plus loin à savoir, déterminer comment les interactions entre les parties de cette connaissance sont responsables des réponses correctes ou incorrectes d'un apprenant lors de la réalisation d'un exercice dans un STI.

Nous pensons que cette réalité doit être appliquée à toute initiative de DC dans un STI. Dans le DC, un résultat de premier niveau n'est pas une finalité en soi. Les opérations qui y apportent une signification pédagogique pour l'orientation de la suite de l'apprentissage sont plus importantes. Nous pensons en effet que ces opérations doivent être intégrées dans le processus de DC car elles influencent la manière dont l'algorithme correspondant doit être implémenté. Par exemple, l'algorithme qui diagnostique les actions de l'apprenant à des fins de planification de curriculum, sera différent de celui qui diagnostique les mêmes actions de l'apprenant pour pallier à ses difficultés immédiates lors de la résolution d'un exercice. Dans le premier cas, ce sont les données agrégées de performance de l'apprenant qui seront traitées par l'algorithme alors que dans le second, il sera probablement question de suivre la trace de chacune de ces actions et d'en analyser les causes.

La deuxième problématique examinée dans cette thèse s'intéresse à la considération explicite du lien entre un algorithme de DC dans un STI et l'exploitation pédagogique qui est faite des résultats de ce DC.

Exprimer formellement cette perspective « pro-pédagogique » et implémenter des programmes génériques correspondants

Les STI sont des artéfacts informatiques dont les fondements théoriques proviennent des sciences cognitives. La troisième problématique de cette thèse est de formaliser les

fondements cognitifs qui sont mis en exergue dans la perspective pédagogique du DC que nous préconisons. Il s'agit d'une formulation systématique et calculable de ces fondements, afin qu'il puissent être interprétables et mis en œuvre par un artéfact informatique tel qu'un STI, lors de son exécution.

Buts et objectifs de la thèse

Les deux premiers buts de cette thèse portent sur la formulation et l'explicitation d'une perspective pédagogique du DC dans les STI à travers: (1) l'élaboration d'un pont sémantique entre un concepteur de STI et un programmeur pour ce qui concerne le module de DC dans un STI, afin de préserver une fidélité pédagogique générale lors de l'exécution de ce module et (2) la considération explicite de la relation entre le DC, son contexte et son exploitation à des fins pédagogiques, dans le but de favoriser une cohérence conceptuelle et une continuité entre le DC et son exploitation pédagogique.

Le troisième but de cette thèse porte sur la formalisation de cette perspective pédagogique du DC à travers la définition d'un algorithme générique de DC et, l'implémentation de cet algorithme par le biais d'une librairie de programmes (objets et procédures) réutilisables.

Conception du DC: cadre de spécification pour le DC

Concernant le premier but plusieurs pistes de solution sont envisageables, par exemple: fournir aux concepteurs pédagogiques et aux programmeurs une liste de recommandations sur la conception et l'implémentation du DC dans un STI, fournir au programmeur une liste répertoriant les correspondances entre les fins pédagogiques du DC dans un STI et les artéfacts informatiques qui les favorisent lors de l'implémentation de ce STI. Dans cette thèse, notre objectif est d'élaborer un cadre de spécification pour le DC (CSDC) dans un STI. Il s'agit de décrire un canevas pour le DC ainsi que les différentes instances qui peuvent être associées à chaque élément de ce canevas.

Pertinence d'un cadre de spécification

L'avantage d'un CSDC par rapport aux listes de recommandations ou de correspondances est qu'il constitue un point de référence pour le concepteur pédagogique. Les informations que ce dernier y trouve lui permettent de caractériser une instance de DC pour un paradigme de cognition particulier.

A priori, la spécification d'une instance de DC constitue un point de départ pour le programmeur: chaque caractéristique du DC traduit une exigence au niveau des approches de représentations de connaissances qui sont nécessaires pour sa mise en oeuvre.

A posteriori, la spécification d'un processus de DC permet au concepteur pédagogique de valider la fidélité d'une implémentation. Par la traçabilité qu'elle implique (au niveau du module de DC correspondant dans un STI), une spécification peut aussi favoriser l'évolution conceptuelle du contenu du CSDC.

De plus, si l'on considère que le CSDC reflète une vision conceptuelle du DC dans un STI, on peut le percevoir comme un modèle de DC. Certains des rôles fondamentaux des modèles dans la recherche en IA et en éducation (*Artificial Intelligence in Education: AIED*) selon Baker (2000) constituent une justification fondamentale pour le choix de définir un CSDC, en l'occurrence, le modèle en tant qu'outil scientifique. En tant qu'outil scientifique, un modèle justifie toute manifestation empirique ou tangible d'un artéfact. Dans le cas du DC dans un STI, un modèle de DC permettrait donc de justifier le choix d'une approche d'implantation. La pertinence d'un modèle ressort aussi lorsqu'il est question de comparer deux artéfacts remplissant la même fonction, en l'occurrence, deux modules de DC dans un STI. En effet, des critères stables, formels et surtout communs doivent guider cette comparaison, ce que fournit un CSDC.

Méthodologie de définition d'un cadre de spécification du DC

Pour établir un cadre de spécification du DC, notre méthodologie consiste à répondre aux questions suivantes :

- 1) Quelles sont les caractéristiques conceptuelles du DC?
- 2) Quelles sont les caractéristiques opérationnelles du DC ?
- 3) Quelles instances (valeurs) peuvent être assignées à chaque caractéristique identifiée ?
- 4) Quelles sont les relations entre ces instances qui permettent de maintenir une cohérence interne au niveau de l'instance de DC qui y correspond ?
- 5) Quelles sont les relations entre les instances de valeurs associées aux caractéristiques du DC et les artéfacts informatiques qui favorisent la fidélité de la mise en œuvre de l'algorithme correspondant. Ici les artéfacts informatiques incorporent surtout les

approches de représentations de connaissances sur lesquelles l'algorithme de DC s'appuie.

Les deux premières questions sont les plus importantes car elles inspirent l'orientation des réponses aux questions subséquentes. Les connaissances de base à propos du DC seront mise en exergue et réorganisées en fonction de ces questions.

Conception du DC: cohérence conceptuelle avec les paradigmes de cognition

Un paradigme de cognition postule des hypothèses à propos de la nature de la cognition. Le diagnostic des actions de l'apprenant dans un contexte d'apprentissage pose des hypothèses à propos de l'état cognitif (ou de la cognition) de l'apprenant. De plus, étant donné l'absence quasi certaine de l'intervention humaine – à court terme du moins - la conception des STI se base sur une approche scientifique. L'utilisation des principes des paradigmes de cognition constitue un aspect caractéristique de cette méthodologie scientifique. Dans cette perspective, notre hypothèse est que le paradigme de cognition en tant qu'élément du contexte pédagogique du DC dans un STI influence la nature et – par conséquent – les opérations de ce processus. Notre deuxième objectif dans cette thèse est d'étudier et de formuler explicitement les implications qui matérialisent cette influence.

Méthodologie d'intégration des paradigmes de cognition au DC

Pour formaliser la relation entre les paradigmes de cognition et le DC, un survol de ces paradigmes est présenté dans un premier temps en s'inspirant de la revue exhaustive proposé par Greeno, Collins et Resnick (1996). Dans un second temps, à partir des prémisses propres à chacun de ces paradigmes et sur la base des caractéristiques des principaux courants de théories d'apprentissage associées, on propose des relations reflétant leur influence sur les caractéristiques conceptuelles et opérationnelles du DC et on intègre ces relations au CSDC.

Conception du DC: continuité du DC dans la remédiation ou la boucle diagnostic-remédiation

Le résultat d'un DC est un des principaux précurseurs de la suite de l'organisation de l'apprentissage à travers un STI. Au niveau microscopique, il permet de déterminer la prochaine action pédagogique. Au niveau macroscopique, il guide la réorganisation d'un plan d'enseignement, du contenu et des outils didactiques. Le principal argument en faveur de la

présence et de l'exploitation quasi permanente du DC réside dans le fait qu'il permet de cibler la pédagogie –à court et moyen termes - en fonction des besoins réels de l'apprenant.

Il existe une autre perspective de cette relation DC/pédagogie dont l'importance semble ne pas avoir été étudiée dans le domaine AIED – explicitement et formellement tout au moins. De fait, les résultats de l'application d'une action pédagogique – inspirée d'un DC – *peuvent aussi être utilisés comme une rétroaction vers ce DC*, ce en termes de:

- 1) Confirmation du DC initialement posé;
- 2) Raffinement du DC initialement posé;
- 3) Ajustement des paramètres de l'algorithme de DC initialement utilisé.

Par exemple, le résultat d'une instance de DC permet de cibler quelles connaissances feront l'objet de mesures pédagogiques de remédiation. Au terme de ces mesures, les performances de l'apprenant devraient indiquer la justesse des inférences posées lors du DC initial (bien que cela dépende aussi de l'efficacité de la remédiation elle-même). Notre troisième objectif dans cette thèse est *d'expliciter la relation co-déterminante ou la boucle entre le DC et la stimulation de la pensée réflexive* de l'apprenant – ou réflexion tout court – comme instance de remédiation de ce DC.

Méthodologie d'intégration de la boucle diagnostic-remédiation dans le DC

Pour analyser la relation co-déterminante qui existe entre le DC et la pensée réflexive (comme instance de remédiation suite à un DC), les principales phases de ce phénomène sont d'abord formulées, en s'appuyant sur les travaux de Dewey (1933) en psychologie éducationnelle. Ces phases doivent cependant être interprétées selon l'instance de DC dont il est question. Dans un deuxième temps, on définit des mises en œuvre génériques de la stimulation de la pensée réflexive dans un STI, à travers les approches "*Interactive Learner Modelling*" (Dimitrova, 2003b) et "*Open Learner Modelling*" (Vassileva, McCalla, et Greer, 2003). Enfin, un ensemble de relations entre ces deux approches et le DC est défini et intégré dans le CSDC.

Implémentation du DC: un algorithme générique basé sur une approche bayésienne

Pour réaliser le troisième but de cette thèse, un algorithme générique de DC basé sur une approche de raisonnement bayésien est développé. Le DC étant un processus de raisonnement basé sur de multiples sources d'évidences donnant lieu à plusieurs interprétations possibles.

L'adoption d'une approche bayésienne a deux avantages: elle fournit des mécanismes qui permettent de gérer l'incertitude qui caractérise les situations où il faut gérer plusieurs sources d'évidences et plusieurs interprétations possibles, elle fournit des mécanismes formels de raisonnement par abduction (propre au DC).

Méthodologie de définition d'un algorithme bayésien générique de DC

Après avoir présenté l'essence du raisonnement bayésien, les principales méthodes d'inférence bayésienne sont décrites. Ensuite, on montre comment ces méthodes sont utilisées pour réaliser les phases de base du DC ainsi que la boucle DC-remédiation (stimulation de la pensée réflexive) préconisée dans cette thèse.

Originalité de la contribution

Dans un STI, la question de la conception et de l'implémentation du diagnostic du comportement de l'apprenant est abordée de manière intuitive et au "cas par cas" selon les objectifs spécifiques de ce STI. D'un point de vue purement opérationnel, cette approche n'est pas vraiment néfaste puisque les STI qui en résultent "fonctionnent". Cependant, lorsqu'il s'agit de rendre au DC sa dimension pédagogique, un regard en perspective sur des considérations plus fondamentales pourrait favoriser la réflexion dans ce sens.

Au terme des prochains chapitres, l'originalité qui ressortira de cette thèse est qu'elle formule, formalise et implémente explicitement une dimension «pro-pédagogique» du DC dans les STI.

Cette dimension «pro-pédagogique» s'exprime principalement à travers deux relations qui n'ont jamais été auparavant, explicitement étudiées et mises en pratique dans la recherche sur le DC dans les STI: la relation entre le DC et les paradigmes de cognition; la relation entre le DC et son exploitation pédagogique, en l'occurrence la remédiation, ce à travers une boucle diagnostic-remédiation. Outre leur originalité conceptuelle, ces relations sont articulées dans un CSDC qui tient lieu de cadre de référence pour ce processus, ce afin d'en préserver la fidélité pédagogique au cours de son implémentation dans un système informatique. Le CSDC en lui-même constitue donc également une contribution originale au niveau de la conception du DC pour un STI.

L'originalité de notre contribution sur le plan informatique est que nous allons proposer une expression formelle de la boucle diagnostic-remédiation à travers une approche

bayésienne générique de raisonnement. L'algorithme correspondant de DC sera implémenté à travers une librairie de programmes génériques et donc réutilisables par tout membre de la communauté AIED désireux d'adopter cette philosophie de diagnostic dans un STI.

Organisation de la thèse

Notre réflexion sur les problématiques soulevées est présentée à travers sept chapitres.

Les deux premiers chapitres portent sur les problématiques du DC qui sont d'intérêt dans cette thèse. Leur but est de préciser conceptuellement notre position pour une approche plus pédagogique du DC dans les STI, et de dégager l'originalité de notre contribution dans ce sens. Le Chapitre I est une ébauche de la vision du DC qui est préconisée dans cette thèse. Nous y introduisons et justifions les principales composantes du CSDC. Le Chapitre II est une revue des travaux relatifs au DC dans les STI selon la perspective pédagogique qui ressort de notre CSDC. En l'espèce, il s'agit de: (1) la conception du DC dans les STI en tenant compte de l'impact d'un paramètre de contexte pédagogique; (2) l'implémentation et la mise en oeuvre du DC dans les STI en tenant compte explicitement de l'impact de son exploitation pédagogique à travers une boucle diagnostic-remédiation, plus particulièrement d'une boucle diagnostic-réflexion.

Les quatre prochains chapitres portent sur le développement de la perspective du DC préconisée dans cette thèse. Ces chapitres ont pour but de préciser en substance la contribution de cette thèse. Le Chapitre III décrit le CSDC en tant que tel afin de détailler la perspective pédagogique du DC soutenue dans cette thèse. En faveur de l'utilité de ce cadre, l'architecture de CD-SPECIES (un système informatique qui servirait de coquille au CSDC, afin de favoriser son exploitation interactive par les concepteurs pédagogiques et par des programmeurs de STI) est également présentée. Le Chapitre IV décrit formellement un algorithme générique de DC associé à la perspective pédagogique décrite au Chapitre III et basé sur une approche de raisonnement bayésien. Le Chapitre V décrit l'implémentation d'un prototype de STI – Prolog-Tutor –, qui a été développé pour concrétiser la perspective pédagogique de DC préconisée dans cette thèse. Un accent est mis sur la concrétisation de l'algorithme générique de DC présenté au Chapitre IV. Au Chapitre VI, notre but est de d'illustrer concrètement les hypothèses présentées dans les chapitres précédents. Ce chapitre décrit un ensemble de spécifications de DC issues du CSDC et illustre leur mise en oeuvre dans Prolog-Tutor, en référence à l'algorithme générique présenté au Chapitre IV.

Le Chapitre VII porte sur l'évaluation des hypothèses sur lesquelles s'appuient les notre principale contribution dans cette thèse. Ce septième chapitre réitère donc ces hypothèses avant de présenter les résultats de leur évaluation.

Chapitre I

DIAGNOSTIC COGNITIF DANS LES STI: UNE PERSPECTIVE PRO-PÉDAGOGIQUE

1.1 Introduction

Un cadre de spécification du diagnostic cognitif (DC) doit avoir un contenu significatif, sans être inutilement exhaustif. Un certain nombre de priorités doit être défini pour obtenir un tel contenu. Le DC dans un STI comporte des caractéristiques intrinsèques: le type d'inférences qui seront posées (type de résultat) et les phases du processus correspondant. Bien que cette thèse présente en détail ces éléments dans un CSDC (Chapitre II), ils ne constituent pas l'originalité de notre proposition.

Ce qui apparaît nouveau dans la perspective mise de l'avant dans cette thèse concerne l'ancrage explicite du DC dans son *contexte* et son *exploitation pédagogiques*. Cet ancrage aura des répercussions sur différentes instances de DC.

Le contexte pédagogique est un concept évasif et le choix de ses éléments dépend de chaque circonstance de conception d'un STI. Néanmoins, nous pensons que la considération de certains paramètres est pertinente: notamment, le paradigme de la cognition qui est considéré (Reigeluth, 1999). Dans cette thèse, nous examinons les implications respectives

des différents paradigmes de cognition sur les instances des caractéristiques intrinsèques du DC.

Les exploitations pédagogiques du DC sont nombreuses et variées. Entre autres, la stimulation de la pensée réflexive (aussi appelée réflexion) est une exploitation pédagogique du DC reconnue pour favoriser un apprentissage mieux fondé (Dewey, 1933). L'application de la réflexion dans les STI est une intégration nouvelle mais prometteuse (Bull, McEvoy, et Reid, 2003). Pour cette raison, nous examinons les implications sur le processus de DC, d'un besoin de stimuler la pensée réflexive chez l'apprenant.

L'objectif de ce chapitre est de décrire et de justifier cette perspective pédagogique du DC (Figure I.1). La première section présente notre vision d'ensemble du DC dans un STI, depuis sa conception jusqu'à la programmation informatique du module correspondant du STI. La seconde section décrit la nature de la pensée réflexive et justifie sa présence dans l'implémentation du DC dans un STI. La troisième section suit le même cours en ce qui concerne les paradigmes de la cognition et les théories d'apprentissage associées à ces paradigmes.

1.2 Une vision pro-pédagogique du DC

Avant de préciser les éléments qui confèrent un caractère plus pédagogique à la perspective de DC mise de l'avant dans cette thèse, ses composantes de base sont rappelées. Il s'agit des composantes du DC qui sont généralement évoquées lors de son implémentation dans les STI.

1.2.1. Le diagnostic cognitif en général

L'on peut comprendre le diagnostic cognitif comme un concept et dans ce sens, il revêt des attributs ontologiques. On peut aussi le considérer comme un processus et dans ce sens, il intègre des opérations qui auront une particularité selon qu'il s'agisse d'un contexte d'apprentissage classique ou à travers un STI.

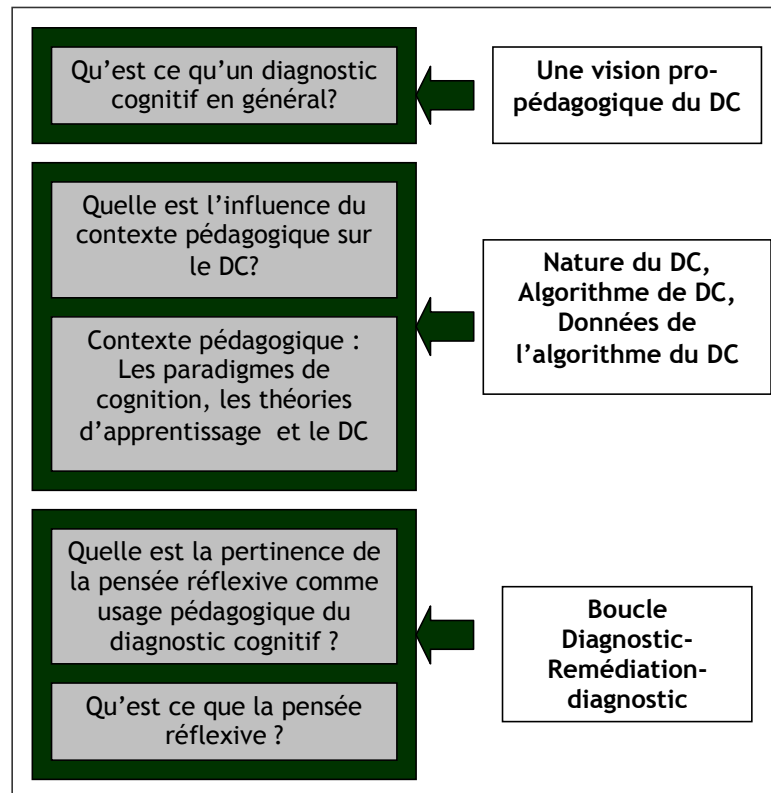


Figure I.1. Démarche du Chapitre 2

1.2.1.1. Le concept de diagnostic cognitif

Dans diagnostic cognitif, on retrouve "diagnostic". Tel que l'indique son étymologie, ce terme signifie "connaissance à travers les signes". Dans un STI, le diagnostic cognitif peut être interprété comme la connaissance de l'état cognitif de l'apprenant à travers les signes qu'il présente (ses actions et ses réactions lors d'une activité d'apprentissage). Plus concrètement, le DC est un algorithme qui utilise des représentations de connaissances sur un apprenant, un domaine de connaissances, un modèle pédagogique pour caractériser et inférer *différents aspects* de son état cognitif: ses actions et son raisonnement, ses connaissances, et finalement ses émotions.

1.2.1.2. Le processus de diagnostic en général

Pour comprendre l'algorithme général de DC dans un STI, on peut établir une analogie avec les situations de diagnostic médical: l'association entre les symptômes et les causes. Dans le

contexte d'apprentissage, il faut associer le comportement de l'apprenant pendant une activité d'apprentissage à l'état cognitif qui explique le mieux ce comportement.

Hoc (1990) a proposé une organisation de l'activité générale de diagnostic à travers cinq sous processus:

- (1) L'acquisition des indices. Cette acquisition doit être contrôlée par les prémisses qui ont été préalablement fixées sur le genre d'inférence qu'on aimerait poser (le contexte du diagnostic);
- (2) La génération d'hypothèses par inférence et l'interprétation des hypothèses;
- (3) La conclusion ou le résultat du diagnostic en tant que tel;
- (4) L'élaboration ou l'ajustement du diagnostic qui peut être utilisé pour orienter les stratégies des prochaines acquisitions d'indices;

L'acquisition des indices correspond à l'acquisition des données observées. La génération et l'interprétation des hypothèses correspondent à l'inférence propre au diagnostic. La conclusion correspond à l'établissement d'un diagnostic proprement dit. Elle consiste à établir par le biais de tests adéquats quelles hypothèses sont valides. Cela permet de déterminer les hypothèses qui seront effectivement confirmées comme résultat final du diagnostic.

1.2.1.3. Le processus de diagnostic cognitif dans un STI

Comment interpréter ce processus générique dans une situation d'apprentissage à travers un STI? Une considération explicite des recommandations des étapes 1 et 4 et des implications de l'étape 2 permet d'introduire les notions de contexte pédagogique et d'exploitation pédagogique du DC dans un STI.

De plus, la génération des hypothèses (étape 2) est un premier niveau d'inférence qui consiste à *établir* des présomptions qui expliqueraient les données observées. L'interprétation des hypothèses (étape 2) est un deuxième niveau d'inférence qui consiste à expliquer et à justifier les hypothèses générées. Dans un STI, cette étape doit être considérée comme facultative car elle n'est possible que lorsque la quantité d'informations disponibles – à propos de l'état cognitif actuel de l'apprenant et à propos du domaine appris – est bien étoffée. Elle est pertinente parce qu'elle peut guider le choix des hypothèses qui seront confirmées: celles dont l'interprétation est la plus plausible ont plus de chance d'être retenues comme résultat final du diagnostic cognitif (étape 3).

1.2.1.4. Le contexte pédagogique

Le contexte pédagogique du DC doit être vu comme un ensemble des paramètres qui influencent résolument la nature de ce processus et la manière dont il se déroule. Dans un STI, ce contexte incorpore plusieurs facteurs, principalement: les facteurs relatifs aux caractéristiques individuelles (émotions et attitudes) de l'apprenant et les facteurs relatifs aux objectifs d'apprentissage. Sans renier l'importance des caractéristiques individuelles de l'apprenant, nous nous intéressons aux facteurs relatifs aux objectifs d'apprentissage.

1.2.2. Pertinence et éléments du contexte pédagogique

Le DC dans un STI est un processus – qui se veut certes flexible – de traitement de l'information. L'information ainsi traitée est une représentation des connaissances nécessaires à la dérivation des inférences recherchées. Par exemple, on peut considérer deux situations où : (1) les inférences recherchées (résultat du DC) concernent les performances cumulatives de l'apprenant par rapport à un objectif d'apprentissage et (2) les inférences recherchées concernent le caractère fondé (rationnel) des arguments de l'apprenant lors de la résolution d'un exercice avec des pairs. Les représentations des connaissances qui sont nécessaires dans ces deux situations seront complètement distinctes. Dans le premier cas, les réponses finales de l'apprenant et leur lien avec les objectifs d'apprentissage constitueront la substance des représentations. Dans le second cas, le lien entre les justifications fournies par l'apprenant et les éléments de sa démarche, les interactions de l'apprenant avec ses pairs, constitueront la substance des représentations nécessaires au DC. Par conséquent, le paradigme de cognition considéré dans un STI influence indirectement la conception et l'implémentation du DC.

Le but de cette section est de justifier le choix des facteurs relatifs aux objectifs d'apprentissage comme éléments principaux du contexte pédagogique du DC dans un STI. Ces facteurs sont: le paradigme de cognition considéré et le courant de théories d'apprentissage associé à ce paradigme. On notera ainsi que les théories d'apprentissage influencent la phase d'acquisition d'indices ou d'observation du DC, tandis que les théories de la cognition influencent leur phase d'inférence.

1.2.2.1. Les théories d'apprentissage et la phase d'acquisition d'indices du DC

L'acquisition des indices est inhérente au DC puisque aucune conclusion ne peut être dégagée sans disposer de données sur l'apprenant. Selon Hoc (1990), l'acquisition des indices doit être guidée par l'objectif spécifique du diagnostic actuellement en cours.

Dans un STI, le DC observe les phénomènes qui extériorisent le processus d'apprentissage. Étant donné le caractère purement computationnel qui caractérise cette situation, ces processus doivent être formellement définis. La référence aux principes d'une théorie d'apprentissage ou d'un courant de théories d'apprentissage favorise cette formalisation. En effet dans un STI, la mise en oeuvre de l'instruction basée sur un modèle pédagogique a pour principal but de déclencher les mécanismes cognitifs permettant la réalisation des objectifs d'apprentissage. Or, ces mécanismes cognitifs sont influencés par la théorie d'apprentissage ou le courant de théorie d'apprentissage adopté. Par exemple, un courant béhavioriste de l'apprentissage suggère que pour favoriser l'apprentissage, l'apprenant devrait être conditionné à répondre d'une manière précise à des stimuli précis. Cette recommandation implique que lors du DC, seules les réponses de l'apprenant face à des stimuli précis, devraient être considérées pertinentes dans la phase d'observation ou d'acquisition d'indices. On peut donc considérer que le courant de théories d'apprentissage choisi lors de la conception d'une activité d'apprentissage dans un STI influence la nature des indices qui seront jugés pertinents à observer, d'une instance de DC à l'autre.

1.2.2.2. Les paradigmes de la cognition et la phase d'inférence d'hypothèses

Les hypothèses inférées dans la deuxième phase du DC sont des hypothèses "à propos" de la cognition de l'apprenant. Une fois de plus, étant donné le caractère computationnel des STI, cette cognition doit être formellement définie. Dans ce sens, les principes contenus dans une théorie ou dans un paradigme de la cognition renseignent sur les différents types d'inférences qui peuvent être posées "à propos" de l'état cognitif de l'apprenant. Elles influencent donc la nature du résultat du DC.

1.2.2.3. Conclusion

Il existe une correspondance entre chaque paradigme de la cognition et des classes de théories d'apprentissage. La section suivante décrit les principaux paradigmes de cognition,

ainsi que cette correspondance avec les classes de théories d'apprentissage. Le chapitre 3 traite de la relation entre ces concepts et le DC dans un STI.

1.3 Paradigmes de cognition et courants de théories d'apprentissage

Les paradigmes de la cognition et les théories de l'apprentissage s'intéressent respectivement à la nature et à l'acquisition du comportement intelligent. L'évolution historique des postulats de ces paradigmes et théories présente un caractère incrémentiel. En ce qui concerne la cognition, l'on a assisté à une évolution allant de l'empirisme au pragmatisme avec ses considérations socio-historiques, en passant par le rationalisme. En ce qui concerne l'apprentissage, ces courants de la cognition ont pu être associés à trois écoles de théories d'apprentissage: le béhaviorisme, le cognitivisme et le socioconstructivisme. Dans les prochaines lignes, nous nous inspirons de l'analyse de Greeno, Collins, et Resnick (1996) pour souligner les caractéristiques clés des paradigmes de cognition, des courants de théories d'apprentissage associés et surtout des relations existantes entre ces éléments et le DC dans un STI.

1.3.1. Fondements

D'obédience empirique tout d'abord, les théories de la cognition ne considéraient que ce qui est observable, déniaient l'intégrité scientifique de toute information relative aux processus "internes" de l'esprit (Locke, 1690; Thorndike, 1912). Certaines caractéristiques du comportement intelligent – principalement l'intentionnalité – demeuraient toutefois inexplicables en termes observables. Le rationalisme a fourni des arguments pour répondre à cette question en attribuant au comportement intelligent un fondement logico-rationnel: l'atteinte d'un but à travers des processus de conceptualisation et de raisonnement. La dimension contextuelle de l'apprentissage a finalement dû être considérée à travers les principes du pragmatisme et les influences socio-historiques. La sociologie et l'anthropologie ont en effet montré que la connaissance et l'apprentissage sont tributaires de l'environnement physique, de la communauté sociale, du bagage historique et culturel dans lequel ils se déploient.

1.3.2. L'empirisme et le béhaviorisme

L'épistémologie empirique prône la connaissance comme résultat d'une expérience *sensible* dans l'environnement d'un sujet cognitif. Seuls les phénomènes tangibles méritent d'être

considérés dans toute tentative d'explication scientifique de la nature et de l'acquisition de la connaissance. La connaissance résulte d'une synthèse *a posteriori* de l'interaction avec le monde sensible. L'associationnisme, le béhaviorisme et le connexionnisme sont des perspectives différentes de l'apprentissage selon ce courant de cognition.

L'associationnisme (Hume, 1748) considère que la connaissance est un ensemble d'associations entre des objets physiques. Ces associations exploitent les analogies, les contrastes et les concordances, et l'apprentissage correspond à l'acquisition de ces associations.

Le béhaviorisme préconise une étude purement scientifique de la cognition par la seule considération des comportements cognitifs tangibles. Les théories de ce courant (Pavlov, 1927; Skinner, 1974; Thorndike, 1932) considèrent généralement la connaissance comme un ensemble d'associations entre une perception physique (le stimulus) et une réponse comportementale observable. L'apprentissage consisterait donc à établir ces connexions, à les renforcer ou à les atténuer à travers le conditionnement.

Le connexionnisme (Rumelhart, 1989) rejoint le béhaviorisme dans le fond, mais dans la forme, les associations se font non pas entre des unités singulières, mais entre des patrons d'activités cérébrales. Ces patrons correspondent à la réponse à un stimulus donné. La connaissance est donc un patron d'activité cérébrale : le connexionnisme prolonge le béhaviorisme à travers une analogie au fonctionnement des neurones.

De ces trois courants, le béhaviorisme est celui qui influencera le plus l'élaboration des pratiques en éducation, avec la définition des objectifs d'apprentissage en terme de comportement observable des apprenants (Bloom, 1968; Gagne, 1992). Pour le reste de nos analyses, le béhaviorisme est donc retenu comme courant représentatif de l'empirisme au titre des théories d'apprentissage.

1.3.2.1. Apprentissage: le courant béhavioriste de théories d'apprentissage

Trois éléments clés caractérisent l'apprentissage béhavioriste: le stimulus, la réponse désirée et l'association stimulus-réponse. L'acquisition de la connaissance dans ce cas explique comment cette association est établie, renforcée et maintenue au fil du temps. Les principaux tenants de ce courant ont répondu à cette problématique: l'apprentissage consiste en la création d'associations stimulus-réponse par *conditionnement*, à leur *renforcement* et à leur *préservation* au fil du temps.

Pavlov (1927) introduit le conditionnement comme première expérience de l'apprentissage. Il consiste à établir une connexion entre un stimulus et une réponse par le biais de la récurrence du stimulus dans des conditions où la réponse voulue apparaît naturellement ou sur commande. Ce conditionnement de base peut être élaboré sous diverses formes: la généralisation (des stimuli similaires à un stimulus originel provoqueront la réponse associée à ce dernier); l'oubli (si la pratique d'un appariement stimulus-réponse est interrompue, l'association correspondante peut disparaître); le rappel spontané (le sujet cognitif se rappelle de la réponse associée à un stimulus après un certain laps de temps, bien que cette réponse pourrait être définitivement oubliée si la pratique n'est pas reprise); la discrimination (avec la pratique, un sujet cognitif peut différencier des stimuli similaires); le conditionnement de haut niveau (les stimuli présentés simultanément provoqueront éventuellement la même réponse lorsque présentés individuellement).

Thorndike (1932) introduit formellement la notion de renforcement avec l'idée selon laquelle les associations stimulus-réponse sont éventuellement consolidées lorsque la réponse désirée est récompensée (*feedback* positif).

Skinner (1938) introduit la notion de conditionnement "opérant" qui concerne le caractère volontaire des comportements. Écartant le facteur de récompense, cette perspective stipule qu'un sujet cognitif opère dans son environnement et que la nature de la réponse de cet environnement (pas nécessairement une récompense) détermine la tendance de ce sujet à répéter ou non ce comportement dans le futur. Cette explication se démarque des précédentes en ce que les sujets cognitifs acquièrent un caractère volontaire, ils ne réagissent plus simplement à l'état de l'environnement. Le conditionnement "opérant" a permis d'élaborer la notion de renforcement à travers quatre concepts sous-jacents:

- (1) Le renforcement positif: les réponses ou les opérations récompensées auront tendance à être répétées;
- (2) Le renforcement négatif: les réponses ou les opérations qui permettent d'éviter un *feedback* négatif ou qui permettent d'éviter des conséquences négatives auront tendance à être répétées;
- (3) L'absence de renforcement: les réponses ou les opérations qui ne reçoivent aucun *feedback* auront tendance à disparaître;

- (4) La punition: les réponses ou les opérations qui reçoivent un feedback négatif ou qui provoquent des conséquences négatives vont éventuellement disparaître, bien qu'elle puissent réapparaître en cas de changement au niveau du feedback qu'elles provoquent.

Toutes ces notions de renforcement sont chapeautées par la notion de façonnage ou apprentissage approximatif, où le sujet cognitif est dirigé ou encouragé progressivement vers la réponse ou le comportement désiré. La mémoire en tant que telle n'est pas importante dans l'apprentissage béhavioriste.

En résumé, le béhaviorisme matérialise les principes de l'apprentissage humain pour une cognition à orientation empiriste. Les facteurs clés de ces principes sont: la nature des stimuli de l'environnement de l'apprenant; la nature des réponses de l'apprenant; le conditionnement de la relation stimulus-réponse chez l'apprenant; les diverses formes de renforcement de la relation stimulus-réponse chez l'apprenant.

1.3.2.2. Empirisme, béhaviorisme et diagnostic cognitif

La théorie d'apprentissage béhavioriste implique que seules les actions observables et tangibles de l'apprenant doivent être considérées valides lors de la phase d'acquisition d'indices. La particularité de cette exigence est que les données observées doivent être directement utilisées pour établir un DC – sans passer par une interprétation.

De plus, selon l'empirisme, le DC devrait caractériser le comportement observable de l'apprenant (réponse) en présence de conditions particulières dans son environnement d'apprentissage; par exemple, les *indices* que présente une question ou un exercice à résoudre. Cette caractérisation devrait être guidée par l'association entre les conditions de la situation d'apprentissage et les réponses prédéfinies. Par exemple, une réponse de l'apprenant peut être caractérisée comme étant correcte si elle correspond à la réponse qui est effectivement associée aux conditions correspondantes d'apprentissage.

1.3.3. Le rationalisme et le cognitivisme

1.3.3.1. Cognition: le paradigme rationaliste

Plus qu'une doctrine épistémologique, le rationalisme (Descartes, 1637) est une philosophie où la connaissance est identifiée à la « raison ». La raison est indépendante de l'expérience avec le monde, elle est innée et ses fondements sont acquis *a priori*. Par conséquent,

l'apprentissage consiste à prendre conscience de cette connaissance, par l'expérience avec le monde sensible et ce dans une perspective rationnelle: toute activité doit être dirigée vers un but prédéfini dans un contexte contrôlé et déterminé (Leibniz, 1714).

Les hypothèses rationalistes sur la nature de la connaissance se penchent sur la nature de l'intellect. Fondamentalement, la connaissance dans ce sens se manifeste à travers:

- (1) La présence d'un ensemble adéquat de structures (représentations) mentales pour la réalisation de diverses tâches cognitives de haut niveau – notamment la compréhension des concepts, la résolution de problème, le traitement du langage;
- (2) La capacité d'utiliser adéquatement ces structures mentales pour réaliser ces tâches cognitives. Cette capacité est aussi dénommée "traitement de l'information" et dans la perspective rationaliste, il s'agit des mécanismes de raisonnement (inductif, déductif, abductif) et des mécanismes métacognitifs.

Les structures mentales sont une impression mentale par un sujet cognitif des perceptions de la réalité quotidienne de son environnement. Le traitement de l'information se réalise à travers des processus mentaux de nature logique, opérant sur les structures mentales afin de produire un comportement intelligent. La relation entre l'esprit et le cerveau, la relation entre le logiciel et la machinerie de l'ordinateur, telles sont des métaphores qui illustrent la perspective rationaliste de l'apprentissage.

La psychologie "gestaltiste" (Wertheimer, 1922) et le cognitivisme (Miller, 1962; Newell, 1990; Pylyshyn, 1984) sont les principaux courants regroupant les théories d'apprentissage associées au rationalisme. La psychologie "gestaltiste" définit et décrit un certain nombre d'habiletés cognitives: organisation optimale de structures basée sur des connexions associatives, organisation, réorganisation et concentration sur les données d'un problème pour en déterminer la solution. Les réalisations contemporaines de l'IA dans le domaine de l'éducation et dans la robotique découlent directement des hypothèses cognitivistes qui sont développées dans les lignes suivantes.

1.3.3.2. Apprentissage: le courant cognitiviste de théories d'apprentissage

L'hypothèse centrale du cognitivisme est la suivante: la cognition doit être perçue comme un ensemble de processus de traitement symbolique et formel de l'information. On parle de traitement symbolique parce qu'on considère qu'un sujet humain se représente l'information sous forme de symboles encastrés dans des structures mentales. Le traitement est formel

parce que l'interprétation des symboles dépend de leur forme physique. L'hypothèse de l'apprentissage comme un processus de traitement de l'information s'inspire du modèle de fonctionnement de l'ordinateur pour :

- (1) Expliquer comment les informations sont recueillies et structurées (ou représentées, conservées et ultérieurement repérées dans ces dispositifs): la représentation des connaissances;
- (2) Expliquer comment les représentations de connaissances sont traitées : le traitement de l'information en tant que tel (le raisonnement);
- (3) Définir les dispositifs physiques qui supportent la représentation des connaissances et le traitement correspondant de l'information: la mémoire. Le cognitivisme se positionne pour une cognition activement construite à travers l'activité *intellectuelle* plutôt que l'assimilation de l'information. Le concept de mémoire joue un rôle clé dans cette optique: la psychologie cognitive l'a désignée comme le support des activités intellectuelles du sujet cognitif. Cependant, seul son contenu (les représentations mentales et les processus mentaux) fait l'objet de l'apprentissage.

La représentation des connaissances

La première dimension de l'apprentissage cognitiviste concerne l'acquisition des structures mentales, c'est-à-dire la représentation des connaissances. L'hypothèse cognitiviste stipule que la connaissance correspond à une mémorisation significative de l'information. Cette nécessité sémantique de l'encodage de l'information impose la nature symbolique des représentations mentales qui en résultent.

Le traitement de l'information

La deuxième dimension de l'apprentissage cognitiviste concerne l'acquisition des processus mentaux. Les nativistes et les constructivistes se sont opposés dans leur explication de ce phénomène. Les nativistes (Chomsky, 1959) attribuent une part importante à l'inné dans la possession des processus mentaux: ils sont présents dès l'existence du sujet cognitif et la seule utilité de l'apprentissage serait d'en prendre conscience. Les constructivistes (Piaget, 1932) considèrent que seule l'expérience phénoménologique avec l'environnement permet l'acquisition des structures logico déductives qui sous-tendent les processus mentaux. Dans les deux cas, apprendre à raisonner nécessite des situations où le sujet cognitif est forcé

d'établir un lien entre un but à atteindre et le contexte (les conditions) qui dans lequel ce but doit être atteint.

Un certain nombre de modèles computationnels de l'apprentissage se sont efforcés de fournir une explication de la capacité de résoudre des problèmes. L'architecture SOAR de Newell (1990) et l'architecture ACT-R de Anderson (1983) sont les exemples les plus marquants dans cette catégorie. ACT-R caractérise l'apprentissage de la résolution d'un problème comme un processus incrémentiel. Premièrement, l'apprentissage consiste à acquérir et à interpréter toute information sous forme déclarative, c'est-à-dire en termes de faits. Ces faits peuvent correspondre à des concepts qui ont déjà été appris. Ces faits peuvent aussi correspondre à des règles qui associent des actions (conséquence de la règle) à des buts ou à des stimuli de l'environnement (conditions de la règle). Deuxièmement, l'apprentissage consiste à prendre conscience que les règles doivent être interprétées comme des productions: l'action associée à la conséquence d'une règle doit être exécutée lorsque les informations du problème à résoudre correspondent aux conditions de cette règle. Troisièmement, l'apprentissage consiste à ajuster les règles (généraliser, spécifier) initialement acquises en fonction du feedback de l'environnement ou de toute autre autorité.

1.3.3.3. Rationalisme, cognitivisme et diagnostic cognitif

La cognition est donc la possession de structures et de mécanismes qui respectivement, représentent des concepts et permettent le raisonnement dirigé vers un but. Dans ce cas, le DC doit établir que le sujet cognitif possède de bonnes structures de représentation interne et de raisonnement, et que ces structures sont adéquatement utilisées lors de la réalisation d'une tâche cognitive.

Dans le contexte particulier d'un STI rationaliste-cognitivist, le DC devra observer le comportement de l'apprenant lorsqu'il essaie de résoudre un exercice, afin de déterminer les habiletés cognitives qui lui manquent pour le faire correctement. Ces habiletés cognitives peuvent être formulées en termes de représentations mentales ou de processus mentaux adéquats. Les théories cognitivistes reconnaissent l'activité mentale et les états mentaux non observables. Le DC doit donc également se baser sur des faits observables relatifs à l'apprenant, mais dans le but éventuel de dériver le non observable.

1.3.4. Pragmatisme et socioconstructivisme

1.3.4.1. Cognition: courant pragmatiste avec contexte socio historique

Introduit par Dewey (1933), le pragmatisme oriente les conceptions de la cognition en accordant une place importante à la liberté de l'individu et à sa dimension socioculturelle. Le pragmatisme réfute les vérités conceptuelles et absolues qui caractérisent les courants qui l'ont précédé. Il prône une cognition teintée par un contexte socioculturel, conditionnée par l'interprétation d'une expérience personnelle et *in situ* du sujet cognitif (Driscoll, 1994). Ici, le terme "expérience" est beaucoup plus élaboré que dans la thèse empiriste car dans ce cas, elle se base sur une notion plus élaborée du contexte du sujet cognitif. Ce contexte intègre les *interactions* du sujet cognitif :

- (1) Selon un *environnement physique*
- (2) À travers l'*activité*, souvent outillée par des instruments technologiques (de la charrue à l'ordinateur). La *théorie de l'activité* est une analyse du rôle de l'activité dans l'acquisition de la connaissance (Nardi, 1996);
- (3) Selon un *contexte socioculturel* défini entre autres par les interactions avec les autres sujets cognitifs et les formes de ces interactions. Dans ce cas, la *théorie de la cognition distribuée* s'est intéressée à l'explication de la connaissance comme étant distribuée à travers les acteurs d'un contexte (Flor et Hutchin, 1991). Les théories de l'activité et de l'action située comportent aussi des éléments relatifs à l'aspect culturel et social de l'acquisition de la connaissance (Lave, 1988);
- (4) Selon un *bagage historique* en tant qu'influence de l'expérience passée. Dans ce cas, les théories de l'activité, de la cognition distribuée et de l'action située se sont également intéressées à l'impact de l'expérience sur l'acquisition de la connaissance.

Les théories évoquées se sont penchées sur un aspect particulier de cette optique mais elles se rejoignent toujours sur un point fondamental: la cognition émerge de l'*interaction rétroactive* du sujet cognitif avec son environnement.

Les interactions comme collaborations constructives : le socioconstructivisme

Selon Vygotsky (1978), la connaissance réside d'abord dans les interactions d'un sujet cognitif avec ses pairs, avant d'être individuellement intériorisée. Cette hypothèse part du fait qu'une seule personne ne peut embrasser les multiples facettes nécessaires à la résolution

d'un problème. Pour développer sa cognition, le sujet cognitif doit donc se servir activement des caractéristiques de son environnement, des outils mis à sa disposition et de l'argumentation constructive avec ses pairs.

Les interactions selon un environnement physique : la cognition distribuée

La connaissance est aussi distribuée à travers les éléments de l'environnement d'un sujet cognitif: ses pairs (sujets de l'interaction), les instruments utilisés (outils d'une activité). Dans la théorie de la cognition distribuée par exemple, l'unité d'analyse est un système constitué des individus et des instruments qu'ils utilisent pour atteindre un but précis. Cette délocalisation de l'information force à un partage qui confère un caractère négocié à la résolution de toute tâche, ce qui corrobore la nature non unilatérale de la connaissance qui est préconisée dans cette optique.

Les interactions à travers l'activité instrumentée: la théorie de l'activité

La nature d'un concept change continuellement au cours des diverses situations de son utilisation. Cette remarque introduit la notion de la relativité de la connaissance. La théorie de l'activité stipule que seule l'activité permet une appréhension utile et significative d'un concept: elle est le véhicule qui permet à un sujet cognitif de transformer l'information en connaissance.

Selon Nardi (1996), la conscience comme ensemble discret d'actes cognitifs désincarnés est une antinomie. En effet d'une part, la connaissance ne se limite pas au fonctionnement du cerveau et d'autre part, la connaissance intègre l'impact des pratiques quotidiennes sur l'environnement du sujet cognitif, ainsi que la rétroaction de cet impact sur ce sujet.

Selon Leontiev (1984), la pratique éveille les sens par lesquels les humains éprouvent réellement le monde environnant. Sa théorie de l'activité considère cette dernière comme l'unité minimale d'analyse cognitive, accompagnée d'un contexte - son motif, son environnement et surtout les instruments/artéfacts utilisés pour l'effectuer -, ainsi que d'une éventuelle communauté participante. Les instruments sont les médiateurs du comportement et de la pensée: les sujets pensants sont stimulés par un objectif et ils utilisent ces instruments pour les atteindre.

Selon Kaptelinin (1996), le sujet cognitif et l'instrument forment un système cognitif plus efficace: un organe fonctionnel. Par exemple, si on considère le rôle des exercices

pratiques dans un laboratoire scolaire de physique-chimie, l'information sera mieux perçue avec la manipulation d'un microscope, d'une paire de pinces, d'une loupe, d'un bain-marie, et cetera que à travers la seule description théorique du processus en étude.

Les interactions situées dans un contexte historique et culturel

La considération des artefacts (en tant que véhicules de l'activité) et de l'interaction collaborative comme composantes de la cognition attribue une dimension historique et culturelle à cette cognition. En effet, la forme et l'utilité d'un outil est influencée par la culture (la vision du monde et de la dynamique de ce monde) dans laquelle il a été créé. De même, l'usage qui est fait d'un outil reflète les besoins et la vision de la communauté cognitive au sein de laquelle il a été créé. Ces outils se voient améliorés, re-conceptualisés ou inventés au fil des découvertes scientifiques. Le langage est un exemple de fonction cognitive fortement influencée par l'histoire et la culture des sujets cognitifs.

1.3.4.2. Apprentissage: courant socioconstructiviste des théories d'apprentissage

L'apprentissage est un processus actif d'où émerge la signification à travers un processus social de négociation (Merrill, 1991). Cette position émane de deux éléments essentiels de la connaissance dans la perspective pragmatiste et socio-historique de la cognition: son caractère distribué et son caractère émergent d'interactions constructives entre le sujet cognitif et les éléments de son environnement.

Maturana et Varela (1980) proposent une métaphore plus générale pour une vue d'ensemble de l'apprentissage dans cette perspective de la cognition: l'autopoïèse. L'autopoïèse est de fait la propriété qu'a un système de produire des processus internes qui contribuent eux-mêmes (rétroactivement) au maintien et à l'évolution de ce système. On peut comprendre cette notion en évoquant *a contrario* celle du système allo-poïétique, à savoir un système dont le maintien et l'évolution ne peut être assumé par ses propres composantes. À travers la notion de couplage structurel, les principes de l'autopoïèse permettent une définition formelle de l'apprentissage pragmatique. Il s'agit d'un processus continu qui se produit quand deux systèmes autopoïétiques subissent une ontogenèse "couplée". Cela survient lorsque ces systèmes sont engagés dans un historique d'interactions récurrentes, lors d'un processus de coordination réciproque. L'ontogenèse est l'histoire des changements structuraux subis par un système vivant sans qu'il perde son identité.

Ramené dans un contexte d'apprentissage, l'apprenant en tant que système autopoïétique répond aux modifications de son contexte d'apprentissage (ses pairs, les outils d'apprentissage, la présentation de l'instruction, etc) en transformant ses représentations mentales. Plus concrètement, Vygotsky (1978) propose trois étapes qui articulent avec concision les aspects de l'apprentissage pragmatique:

- (1) Une mise en situation correspondant à la formulation d'un problème concret, voire réel;
- (2) Une collaboration manifeste avec les pairs (extériorisation);
- (3) Enfin une phase réflexive (intériorisation) qui correspond à une synthèse de l'expérience de collaboration pour négocier la connaissance.

Mise en situation ou enracinement culturel

Conceptuellement, la mise en situation doit favoriser la perception de la situation associée à un problème à résoudre. Cette étape permet la création des représentations mentales *relatives* à cette situation. On parle d'enracinement dans la culture du problème: la mise en situation impose à l'apprenant de s'imprégner dans la logique de pensée qui sous-tend le problème.

Par exemple, l'apprentissage des concepts mathématiques ne doit pas être une entreprise d'assimilation ou de prise de conscience de structures mentales unilatérales. Au fond, il s'agit d'acquérir des mécanismes de raisonnement qui sont utilisées ultérieurement pour résoudre des problèmes réels. On devrait donc commencer par conditionner l'apprenant à la logique de pensée du mathématicien, par rapport au problème à résoudre. Ce type d'apprentissage ne peut être véhiculé que par une *activité authentique* (Brown, Collins, et Duguid, 1988): une activité où il faut agir dans un contexte correspondant à la culture du domaine. Le Tableau I.1 illustre cette vision dans l'apprentissage du calcul des fractions en arithmétique en confrontant le contexte rationaliste et le contexte pragmatiste et socioculturel.

Tableau I.1. Situation d'apprentissage: rationaliste versus pragmatiste

Définition du problème : Hélène désire partager un gâteau entre ses 2 enfants Ludovic et Valérie. Ludovic doit obtenir les 2/3 du gâteau et il désire en plus conserver les 3/4 de sa partie pour son goûter. Quelle sera la part du gâteau d'Hélène qui sera réservée au goûter de Ludovic?	
Approche cognitiviste ou béhavioriste	Approche pragmatiste
Données conceptuelles $X = 2/3$ de Z $Y = 3/4$ de X Z = Gâteau en entier	Contexte Culture : Alimentation Situation : Partage Activité : Découpage du gâteau Instrument : Couteau à gâteau
But $Y = ?$	But Quelle part Ludovic mangera immédiatement et quelle part gardera-t-il pour son goûter?
Solution Traitement de l'information $Y = 3 \times 2/4 \times 4 = 1/2 Z$	Solution : Action instrumentée Ludovic découpe le gâteau en 3 parts égales; Ludovic prend les 2 de ces parts; Ludovic divise une de ces 2 parts en 2 parts égales; Ludovic mange une de ces parts et garde les 3 autres pour son goûter.
Apprentissage Entraînement à travers l'application d'un principe de calcul arithmétique (multiplication de 2 fractions)	Apprentissage Application concrète des principes liés aux fractions

Découverte ou construction de la connaissance à travers la collaboration

La phase de collaboration favorise la découverte d'une solution à un problème plus complexe, une fois que son contexte a été introduit par un problème simple. Les apprenants peuvent collaborer à travers des *élaborations cognitives* qui consistent à:

- Articuler leurs arguments;
- Rationaliser ou objectiver leurs arguments;
- Critiquer les arguments des pairs par : leur évaluation formelle, la confrontation d'arguments opposés ou contradictoires;
- Négocier une solution à la lumière des critiques qui ont été partagées.

Ce processus de collaboration aboutit à l'émergence et à la découverte de la connaissance. Chaque apprenant apprend de ses pairs en portant d'abord attention à leurs *élaborations respectives*. Ensuite, l'apprentissage est mis en branle à travers un processus de discussion

constructive: les apprenants posent des questions sur les arguments de leurs pairs, confrontent ces arguments en y opposant d'autres points de vue, en y identifiant des contradictions ou des incohérences. Ultimement, les apprenants qui collaborent coordonneront leurs points de vue respectifs afin de négocier une solution acceptable à un problème.

L'élaboration cognitive favorise surtout le développement des capacités métacognitives et sociales telles que l'autogestion, l'autocritique, l'autoévaluation, l'engagement, l'organisation, le sens critique, la collaboration. L'élaboration cognitive force également les apprenants à une introspection sur leurs arguments, ce qui en favorise le fondement.

Autogestion métacognitive

Une fois qu'une solution a été négociée, chaque sujet cognitif doit tirer les leçons de son expérience. Il s'agit d'intégrer les points saillants évoqués lors de la collaboration en termes de concepts et d'approche de raisonnement. Il s'agit aussi de généraliser s'il y a lieu les connaissances nouvellement construites, par leur association à des problèmes similaires. Enfin il s'agit d'évaluer les autres composantes du système cognitif entier qui a contribué à l'émergence de la connaissance.

1.3.4.3. Pragmatisme, socioconstructivisme et diagnostic cognitif

La connaissance émerge de l'activité instrumentée du sujet cognitif dans son environnement, de son interaction sociale avec ses pairs sur la toile de fond de l'histoire et de la culture de la communauté qu'ils forment. Dans cette perspective, la connaissance n'est pas unilatérale, ce qui exclut la nécessité de diagnostiquer les actions d'un apprenant par rapport à un ensemble préfixé de connaissances. Par contre, ce qui demeure stable dans la perspective pragmatiste est le processus de construction de la connaissance. Dans cette logique pragmatiste et socioconstructiviste, le DC devrait se pencher sur :

- (1) Les capacités d'argumentation de l'apprenant lors du processus de construction ou de négociation d'une connaissance;
- (2) La capacité de collaboration de l'apprenant avec ses pairs;
- (3) Les capacités métacognitives qui favorisent l'attitude pragmatiste lors de l'apprentissage.

D'après l'optique socioconstructiviste, l'apprentissage traverse ne se limite pas au travail intellectuel et individuel. Il se révèle également à travers l'activité, les instruments, l'environnement et l'interaction avec une communauté d'apprenants. Le DC doit observer l'apprenant sous tous ses angles (mental, environnemental, culturel), par exemple :

- (1) La pertinence des actions posées et des instruments utilisés;
- (2) La nature et la pertinence des interactions entretenues avec d'éventuels pairs;
- (3) La nature des instruments utilisés lors de la construction d'une solution à un exercice;
- (4) La nature et les facteurs déterminants de l'environnement physique et socioculturel des apprenants considérés.

1.3.5. Conclusion

Les sections précédentes ont permis de rappeler la relation entre la nature de la cognition, les théories d'apprentissage et le diagnostic cognitif dans un STI. Cela met en exergue l'importance de ces paramètres dans cette fonction pédagogique, d'autant plus qu'elle est appliquée dans un contexte contrôlé par une machine plutôt que par un humain. Le questionnement occasionné par l'explicitation de cette relation présage une contribution importante et originale dans la perspective des sciences cognitives appliquées au domaine AIED. En effet, les réponses possibles conféreront au DC dans un STI une cohérence conceptuelle avec le contexte pédagogique de ce STI (en l'espèce le paradigme de cognition qui y est associé). Dans la prochaine section, l'on introduit le deuxième volet de notre contribution dans la conception et la mise en œuvre du DC dans un STI. Il s'agit de l'intégration au processus de DC, de l'exploitation pédagogique qui est faite de ses résultats, ce à travers une boucle diagnostic-remédiation; la stimulation de la pensée réflexive chez l'apprenant tenant lieu d'approche de remédiation. La boucle diagnostic-remédiation est un autre aspect de l'originalité de cette thèse. Dans la perspective des sciences cognitives appliquées au domaine AIED, cette boucle montre comment le DC et son exploitation pédagogique se co-déterminent favorisant ainsi une amélioration progressive de la compréhension des difficultés de l'apprenant. Dans la perspective informatique, l'application concrète de cette boucle dans un STI est réalisée en deux étapes: premièrement, une formalisation de sa dynamique basée sur une approche d'inférence bayésienne sera effectuée; ensuite l'implémentation informatique du modèle formel correspondant sera élaborée à travers une librairie de programmes génériques et réutilisables. Une telle librairie pourrait

alors être importée par tout module de DC dans un STI où l'on désire adopter la perspective de la boucle DC-remédiation.

1.4 Intégration DC-exploitation pédagogique: la boucle diagnostic-remédiation

Le DC n'est pas une fin en soi. Le résultat du DC est exploité pour renforcer l'efficacité des actions pédagogiques, pour un apprentissage plus profond. De plus, les conséquences de l'exploitation pédagogique des résultats d'un DC doivent être utilisées ou interprétées de manière à permettre une rétroaction sur la qualité de ce DC. Ceci donne lieu à une boucle rétroactive DC-remédiation-DC (Tchetagni et Nkambou, 2004c). Ici, le terme "remédiation" est utilisé au sens large. Il s'agit de toute action pédagogique du tuteur, dont le but est de résoudre les difficultés de l'apprenant qui ont été identifiées lors d'un DC.

Le deuxième élément de la vision «pro-pédagogique» du DC soutenue dans cette thèse découle directement de cette boucle entre le DC et la remédiation. En fait, lorsque l'environnement de déploiement du DC est un STI, cette optique cyclique implique que, le contenu de l'algorithme correspondant au DC est influencé par les mesures réparatrices qui suivront. Par conséquent, la remédiation devient un facteur déterminant de la nature du DC.

Pour concrétiser cette vision, le phénomène de "pensée réflexive" (ou encore la réflexion) a été utilisé comme instance d'approche de remédiation après un DC. Avant d'examiner la nature de ce phénomène, il faut comprendre pourquoi il peut être considéré comme action permettant de remédier aux difficultés d'un apprenant.

1.4.1. Pourquoi intégrer la remédiation au DC? La boucle diagnostic-remédiation

Lorsqu'il a été établi que l'apprenant éprouve des difficultés par rapport à un élément de connaissance ou par rapport à l'utilisation d'une capacité, une approche de remédiation consiste à l'amener activement à comprendre et à corriger son erreur. Deux avantages correspondent à une telle pédagogie lorsque des difficultés de l'apprenant ont été identifiées. Premièrement, l'implication active de l'apprenant dans la correction de son erreur favorise une compréhension plus profonde de la connaissance ou de la capacité sous-jacente (Self, 1999); on évite ainsi les résultats superficiels où l'apprenant est surtout amené à fournir une solution correcte, sans s'assurer qu'il "comprend vraiment". Deuxièmement, la participation active de l'apprenant à la correction de son erreur permet éventuellement d'ajuster

continuellement les hypothèses initiales de DC, permettant des représentations de plus en plus précises de ses besoins : raffiner, confirmer ou infirmer les hypothèses initiales.

Les opérations associées à la participation de l'apprenant dans la correction de son erreur *pour comprendre* (et pas seulement *pour corriger*) embrassent le phénomène de la réflexion. Étant donné leurs implications susmentionnées, elles constituent un usage pédagogique légitime du DC.

Par ailleurs, bien qu'un STI ne soit pas un système cognitif à proprement parler, nous avons décidé de désigner par "réflexion du côté du système", l'ensemble des ajustements effectués par ce système suite aux effets de la réflexion de l'apprenant après un DC. Ces ajustements peuvent s'appliquer à l'échelle microscopique aux résultats initiaux de ce DC. Ils peuvent aussi concerner des données macroscopiques reliées à l'algorithme associé à ce DC: ajustement des paramètres, ajustement des informations qui y sont utilisées, etc.

1.4.2. Qu'entend on par réflexion?

La réflexion est avant tout un phénomène générique qui rassemble les processus cognitifs reliés au jugement et à la pensée (Larousse, 2004). En sciences cognitives et en éducation, ce terme a souvent pris une signification particulière, soit en tant que phénomène métacognitif, soit en tant qu'analyse consciente.

1.4.2.1. La réflexion en tant que métacognition

Flavell (1979) définit la métacognition comme "une prise de conscience de l'expérience cognitive" et "une prise de conscience des connaissances acquises". Il précise que cette prise de conscience permet la sélection, la révision ou l'abandon de certaines tâches cognitives, buts ou stratégies lorsque mis en relation entre eux et en relation avec les habiletés de l'apprenant et ses intérêts actuels. Noël (1997) propose la définition suivante de la métacognition: un processus mental dont l'objet est soit une activité cognitive, soit un ensemble d'activités cognitives que le sujet vient d'effectuer ou est en train d'effectuer, soit un produit mental de ces activités cognitives. La réflexion peut être vue comme une activité métacognitive dans la mesure où le sujet cognitif porte un regard sur son processus d'apprentissage.

1.4.2.2. La réflexion en tant qu'analyse consciente:

Historiquement, la pensée réflexive comme analyse consciente pendant l'apprentissage a été étudiée selon plusieurs écoles de pensée. Le modèle de Lewin de recherche-action met un accent sur la réflexion pendant l'activité de résolution de problème, dans un contexte social et organisationnel (Lewin, 1948). Le travail de John Dewey est une autre base théorique pour la réflexion en tant qu'analyse consciente dans les us et pratiques de l'apprentissage et de l'éducation (Dewey, 1933). La théorie de Dewey est fondamentale puisqu'elle a été construite sur une argumentation logico philosophique, dans le sens d'articuler explicitement les principales composantes du processus de pensée réflexive en tant qu'analyse consciente. Enfin, des recherches plus contemporaines se sont également penchées sur la nature de la réflexion consciente, que ce soit dans le contexte de l'apprentissage par l'expérience (Kolb, 1984) ou dans le contexte de l'éducation professionnelle (Schön, 1983). La majorité de ces modèles sont inspirés de la théorie de Dewey. De plus, dans cette thèse, l'on s'efforce de rester le plus générique possible en évitant entre autres de se limiter à un style d'apprentissage particulier (par expérience pour le modèle de Kolb, professionnel pour le modèle de Schön, socio-organisationnel pour le modèle de Lewin). Dans ce sens, la théorie de la pensée réflexive selon Dewey a été retenue dans cette thèse comme cadre de référence lors de l'intégration de ce phénomène au DC dans un STI, comme mesure de remédiation.

Selon Dewey, l'idée et la formulation d'un jugement « dans notre pensée » ne peuvent à elles seules témoigner d'un apprentissage « réel ». L'apprentissage réel se caractérise par des processus mentaux *logiques* - et les changements d'états mentaux correspondants - qui mènent à une idée ou à un jugement. Avant de proposer une définition formelle de la pensée réflexive, il est nécessaire d'en saisir les deux points de vue ainsi que le caractère logique correspondant: le processus de la pensée réflexive qui se caractérise par *la méthode logique sur laquelle il se fonde (rationnelle)* et le produit de la pensée réflexive qui se caractérise par *sa forme ou son expression logique (argumentative au sens de syllogisme)* (Dewey, 1933, p. 75). À la suite d'un DC, l'objectif est de provoquer chez l'apprenant une articulation mentale des connaissances ou des capacités diagnostiquées comme source de difficulté pour l'apprenant. Cette articulation mentale doit témoigner que l'apprenant comprend réellement ce qu'il fait; la pensée réflexive en tant qu'analyse consciente est un moyen adéquat pour la

réaliser. La réflexion en tant qu'analyse consciente s'inscrit donc au cœur de l'action dans le processus d'apprentissage.

1.5 Conclusion

Dans un STI, le transfert de la responsabilité pédagogique de l'homme vers la machine implique une définition formelle de cette pédagogie. Étant donné le caractère computationnel de son déploiement, cette conception ne peut s'échafauder qu'à travers des recommandations formelles ou des principes. Les théories de la cognition, les théories d'apprentissage et les théories d'instruction sont les moyens communément utilisés à cette fin, lors de la conception pédagogique des STI. Pour la fonction pédagogique particulière qu'est le DC, ce chapitre a permis de réaliser que les théories de la cognition – et par ricochet – les théories d'apprentissage fournissent des principes qui ne peuvent pas être ignorés dans la conception de ce processus. En effet, on a pu constater que: les théories de la cognition indiquent ce qu'il est pertinent de diagnostiquer, d'autant plus que cette pertinence varie significativement d'une théorie à l'autre; ces théories indiquent indirectement la nature et le type des informations qui doivent être représentées dans un algorithme de DC; les théories d'apprentissage indiquent ce qu'il est pertinent d'observer chez l'apprenant, afin de poser un DC.

La vision «pro-pédagogique» du DC dans un STI prônée dans cette thèse stipule que (Figure I.2): il existe une relation entre les théories de la cognition assumées dans un STI et la nature du DC dans ce STI. En effet, ces théories délimitent les besoins réels qui permettent une implémentation efficace du DC: seules les informations relatives à la vision de ce que sont la connaissance et l'apprentissage sont modélisées, représentées et utilisées. Elles favorisent aussi une implémentation cohérente du DC: seules les informations relatives à la vision de ce qu'est la connaissance sont inférées comme résultat du DC.

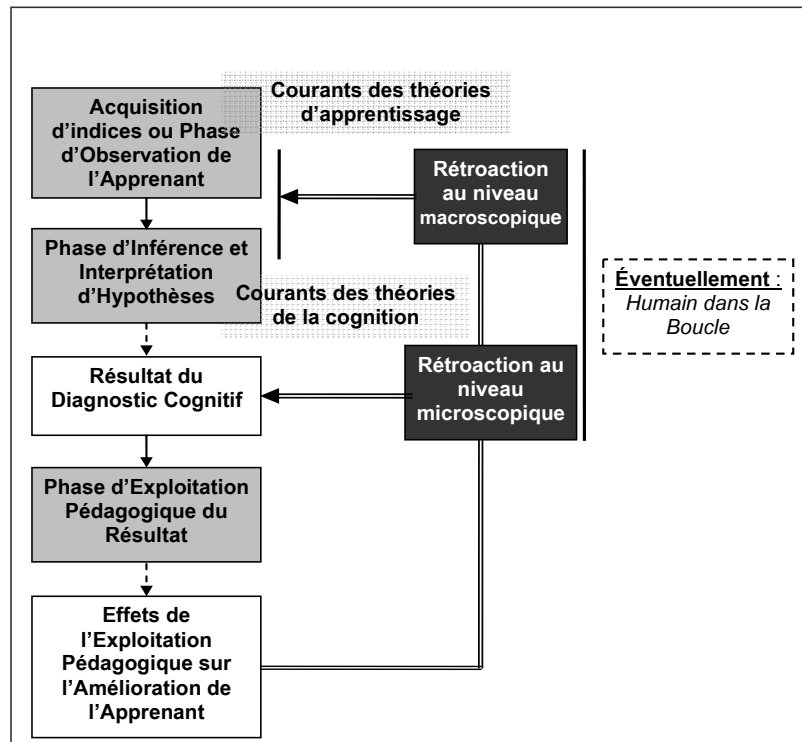


Figure I.2. Perspective «pro-pédagogique» du Diagnostic Cognitif

Cette vision préconise aussi qu'il existe un lien important entre le DC dans un STI et l'exploitation qui est faite de son résultat. En effet, on anticipe que la nature de cette exploitation influence la manière dont se déroule l'algorithme de DC. De plus les conséquences de cette exploitation peuvent à leur tour influencer le résultat dont elle est issue. Dans ce chapitre, le phénomène de la "pensée réflexive" a été introduit et justifié comme instance d'exploitation pédagogique du DC ou encore comme approche de remédiation après un DC.

Du reste, un cadre de spécification du DC favorise la considération de ces aspects dans la conception et l'implémentation de ce processus dans un STI. Avant d'élaborer ce cadre, nous examinons dans le prochain chapitre: (1) dans quelle mesure la recherche dans le domaine AIED a proposé des solutions relatives à la conception formelle du DC pour les STI; (2) dans quelle mesure toute conception formelle du DC a explicitement abordé les aspects «pro-pédagogique»s du DC dans un STI, en l'occurrence à travers une boucle diagnostic-remédiation. Le prochain chapitre est une synthèse de cet examen.

Chapitre II

ÉTAT DE L'ART

2.1 Introduction

Ohlsson (1987) définit une stratégie pédagogique comme un objectif générique d'instruction, réalisé par plusieurs séquences d'actions concrètes: les tactiques pédagogiques. Le diagnostic cognitif dans les STI est une instance de stratégie pédagogique dont la finalité est d'expliquer les actions correctes et incorrectes de l'apprenant en terme de capacités acquises ou non acquises, de raisonnement adéquat ou inadéquat, etc. Ainsi, plusieurs tactiques de DC peuvent réaliser le même objectif de DC. Tel qu'il a été montré au cours du chapitre précédent, les opérations qui permettent de réaliser le DC dans un STI varient selon l'exploitation pédagogique qui sera faite de leur résultat. De même, la nature de ce résultat varie selon le contexte pédagogique du STI dans lequel le DC est effectué.

Le propos de ce chapitre porte sur les travaux de modélisation formelle du DC dans les STI (section 2.2), avec un accent particulier sur certaines questions. Nous allons ainsi analyser dans quelle mesure ces travaux considèrent l'influence des théories de cognition et d'apprentissage dans la conception du DC (section 2.3). Nous allons également analyser dans quelle mesure ces travaux considèrent l'influence de l'exploitation pédagogique du résultat

d'un DC sur la nature des opérations (ou tactiques) concrètes qui permettent sa mise en œuvre (section 2.4).

2.2 Conception formelle du DC dans les STI

Dans quelle mesure la conception du DC dans un STI a été abordée sur la base d'un cadre de spécification de ce processus? Trois créneaux de recherche pourraient fournir des réponses possibles: les *shells* de modélisation de l'apprenant, les STI modulaires et dans une moindre mesure, les systèmes multi-agents (SMA).

Les *shells* de modélisation de l'apprenant sont des cadres génériques qui permettent de spécifier: le contenu désiré pour le modèle de l'apprenant dans un STI, les fonctions qui permettront de gérer ce contenu. Certaines de ces fonctions correspondent au DC.

L'intérêt des STI modulaires est qu'en théorie, leur conception en favorise la réutilisation dans d'autres STI. À notre avis, cette réutilisation ne peut être possible que si chaque module – en particulier le module de DC - est élaboré sur la base d'une spécification formelle, voire standardisée. En effet, une telle approche de conception favorise l'intelligibilité (connaissance et compréhension des composantes et des fonctions d'un module), donc la portabilité et par conséquent, la réutilisation.

Enfin, les SMA peuvent être vus comme des cas particuliers de systèmes modulaires, et la pertinence de les aborder découle donc des mêmes arguments.

2.2.1. Les *Shells* de modélisation de l'utilisateur

Un *shell* est une implantation générique d'une composante d'un logiciel. Il doit satisfaire les besoins de représentation d'un maximum de systèmes, parmi ceux dont il permet de construire une instance (Pohl, 1999). Un *shell* pour la modélisation de l'apprenant doit donc entre autres, permettre d'indiquer une approche de représentation des connaissances de l'apprenant, et une approche de construction de ces représentations.

BGP-MS (Kobsa et Pohl, 1995) est un *shell* de modélisation de l'utilisateur basé sur le formalisme AsTRA (*Assumption Type Representation Framework*). La représentation du contenu avec AsTRA exige que la base de connaissances correspondant au modèle de l'apprenant permette: la vérification de la justesse d'une proposition à propos de l'état des connaissances de l'utilisateur; la création d'une proposition à propos de l'état des connaissances de l'utilisateur. Dans le contexte d'un STI, BGP-MS permet une représentation formelle et

logique de l'état cognitif de l'apprenant. Le DC quant à lui est réduit à une simple opération d'ajout de proposition dans la base de connaissances associée au modèle de l'apprenant.

Le système GUMS (Finin, 1989) utilise une arborescence de stéréotypes pour regrouper les modèles d'usagers selon des traits communs. Les feuilles de cette arborescence correspondent à des modèles individuels d'usagers. Ces modèles sont essentiellement constitués d'assertions sur ces usagers. GUMS offre une interface de six commandes dont quatre sont reliées à l'administration des profils usagers. Les deux autres commandes permettent respectivement de questionner un modèle individuel et d'y ajouter une assertion.

TAGUS (Paiva et Self, 1995) est un *shell* pour la modélisation de l'apprenant dans un STI. Il utilise un formalisme logique pour représenter les connaissances, le raisonnement et les stratégies cognitives de l'apprenant. TAGUS fournit une interface avec six services: deux services pour s'enquérir d'une caractéristique ou d'une connaissance, et quatre services pour la fonction de mise à jour du modèle de l'apprenant. La particularité de chacun de ces quatre services varie selon qu'on désire maintenir la cohérence du modèle de l'apprenant lorsqu'une assertion sur son état de connaissances y est ajoutée.

FITS (Ikeda et Mizogouchi, 1994) propose une architecture générique à travers laquelle on peut créer une instance d'un STI. FITS est basé sur trois modules: le module de l'apprenant, le module d'expertise et le module du tuteur. Les rôles du DC se retrouvent dans le module de l'apprenant puisque ce dernier inclut: la construction du modèle de l'apprenant, l'identification des erreurs de l'apprenant et enfin l'analyse des erreurs de l'apprenant. FITS offre une représentation très précise des algorithmes qui pourraient supporter chacune de ces fonctions. Toutefois, ce système ne fait aucune référence aux liens entre les opérations de DC qui ont été fixées et le contexte dans lequel elles seront mises à contribution.

Dans le contexte d'un STI, l'ajout d'une assertion peut être assimilé entre autres à une mise à jour du modèle de l'apprenant suite à un DC. Cette opération est trop générale pour être considérée comme un diagnostic en bonne et due forme, d'autant plus qu'elle ne nécessite aucun mécanisme d'inférence. Aucune indication sur l'impact pédagogique de ces ajouts, le type de données sur l'apprenant qui pourraient supporter ces ajouts, les variations de ces données selon un contexte pédagogique, n'est mentionnée. Dans ce sens, les *shells* de modélisation de l'utilisateur et leur utilisation particulière pour la modélisation de l'apprenant dans un STI offrent des solutions trop génériques à la modélisation formelle du DC.

2.2.2. Les STI modulaires et les SMAs

L'idée d'architecture modulaire pour les STI a suscité un intérêt remarquable dans la recherche du domaine AIED (Frasson, Mengelle, Aimeur, et Gouarderes, 1996; Mengelle, de Léan, et Frasson, 1998). Dans cette optique, les STI sont perçus comme un ensemble orchestré de logiciels indépendants. L'intérêt d'une telle approche est la réduction des coûts d'implantation et des risques de complications potentielles lors des mises à jour. La notion de tuteurs connectables (*plug-in tutors*) a été introduite dans ce sens. La construction de ces tuteurs se base sur une conception indépendante de chacune de leurs composantes spécialisées, avant de les intégrer par le biais d'une plateforme de communications. L'évolution de l'idée des tuteurs modulaires s'est centrée sur la définition de standards de communication entre les composantes (Koedinger, Suthers, et Forbus, 1999; Ritter, 1997) d'un STI et non sur les standards relatifs au contenu et au fonctionnement interne de ces composantes. En effet, certains travaux illustrent concrètement un STI modulaire (Ritter et Koedinger, 1996). Cependant, aucune spécification formelle des composantes et du fonctionnement interne d'un module de DC (ni des autres modules) ne soutient cette démonstration.

Dans la même veine que les STI modulaires, les STI multi-agents doivent être conçus comme une "société d'esprits" où chaque agent proactif a un but réalisable par l'interaction avec les autres agents. Existe-t-il des agents de DC dans les STI multi-agents? Avant de répondre à cette question, il faut savoir que le fonctionnement des systèmes multi-agents s'appuie sur l'envoi, la réception et le traitement de messages. Pour notre travail, il est donc question de déterminer si les contenus des messages associés à un agent de DC peuvent être interprétés comme des éléments formels du DC.

ABITS par exemple (Capuano, Marsella, et Salerno, 2000) est un STI multi-agents pour la formation à distance. ABITS a été élaboré à partir de l'identification d'un certain nombre de services – dont celui de modélisation de l'apprenant - pour créer les agents pertinents en conséquence. ABITS indique que l'agent de modélisation de l'apprenant peut fournir des informations sur son état de connaissances. Cependant, cet agent n'intègre pas une structure et des caractéristiques formelles qu'on aurait pu regrouper dans un cadre conceptuel du DC pour un STI.

De nombreuses autres applications du concept multi-agents dans les STI ont contribué à l'avancement des connaissances dans le domaine AIED (Frasson, Mengelle, Aimeur, et Gouardères, 1996; Gouaderes et Minko, 2000). Les agents de modélisation de l'apprenant ont été élaborés pour plusieurs fins: pour la collaboration entre les apprenants (Greer et al., 2001), pour une approche émergente du DC (Webber, 2004; Webber, Pesty, et Balacheff, 2002), pour la mise à jour du modèle de l'apprenant (Hospers et al., 2003), ou pour la planification de l'apprentissage (Nkambou et Kabanza, 2001). Les choix d'implémentation de ces agents paraissent découler d'une analyse *ad hoc* de la dynamique du STI en question. La majorité du temps, l'argument avancé en faveur de l'approche multi-agents est la possibilité de les réutiliser, étant donné leur indépendance – au moins théorique –. À notre avis, la réutilisation n'est possible qu'en présence d'une référence conceptuelle définissant formellement le contenu et les fonctions de l'entité réutilisée.

2.3 Conception formelle du DC et contexte pédagogique

Un système auteur fournit à un tuteur humain ou à un concepteur pédagogique le moyen de concevoir l'expertise pédagogique désirée pour un STI et dans certains cas de créer les artefacts correspondants dans un STI. Les systèmes auteurs s'élaborent généralement en tenant compte d'une théorie d'apprentissage et d'une théorie de la conception de l'instruction (Reigeluth, 1999). Une analyse des systèmes auteurs permet d'examiner dans quelle mesure l'influence des théories de cognition et d'apprentissage sont considérés dans la conception du DC.

Murray (2003b) classe les systèmes auteurs dans deux catégories, selon le type de STI qu'ils produisent: les systèmes orientés vers la modélisation pédagogique et les systèmes orientés vers la création des STI qui concrétisent les modèles pédagogiques.

2.3.1. Les systèmes auteurs orientés vers la modélisation pédagogique

Avec IDE (Merrill, 2003), il est question d'utiliser des objets de connaissance pour gérer l'instruction dans un STI basé sur la simulation. Il s'agit d'un point de vue pragmatiste de la cognition. Un objet de connaissance décrit les entités (objets, actions, processus de l'environnement de simulation) que les apprenants vont manipuler. Les objets de connaissance servent de support à une instruction préfixée. Le DC est uniquement basé sur les résultats de l'interaction de l'apprenant avec ces objets, sans tenir compte de la nature de

ces interactions. En effet, tenir compte de la nature de ces interactions aurait impliqué une considération du contexte pédagogique sous-entendu dans IDE. Ce contexte préconise effectivement une cognition ancrée dans la réalité et l'action.

GTE (Van Marcke, 1998) est un système auteur qui propose une épistémologie de l'expertise pédagogique et des modèles pédagogiques associés. Pour définir les connaissances qui composent cette expertise, on utilise la notion de tâche pédagogique. Il s'agit d'une description générique de ce qu'un tuteur fait dans une situation particulière d'instruction. Dans GTE, le DC n'est pas considéré en fonction de son contexte pédagogique. Il est plutôt mentionné comme une suite prédéterminée d'opérations associées à l'analyse et à la caractérisation des réponses de l'apprenant dans un STI.

Cream-Tools (Nkambou, Frasson, et Gauthier, 2003b) est un système auteur qui propose un ensemble de modèles pédagogiques dont l'interprétation permet la planification formelle et intelligente d'un cours.

REDEEM (Ainsworth et al. 2003) est centré sur la description des connaissances reliées à l'expertise d'enseignement, notamment en ce qui concerne le choix d'une stratégie tutorielle en fonction de l'état cognitif d'un apprenant. REDEEM utilise une phase d'interview de tuteurs humains. Ces tuteurs y sont invités à spécifier dans quelles conditions une stratégie pédagogique prédéfinie doit être sélectionnée ou modifiée. L'approche de modification de stratégie d'enseignement dans REDEEM tient compte des informations statiques du modèle de l'apprenant (état des connaissances, historique des ressources consultées, etc).

EON (Murray, 2003a) intègre un ensemble d'instruments pour supporter la conception d'un STI selon l'approche classique: le modèle du domaine, le modèle pédagogique, le modèle de l'apprenant et le modèle de l'interface.

Dans ces trois derniers systèmes, il est question de modéliser des activités pédagogiques qui auraient pu bénéficier ou même faire partie d'un processus de DC (planification des opérations de DC avec Cream-Tools, planification conceptuelle des opérations de DC avec REDEEM, modélisation formelle avec EON). Seuls les modèles proposés dans Cream-Tools tiennent compte des théories de la cognition, des théories d'apprentissage et des théories d'instruction et de conception de l'instruction. Cependant, Cream-Tools n'offre pas un cadre formel de spécification du DC pour un STI.

2.3.2. Les systèmes auteurs orientés vers la performance

DEMONSTR8 (Blessing, 2003)) fait partie d'une famille de systèmes auteurs pour la conception des tuteurs cognitifs. Dans ces tuteurs, l'approche de DC utilise la technique dite de "*model tracing*" (voire Chapitre 3), basée sur le modèle psychologique ACT* (Anderson, 1983, 1993). DEMONSTR8 permet seulement la construction des éléments nécessaires à l'application du "*model tracing*" dans l'apprentissage de l'arithmétique. De plus, le modèle ACT* se base sur une vision purement rationaliste de la cognition. Néanmoins, aucune référence conceptuelle n'assure que l'usage de DEMONSTR8 sera toujours effectué lors de la conception d'un STI basé sur cette optique de la cognition.

DIAG (Towne, 2003) est un environnement permettant de construire des simulations d'appareils électroniques. Dans ce système, la pédagogie est implicite et il n'y a pas de modèle de l'apprenant et donc pas de diagnostic approfondi. La seule information que le tuteur exploite est la performance de l'apprenant lors de son dernier exercice. Cette information est utilisée pour ajuster le niveau de difficulté du prochain exercice. Malgré l'absence de formalisation du DC, on peut s'inspirer de DIAG puisque son fonctionnement en lui-même évoque certains des principes et des concepts intervenant dans le DC: spécification des relations de causes à effets, classement des causes probables ou localisation de la cause et raffinement de ce classement, détermination de la cause la plus probable. Cependant, ces éléments du DC sont assez génériques. Il faudrait en plus une référence permettant de définir leurs différentes instances en fonction du contexte pédagogique du STI où sera mis en œuvre le futur module de DC.

2.3.3. Les systèmes auteurs hybrides

Il s'agit des systèmes auteurs qui permettent la modélisation d'une fonction pédagogique, ainsi que la construction d'outils informatiques permettant concrétisation de cette fonction. IRIS (Arruarte et al., 2003) se base sur une théorie de l'instruction héritée du modèle CLAI (Arruarte, Fernandez-Castro, et Greer, 1996). Ce modèle intègre les processus cognitifs de l'apprenant, les événements et les actions d'instruction pour générer automatiquement un STI. Chacun de ces aspects est explicitement défini dans le système auteur, notamment le DC qui est une forme d'action d'instruction. La composante de DC dans IRIS est basée sur deux éléments principaux: la création des exercices en fonction des objectifs poursuivis et

l'association de chaque exercice à un processus générique de DC issu du système DETECTIVE (Ferrero, Fernandez-Castro, et Urretavizcaya, 1999).

Malgré cet abord explicite et formel du DC et de son lien avec un contexte pédagogique, DETECTIVE fournit une définition unilatérale et algorithmique du DC. Notre perspective se démarque d'IRIS dans la mesure où notre but est de formuler les caractéristiques du DC ainsi que leurs relations avec le contexte dans lequel il est déployé.

2.4 . La boucle diagnostic-remédiation: stimulation de la pensée réflexive comme approche de remédiation

Dans cette section, nous examinons les principales recherches relatives à l'exploitation pédagogique du DC en général. Nous nous penchons ensuite sur l'intégration de la pensée réflexive dans l'application du DC dans un STI. En particulier, l'on se pose la question de savoir si la relation co-déterminante entre ces deux processus a déjà été soulignée et mise en œuvre lors de l'application du DC dans les STI.

2.4.1. Exploitation des résultats d'un DC en général

Dans ses approches formelles de modélisation de l'apprenant, Self (1994) relève que la raison d'être du DC est de supporter l'adaptation de certaines activités pédagogiques selon les besoins réels de l'apprenant. Nous pensons que la nature de cette activité pédagogique influence la nature des opérations qui correspondent le DC dans un STI. Par exemple, pour une tâche de révision du plan d'enseignement, il est probable que le tuteur ou le planificateur d'un STI ait besoin de savoir le niveau d'acquisition de l'apprenant par rapport à un ou plusieurs objectifs d'apprentissage (Nkambou, Frasson, et Gauthier, 1996). Par contre, si l'activité pédagogique qui exploite le résultat du DC concerne une explication ciblée, il est probable que le DC se soucie plutôt de raffiner ses hypothèses relatives aux capacités ou aux connaissances avec lesquelles l'apprenant éprouve des difficultés. Selon Self, les résultats du DC dans un STI sont exploités par trois fonctions pédagogiques principales: la planification des objectifs d'apprentissage ou des actions pédagogiques; la prédiction; et la remédiation des actions de l'apprenant.

2.4.1.1. Exploitation du DC pour la planification dynamique

Un STI devrait pouvoir planifier dynamiquement l'apprentissage (Wasson, 1998). Il s'agit de créer ou de réviser "en ligne" (en cours de session d'apprentissage) des plans d'instruction

afin de s'adapter aux besoins courants de l'apprenant. Certains auteurs ont étudié les problématiques classiques de cette nécessité, à savoir la modification des activités que l'apprenant doit réaliser pendant une session donnée (Brusilovsky et Peylo, 2003). D'autres auteurs ont attribué une place plus importante à la planification en l'associant à de nombreuses tâches tutorielles: la macro planification (planification globale des objectifs d'une session d'apprentissage, planification des leçons pour une session d'apprentissage) et la micro planification (planification des séquences d'activités de l'apprenant, planification de l'interaction avec l'apprenant au cours d'une activité: (Nkambou et Kabanza, 2001)).

2.4.1.2. Exploitation du DC pour la prédiction des actions et la remédiation

Les résultats du DC déterminent à long terme la substance du contenu du modèle de l'apprenant. Or ce modèle peut être utilisé pour prédire le comportement de ce dernier dans une tâche précise. Lors de la conception d'un STI, on peut effectuer une analyse cognitive pour comprendre les manières les plus communes de résoudre un problème. Généralement assurée par des psychologues de l'éducation (Lovett, 1998)), l'analyse cognitive permet de construire des modèles computationnels de résolution de problème. Ils sont associés à un problème ou à une activité d'apprentissage particulière. Ces modèles computationnels s'expriment soit sous la forme d'un ensemble de règles de production, soit sous la forme d'un réseau d'influences probabilistes (Mislevy, 1994).

Si l'on considère les modules de DC basés sur les règles de production, ces règles associent une action de l'apprenant avec un but. Une fois que ce but a été inféré, la prochaine étape du DC consiste à inférer la ligne de raisonnement que l'apprenant est probablement en train de suivre (Anderson et al., 1995; Conati, Gertner, et VanLehn, 2002; Conati et al., 1997; Koedinger et al., 1997). Avec cette information, on peut prédire la prochaine action qu'il va poser. D'autres approches de DC à des fins de prédiction des actions de l'apprenant se basent sur des modèles de tâches à base de contraintes. Dans ce cas, la violation d'une contrainte indique indirectement la ligne de raisonnement que l'apprenant est en train de suivre (Mitrovic, 2003).

Les représentations agrégées telles que les graphes d'influences probabilistes (dont les réseaux bayésiens sont une instance) permettent d'inférer par déduction ou par abduction les sources de difficultés de l'apprenant et donc de cibler différentes approches de remédiation (Martin et VanLehn, 1995; Nkambou et Tchétagni, 2002, 2004; Tchétagni et

Nkambou, 2002; Zapata-Rivera et Greer, 2002). Ces inférences sont basées sur des algorithmes de propagation bayésienne (Pearl, 1988).

La remédiation est l'une des principales retombées pédagogiques du DC. Dans un sens général, elle peut prendre la forme d'un indice (Conati et Van Lehn, 1996), d'une explication détaillée basée sur des exemples ou des analogies (Weber et Brusilovsky, 2001).

Sous un angle différent, l'implication de l'apprenant dans le DC constitue une approche plus profonde à la remédiation de ses difficultés, parce qu'elle favorise un suivi des hypothèses posées à propos de l'état cognitif de l'apprenant. Ce suivi peut être mis en œuvre par un mécanisme où les réactions de l'apprenant lors de la remédiation sont observées dans le but d'améliorer la qualité du diagnostic. Ce mécanisme s'élabore à travers une boucle entre le DC et la remédiation (Tchetagni et Nkambou, 2004c). L'implication de l'apprenant dans la remédiation de ses difficultés a été matérialisée en particulier par deux stratégies pédagogiques, selon l'état de l'art du domaine AIED sur cette question.

La première stratégie consiste à amener l'apprenant à s'expliquer sa propre démarche ou à l'expliquer au tuteur: il s'agit de la méthode communément appelée *self explanation* (Chi et al., 1994). Cette méthode a été largement appliquée dans les STI (Aleven, Popescu, et Koedinger, 2003; Conati et Van Lehn, 2000).

La seconde stratégie peut être considérée comme un cas particulier de l'approche *self explanation*. En bref, elle consiste à amener l'apprenant à justifier ses réponses à un exercice en faisant explicitement référence aux faits qui caractérisent cet exercice et à son propre raisonnement: il s'agit de la pensée réflexive, selon une perspective Deweyenne (Dewey 1933).

Le deuxième objectif de cette thèse est de considérer explicitement le lien entre différentes formes de pensée réflexive et le DC, afin d'en traduire les répercussions dans une conception formelle de ce processus.

2.4.2. La pensée réflexive: définition et exploitation dans le DC

Un courant récent de l'application du DC dans les STI insiste sur l'importance d'impliquer les apprenants (Kay, 1997)). Cette perspective est principalement supportée par un ensemble d'approches qui encouragent et stimulent le phénomène de la pensée réflexive chez les apprenants dans un STI. Schön (1983) décrit deux types de réflexion: la réflexion pendant l'action (*reflection in action*) et la réflexion sur l'action (*reflection on action*).

La réflexion pendant l'action survient pendant l'exécution d'une tâche cognitive: elle permet au sujet de surveiller ou de questionner ses actions pendant qu'elles sont posées.

La réflexion sur l'action survient lorsque le sujet examine et questionne ses actions rétrospectivement, après leur exécution.

L'intégration de la réflexion dans le déploiement du DC se manifeste à travers deux principales approches de modélisation de l'apprenant: l'ouverture du modèle apprenant encore appelée *Open Learner Modeling* (OLM) et la modélisation interactive de l'apprenant encore appelée *Interactive Learner Modeling* (ILM). Intuitivement, on peut associer l'approche OLM à la réflexion sur l'action, et l'approche ILM à la réflexion pendant l'action.

2.4.2.1. L'approche *Open Learner Modeling*

Avec l'approche OLM, l'apprenant peut accéder au contenu de son modèle, c'est-à-dire aux informations sur son état cognitif, telles que interprétées dans le STI. La valeur pédagogique de l'OLM repose sur les techniques de représentation des connaissances utilisées pour représenter l'état cognitif de l'apprenant.

Les représentations graphiques - par exemple, les cartes conceptuelles (Cimolino, Kay, et Miller, 2003)- permettent de représenter l'information à travers des relations conceptuelles. Elles favorisent une appréhension de la sémantique du domaine étudié à travers des connaissances déclaratives.

Les structures hiérarchiques - par exemple les espaces problèmes hiérarchiques - sont un cas particulier des représentations graphiques où la relation conceptuelle est de nature hiérarchique (Rueda et al., 2003). Les structures hiérarchiques favorisent la compréhension des relations d'ordre entre les éléments représentés. Elles peuvent donc être utilisées tant pour représenter un domaine de connaissances que pour représenter la chronologie des actions de l'apprenant.

Les apprenants peuvent aussi examiner leur niveau d'acquisition ou de maîtrise par rapport à une liste de capacités visées dans un STI (Bull, McEvoy, et Reid, 2003; Mitrovic et Martin, 2002). Dans ce cas l'objectif pédagogique de la réflexion est d'encourager les apprenants à reconnaître leurs forces et leurs faiblesses.

L'approche OLM est aussi un outil important dans l'apprentissage collaboratif. On retrouve dans la littérature des taxonomies d'approche OLM pour des besoins spécifiques de collaboration et des applications concrètes d'apprentissage collaboratif à travers des réseaux

d'entraide entre apprenants (Vassileva, McCalla, et Greer, 2003). L'approche OLM est réalisée dans un STI à travers deux méthodes principales. Un apprenant peut visualiser et modifier librement le contenu de son modèle (Zapata-Rivera et Greer, 2003), ou alors visualiser et s'impliquer dans la construction du contenu de ce modèle (Dimitrova, Self, et Brna, 2001).

2.4.2.2. L'approche *Interactive Learner Modeling*

Lorsqu'un DC est posé au cours d'une activité d'apprentissage, on peut stimuler la réflexion de l'apprenant sur les difficultés qui ont été diagnostiquées. Une façon de le faire est d'exécuter un DC interactif, c'est-à-dire un DC où les données sont explicitement demandées et recueillies auprès de l'apprenant. Les dialogues tutoriels constituent la méthode la plus couramment utilisée dans ce sens (Dimitrova, 2003a; Tsovaltzi et Matheson, 2002). Généralement, l'usage de ces dialogues permet un DC précis à travers une mise en oeuvre moins coûteuse (Self, 1999).

Certains STI implémentent des modules de traitement de langage naturel pour supporter les dialogues tutoriels. Il s'agit principalement de Auto-Tutor (Person et al., 2001), Atlas-Andes (Rosé et al., 2001), Cirsim (Evens et al., 2001), CycleTalk (Rosé et Torrey, 2004), Geometry-Explanation-Tutor (Alevan, Popescu, et Koedinger, 2003), Why-Andes (Makatchev, Jordan, et VanLehn, 2004). D'autres STI implémentent des modules de gestion de dialogue basé sur des menus ou des interfaces prédéfinis. Il s'agit principalement de: STyLe-OLM (Dimitrova, 2002), Ms Lindquist (Heffernan et Koedinger, 2002), CATO (Ashley, Desai, et Levine, 2002), Prolog-Tutor (Tchetagni, Nkambou, et Bourdeau, 2005a).

2.4.2.3. Discussion

Pour pouvoir tenir compte explicitement du lien entre le DC et son exploitation pédagogique par la réflexion de l'apprenant, il faudrait déjà savoir en quoi elle consiste exactement. Si l'on considère par exemple les dialogues tutoriels dans un STI, l'on ne saurait juger si la réflexion qui émerge de ces dialogues n'est qu'un effet de bord de leur nature interactive, ou si elle correspond à une tentative réelle de l'apprenant de comprendre de manière fondée un problème et sa solution. En fait, ces dialogues ne devraient pas seulement amener l'apprenant à construire la solution correcte à un exercice. Ils doivent surtout mener à

une compréhension des fondements de cette solution, par rapport au problème posé dans l'exercice, aux capacités associées au domaine étudié.

La majorité des travaux susmentionnés ne définissent pas explicitement ce qui est entendu par réflexion, encore moins le lien entre la réflexion et l'approche de DC qu'ils décrivent. Considérant par exemple les dialogues tutoriels, il est certain qu'ils permettent à l'apprenant de construire la solution à un exercice. En ce qui concerne leur compréhension, une question demeure : l'apprenant a-t-il deviné la bonne réponse sur la base des questions du tuteur ou a-t-il vraiment compris les enjeux et les raisons de la solution de l'exercice ?

Les dialogues tutoriels de Auto-Tutor sont basés sur une structure formelle, mais trop générique pour garantir que le résultat du DC correspondant permet une réflexion effective.

Les dialogues de construction de connaissances (KCD) d'Atlas-Andes amènent l'apprenant à construire la solution à un exercice. Cependant, la nature de ce qui est réellement appris à travers ce dialogue est floue. En effet, le contenu de ces dialogues est fortement centré sur les caractéristiques du problème et de sa solution. Aucune définition formelle de la pensée réflexive ou de son résultat ne guide la mise en œuvre de ce dialogue.

Les dialogues de Geometry-Explanation-Tutor ont une structure conceptuelle qui s'approche de la description formelle du processus de pensée réflexive par Dewey. Cependant, ces dialogues sont utilisés à des fins de réflexion sur l'action, et non une réflexion survenant après un DC.

2.5 Conclusion

La conception et la mise en œuvre du DC dans les STI ont été analysées de manière globale dans ce chapitre. Les travaux sur les *shells* de modélisation de l'apprenant, les STI modulaires et les STI multi-agents ont montré qu'en général, le DC n'est pas une fonction dont la conception formelle est considérée à part entière dans la conception globale des STI.

Les contributions récentes dans le sous-domaine des systèmes auteurs ont également été examinées, ces derniers étant reconnus pour concevoir les STI sur la base de principes issus de théories de cognition, d'apprentissage et d'instruction. En dehors du système IRIS (Arruarte et al., 2003), aucun des systèmes examinés ne propose un cadre de conception formelle pour le DC. De plus, aucun des systèmes auteurs analysés ne spécifie le lien formel qui existe entre le DC et les théories de cognition.

Après avoir rappelé les différentes façons dont le résultat d'un DC est exploité dans un STI, nous nous sommes penchés sur le cas de la pensée réflexive. Dans un souci «pédagogique» par rapport au fonctionnement des STI, l'on encourage de plus en plus la stimulation de la pensée réflexive chez l'apprenant, en particulier après un DC. On a pu constater que de nombreux travaux ont adopté cette perspective, mais un défaut est demeuré au fil des contributions analysées: aucune définition formelle du processus de pensée réflexive n'est donnée (bien que des repères conceptuels existent). Il va de soi que cet état de fait rend impossible l'explicitation du lien entre les instances de ce phénomène et l'algorithme de DC qui en favoriserait la stimulation effective.

Cette analyse nous permet donc de confirmer une absence de formalisme au niveau de la conception du module de DC dans un STI. Dans le domaine AIED, il n'existe pas d'artéfact (informatique ou conceptuel) dont le but est de permettre une spécification formelle du DC dans le sens de favoriser une fidélité de l'implémentation et de la mise en oeuvre dans un STI par rapport à l'intention pédagogique de départ. De même, il n'existe pas de travaux qui préconisent une vision du DC où l'intention pédagogique est matérialisée par une boucle co-déterminante diagnostic-remédiation, en l'occurrence lorsque cette remédiation correspond à la stimulation de la pensée réflexive. Ceci confirme l'originalité de la contribution de cette thèse sur ces points d'interrogation.

Les prochains chapitres développent la substance de cette contribution en proposant un cadre de spécification pour le DC dans un STI. Ce cadre s'appuie sur: (1) des travaux récents sur la formalisation des éléments du DC, du processus de DC, de leurs liens avec les différentes théories de la cognition et des implications correspondantes dans l'implémentation du DC dans un STI (Tchetagni, Nkambou, et Bourdeau, 2005b) et (2) des travaux récents relatifs à une description formelle du processus de la pensée réflexive et de son lien avec le DC qui le précède (Tchetagni, Nkambou, et Bourdeau, 2005a).

Chapitre III

CADRE DE SPÉCIFICATION POUR LE DIAGNOSTIC COGNITIF DANS UN STI

3.1 Introduction

Dans cette thèse, nous posons comme hypothèses que: (1) les implications inhérentes à la nature des exploitations pédagogiques des résultats du DC dans un STI font que ces phases d'exploitation doivent y être intégrées, en l'occurrence à travers une boucle diagnostic-remédiation, (2) l'influence du paradigme de cognition qui sous-tend une activité d'apprentissage dans un STI, sur le DC qui sera appliqué au cours de cette activité, doit être prise en compte lors de la conception de cette fonction pédagogique et (3) un cadre de spécification du DC (CSDC) doit expliciter et représenter les incidences des deux hypothèses précédentes, pour favoriser leur considération dans les pratiques de conception et d'implémentation du DC dans les STI. Ces perspectives interpellent une certaine réflexion où il convient entre autres: de définir ce que l'on entend par "cadre de spécification", d'en préciser l'utilité dans la conception et dans l'implémentation du DC pour un STI et enfin, d'en décrire les éléments ainsi que leurs interrelations. Le but de ce chapitre est de

développer ces points pour mettre en exergue la contribution de cette thèse au niveau de la conception du DC pour les STI.

3.2 Cadre de spécification: signification et utilités

Le but de cette section est de préciser la notion de "cadre de spécification", telle que abordée dans cette thèse. Dans la même lancée, nous mettons en relief les bénéfices d'un tel concept lors de la construction d'un artefact en général et en particulier, lors de la construction du module de DC dans les STI.

3.2.1. Qu'entend-on par "cadre de spécification" ?

Par ce vocable, l'on essaie de rendre en français, le mot anglais "*framework*". Le dictionnaire Oxford Anglais/Français (Corréard et al., 2004) propose les traductions suivantes pour ce mot: charpente, structure, armature, encadrement, châssis, trame. Leur point commun est qu'elles réfèrent toutes à une structure de base sur laquelle s'élabore la construction d'une entité; en d'autres termes il s'agit d'*une base de construction*.

D'un point de vue informatique, un cadre est une structure de données (schéma) permettant de décrire les connaissances relatives à une entité, sous forme d'un ensemble d'attributs, et de procédures liées à ces attributs. Cette définition capture bien notre objectif d'établir une structure décrivant les caractéristiques du DC.

En fait, on parle de "cadre de spécification" plutôt que de "cadre" tout court. En effet, cet objet doit être utilisé par un concepteur pédagogique pour spécifier une instance de DC à implémenter dans un STI. Ce cadre devrait donc permettre de mieux comprendre la signification des caractéristiques du DC, les instances possibles de ces caractéristiques, et les circonstances du choix de chaque instance.

3.2.2. Utilité d'un CSDC

L'originalité du concept de CSDC est qu'il supporte individuellement le concepteur d'un STI et le programmeur d'un STI, tout en instaurant un pont sémantique entre ces deux intervenants. Dans les prochaines lignes, nous expliquons comment la notion de cadre telle que conçue dans cette thèse favorise *a priori* ces fonctionnalités.

3.2.2.1. Utilité du CSDC pour la conception du DC pour un STI

Le *grand dictionnaire terminologique* (Office québécois de la langue française (1997)) associe trois définitions à la notion de « cadre »: un cadre est une « bordure », un « plan », un « arrangement » et enfin un « conteneur ».

En tant que « *bordure* », le cadre définit les frontières d'une entité. Dans cette perspective, le CSDC revêt une utilité conceptuelle car il situe les limites dans lesquelles le DC dans un STI doit être compris. Ces limites ne doivent pas être considérées comme des restrictions qui sont imposées au concepteur pédagogique. Au contraire, le CSDC doit plutôt être perçu comme un point de départ à la conception du DC en exposant de manière explicite les éléments clés de ce processus.

En tant que plan, le CSDC définit explicitement le déroulement du DC dans une situation particulière d'apprentissage.

En tant qu'arrangement, le CSDC définit aussi l'organisation logique des composantes du DC. Dans ce sens, son rôle est de permettre au concepteur pédagogique de comprendre les relations entre les différentes composantes du DC. Une appréhension correcte de ces relations favorise la définition d'un processus cohérent.

Pour résumer, le CSDC permet d'articuler explicitement les connaissances relatives au DC, connaissances qui jusqu'ici demeurent implicites « dans l'esprit » d'un concepteur pédagogique. Plus concrètement, le CSDC permettrait:

- En tant que « bordure », de prendre connaissance des éléments du DC et de leurs interrelations;
- En tant que « plan et arrangement », de prendre conscience de certains éléments du DC qui auraient pu être ignorés ou implicitement pris pour acquis, alors que leur considération devrait être formelle dans le déroulement du DC;
- En tant que « plan et arrangement » encore, de comprendre les interrelations entre les éléments du DC;
- En tant que conteneur, d'instancier ou de sélectionner des éléments du DC afin de guider l'implémentation de ce processus – par un programmeur de STI.

3.2.2.2. Utilité du CSDC pour l'implémentation du DC dans un STI

Le DC est un des sujets qui suscitent le plus d'intérêt dans la recherche sur la modélisation de l'apprenant dans les STI. Trois difficultés majeures émergent lorsqu'il s'agit de l'implémentation « computationnelle » de ce processus: le choix et la mise en œuvre de la technique d'intelligence artificielle (IA) ou de l'algorithme informatique qui assurera le processus en tant que tel, (2) le choix du type de données à observer pour effectuer le DC et, (3) l'exploitation pédagogique qui sera fait de ces inférences.

En fait, le contexte pédagogique influence la nature des indices qui seront observés, la nature du résultat du diagnostic en tant que tel et dans une certaine mesure, le choix des techniques de représentation des connaissances qui seront utilisées. L'exploitation pédagogique du DC influence la manière dont l'algorithme de DC devrait se dérouler. Une certaine réflexion lors de la conception du DC est donc nécessaire pour poser des choix judicieux par rapport à ces aspects.

Le rôle d'un cadre de spécification est d'affranchir le concepteur et le programmeur du module de DC d'un STI de cette étape de réflexion, ou tout au moins de leur offrir un point de départ. En effet, l'une des justifications de cette thèse réside dans le fait que les trois points susmentionnés ne sont pas explicitement considérés lors de la mise en oeuvre du DC dans les STI. Le CSDC doit expliciter des recommandations sur ces relations, pour permettre au concepteur pédagogique de définir des instances cohérentes du DC. Par exemple :

- ⇒ *si le DC s'élabore dans un contexte rationaliste, on recommande un DC de type comportemental (voir section 3.3);*
- ⇒ *si l'activité d'apprentissage vise la résolution d'une tâche précise, on recommande également de représenter l'état cognitif de l'apprenant en terme de son raisonnement, en utilisant par exemple un système de règles de production.*

3.3 CSDC de base

Cette section présente les composantes de base du DC appliqué dans un STI. L'objectif est d'articuler les éléments acceptés de manière consensuelle dans le domaine AIED comme faisant partie de ce processus. Après avoir décrit les principaux types de DC, les principales composantes du DC, et les principales relations du DC avec les modules d'un STI, nous proposons une analyse permettant de recommander pour chaque type de DC, les instances de composantes (de DC) les plus appropriées. Le but de cette analyse est de maintenir une

cohérence entre le but pédagogique d'un DC (implicitement défini à travers son type) et les mécanismes qui permettront de l'atteindre dans un STI. Cette analyse constitue un aspect de l'originalité de notre contribution au niveau de la conception du DC pour les STI car les relations qui en ressortent n'ont jamais été explicitement évoquées et justifiées dans le domaine AIED, relativement au DC.

3.3.1. Composantes principales du DC

Les prochaines lignes décrivent les caractéristiques conceptuelles et opérationnelles fondamentales du DC dans les STI. Un accent particulier est mis sur la nature et l'implémentation des deux principaux modules pourvoyeurs d'informations au DC dans un STI: le module du domaine et le module de l'apprenant. L'implémentation de ces modules se base surtout sur des approches précises de représentation des connaissances en IA. On constatera de fait que pour implémenter adéquatement chaque type de DC, certaines de ces approches sont plus souhaitables que d'autres, d'où l'importance de les mentionner explicitement.

3.3.1.1. Types et opérations de diagnostic

Dans un STI, le DC doit être envisagé selon deux points de vue: le point de vue conceptuel et le point de vue opérationnel.

Conceptuellement, le DC possède plusieurs caractéristiques qui qualifient la nature de son résultat et la nature des informations qu'il utilise. Par exemple, la caractéristique "type de diagnostic" indique l'aspect cognitif de l'apprenant auquel les hypothèses de DC réfèrent (Wenger, 1987a): soit l'état de ses connaissances (*diagnostic épistémique*); (2) soit son comportement ou sa stratégie de raisonnement lors de la résolution d'un exercice (*diagnostic comportemental*); soit ses attitudes et ses émotions lors de l'apprentissage (*diagnostic individuel*). Dans le reste de ce chapitre, le terme "connaissance(s)" est utilisé au sens large. Dans la conception pédagogique, on distingue clairement les connaissances, les habiletés et les capacités (Paquette, 2002). Pour des raisons de simplicité de langage, le terme "connaissance(s)" est utilisé pour référer à l'état cognitif de l'apprenant *relativement au contenu du domaine étudié* dans un STI. Les prochains chapitres abordent concrètement l'application du DC dans les STI et dans ce cas, une définition claire des notions de connaissance, habileté et capacité sera proposée.

Opérationnellement, le DC incorpore des mécanismes et des opérations concrètes qui mènent effectivement à des hypothèses sur l'état cognitif de l'apprenant. Selon Mislevy (1994), on peut considérer qu'analogiquement au diagnostic médical, le DC est une sorte d'abduction. Son résultat découle de deux opérations de base (voir section 1.2.1.3. ci-dessus, en page 16): (1) l'acquisition d'indices sur les actions de l'apprenant; (2) l'inférence d'hypothèses à propos de l'état cognitif de l'apprenant. (Tableau III.1).

Le diagnostic épistémique infère l'état des connaissances de l'apprenant qui pourrait expliquer son comportement ou sa performance lors de la résolution d'un exercice.

Le diagnostic comportemental infère le comportement observable et le comportement non observable de l'apprenant (ce sont des inférences nécessaires dans un STI, bien qu'elle n'ont pas lieu d'être avec un tuteur humain).

Le diagnostic individuel aborde la dimension de la personnalité propre de l'apprenant et son impact sur le processus d'apprentissage: motivation, préférences, attitudes, émotions. L'individualité a été formellement reconnue comme facteur déterminant de l'apprentissage (Faivre, Nkambou, et Frasson, 2002). Toutefois, les principes conceptuels qui caractérisent cette influence dans un STI demeurent un sujet de recherche récent, bien que prometteur. Pour cette raison, nous avons choisi de nous concentrer sur les deux premiers types de DC dans un STI, leurs propriétés étant mieux comprises du fait qu'ils ont été l'objet de nombreuses recherches dès la naissance du domaine AIED.

3.3.1.2. Représentation des connaissances nécessaires à la réalisation d'un type de DC

Un STI est basé sur quatre modules principaux: le module du domaine (ou de l'expert), le module de l'apprenant, le module pédagogique et le module de communication. Chacun de ces modules intègre une ou des bases de connaissances matérialisées à travers des approches de représentations connaissances. Nous considérons que le DC est une fonction pédagogique réalisée par un tuteur et pour cette raison, nous considérons le module de DC dans un STI comme une partie de son module pédagogique. Dans un STI, le DC s'appuie sur deux principales sources d'informations (Figure III.1): la représentation des connaissances relatives au domaine étudié (partie du module du domaine) et la représentation de l'état cognitif de l'apprenant (partie du module de l'apprenant).

Tableau III.1. Un cadre de spécification de base pour le DC dans un STI

	Type de diagnostic cognitif				
	Comportemental		Épistémique		
	Évaluation sans inférence	Inférence du comportement non observable	Basé sur l'évaluation sans inférence	Basé sur l'inférence du comportement non observable par « <i>Model Tracing</i> »	Basé sur l'inférence du comportement non observable par l'interprétation reconstructive pure
Étapes du DC					
Acquisition d'indices	✓ Solution finale de l'apprenant à un exercice	✓ Solution finale ✓ Solution construite incrémentielle	✓ Solution finale ✓ Solution construite de manière incrémentielle	✓ Solution construite de manière incrémentielle	✓ Solution finale ✓ Solution construite de manière incrémentielle
Inférence d'hypothèses	✓ Caractérisation de la solution de l'apprenant (Ex : Évaluation de la solution) ✓ Classification de la solution ✓ Reconnaissance du processus de	✓ Reconnaissance de la stratégie de résolution ✓ Reconnaissance du raisonnement de l'apprenant	Premier niveau	✓ Connaissance nécessaire à la génération de la bonne solution	✓ Connaissances associées aux buts de l'apprenant
				✓ Connaissances associées aux buts de l'apprenant ✓ Connaissances compilées associées aux actions identifiées	

	Type de diagnostic cognitif				
	Comportemental		Épistémique		
	Évaluation sans inférence	Inférence du comportement non observable	Basé sur l'évaluation sans inférence	Basé sur l'inférence du comportement non observable par « <i>Model Tracing</i> »	Basé sur l'inférence du comportement non observable par l'interprétation reconstructive pure
			Cohérence structurale Sous connaissances dont l'absence est responsable d'une solution incorrecte ✓ Relations entre des sous connaissances dont l'absence est responsable d'une solution incorrecte	✓ Connaissances décompilées dont l'absence est responsable d'une solution incorrecte	✓ Sous connaissances associées à la poursuite d'un but dont l'absence est responsable d'une solution incorrecte ✓ Relations entre des sous connaissances associées à la poursuite d'un but responsables d'une solution incorrecte
			Cohérence Inocitidinale	Mise à jour de l'état des connaissances de l'apprenant à partir des connaissances diagnostiquées	

	Type de diagnostic cognitif				
	Comportemental		Épistémique		
	Évaluation sans inférence	Inférence du comportement non observable	Basé sur l'évaluation sans inférence	Basé sur l'inférence du comportement non observable par « <i>Model Tracing</i> »	Basé sur l'inférence du comportement non observable par l'interprétation reconstructive pure
Recommandations relatives à l'implémentation					
Techniques de représentation des connaissances en IA					
Module du domaine	<ul style="list-style-type: none"> ✓ Représentation structurelle (<i>Frames</i>) des patrons possibles de solution 	<ul style="list-style-type: none"> ✓ Représentation structurelle (<i>Frames</i>) des patrons possibles de solution ✓ Système à base de règles de production (et Espaces problèmes) ✓ Bibliothèques de plans 	Identiques aux représentations nécessaires pour l'Évaluation sans Inférence	Identiques aux représentations nécessaires pour l'Inférence du comportement non observable	Identiques aux représentations nécessaires pour l'Inférence du comportement non observable

	Type de diagnostic cognitif				
	Comportemental		Épistémique		
	Évaluation sans inférence	Inférence du comportement non observable	Basé sur l'évaluation sans inférence	Basé sur l'inférence du comportement non observable par « <i>Model Tracing</i> »	Basé sur l'inférence du comportement non observable par l'interprétation reconstructive pure
Composantes de l'Algorithme de DC	<ul style="list-style-type: none"> ✓ Heuristiques <i>Machine-Learning</i> pour la reconnaissance et la catégorisation des solutions 	<ul style="list-style-type: none"> ✓ <i>Machine-Learning</i> ou <i>Pattern Matching</i> pour reconnaître les solutions ✓ Systèmes experts pour les règles de production ✓ Arbres et algorithmes de recherche dans les arbres pour les espaces problèmes 	<ul style="list-style-type: none"> ✓ Idem Évaluation Sans inférence ✓ Relier explicitement les connaissances à une catégorie ou à une caractéristique de solution ✓ Relier explicitement les connaissances aux sous connaissances 	<ul style="list-style-type: none"> ✓ Idem Inférence du comportement non observable ✓ Relier les connaissances compilées aux connaissances du domaine 	<ul style="list-style-type: none"> ✓ Idem Inférence du comportement non observable ✓ Relier explicitement les connaissances à une catégorie ou à une caractéristique de solution ✓ Relier explicitement les connaissances aux sous connaissances
Domaine étudié	Domaines déclaratifs et procéduraux	Domaines procéduraux et de résolution de problèmes	Domaines déclaratifs	Domaines procéduraux et de résolution de problèmes	Domaines déclaratifs, procéduraux et de résolution de problème

Les informations du *module du domaine* fournissent au DC des descriptions sur les interrelations épistémiques entre les connaissances, de même que sur les relations entre ces connaissances et divers comportements (actions) de l'apprenant. Les relations entre les éléments du module du domaine et le comportement de l'apprenant sont formalisées par le biais d'une structure appelée "Modèle Causal" (Tchetagni et Nkambou, 2004b).

Les informations du *module de l'apprenant* fournissent au DC des descriptions sur l'état des connaissances de l'apprenant. Ces informations permettent par exemple au DC de choisir entre des hypothèses de diagnostic, celle qui est la plus plausible.

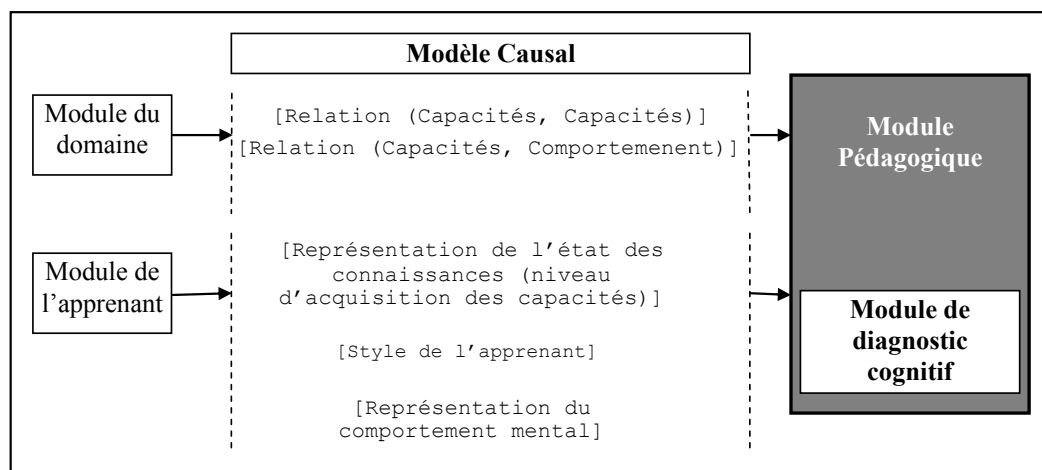


Figure III.1 Relation entre le DC et les modules d'un STI

3.3.1.2.1. *Le module du domaine et le diagnostic cognitif*

Les STI sont mis en œuvre autour de la réalisation d'activités d'apprentissage. Certaines de ces activités présentent les informations relatives au contenu du domaine tandis que d'autres abordent directement des exercices à travers lesquels les apprenants acquerront des capacités par rapport aux connaissances du domaine étudié. Dans tous les cas, des connaissances sont explicitement associées à toute information présentée à l'apprenant à travers une modélisation du domaine étudié. La modélisation cognitive d'un domaine de connaissances peut porter sur: (1) le contenu de ce domaine et dans ce cas, on parle du modèle des connaissances; (2) le contenu relatif à la manière de résoudre un problème particulier dans le domaine et dans ce cas on parle du modèle de l'expert.

La modélisation cognitive des connaissances d'un domaine comprend deux tâches principales: l'analyse du contenu du domaine en termes de concepts, de principes et de procédures (Gagne, 1996); l'organisation du contenu formulé en termes de leurs interrelations: par exemple, les relations de pré-requis, de composition, d'analogies, etc.(Nkambou, Frasson, et Gauthier, 1996). D'un point de vue informatique, quatre principales approches de représentation des connaissances en intelligence artificielle (IA) peuvent être utilisées pour représenter les capacités visées par rapport à un domaine appris par le biais d'un STI:

- les représentations purement logiques (Anderson, 1983);
- les représentations graphiques : les réseaux sémantiques (y compris les ontologies), les graphes conceptuels (Sowa, 2000), les réseaux d'influence (Pearl et Russell, 2003);
- les représentations structurelles encore appelées *frames* (Minsky, 1975);

La modélisation cognitive de l'expert permet de représenter la manière dont un expert du domaine étudié résout une tâche particulière au domaine (Anderson et al., 1987). Ce processus intègre généralement: l'analyse cognitive de la tâche à travers laquelle on représente l'expert; l'expression formelle d'un modèle de tâche dont l'exécution devrait normalement reproduire le comportement de l'expert. Les systèmes à base de règles de production, les espaces problèmes, les plans avec les générateurs de plans, les représentations structurelles (*frames*) sont les principales techniques de représentation des connaissances en IA qui sont utilisées pour représenter le processus de « résolution de problème » d'un expert dans un domaine. Les graphes de tâches prédéfinis (car en fait ils peuvent être dynamiquement générés à partir d'un système de règles de production) peuvent aussi supporter ce mécanisme (Newell, Rosenbloom, et Laird, 1989; Newell et Simon, 1972). Selon les objectifs du DC, des utilisations combinées du modèle du domaine et du modèle de l'expert sont possibles.

3.3.1.2.2. *Le module de l'apprenant et le diagnostic cognitif*

Dans un STI, le module de l'apprenant comporte deux principaux aspects (Wenger, 1987a). Premièrement, il renferme de l'information générale sur l'apprenant: les exercices qu'il a réalisés, la manière dont il les a réalisés (raisonnement, processus de solution), la performance à ces exercices, les sections du système qu'il a parcourues, le temps passé dans chaque section, etc. Ces informations peuvent être regroupées dans le *modèle*

comportemental de l'apprenant. Deuxièmement, il renferme une représentation explicite de l'état des connaissances de l'apprenant qui ont été inférées. Ces informations peuvent être regroupées dans le *modèle épistémique de l'apprenant*. Ces composantes du modèle de l'apprenant correspondent aux niveaux de granularité de représentation de l'apprenant dans les STI.

D'un point de vue concret, certains STI modélisent la manière dont l'apprenant résout les problèmes. Ces systèmes sont communément appelés "*Cognitive Tutors*" (Anderson et al., 1995). Dans ce cas, le modèle comportemental de l'apprenant est établi sur la base d'une modélisation cognitive de l'expert augmenté de quelques éléments correspondant à un comportement erroné (pour pouvoir interpréter les erreurs de l'apprenant).

D'un point de vue plus abstrait, certains STI modélisent l'état des connaissances de l'apprenant inféré à partir de ses résultats aux problèmes résolus. Dans ce cas, le modèle épistémique est une fonction de la représentation des connaissances du domaine.

Les travaux dans le domaine AIED ont permis de distinguer quatre principales approches de modélisation de l'apprenant dans un système tuteur intelligent : le modèle *Overlay* (Carr et Goldstein, 1977), le modèle des erreurs (Brown et Burton, 1978), le modèle du comportement cognitif (Anderson et al., 1995), et le modèle à base de contraintes (Ohlsson, 1994).

Les modèles de type "overlay"

Dans cette approche, le modèle de l'apprenant est conceptualisé comme un sous-ensemble des connaissances du domaine ou du comportement de l'expert. La formulation de ce sous-ensemble peut varier:

- en termes du niveau d'acquisition par rapport à chaque connaissance (probabilités, niveaux discrets)
- en termes d'un sous-ensemble des connaissances proprement dites;
- en termes d'un sous-ensemble des comportements de l'expert;

Les modèles de type *overlay* sont généralement construits à partir d'un *diagnostic cognitif épistémique* (et vice-versa). Même lorsqu'ils correspondent à un sous-ensemble des comportements de l'expert, ses comportements sont considérés comme des connaissances "compilées" (voir ci-dessous).

Il n'est pas toujours question d'acquisition ou de non acquisition dans la modélisation de l'apprenant. En effet, une notion peut être acquise ou comprise de façon erronée (*misconception*) et cet état de fait devrait pouvoir être modélisé. Les modèles de simulations d'erreurs ou à base de bibliothèques d'erreurs (*bugs catalog models*) permettent de répondre à cette question.

Les bibliothèques d'erreurs

Les bibliothèques d'erreurs consignent les différentes façons dont les étudiants commettent des erreurs dans une tâche particulière (Nwana, 1993). Chaque erreur est reliée à un concept ou à une règle mal comprise. L'erreur ainsi modélisée permet d'expliquer les difficultés rencontrées par les apprenants et par conséquent, d'apporter une remédiation plus précise de l'erreur diagnostiquée. Les bibliothèques d'erreurs impliquent naturellement une observation de l'apprenant lors de la résolution d'un exercice. Le modèle de l'apprenant qui consigne les erreurs reconnues grâce à une bibliothèque d'erreurs est donc construit à partir d'un *diagnostic cognitif comportemental* (et vice-versa).

Le modèle du comportement cognitif

Cette approche a pour but de représenter le comportement de l'apprenant. Pour pouvoir reconnaître le comportement de l'apprenant, on utilisera généralement un modèle de l'expert augmenté de quelques éléments correspondant à un comportement erroné (pour pouvoir interpréter les erreurs de l'apprenant). Au terme de cette démarche, on obtient idéalement un modèle représentant le comportement de l'apprenant lors de la réalisation d'exercice:

- une suite d'actions posées;
- une suite de principes et de règles appliqués, de concepts utilisés;
- une suite de buts poursuivis;
- une structure de solutions.

Le modèle du comportement cognitif de l'apprenant est construit sur la base d'un *diagnostic cognitif comportemental* (et vice-versa). Une des tactiques de DC comportemental résulte d'une théorie de l'apprentissage appelée ACT* (Anderson, 1993). ACT* propose un modèle de l'apprentissage où les connaissances d'un domaine sont peu à peu *compilées* afin d'être directement utilisables dans un contexte de résolution de problème. La compilation des connaissances désigne la transformation des connaissances d'un domaine dans un format de règle de production directement utilisable dans la pratique ou dans l'action.

Les modèles à base de contraintes

Tout en rappelant que l'utilité du modèle de l'apprenant est avant tout de guider et d'adapter les choix pédagogiques en fonction des besoins de l'apprenant, Ohlsson (1994) soutient que seuls certains maillons de sa démarche sont réellement importants pour la compréhension et le suivi de son comportement. Un domaine ou un exercice peut être modélisé en terme de contraintes. Chaque contrainte permet de définir les classes d'actions de l'apprenant correspondant à un certain type d'erreurs. Dans ce cas, le module apprenant, de manière analogue aux modèles de type *overlay* correspond à l'ensemble des contraintes qu'il ne respecte pas. Le modèle de l'apprenant à base de contraintes est construit à partir d'un *diagnostic cognitif comportemental* (et vice-versa) puisqu'il porte sur une catégorie particulière du comportement de l'apprenant (les actions correspondant aux contraintes non respectées).

3.3.2. Méthodologie de définition du CSDC

Les caractéristiques conceptuelles et opérationnelles du DC sont connues et relativement faciles à comprendre. L'objectif du cadre de base est surtout de dresser un réseau de relations entre ces caractéristiques, afin de favoriser des instances cohérentes de DC. Les deux prochaines sections développent un CSDC de base (Tableau III.2).

Premièrement, la description des opérations associées à chaque type conceptuel de DC est exposée.

Deuxièmement, les principaux mécanismes de représentation des connaissances et les principaux éléments algorithmiques, associés à chaque grande classe d'instances de DC sont recommandés. Ces recommandations symbolisent la traduction des exigences pédagogiques du concepteur dans un jargon computationnel compréhensible par le programmeur. Il s'agit d'un des plus importants aspects de l'utilité du CSDC: combler le fossé sémantique entre le concepteur pédagogique et le programmeur.

Finalement, chaque type de DC introduit dans le CSDC de base est illustré à travers une activité de résolution d'exercice sur un problème d'arithmétique, en l'occurrence la soustraction (Tableau III.2).

Tableau III.2. Illustrations concrètes des types de diagnostic cognitif

Étapes du diagnostic	Types de diagnostic cognitif		
	Évaluation sans inférence	Inférence du comportement non observable : Interprétation Reconstructive de la solution	Diagnostic Épistémique
Observation	Solution finale de l'apprenant : 65		
Inférence	Caractériser la solution finale		
	1	5	4
	0	8	9
	Centaines	Dizaines	Unités
	0	6	5
Mise en œuvre (Technique d'IA)	Algorithmes d'analyse du triplet (<i>machine learning</i>) [centaines/dizaines/unités]	Algorithmes d'analyse du triplet [centaines/dizaines/unités] Base de règles de production : Si But = Soustraction ($Op1_{[1,...,n]}$, $Op2_{[1,...,n]}$) → Soustraire la colonne la plus à droite Si But = Soustraire la colonne la plus à droite $\wedge Op1_n > Op2_n \rightarrow$ Emprunter dizaine $Op1_{(n-1)}$	Résultat de l'inférence du comportement non observable
		Calcul du résultat pour les unités Soustraction de 9 à 14 <i>Etc.</i>	État des connaissances de l'apprenant
			Phase 1 : Connaissances directement responsables : Principe de l'ordre de soustraction des colonnes ; Principe d'emprunt des dizaines Phase 2 : cohérence structurale : Principe de l'ordre de soustraction est analogue à celui de l'ordre d'addition des colonnes ; ... Phase 3 : cohérence longitudinale : Augmenter la probabilité d'acquisition du principe de l'ordre d'addition des colonnes ; ...
			Graphe de décompilation des connaissances compilées dans la base de règle : Association connaissances compilées et les capacités du domaine ou les objectifs d'apprentissage Représentation du modèle de l'apprenant : Base de faits indiquant le niveau de maîtrise de l'apprenant par rapport à chaque objectif d'apprentissage

3.3.3. Diagnostic cognitif: concept, opérations et exigences d'implémentation

Cette section souligne de manière plus détaillée les caractéristiques des différents types de DC précédemment mentionnés. Le but est d'expliciter la forme que prendra le processus de DC pour chaque type de DC ainsi que les mécanismes de représentations qui seront nécessaires en conséquence dans les modules du domaine et de l'apprenant d'un STI.

3.3.3.1. Le diagnostic cognitif comportemental

Dans sa forme la plus primitive, le DC dans un STI consiste à reconnaître, caractériser, classer et juger les actions observables de l'apprenant (Wenger, 1987a). Le DC comportemental établit le processus de raisonnement ou les actions qui auraient mené à la réponse fournie par l'apprenant pour un exercice donné. Deux types de diagnostic comportemental sont reconnus: l'évaluation sans inférence et l'inférence du comportement non observable.

3.3.3.1.1. *L'évaluation sans inférence*

Aperçu conceptuel et opérations

Dans certains STI, l'objectif du DC est de fournir un feedback qui tient compte le plus possible de la nature de l'action ou de la solution de l'apprenant. Pour ce faire, le système informatique doit *évaluer* puis *comprendre* les données de l'apprenant.

L'évaluation de la réponse de l'apprenant à un exercice consiste à établir si elle est correcte ou pas. Elle consiste aussi à caractériser cette réponse de manière à en obtenir une représentation formelle plus détaillée. Par rapport au processus générique de DC, cette étape peut être perçue comme la phase d'observation ou d'acquisition d'indices.

Dans le contexte computationnel d'un STI, «comprendre» la réponse de l'apprenant consiste à associer cette dernière à une solution standard. En fait, un ensemble de mesures pédagogiques est souvent associé à une solution standard. Par rapport au processus générique de diagnostic cognitif, cette étape peut être perçue comme la phase d'inférence d'hypothèses, sauf que dans ce cas, l'état cognitif de l'apprenant qui est inféré ne concerne que la forme externe de son comportement (sa réponse, sa solution).

La particularité de l'évaluation sans inférence est que le DC n'effectue pas d'inférences à proprement parler. Il s'agit d'un DC centré sur des hypothèses à propos du comportement externe et observable de l'apprenant- les réponses qu'il fournit - sans aucune référence à un modèle de son état mental.

Exigences d'implémentation: techniques appropriées d'IA

Pour réaliser une évaluation sans inférence, le module de DC d'un STI devrait disposer d'un ensemble fourni de représentations à propos de la nature correcte des réponses. Des représentations explicites des connaissances à propos de la forme ou des caractéristiques de diverses catégories (standards) de réponses sont aussi nécessaires. Étant donné ces représentations, des algorithmes de comparaison permettent d'évaluer une solution de l'apprenant: on peut utiliser des heuristiques de type "*machine learning*" pour caractériser une solution et l'associer à un standard de solutions.

L'évaluation sans inférence ne permet pas d'inférer le raisonnement, les processus mentaux ou les connaissances qui sous-tendent la génération des réponses qu'elle analyse, qu'elles soient correctes ou incorrectes.

3.3.3.1.2. *Inférence du comportement non observable*

Dans certaines situations d'apprentissage, le raisonnement qui mène à l'action de l'apprenant est plus important ou plus révélateur que le succès de cette action. Des besoins pédagogiques typiques justifient la nécessité de poser des inférences relatives au raisonnement de l'apprenant: par exemple la nécessité de comprendre comment il génère une solution – plutôt que de comprendre uniquement la nature de cette solution.

Certains domaines de connaissances sont tels qu'une approche de raisonnement peut directement révéler l'état des connaissances par rapport à ce domaine. Dans ce cas, le diagnostic du comportement non observable est d'autant plus pertinent, qu'il pourrait constituer les données de départ du diagnostic épistémique.

Enfin, des activités de résolution de problème – au sens cognitif - sont indispensables à la réalisation de certains types d'objectifs d'apprentissage (dans les domaines scientifiques notamment: la physique, la chimie, l'informatique). Les solutions produites dans de telles activités acquièrent des formes tellement complexes et variées que leur interprétation ne peut se faire indépendamment du raisonnement correspondant.

Dans un STI, les tactiques d'inférence ou de reconstruction du raisonnement évoluent sur un continuum dont le paramètre clé est la granularité de la solution ou de l'action observée. Ce continuum varie de la reconstruction pure (à partir uniquement de la solution finale de l'apprenant, sans observation intermédiaire) à la méthode appelée "*model tracing*", où chaque action est observée.

Aperçu conceptuel et opérations

En assumant que la réponse produite découle d'une succession d'intentions, la solution de l'apprenant est interprétée en termes d'une hiérarchie de buts. L'inférence de cette hiérarchie de buts se base sur un mécanisme appelé «interprétation reconstructive».

Dans l'interprétation *purement* reconstructive, le module de diagnostic «observe» d'abord la réponse ou l'action *finale* de l'apprenant. L'inférence consiste à déterminer quel plan pourrait avoir généré cette réponse. Cette inférence peut être de nature déductive ou inductive. Les inférences de nature déductive utilisent les conditions ou les contraintes caractérisant l'exercice que l'apprenant doit résoudre. Un ensemble d'hypothèses à propos de l'apprenant peut également être défini, augmentant ainsi les contraintes qui permettront de raffiner la recherche du plan le plus plausible. Par exemple: la probabilité que la réponse de l'apprenant soit erronée permet de concentrer la recherche sur les plans incluant des buts ou des sous buts erronés. Fort de ces informations, le DC infère un plan dont l'exécution devrait aboutir à la solution de l'apprenant. Les inférences de nature inductive utilisent les caractéristiques et les propriétés de la réponse de l'apprenant pour inférer sa stratégie de raisonnement (son plan).

L'interprétation reconstructive incrémentielle consiste à permettre à l'apprenant de formuler sa réponse à travers certaines étapes. Cela favorise un suivi progressif de la construction d'une solution, avec l'obtention d'informations plus précises ou plus raffinées; ce qui restreint l'espace de recherche du plan de l'apprenant. L'approche *model tracing* est un cas particulier de l'interprétation reconstructive incrémentielle: les solutions partielles de l'apprenant sont observées à *toutes* les étapes modélisées dans la résolution d'un exercice.

Exigences d'implémentation: techniques appropriées d'IA

Comment se concrétise l'inférence du raisonnement de l'apprenant dans un STI? On peut définir une librairie de plans pour chaque exercice. L'algorithme de reconstruction se déploie sur cette structure pour construire le raisonnement de l'apprenant. Par ailleurs, il est tout à fait possible d'inclure "exprès" des plans correspondant à un raisonnement erroné. Cela permet d'interpréter les réponses incorrectes de l'apprenant.

L'usage d'une librairie n'est pas toujours souhaitable, celle-ci ne pouvant pas être exhaustive. Les *espaces problèmes* permettent une reconstruction dynamique du raisonnement de l'apprenant avec des systèmes à base de règles de production. Une règle

de production associe une action ou une réponse de l'apprenant avec un but, ou un ensemble de conditions définissant sa situation actuelle dans la résolution d'un exercice. Plutôt que de faire une recherche sur une librairie de plans, le module de diagnostic peut associer chaque action de l'apprenant à une règle de production. La suite des règles associées à toutes les actions de l'apprenant permettent d'établir l'ensemble des buts qu'il a poursuivis dans son raisonnement.

3.3.3.2. Le diagnostic cognitif épistémique

Le diagnostic épistémique établit l'état des connaissances et des capacités de l'apprenant par rapport à un domaine de connaissances (Wenger, 1987a). À l'instar du diagnostic comportemental, tout part de l'observation d'une action ou d'une réponse de l'apprenant. Ensuite, l'inférence des connaissances - correctes, incorrectes (mal interprétées) ou absentes - qui expliquent le mieux les observations est déclenchée sur trois niveaux.

Aperçu conceptuel et opérations

Le premier niveau d'inférence du diagnostic épistémique doit déterminer la connaissance ou la capacité directement responsable de la réponse que l'apprenant a fournie. Ce niveau se développe sur la base d'un diagnostic comportemental: soit l'évaluation sans inférence de la réponse de l'apprenant, soit l'inférence de son raisonnement. L'évaluation de la solution de l'apprenant peut être utilisée si les questions qui lui sont présentées sont explicitement associées à une connaissance ou à une capacité. Par ailleurs, on peut considérer que le raisonnement impliqué dans la génération d'une solution ou d'une action se base sur un usage interne (non observable) des connaissances adéquates. Ainsi, lorsqu'un élément de connaissance (concepts, principes, lois, procédures) participe à la poursuite d'un but dans le plan de l'apprenant, l'on peut considérer que cet élément de connaissance a été observé.

Le deuxième niveau d'inférence du diagnostic épistémique doit interpréter le résultat du premier niveau. Ce deuxième niveau d'inférence permet d'expliquer le diagnostic d'une connaissance en termes de l'état des connaissances de l'apprenant par rapport aux composantes de cette connaissance. Ce processus est aussi désigné "cohérence structurale" du DC, mais peu de systèmes font état de sa mise en œuvre pendant le DC.

Le troisième niveau d'inférence concerne la mise à jour globale du modèle de l'apprenant (cohérence longitudinale). Il s'agit de propager l'effet du diagnostic posé en

mettant à jour les indicateurs de l'état des connaissances ou des capacités représentées dans le modèle de l'apprenant. Ces indicateurs peuvent être quantitatifs ou qualitatifs.

Exigences d'implémentation: techniques appropriées d'IA

Étant donné que le DC épistémique se base sur les résultats du DC comportemental, les mêmes techniques d'IA seront utilisées dans ce cas: librairie de plans pour la reconstruction interprétative, base de règles de productions pour l'approche *model tracing*. De plus, le DC épistémique nécessite de relier explicitement les connaissances du domaine au raisonnement de l'apprenant. Dans ce cas, l'application de certaines opérations dans l'*espace problème*, la mise en œuvre des buts dans un plan, doivent être explicitement reliées à l'utilisation d'une connaissance.

3.4 Intégration de la pensée réflexive dans la spécification du DC

Dans cette thèse, la stimulation de la pensée réflexive représente un cas concret d'exploitation pédagogique du résultat du DC. La stimulation de la réflexion influence la manière dont se déroulera le DC. Il est donc nécessaire de décrire explicitement cette influence et de l'intégrer dans la mise en œuvre de ce processus. Pour réaliser cette intégration, notre but est de montrer le lien qui doit exister entre la pensée réflexive et les éléments fondamentaux du DC. Plus spécifiquement, notre objectif porte sur la formulation de ce lien à travers un ensemble de relations qui sera ajouté dans le CSDC de base. Le but de cette analyse est de conditionner les combinaisons possibles des instances de composantes du DC afin de préserver la cohérence du processus.

3.4.1. Méthodologie d'intégration de la pensée réflexive

La pensée réflexive (aussi désignée par le terme "réflexion") dans les STI s'élabore à travers les approches *Open Learner Modelling (OLM)* et *Interactive Learner Modelling (ILM)*. Si l'on adapte les concepts de Schön (1983), l'OLM s'associe à une réflexion hors de l'action. Quant à l'ILM, cette approche s'associe à une réflexion « pendant l'action ». Dans les deux cas, l'"action" se rapporte à la réalisation d'une activité d'apprentissage dans un STI. Ces approches se rejoignent sur une caractéristique: l'introspection. Au minimum, elles induisent *une introspection tacite de l'apprenant* à propos de son état cognitif par rapport à un objectif d'apprentissage. Idéalement, elles déclenchent une extériorisation explicite et tangible de cette introspection. Il ne s'agit pas d'une introspection au sens général. Plutôt, la pensée réflexive se caractérise par l'établissement d'une réponse à un problème, sur la base d'une suite de déductions logiques déclenchées

par des faits observés et supportées par les principes applicables à ces faits. Pour comprendre son influence sur le processus de DC et l'y intégrer, l'on se doit de caractériser formellement ce phénomène de réflexion.

Pour ce faire, notre méthodologie consiste à: (1) expliciter les étapes de la pensée réflexive telle que conçue par Dewey; (2) présenter les approches OLM et ILM en général avant de les reformuler dans la perspective de Dewey. Cette reformulation a pour but de montrer comment le phénomène de pensée réflexive peut être concrètement intégré dans le DC.

3.4.2. Les étapes de la pensée réflexive selon Dewey

Le phénomène de pensée réflexive est purement psychologique. Ramené dans un contexte d'apprentissage dans un STI, sa valeur pédagogique est qu'il permet à l'apprenant de "comprendre réellement". En effet, les moyens de communication entre l'apprenant humain et le système informatique sous-jacent à un STI, sont limités. Par conséquent, il est difficile de savoir si cet apprenant devine ses réponses ou s'il les formule sur la base d'un raisonnement fondé sur la sélection et la manipulation de l'information pertinente dans l'exercice qu'il est en train de résoudre. La pensée réflexive permet de stimuler ce genre de raisonnement fondé.

Cinq composantes constituent la substance de ce phénomène cognitif (Figure III.2): (1) la présence d'une situation *problématique*; (2) l'intellectualisation de la situation; (3) l'établissement d'une hypothèse de base comme solution de la situation problématique, selon un raisonnement fondé; (4) un raisonnement accentué permettant d'élaborer l'hypothèse de base, vers une solution plus raffinée et mieux fondée; (5) l'évaluation de la solution obtenue, afin de corroborer ses implications avec les caractéristiques de la situation problématique initiale.

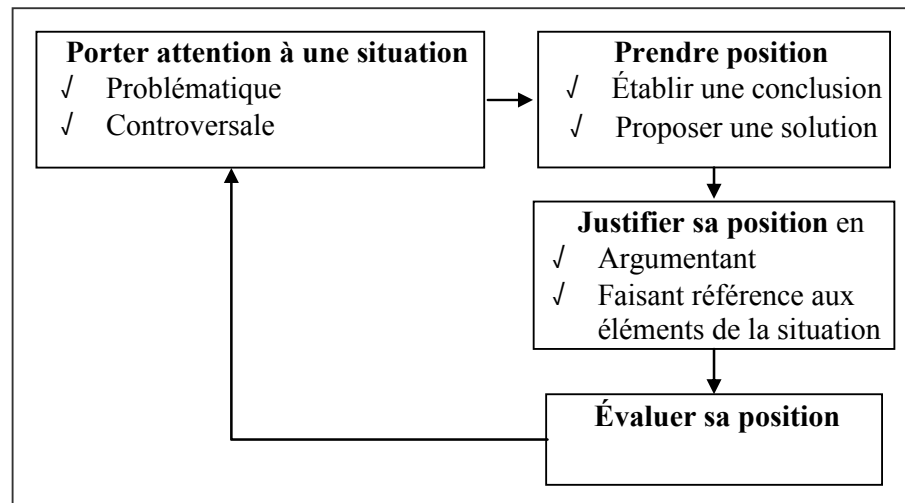


Figure III.2 La pensée réflexive selon Dewey

3.4.2.1.1. *La situation problématique*

Pour stimuler un apprentissage authentique, il ne suffit pas d'assigner une tâche à l'apprenant et de la définir comme un « problème » *par défaut*. Une tâche d'apprentissage doit par définition soulever une situation problématique. Elle doit être définie à travers une situation avec au moins une des caractéristiques suivantes: une confrontation à plusieurs conclusions possibles, une contradiction apparente ou réelle impliquant une controverse à analyser. Ces caractéristiques particulières sont nécessaires car selon Dewey, l'inhibition d'une réponse directe, triviale ou spontanée provoque un délai. Ce délai serait essentiel pour déclencher le questionnement caractéristique d'un raisonnement fondé et explicite.

Dans ce sens, tout repose sur la formulation d'une tâche, aussi triviale qu'elle puisse paraître: le but est de provoquer la curiosité de l'apprenant. Le Tableau III.3 illustre deux formulations d'un problème. La première formulation, ne laisse pas de place à la curiosité et il est fort probable que la maladie la plus communément associée aux symptômes énoncés sera proposée. La deuxième formulation nuance la situation en adjoignant aux symptômes normalement observés des paramètres qui resserrent le domaine d'analyse. Par exemple, en plus de l'analyse des symptômes de base, il faudrait évaluer la plausibilité de chaque combinaison des symptômes observés étant donné les maladies soupçonnées.

Intellectualisation de la situation

Un problème bien posé et correctement saisi est à moitié résolu. Face à une situation problématique, le blocage initial nécessite obligatoirement une inspection des conditions (les faits) de la situation *qui la rendent problématique*, leur caractérisation et si possible leur formalisation. C'est l'*intellectualisation* du problème.

Dans l'exemple du Tableau III.3, ce n'est pas la nécessité de diagnostiquer une maladie par l'analyse des symptômes qui provoque une réflexion poussée. L'apprenant doit étudier les différentes subtilités des symptômes entre les deux maladies avancées comme causes possibles. Un exemple d'intellectualisation dans ce cas est une formulation mentale des relations entre les symptômes observés et les maladies suggérées.

Tableau III.3. Deux situations de stimulation de la pensée réflexive

Situation 1	Situation 2
Le patient «X» présente les symptômes suivants : fatigue, toux, essoufflement, douleurs thoraciques. À votre avis, de quoi « X » est il atteint?	Le patient « X » présente les symptômes suivants: fatigue, toux, essoufflement, douleurs thoraciques. On soupçonne une maladie pulmonaire obstructive chronique (MPOC) du fait que le patient est un gros fumeur. Toutefois, d'autres éléments (hypertension artérielle) du dossier médical évoquent la possibilité d'une maladie cardiaque athéroscléreuse (MCAS). À votre avis, de quoi « X » est il atteint?

3.4.2.1.2. Rationalisation de base et proposition d'une solution ou d'une hypothèse explicative

L'intellectualisation est une expression formelle des caractéristiques pertinentes de la situation problématique. Cette étape est importante parce qu'elle permet la sélection des informations sur lesquelles doit se fonder le raisonnement qu'on désire provoquer explicitement. Ce raisonnement consiste à utiliser les connaissances, les lois, les principes et les heuristiques du domaine de connaissances associés au problème posé - et bien sûr applicables à la situation intellectualisée - pour établir une hypothèse qui pourrait expliquer ou résoudre la situation problématique.

3.4.2.1.3. Rationalisation avancée (composante facultative)

Dans certaines situations à riche contenu, un raisonnement plus poussé permet de raffiner la solution initialement obtenue lors du raisonnement de base. Le sujet cognitif exploite des analogies, des généralisations, des cas passés, le sens critique, l'autoévaluation, pour arriver à une solution alternative ou plus satisfaisante. Dans la réalité, la majorité des

personnes réfléchissent ainsi. Dans le contexte d'une interaction entre une machine et un humain, la modélisation d'un contenu riche en faits, en lois et surtout en historique est nécessaire, sans compter la modélisation du raisonnement de haut niveau qui sera nécessaire.

3.4.2.1.4. *Évaluation (composante facultative)*

Lorsqu'une solution est proposée pour un problème, son application devrait permettre de le résoudre. Lorsqu'une hypothèse a été établie comme explication d'un phénomène inconnu ou inexplicé, son application devrait reproduire les conditions de ce phénomène. Lorsque l'apprenant formule une réponse ou une solution à une situation problématique, il peut donc aller évaluer la justesse de cette réponse. D'un point de vue pédagogique, cette phase est d'une utilité particulière lorsque l'objectif est d'amener l'apprenant à se contredire, lorsque la solution qu'il propose n'est pas correcte.

3.4.2.1.5. *Autres considérations sur la pensée réflexive*

En plus des cinq caractéristiques principales que nous venons de mentionner, la pensée réflexive a des caractéristiques additionnelles. Il s'agit principalement des propriétés de son résultat et des phénomènes observables qui témoigneront de sa présence.

Le produit de la pensée réflexive: une formulation argumentative

Le produit de la pensée réflexive est l'idée ou le jugement qui en résulte et sa formulation doit être argumentée. En effet, cette formulation doit relier des effets (les conditions et les faits sélectionnés lors de l'intellectualisation) à des causes (les solutions, les explications) par le biais de l'utilisation des connaissances d'un domaine. La formulation argumentative est importante car elle représente une extériorisation de la troisième composante de la pensée réflexive (la rationalisation). D'un point de vue pédagogique, il s'agit du résultat observable de la pensée réflexive de l'apprenant.

L'observation de la pensée réflexive: tacite ou manifeste?

Dans un contexte d'apprentissage, il faudrait légitimer ou confirmer que le phénomène de pensée réflexive survient réellement chez l'apprenant. La pensée réflexive peut être tacite, et dans ce cas, on parle de « réflexion implicite ». La réflexion implicite n'est pas confirmable par des faits observables. Elle est hypothétique, assumée, inférée sur la base de la présence de certains catalyseurs appropriés. Dans un STI, l'accent doit donc être mis sur l'élaboration de ces catalyseurs en termes de tactiques pédagogiques. Le déploiement de ces tactiques favoriserait - *a priori* - le déclenchement de la réflexion (voir discussion ci-dessous).

La pensée réflexive peut aussi être manifeste et dans ce cas on parle de « réflexion explicite ». La réflexion explicite se caractérise par une extériorisation tangible du raisonnement; elle est donc observable. En fait, il va de soi que la pensée est intrinsèquement hermétique. Toutefois d'un point de vue pédagogique, on peut utiliser des techniques pour en favoriser l'extériorisation.

3.4.2.2. Pensée réflexive et "*Open Learner Modelling*"

Arrimer une erreur de l'apprenant à une « règle de production mal interprétée » (*buggy rule*) ou à un « concept mal compris » (*misconception*), fournir un feedback précis, sont les deux exploitations classiques du DC et du modèle de l'apprenant dans les STI. L'OLM est une conception différente du DC où le but est d'amener l'apprenant à s'auto diagnostiquer à travers un examen et une analyse générale du domaine étudié.

L'OLM est une intégration du DC avec la stimulation de la pensée réflexive. Pour comprendre comment cela est possible, il convient de se pencher sur deux questions. D'abord, comment l'approche OLM de DC favorise-t-elle la pensée réflexive? Ensuite, comment la pensée réflexive telle que précédemment formalisée s'intègre-t-elle au processus de DC dans un STI? Pour répondre à cette question, on devrait considérer deux niveaux d'intégration: l'intégration conceptuelle et l'intégration opérationnelle. L'intégration conceptuelle consiste à *associer* une forme de pensée réflexive aux caractéristiques conceptuelles du DC. L'intégration opérationnelle consiste à *insérer* une ou plusieurs phases de la pensée réflexive dans les opérations du DC.

3.4.2.2.1. "*Open Learner Modelling*" dans la perspective de Dewey

Le Tableau III.4 illustre notre interprétation de l'OLM dans la perspective de la pensée réflexive selon Dewey. Pour l'apprenant, la *situation problématique* de l'approche OLM correspond à une auto analyse de son état cognitif. Comment interpréter cette auto-analyse en termes de réflexion au sens de Dewey?

Premièrement, l'apprenant doit comprendre de quoi il est question dans le contenu de son modèle. Par exemple, il peut s'agir d'une représentation des caractéristiques de sa réponse, telle que proposée lors de la résolution d'un exercice. Il peut aussi s'agir d'une représentation de son niveau d'acquisition des différentes connaissances et capacités visées dans le STI (Tableau III.4). Ceci peut être considéré comme une *intellectualisation* du problème puisqu'il est question de relever les éléments clés sur lesquels l'auto-analyse va s'élaborer.

Tableau III.4. OLM et pensée réflexive d'après Dewey

Composante de la pensée réflexive selon Dewey	Interprétation dans le processus d'ouverture du modèle de l'apprenant (OLM)
Situation problématique	"Quel est mon état cognitif ?" Quel est la forme de ma réponse ?; Quel raisonnement a généré ma réponse ? Quelles connaissances compilées/du domaine ont participé à la génération de ma réponse ?
Intellectualisation	Prises de connaissance du contenu du modèle selon les inférences du système Quelles sont les caractéristiques de ma réponse selon l'interprétation du système? Quel était mon plan de résolution selon l'interprétation de ma réponse par le système ? Quel est mon niveau d'acquisition des connaissances ou des capacités selon l'interprétation
Raisonnement fondé et conclusion	Exemple : Contenu du modèle : Niveau d'acquisition de l'apprenant par rapport à l'ensemble des capacités relatives à la connaissance "soustraction" ; Raisonnement de l'apprenant fondé sur : ⇒ le contenu dont il a pris connaissance et le résultat de ses activités d'apprentissage dans le STI: "J'ai réussi à résoudre tous les exercices sur la soustraction" Conclusion : Je suis d'accord avec l'interprétation du système Exemple Contenu du modèle Le système a inféré le raisonnement de l'apprenant lors d'une opération de soustraction : "95-19". Représentation du Raisonnement : Liste des buts de l'apprenant en association à chacune de ses actions pendant l'exécution de la soustraction. La liste ne contient pas le But "Emprunter une dizaine à la colonne de gauche" Raisonnement fondé de l'apprenant : ⇒ "Je n'ai pas explicitement indiqué qu'une dizaine a été emprunté dans la deuxième colonne de la soustraction "95-19", mais je l'ai considéré dans ma solution qui est justement "76" Conclusion Je suis en désaccord avec l'interprétation du système

Deuxièmement, l'apprenant doit examiner chaque élément du contenu de son modèle pour évaluer l'interprétation du système sous-jacent au STI. Deux situations sont possibles selon que l'apprenant est en accord ou en désaccord avec cette interprétation. Dans tous les cas, la position de l'apprenant doit être supportée par des faits ou des principes appropriés. Pour une capacité donnée par exemple, si l'apprenant estime son niveau d'acquisition supérieur à celui inféré par le système, cette position doit se justifier, par exemple à travers: une perception personnelle, une trace de ses succès à un ensemble

d'exercices dont la résolution fait appel à la capacité en question, etc. Cette phase correspond au *raisonnement* en tant que tel.

À elle seule, la visualisation du modèle de l'apprenant permet nécessairement une réflexion (Bull et Nghiem, 2002). Toutefois, cette réflexion demeure implicite et ne peut qu'être tacitement présumée comme ayant lieu. En effet, l'interaction entre l'apprenant et le système est unidirectionnelle: le système fournit des informations à l'apprenant sans retour. On assume donc que la visualisation de ces informations déclenche chez l'apprenant un questionnement, sur la nature de ces informations et ensuite sur son état cognitif en regard de la substance de ces informations. Si aucun outil ne permet une extériorisation de ce questionnement, l'OLM n'est d'aucune utilité pour le système. Pour permettre cette extériorisation, on peut envisager un scénario de visualisation du modèle apprenant avec une phase d'édition du contenu du modèle. Dans cette phase, l'apprenant pourra fournir à son tour des informations au système. Deux cas de figures sont possibles: (1) l'apprenant modifie directement les informations de son modèle; (2) l'apprenant "négocie" avec le système la modification qu'il souhaite apporter au contenu de son modèle. La négociation confère une légitimité à l'édition du modèle d'un apprenant: ce dernier doit justifier chaque modification qu'il désire y apporter.

3.4.2.3. Intégration de l'approche OLM dans le diagnostic cognitif

De manière plus spécifique, on s'interroge sur la manière dont le DC et l'approche OLM reformulée en termes de pensée réflexive s'harmonisent. Pour cela, deux analyses sont proposées respectivement dans le cas de l'édition libre du modèle de l'apprenant et dans le cas de l'édition négociée du modèle de l'apprenant.

3.4.2.3.1. Édition libre du contenu du modèle de l'apprenant

Aperçu dans un STI

Lorsque l'apprenant apporte librement des modifications au contenu de son modèle, le DC est un autodiagnostic pur puisque le système n'intervient à aucun moment. Le système est un réceptacle des informations provenant de l'apprenant.

Dans la logique de l'apprentissage à travers un STI, l'autodiagnostic doit être interprété comme une initialisation ou un ajustement sporadique du modèle de l'apprenant. Le but est de maintenir une certaine continuité entre la vision de l'apprenant et l'interprétation de son état cognitif par le système. L'autodiagnostic peut aussi constituer la méthode de DC par défaut dans un STI et dans ce cas, le système sous-jacent

est insensible à ses interactions avec l'apprenant et seules les informations fournies par ce dernier constituent le DC.

Association conceptuelle entre l'édition libre et le type de DC

Conceptuellement, l'édition libre du modèle de l'apprenant s'associe naturellement aux deux types de diagnostic définis dans le cadre de base. Que l'état cognitif représenté dans le modèle soit relatif à la caractérisation des actions observables de l'apprenant (diagnostic comportemental), à son raisonnement (diagnostic comportemental) ou à ses connaissances (diagnostic épistémique), la possibilité de modifier les informations correspondantes est cohérente.

En effet, si le modèle de l'apprenant consigne une représentation de la solution de l'apprenant pour un exercice donné, il est tout à fait acceptable de lui donner la possibilité de modifier cette représentation. Cela permet d'obtenir une représentation plus précise et plus fiable de cette information.

Pour les mêmes raisons, si le modèle de l'apprenant contient une représentation de son raisonnement, on peut lui donner la possibilité de préciser sa ligne de raisonnement: indication des principes qu'il utilise ; indication des connaissances compilées qu'il applique (règles de production associées à son processus de résolution d'un exercice).

Enfin, si le modèle de l'apprenant reflète son niveau d'acquisition par rapport à des capacités du domaine de connaissance étudié, ce dernier peut aussi avoir la possibilité de modifier ce niveau.

Toutefois, pour assurer une certaine légitimité à l'autodiagnostic, il est préférable de permettre une édition libre du modèle de l'apprenant lorsque ce dernier est autonome et mature. En effet, le contenu de ce modèle doit être véridique et fiable en tout moment car il détermine le choix des stratégies pédagogiques dans un STI.

Association opérationnelle de l'édition libre avec le DC: la boucle diagnostic – réflexion – diagnostic

L'autodiagnostic qui émane de l'édition libre du modèle de l'apprenant entraîne une situation particulière: on peut le considérer comme une forme spéciale de DC ou alors uniquement comme un simple processus de réflexion. Selon le cas, son intégration dans le processus de DC proprement dit- le DC de l'état cognitif de l'apprenant, *par le système* - est différente. On a choisi de le considérer comme une phase de réflexion de l'apprenant à

part entière, étant donné que notre problématique concerne le DC effectué par le système (et non pas par l'apprenant lui-même) sous-jacent à un STI (Tableau III.5).

Tableau III.5. Intégration de l'édition libre du modèle de l'apprenant dans le DC

Vision pro-pédagogique du DC dans un STI	Instance dans le cas de la pensée réflexive avec l'approche OLM	Description
Observation	Session d'apprentissage	Le DC du système a lieu selon diverses approches. Il contribue à la construction du modèle de l'apprenant qui est inspecté par la suite par ce dernier.
Inférence		
Usage pédagogique du DC	Visualisation et Édition libre de son modèle construit au fil des sessions d'apprentissage	Inspection du contenu de son modèle par l'apprenant (Réflexion implicite)
Effets de l'exploitation pédagogique	Autodiagnostic de l'apprenant	Résultat de l'inspection de l'apprenant (peut nécessiter une justification de l'apprenant, préalable à une confirmation par le système)
Rétroaction des effets de la réflexion (Réflexion du côté du système)	Prochaine session d'apprentissage : Planification des actions pédagogiques dans la prochaine session d'apprentissage	Boucle DC-Réflexion-DC
	Ajustement des paramètres de l'algorithme de DC	
	Conditions d'ajustement	Nombre de caractéristiques de solution non identifiées Nombre maximal de Buts non identifiés lors de l'inférence du raisonnement de l'apprenant Différence maximale entre le niveau d'acquisition estimé par l'apprenant et le niveau d'acquisition inféré par le système pour chaque capacité ou connaissance représentée dans le modèle
	Ajustement : Humain dans la boucle	L'algorithme doit sauvegarder une trace des seuils d'ajustement qui ont été dépassés et des activités d'apprentissage correspondantes

Quelle est la rétroaction de l'autodiagnostic sur le système? L'édition libre d'un modèle de l'apprenant s'élabore soit au début d'une session d'apprentissage (initialisation du modèle de l'apprenant), soit entre deux sessions d'apprentissage (ajustement du

modèle de l'apprenant). Au demeurant, elle doit toujours précéder le DC du système proprement dit puisque l'autodiagnostic correspondant permet d'anticiper les décisions pédagogiques dans la session d'apprentissage suivante. L'autodiagnostic entraîne également un ajustement des hypothèses du système à propos de l'état cognitif de l'apprenant. Dans sa forme la plus simple, il consiste à remplacer les anciennes données par celles fournies par l'apprenant. Dans sa forme plus complexe, il consiste à ajuster les paramètres de l'algorithme de DC. Cela survient lorsque la différence entre les informations de l'apprenant et celles inférées par le système est significative. Étant donné la complexité de cette opération, il est fort probable que l'humain intervienne dans cette rétroaction. L'algorithme de DC doit explicitement implémenter cet ajustement ou alors sauvegarder une trace indiquant la nécessité d'un tel ajustement.

Exemple

Le Tableau III.2 (ci-dessus, en page 67) illustre trois instances de DC correspondant aux principaux types de DC proposés dans le CSDC de base. La première instance est associée à l'évaluation sans inférence: l'apprenant caractérise sa réponse en y indiquant les chiffres correspondant aux unités, aux dizaines et aux centaines respectivement. La deuxième instance est associée à l'interprétation reconstructive et dans ce cas l'apprenant exprime son raisonnement en indiquant la démarche utilisée pour obtenir le chiffre correspondant aux unités, aux dizaines et aux centaines dans sa réponse. La troisième instance est un diagnostic épistémique et dans ce cas, l'apprenant indique sur une échelle de 0 à 10 (0 étant le minimum et 10 étant une maîtrise totale) le niveau d'acquisition qu'il estime avoir atteint par rapport à chaque principe associé à la soustraction. La session d'apprentissage peut alors être déclenchée ou reprise sur la base de ces nouvelles informations.

3.4.2.3.2. Implication de l'apprenant dans la construction du contenu de son modèle: négociation

Aperçu dans un STI

La négociation du contenu du modèle de l'apprenant introduit une approche différente de la réflexion. En effet, elle impose à l'apprenant un questionnement concret sur la nature de chaque élément du modèle dont il désire négocier le contenu. Par exemple, considérons que le contenu du modèle d'un apprenant représente son niveau d'acquisition par rapport à un ensemble de capacités. S'il désire augmenter son niveau d'acquisition (par exemple une probabilité d'acquisition), le système lui permet de se justifier à travers

la résolution d'un exercice dont le niveau de difficulté reflète le niveau d'acquisition estimé. Une autre façon serait d'indiquer un historique d'apprentissage (dans le STI) qui justifierait une telle modification. Dans les deux cas, la justification impose une réflexion plus concrète à l'apprenant dans la mesure où il doit fournir des faits tangibles pour supporter sa requête de modification.

La négociation du contenu du modèle de l'apprenant s'intègre au DC dans un STI de manière similaire à l'édition libre de ce modèle.

Association conceptuelle de la négociation du modèle de l'apprenant à un type de DC

Au niveau conceptuel, la négociation du contenu du modèle de l'apprenant s'associe naturellement au diagnostic épistémique. Si l'on considère que le contenu du modèle de l'apprenant concerne ses connaissances ou ses capacités par rapport à un domaine à étudier, les exercices de performance qui permettent à l'apprenant de se justifier constituent une approche naturelle de négociation. Ces exercices favorisent *de facto* un examen des caractéristiques conceptuelles et contextuelles de la connaissance ou de la capacité qu'ils permettent d'évaluer: sa nature, les conditions de son usage, les principes associés à son application.

Le diagnostic comportemental porte sur le raisonnement ou la démarche non observable qui soutient les actions observables de l'apprenant. Une modification d'un tel diagnostic par l'apprenant ne nécessite pas de justification de sa part *a priori*. Le raisonnement ou la démarche cognitive peuvent se justifier dans un contexte argumentatif et non pas dans un contexte informatif, tel que c'est le cas dans l'édition du modèle de l'apprenant. En effet, le contexte argumentatif s'inscrit dans une circonstance où les objectifs pédagogiques s'orientent vers la cognition de niveau supérieur: l'apprenant argumente pour justifier le caractère raisonnable, logique et approprié de sa démarche.

3.4.3. Pensée réflexive et *Interactive Learner Modeling*

La modélisation interactive de l'apprenant implique l'apprenant dans son propre processus de DC. Il est vrai que l'édition négociée de son modèle peut être considérée comme une modélisation interactive de l'apprenant. La particularité de l'approche ILM est que les interactions se déroulent au cours de la résolution d'un exercice. Elle stimule donc une réflexion pendant l'action. Les dialogues tutoriels sont la tactique pédagogique utilisée pour supporter l'ILM. Toutefois, un dialogue tutoriel générique ne peut à lui seul garantir l'occurrence de la réflexion au sens de Dewey. En effet, la nature interactive de ces dialogues entraîne qu'il est effectivement possible de déclencher un questionnement

chez l'apprenant. Cela dit, il faut s'assurer que ce questionnement l'aide à comprendre un exercice, c'est-à-dire, à comprendre les fondements de sa réponse, en regard du problème qui y est associé. Une façon d'y arriver est d'utiliser des dialogues tutoriels dont la structure favorise l'occurrence des composantes de la pensée réflexive.

3.4.3.1. Interactive Learner Modeling dans la perspective de Dewey

Les dialogues tutoriels sont généralement utilisés pour stimuler une réflexion lors de la complétion d'une activité d'apprentissage. Pour favoriser l'occurrence de la pensée réflexive, on peut concevoir un dialogue tutoriel de manière que les interactions qui en découlent correspondent à une ou plusieurs composantes de ce phénomène. Autrement dit, le contenu de ces interactions doit pouvoir:

1. Présenter une situation problématique;
2. Amener l'apprenant à observer et à sélectionner les points ou les faits responsables de ce caractère problématique;
3. Amener l'apprenant à déterminer et à utiliser: des concepts, des lois, des principes *applicables* à la situation pour proposer sa réponse;
4. Amener éventuellement l'apprenant à évaluer sa réponse.

Pour favoriser de telles propriétés dans un dialogue tutoriel, on peut le concevoir de manière qu'il ait deux propriétés principales (Tableau III.6):

- ⇒ Sa structure correspond aux composantes de la pensée réflexive: chaque interaction doit avoir pour objectif de réaliser une de ces composantes;
- ⇒ Son contenu instancie chaque composante de la pensée réflexive en utilisant le contexte associé à l'exercice à résoudre.

La première propriété assure la cohérence du flot séquentiel d'interactions entre le système et l'apprenant: ce flot a un but déterminé - amener l'apprenant à répondre de manière fondée -; ce flot suit une logique découlant d'une théorie - la théorie de la pensée réflexive de Dewey.

La deuxième propriété assure la présence effective, voir explicite du phénomène de pensée réflexive chez l'apprenant. On entend par instancier une composante de la pensée réflexive le fait de concrétiser cette composante en utilisant les données de l'exercice que l'apprenant est en train de résoudre. Par exemple, si l'on considère la phase d'intellectualisation dans une situation de "soustraction avec retenue", le système doit – au moins - amener l'apprenant à identifier les colonnes des opérands où il sera nécessaire de faire une retenue.

Tableau III.6. Structure d'un dialogue tutoriel explicite pour la pensée réflexive

Structure du dialogue tutoriel	Recommandations générales pour le contenu du dialogue
[Question introductive]	Chaque question doit présupposer une assertion dans le contexte de l'exercice que l'apprenant est en train d'effectuer. Cette assertion a pour but de pousser l'apprenant à justifier sa position selon qu'il est d'accord ou pas.
[Favoriser l'observation chez l'apprenant]	Demander directement à l'apprenant d'identifier les faits importants, c'est-à-dire les informations qui favorisent le caractère problématique du problème
[Favoriser le raisonnement sur la base des Faits correspondant aux observations] et [Amener l'apprenant à évaluer la conclusion à laquelle il arrive]	Amener l'apprenant à utiliser un concept, une loi, un principe ou le sens commun approprié pour résoudre la situation problématique qui a été identifiée ⇒ Amener l'apprenant à utiliser une loi ou un concept ou un principe pour justifier la conclusion à laquelle il arrive ⇒ Amener l'apprenant à appliquer un principe pour vérifier sa conclusion (la confirmer ou arriver à une contradiction au cas où elle serait erronée)
[Favoriser un raisonnement interne poussé sur la base d'idées émanant du raisonnement sur les Faits]	✓ Amener l'apprenant à faire un raisonnement de niveau supérieur. Il s'agit d'un raisonnement qui ne se base pas nécessairement sur les données directes du problème. Plutôt, ce raisonnement consiste à utiliser les expériences passées et le contexte pour arriver à résoudre une question. Le raisonnement par analogie et le raisonnement à base de cas sont des exemples. Il demeure que leur déploiement doit être explicite (basé sur un modèle formel par exemple) afin de permettre à un STI de confirmer que l'apprenant l'a effectivement exécuté. ⇒ Ce type de raisonnement nécessite un contexte et un domaine riches en informations et en relations entre les informations

3.4.3.2. Intégration de l'approche ILM dans le diagnostic cognitif

Association conceptuelle de l'ILM à un type de DC

Le dialogue à structure réflexive a pour but d'assurer que toute réponse à laquelle parvient l'apprenant par le biais de ce dialogue est fondée. Nous considérons qu'il existe deux façons d'amener un apprenant à justifier sa réponse ou sa solution à un exercice.

La première façon consiste lui poser une suite de questions (suivant le flot des étapes de la pensée réflexive) dont les réponses nécessitent l'utilisation de connaissances et de capacités. Dans ce cas, ces réponses permettront de poser un DC épistémique. L'évaluation sans inférence peut aussi exploiter ce type de dialogue tutoriel dans la mesure où les réponses de l'apprenant révèlent des parties de sa solution.

La seconde façon consiste à lui poser une suite de questions (encore suivant le flot étapes de la pensée réflexive) dont les réponses correspondent aux buts associés à ses actions. Dans ce cas, ces réponses constituent une source d'information pour un DC comportemental sur le raisonnement de l'apprenant (et non sur la forme de sa solution).

Par conséquent, l'approche de réflexion par ILM s'associe au DC comportemental et au DC épistémique (Tableau III.7).

Association opérationnelle de l'ILM au DC: la boucle diagnostic-réflexion

Dans un STI, les dialogues tutoriels sont surtout utilisés pour raffiner un DC initialement posé par d'autres moyens. Chaque fois qu'une inférence à propos de l'état cognitif de l'apprenant est effectuée, un processus de réflexion est déclenché – si jugé nécessaire –. Au cours d'un tel dialogue, on pense que les interactions entre le système et l'apprenant provoquent les phénomènes suivants:

- (1) Les questions posées par le système obligent l'apprenant à justifier ses actions soit par des buts qu'il poursuit, soit en utilisant adéquatement des connaissances et des capacités. La structure des interactions assure en théorie la présence de la réflexion. De plus, l'hypothèse selon laquelle un état de remédiation ou d'amélioration de compréhension survient comme effet de bord de cette réflexion est raisonnable (Dimitrova, 2002);
- (2) Les réponses de l'apprenant permettent au système de confirmer son diagnostic initial et de le raffiner. Grâce à ces réponses, le système peut identifier les aspects de l'élément initialement diagnostiqué qui sont réellement problématiques. Cela peut être interprété comme une réflexion du côté du système.

On peut donc qualifier le flot d'interactions entre le système et l'apprenant comme une suite de boucles. Chaque boucle est amorcée par un DC initial sur l'état cognitif de l'apprenant, s'élabore à travers la stimulation de la pensée réflexive chez l'apprenant et se termine par une rétroaction des effets de cette stimulation sur le DC initial.

Tableau III.7. CSDC: intégration explicite de la stimulation de la pensée réflexive

Réflexion dans un STI	Type de modélisation de l'apprenant	Manifestation de la réflexion	Association conceptuelle au DC		Association Opérationnelle au DC
			Association	Condition	
Visualisation et Édition libre du Modèle Apprenant	Open Learner Modelling	Implicite	Béavioral	Aucune	<ul style="list-style-type: none"> ✓ Intégration - Avant une activité de résolution d'exercice Sporadiquement au cours d'une activité de résolution d'exercice ✓ Réflexion du système Confirmation/Infirmité de l'Autodiagnostic de l'apprenant et des diagnostics précédents Ajustement des paramètres de l'algorithme de propagation des inférences si nécessaire
			Épistémique	Autonomie et Maturité de l'apprenant par rapport à l'apprentissage (section 4)	
Visualisation et Négociation du contenu du Modèle Apprenant	Open Learner Modelling	Implicite	Béavioral	Contexte argumentatif, Apprentissage constructiviste (section 4)	<ul style="list-style-type: none"> ✓ Intégration - Avant une activité de performance ou de révision par rapport à un objectif d'apprentissage ✓ Réflexion du côté du système : Confirmation/Infirmité de l'Autodiagnostic et des diagnostics précédents Ajustement de l'algorithme de propagation des inférences
			Épistémique	Aucune	
Dialogues tutoriaux	Interactive Learner Modelling	Implicite/Explicite	Béavioral	Les justifications de l'apprenant doivent correspondre à des Buts ou à des concepts et principes du domaine	<ul style="list-style-type: none"> ✓ Intégration - À la fin du diagnostic cognitif du système ✓ Réflexion du côté du système : Raffinement des hypothèses
			Épistémique	Aucune	

3.5 Considération explicite du contexte pédagogique du DC

Nous venons de suggérer une intégration de la pensée réflexive dans le CSDC de base. En outre, il est nécessaire d'explicitier les implications d'une théorie de la cognition - et des théories d'apprentissage associées – sur les éléments du CSDC. En effet, la conception et la mise en œuvre des STI devraient s'ancrer dans un contexte pédagogique *basé sur des principes* dont ceux qui sont issus des paradigmes de la cognition (Reigeluth, 1999).

Dans la suite de ce chapitre nous proposons une interprétation de l'influence des paradigmes de cognition - et des théories d'apprentissage associées- sur le DC. Cette influence est explicitée dans le CSDC à travers des règles de cohérence sur les instances de DC.

3.5.1. DC et paradigme de cognition: que doit-on inférer? Sur quoi orienter la pensée réflexive?

Dans la psychologie de l'apprentissage, trois principaux paradigmes ont marqué la perception de la nature de la cognition: l'empirisme avec les théories behavioristes d'apprentissage, le rationalisme avec les théories cognitivistes de l'apprentissage et le pragmatisme socio-historique avec les théories socioconstructivistes de l'apprentissage. Sur la base des principes de ces paradigmes, l'on peut circonscrire la nature de ce qui doit être diagnostiqué, de ce sur quoi devrait porter le processus de pensée réflexive mis en branle suite au DC (Tableau III.8, Tableau III.9).

3.5.1.1. DC et Empirisme

Dans la perspective d'une cognition empirique et d'un apprentissage behavioriste, le DC consiste à établir les comportements désirables (observables) déjà acquis par l'apprenant et ceux qui restent à acquérir. Ces comportements sont généralement présentés sous forme d'objectifs d'apprentissage dans un domaine de connaissances précis. Pour cette raison, le *diagnostic épistémique* représente le type de DC qui est le plus cohérent dans un tel contexte pédagogique.

Tableau III.8. CSDC: Paradigmes de cognition et inférence du DC

Étapes du DC	Paradigmes de cognition		
	Type d'influence	Nature de l'influence	
		Paradigme	Influence
Inférence	Quels types de connaissances ou de capacités sont pertinents à inférer par le DC ?	Empirisme	<ul style="list-style-type: none"> ✓ Diagnostic comportemental par évaluation sans inférence ✓ Diagnostic comportemental par Interprétation reconstructive des actions observables en termes de buts ✓ Diagnostic épistémique par interprétation reconstructive des actions observables en termes de connaissances compilées (ancrées dans le contexte réel), i.e. spécifiquement l'approche <i>Model Tracing</i> ✓ Diagnostic épistémique par évaluation sans inférence
		Rationalisme	<ul style="list-style-type: none"> ✓ Interprétation Reconstructive des actions observables ✓ Diagnostic Épistémique
		Pragmatisme et Socio-Historisme	<ul style="list-style-type: none"> ✓ Interprétation reconstructive des actions observables en termes de raisonnement

Tableau III.9. CSDC: Paradigmes de cognition et pensée réflexive

Processus de diagnostic	Paradigmes de cognition		
	Type d'influence	Nature de l'influence	
Pensée réflexive	Quelles approches de réflexion stimulent les éléments cognitifs clés dans cette perspective de la cognition ?	Empirisme	<ul style="list-style-type: none"> ✓ Apprenant <ul style="list-style-type: none"> ▪ Édition libre ou négociée du modèle de l'apprenant ✓ Système <ul style="list-style-type: none"> ▪ Aucune influence particulière
		Rationalisme	<ul style="list-style-type: none"> ✓ Apprenant <ul style="list-style-type: none"> ▪ Tous les types de réflexions ✓ Système <ul style="list-style-type: none"> ▪ Raffinement d'hypothèses
		Pragmatisme et Socio-Historisme	<ul style="list-style-type: none"> ✓ Réflexion pendant l'action <ul style="list-style-type: none"> ▪ Édition libre ou négociée du modèle de l'apprenant pour l'autoévaluation ✓ Système <ul style="list-style-type: none"> ▪ Amélioration des outils d'apprentissage constructiviste (<i>inquiry learning</i>), ▪ Amélioration des outils métacognitifs

La perspective behavioriste ignore les états mentaux. La pensée réflexive étant un phénomène mental par nature, il apparaît contradictoire de la considérer dans un contexte behavioriste d'apprentissage. Toutefois, on peut quand même recourir à l'édition (libre ou négociée) du contenu du modèle de l'apprenant. En effet, l'édition impose à l'apprenant d'effectuer un processus d'*association de faits tangibles* (événements d'apprentissage, performances) à une *conséquence* (changement du contenu du modèle). *A priori*, elle ne nécessite pas l'intervention directe de processus mentaux.

En ce qui concerne la réflexion du côté du système, elle ne revêt pas de caractéristique particulière dans cette instance de contexte pédagogique.

3.5.1.2. DC et Rationalisme

Dans une perspective rationaliste et cognitiviste, le DC consiste surtout à établir la compréhension conceptuelle chez l'apprenant et l'utilisation judicieuse des principes. À un niveau plus abstrait, le DC s'intéresse aussi au transfert de cette compréhension dans des contextes différents, à la possession et à l'utilisation judicieuse de capacités métacognitives. Dans la compréhension conceptuelle, un DC examine les représentations mentales de l'apprenant telles qu'interprétées par le système, afin d'en déterminer la justesse; il s'agit donc d'un DC épistémique. L'utilisation judicieuse d'un concept ou d'un principe correspond à un raisonnement basé sur une représentation des processus mentaux appropriés; il s'agit donc d'un DC comportemental. Ainsi le DC comportemental et le DC épistémique sont cohérents avec un contexte pédagogique rationaliste/cognitiviste, selon qu'on désire diagnostiquer la compréhension conceptuelle ou l'utilisation des concepts et des principes.

En ce qui concerne la pensée réflexive, elle consiste pour l'apprenant à raisonner à propos de la nature d'un concept, d'un principe et d'une capacité métacognitive ainsi qu'à propos de leur utilisation judicieuse. Les approches OLM et ILM de stimulation de la pensée réflexive sont donc celles qui s'appliquent à ce contexte pédagogique. Dans l'approche OLM: la visualisation et l'édition du contenu du modèle de l'apprenant lui permettent de raisonner sur la nature d'un élément de connaissance ou d'un objectif d'apprentissage; la visualisation et la négociation, correspondent plus à un raisonnement à propos de l'utilisation d'un concept ou d'un principe lors de la résolution d'un exercice. Quant à l'approche ILM, les dialogues tutoriels amènent aussi l'apprenant à raisonner à propos de l'utilisation d'un concept ou d'un principe lors de la résolution d'un exercice.

En ce qui concerne la réflexion du côté du système, elle consiste à ajuster le DC qui a été effectué, en tenant compte des conséquences de la stimulation de la pensée réflexive chez l'apprenant.

3.5.1.3. DC et la perspective Pragmatiste et socio-historique de la Cognition

Dans une perspective pragmatiste et socio-historique de la cognition, le DC consiste à diagnostiquer les capacités propres à l'approche empirique de l'apprentissage (*inquiry skills*): hypothèse, expérimentation, synthèse, collaboration. Il consiste aussi à établir, les capacités métacognitives: planification, réflexion au sens métacognitif, autoévaluation. Dans un tel contexte pédagogique, l'accent est mis plus sur les capacités en elles-mêmes et moins sur le contenu auquel elles s'appliquent. Selon ce paradigme, la démarche d'apprentissage est souvent plus importante que le contenu ou le résultat relatifs à cet apprentissage. Étant donné que certains sous-types de DC comportemental s'intéressent à la démarche cognitive dont découlent les actions observables d'un apprenant, il constitue l'approche de diagnostic la mieux indiquée dans ce contexte pédagogique.

La pensée réflexive au sens métacognitif fait partie intégrante des objectifs d'apprentissage dans un contexte basé sur le pragmatisme et les influences socio-historiques. De plus, la pensée réflexive telle qu'intégrée dans le DC correspond naturellement au processus d'apprentissage selon une approche empirique. En effet, si l'on s'en tient à la définition de Dewey, on constate qu'elle est analogue à ce type d'apprentissage: examen des données d'un problème (sélection des informations), raisonnement sur les données pour établir une conclusion par rapport à un problème de départ ou encore pour corroborer une hypothèse de départ (analyse), évaluation de la conclusion (expérimentation). Dans ce sens, un contexte pragmatiste d'apprentissage s'associe naturellement à une réflexion « pendant l'action » (approche ILM).

En quoi consiste la réflexion du côté du système? L'évaluation du niveau de connaissances de l'apprenant a moins d'importance dans ce contexte pédagogique. La pédagogie met plutôt l'accent sur les outils qui favorisent le processus d'apprentissage. Ces outils correspondent à une gamme assez variée mais deux principaux types d'action ont été identifiés par rapport à ces outils: (1) l'ajout ou l'ajustement des outils d'enquête; (2) l'ajout ou l'ajustement d'outils associés aux activités métacognitives. L'ajustement des outils d'enquête intervient lorsque le résultat d'un DC fait état d'un défaut au niveau du processus d'enquête. Par exemple, le système devrait fournir à l'apprenant un instrument de collecte de données ou d'analyse à l'interface plus compréhensible et aux

fonctionnalités plus explicites, lorsque ce dernier trahit des lacunes par rapport à cette étape du processus d'enquête. Les outils métacognitifs doivent être mis en valeur lorsqu'un diagnostic est posé par rapport à une capacité métacognitive. Par exemple, le système devrait ajouter un outil de collaboration (ou mettre sa présence en valeur) lorsqu'un apprenant a besoin d'aide dans un contexte collaboratif.

3.5.2. Diagnostic cognitif et théories d'apprentissage: que doit-on observer?

Étant donné un paradigme de cognition, la théorie d'apprentissage regroupe un ensemble de principes qui expliquent les processus mentaux et les comportements observables qui mènent à l'acquisition de cette cognition. La théorie d'apprentissage favorise la réalisation d'un objectif d'apprentissage en guidant la conception des activités d'apprentissage associées. Pour cette raison, lorsque les caractéristiques d'une activité d'apprentissage sont élaborées sur la base d'une théorie d'apprentissage, le DC qui se déroule lors de cette activité doit tenir compte de cette théorie. La théorie d'apprentissage indique quels processus mentaux et quelles actions observables de l'apprenant sont pertinents à relever lors de la phase d'acquisition d'indices du DC (Tableau III.10).

Trois grandes classes de théories d'apprentissage peuvent être associées aux trois paradigmes de cognition qui ont été considérés: les théories d'apprentissage behavioristes, les théories d'apprentissage cognitivistes et les théories d'apprentissage socioconstructivistes.

3.5.2.1. Acquisition d'indices dans l'apprentissage behavioriste

Les théories d'apprentissage behavioristes considèrent que la connaissance doit être *communiquée ou transférée* vers l'apprenant. L'apprentissage dans cette optique consiste à présenter à l'apprenant une association entre des entités ou des événements. Au sein de cette association, les entités dites « stimuli » correspondent aux conditions ou aux indices dont la présence implique celle des entités dites « réponses ». L'objectif d'apprentissage est réalisé lorsque l'apprenant produit la « réponse » attendue pour chaque ensemble de stimuli présenté.

Lors de la phase d'acquisition d'indices du DC, les seules observations pertinentes sont donc les réponses de l'apprenant.

3.5.2.2. Acquisition d'indices dans l'apprentissage cognitiviste

Aussi appelées théories de traitement de l'information, les théories d'apprentissage cognitivistes sont orientées sur le processus de pensée qui sous-tend le comportement

observable. Les réponses de l'apprenant sont donc également observées, mais elles ne sont pas une finalité. En effet, elles constituent un indicateur des processus mentaux (traitement de l'information) et des représentations mentales qui les ont produites.

Tableau III.10. CSDC: Paradigmes de cognition et acquisition d'indice dans le DC

Étapes du DC	Théorie d'apprentissage		Théories d'instruction et du design de l'instruction		
	Type d'influence	Influence		Type de l'influence	Influence
Observation	Quels phénomènes observables ou non observables sont pertinents à observer afin d'établir des inférences ?	Courant théorique	Influence	Quand déclencher le diagnostic dans un STI ?	À l'instant indiqué par le concepteur pédagogique. Note : Cet instant doit être compris dans le cours d'une <i>activité d'apprentissage</i>
		Béavioriste	✓ Réponse de l'apprenant		
		Cognitiviste	✓ Protocole de "penser à voix haute" (humain dans la boucle) ✓ Réponses structurées ou explicitées de l'apprenant		
		Socio-constructiviste	✓ Actions et analyse en contexte concret ✓ Interaction avec pairs ✓ Actions influencées par le bagage historico social		

L'apprentissage passe donc par une représentation mentale de l'information. Cette représentation permet de bien comprendre l'information, de l'organiser dans des structures cognitives telles que la mémoire, de l'associer ou de l'intégrer à des informations déjà existantes. L'apprentissage passe aussi par des processus de traitement des représentations mentales qui permettent: de se rappeler (extraire une représentation de la mémoire), de raisonner sur ces représentations, etc.

Dans un tel contexte pédagogique, lorsqu'un humain intervient effectivement dans la boucle du diagnostic, l'exécution d'un protocole de « penser à voix haute » pourrait supporter la phase d'observation du DC. Sinon, on peut aussi observer la réponse de l'apprenant à divers niveaux de granularité, pour recueillir le plus d'informations

possible. En effet, cette réponse doit être suffisamment structurée ou suffisamment raffinée pour pouvoir être associée à des représentations mentales ou à des processus mentaux, étant donné que ces derniers ne sont pas directement observables.

3.5.2.3. Acquisition d'indices dans l'apprentissage socioconstructiviste

Les théories socioconstructivistes privilégient une approche de l'apprentissage selon laquelle la cognition *émerge*:

- (1) à travers une expérience concrète et personnelle dans une situation spécifique et réelle;
- (2) à travers divers types d'interactions avec des pairs (collaboration, coopération)
- (3) selon un contexte social
- (4) selon un bagage historique.

L'apprentissage est basé sur une interprétation individuelle d'une expérience concrète. L'activité est concentrée du côté de l'apprenant et dans ce sens, la pédagogie est plus observationnelle et stimulatrice qu'interventionniste. L'apprenant peut être appelé à interagir avec des pairs pour partager ses connaissances, confronter son interprétation d'une expérience concrète, co-construire la solution à un problème. Étant donné ce centrage autour de l'apprenant, les retombées de l'apprentissage seront différentes pour chaque apprenant. Dans un contexte pédagogique socioconstructiviste, il faut donc observer:

- (1) Les actions de l'apprenant dans la situation concrète d'apprentissage à laquelle il est soumis: quels éléments retiennent son attention? Sur quoi se base son interprétation étant donné ces éléments?
- (2) Les interactions de l'apprenant avec ses pairs ;
- (3) Les actions et les interactions directement influencées par un contexte socio-historique.

3.6 Exploitation du CSDC: CD-SPECIES ou un outil de support à la conception du DC

Un CSDC demeure un instrument introductif et indicatif pour le concepteur pédagogique. Nous pensons que le cadre de spécification proposé dans cette thèse devrait être exploité à travers un outil informatique pour trois raisons principales: la flexibilité d'utilisation du CSDC, l'évolution conceptuelle à court et moyen termes du contenu du CSDC, la réutilisation des spécifications, la génération d'une architecture de base (physique ou

conceptuelle) pour un module de DC dans un STI, selon les caractéristiques d'une spécification donnée.

Le but de cette section est de décrire l'architecture de CD-SPECIES (pour *Cognitive Diagnosis Specifications Tool*), un outil qu'on peut concevoir comme une coquille pour le CSDC (Tchetagni, Nkambou, et Bourdeau, 2005b). Munie d'un certain nombre de fonctionnalités, cette coquille permettrait une exploitation plus représentative des objectifs de flexibilité, d'évolution conceptuelle et de réutilisation. Dans les prochaines lignes, nous précisons la pertinence de CD-SPECIES. Ensuite, nous proposons une perspective du processus général de conception d'une composante de diagnostic cognitif pour un STI. C'est à travers ce processus que les interactions qui représentent la flexibilité, favorisent l'évolution conceptuelle et la réutilisation, sont explicitement mises en évidence. Enfin, les composantes de CD-SPECIES sont présentées ainsi que leurs interrelations.

3.6.1. Pertinence de CD-SPECIES

Cette section développe les deux principaux points qui ont été avancés pour justifier la pertinence d'un outil comme CD-SPECIES: la flexibilité d'usage du cadre de spécification de DC; l'évolution conceptuelle du contenu et la réutilisation des spécifications.

3.6.1.1. Flexibilité d'usage: spécification par focus et sur demande

De notre point de vue, la flexibilité d'usage est désirable par ce qu'elle permettrait un certain nombre de facilités lors de la définition d'un module de DC pour un STI. CD-SPECIES peut offrir une interface permettant au concepteur pédagogique de spécifier une composante de DC à la fois, tout en tenant compte des valeurs associées aux composantes qui ont déjà été spécifiées. Cela devrait permettre une focalisation de l'effort de spécification sur chaque élément du DC.

On considère qu'avec CD-SPECIES, la définition d'une composante de diagnostic se fait: soit en instanciant directement les caractéristiques du DC, soit en interrogeant le système sur la nature de cette composante, de ses différentes valeurs possibles et des conditions associées au choix de chacune de ces valeurs. Par exemple, le concepteur pédagogique peut se focaliser tour à tour sur la spécification du type de DC qu'il désire mettre en œuvre, ou encore sur la spécification de l'étape correspondante d'acquisition d'indices. Pour chaque élément spécifié, le concepteur peut s'enquérir de la nature de cet élément en interrogeant le système.

3.6.1.2. Évolution conceptuelle: développement progressif du CSDC

Par ailleurs, le concepteur doit pouvoir modifier les données qui lui sont fournies par CD-SPECIES dans un sens plus fidèle à ses vues personnelles du DC ou à son contexte particulier. Par exemple, dans le CSDC, la majorité des données à retenir lors de la phase d'acquisition d'indices du DC concernent la solution fournie par l'apprenant. Certaines instances de DC dans un STI considèrent la quantité d'explications que l'apprenant a sollicitée avant de répondre à une question ((Conati et al., 1997) dans "Andes"). Un autre exemple: sur le chapitre du contexte pédagogique, certains travaux indiquent que les préférences et les données anthropologiques de l'apprenant influencent l'éventail de paramètres qui peuvent être considérés comme des intrants du DC (Shute, Lajoie, et Gluck, 2000).

La possibilité de modifier les recommandations du CSDC permet au concepteur pédagogique d'élaborer des spécifications plus détaillées et plus spécifiques sur la toile de fond des patrons classiques offerts par ce cadre. En outre dans certaines conditions, certaines modifications peuvent être directement répertoriées comme ajouts d'information dans le cadre de spécification initial, favorisant ainsi une évolution conceptuelle de son contenu. Cela peut arriver lorsque le concepteur pédagogique qui utilise le système a des droits de modification, sur la base de son expérience par exemple. Il va de soi que cette thèse jette seulement les fondations d'un cadre de spécification pour le DC. Ce cadre ne saurait donc être exhaustif et sa présentation à travers un outil informatique interactif en favoriserait l'évolution.

3.6.1.3. Réutilisation des spécifications de DC

Au-delà de favoriser la communication des attentes d'un concepteur pédagogique par rapport à une composante de DC, à un programmeur, une spécification peut être ultérieurement utilisée comme:

- point de repère pour l'évaluation d'un module de DC dans un STI (correspond-il à la spécification?);
- point de départ pour une spécification de processus de DC aux propriétés similaires (même contexte pédagogique par exemple);
- point de référence dans la comparaison de deux modules de DC.

Les mécanismes de sauvegarde et d'indexation des spécifications de DC pourraient en favoriser la réutilisation. CD-SPECIES en tant qu'outil informatique pourrait intégrer ces mécanismes.

3.6.2. Survol de la conception du DC avec CD-SPECIES

Le but de cette section est d'illustrer les usages potentiels d'un outil tel que CD-SPECIES, lors de la conception du DC dans un STI. Dans un premier temps, l'on se propose de positionner l'utilité de CD-SPECIES dans le cadre de la conception pédagogique. Les fonctionnalités de cet outil sont abordées par la suite.

3.6.2.1. Usage de CD-SPECIES dans la conception pédagogique

Selon Morrison, Ross et Kemp (2004), le processus de conception pédagogique représente un développement systématique de spécifications pédagogiques, en utilisant des théories d'apprentissage et d'enseignement pour assurer la qualité de l'enseignement. Dans le cas particulier des STI, le design de l'enseignement s'amorce par l'analyse des objectifs d'apprentissage et évolue vers le développement des artéfacts qui représenteront le matériel didactique favorisant l'atteinte de ces objectifs. Ce matériel didactique inclut toujours un certain nombre d'activités d'apprentissage, spécifiques à l'atteinte de chaque objectif d'apprentissage.

Le DC est un moyen pédagogique permettant l'interprétation des actions de l'apprenant pendant qu'il exécute une activité d'apprentissage. Cette interprétation est par la suite exploitée à des fins pédagogiques. Dans ce sens, un outil comme CD-SPECIES doit être mis à contribution dans la phase de définition des activités d'apprentissage, lors de la conception pédagogique globale d'un STI. Cependant, tout processus de diagnostic n'est possible que lorsqu'il s'appuie sur des phénomènes observables. Par conséquent, CD-SPECIES ne peut être utilisé que lors de la définition d'une activité d'apprentissage où il est possible d'observer des faits tangibles reliés au comportement de l'apprenant.

Par exemple, ID Expert (IDE) est un outil auteur basé sur la théorie de conception pédagogique appelée "Instructional Transaction Theory" (Merrill et ID2-Research-Group, 1998). Dans IDE, la pédagogie est conçue comme une série de "transactions" entre l'apprenant et le STI. Une transaction élabore plusieurs types d'interactions pédagogiques: exploration guidée d'un domaine de connaissances, pratique assistée à travers des exercices, évaluation formelle à travers des tests, etc. Dans cette circonstance, CD-SPECIES peut être utilisé avant la définition de chaque transaction pour définir le processus de DC qui doit être appliqué lors de l'exécution de cette transaction par l'apprenant. Dans cette optique, le DC peut être vu comme une interaction intégrée dans une transaction ou comme une transaction à part entière, automatiquement déclenchée à la suite de celle où l'apprenant a été observé.

3.6.2.2. La conception du DC avec CD-SPECIES

Quelle méthodologie de spécification d'une composante de DC peut-on envisager à travers CD-SPECIES? On considère qu'un concepteur pédagogique *utilise* CD-SPECIES pour produire une spécification de DC, selon le contexte pédagogique du STI auquel il est destiné. CD-SPECIES doit soutenir le concepteur pédagogique afin de ménager deux propriétés désirables au niveau de la conception du DC: l'articulation explicite des composantes de DC dans un STI et la cohérence entre les composantes choisies (Tableau III.11).

Tableau III.11. Utilisation de CD-SPECIES

	Permettre la création d'une spécification	Répondre aux requêtes du concepteur pédagogique	Répondre aux requêtes du programmeur
Fonctionnalités de l'interface de CD-SPECIES	Permettre au concepteur pédagogique de : <ol style="list-style-type: none"> 1. Sélectionner un paradigme de cognition 2. Sélectionner une classe de théories d'apprentissage 3. Sélectionner la composante du DC qu'il désire spécifier (caractéristique conceptuelle ou opérationnelle) 4. Ajouter des notes et détails personnels pour chaque élément sélectionné 5. Ajouter un nouvel élément dans la structure du cadre de spécification 6. Ajouter une nouvelle donnée dans le contenu du cadre de spécification 	Permettre au concepteur de : <ul style="list-style-type: none"> ▪ Demander de l'information sur un élément du DC à la demande du concepteur ▪ Demander une vérification de la cohérence des composantes instanciées du DC pour une spécification ▪ Demander une sauvegarde une spécification dans un fichier formaté sur la machine locale de CD-SPECIES ▪ Demander la visualisation d'une spécification 	<ul style="list-style-type: none"> ▪ Permettre au programmeur de ▪ Demander la génération d'un pseudo algorithme générique de DC, associé à une spécification précise ▪ Générer une architecture conceptuelle du module de DC associée à une spécification précise ▪ Demander la visualisation d'une spécification

En ce qui concerne l'explicitation, CD-SPECIES devrait permettre à un concepteur pédagogique de "prendre conscience" de la nature des composantes du DC lorsque mis en œuvre spécifiquement dans un STI. Ceci est important dans la mesure où les éléments de l'expertise pédagogique sont implicites pour l'expert pédagogique la majorité du temps. Lorsqu'il faut communiquer cette expertise dans une structure compréhensible pour un programmeur – c'est-à-dire sous une forme quasi-opérationnelle – l'articulation de ces connaissances peut être difficile, voire impossible. CD-SPECIES doit permettre au concepteur pédagogique de formuler clairement ses intentions à propos du déroulement du DC dans un STI.

Sur le chapitre de la cohérence, CD-SPECIES doit recommander à un concepteur pédagogique comment les composantes du DC peuvent être instanciées de manière conforme. Ces recommandations assurent que les valeurs choisies s'harmonisent avec le contexte pédagogique – lorsqu'il a été préalablement défini par le concepteur. Par exemple, il faudrait indiquer à ce dernier que si une approche socioconstructiviste de l'apprentissage est adoptée, l'évaluation sans inférence est un type de diagnostic inapproprié. Les recommandations doivent aussi assurer que les valeurs choisies s'harmonisent avec leur nature intrinsèque ou avec les valeurs d'autres éléments qui ont déjà été définis. Par exemple, si le type de DC qui a été choisi est épistémique, il va de soi qu'au moins un des artefacts implémentés dans le futur STI devrait inclure une représentation explicite des connaissances du domaine appris.

La seule façon d'assurer la présence de ces interactions entre un concepteur pédagogique et un cadre de spécification du DC est d'élaborer une interface appropriée dans CD-SPECIES. Les fonctions de cette interface devraient: (1) permettre au concepteur pédagogique d'entrer l'information relative à une spécification et de la sauvegarder; (2) permettre au concepteur pédagogique de demander de l'aide à CD-SPECIES; (3) permettre à CD-SPECIES de fournir de l'information à un programmeur.

3.6.2.3. Création d'une spécification

La Figure III.3 illustre les principales opérations qui sont proposées à l'interface de construction d'une spécification dans CD-SPECIES. Il s'agit principalement de:

- (1) La définition d'un contexte pédagogique par le concepteur pédagogique. Il faut noter que le concepteur pédagogique ne devrait pas être obligé de spécifier un contexte pédagogique. Si tel est le cas, CD-SPECIES devrait vérifier la cohérence entre les valeurs proposées pour les composantes du DC. Si par contre un contexte pédagogique est indiqué, CD-SPECIES doit en plus s'assurer de la cohérence des valeurs des composantes du DC avec le contexte pédagogique;
- (2) La définition des valeurs associées aux composantes d'une instance de DC, nommément son type (composante conceptuelle) et les phases d'acquisition d'indices et d'inférence (composante opérationnelle). Il s'agit donc de fournir au concepteur une interface lui permettant d'indiquer ces valeurs;
- (3) L'indexation d'une spécification où le concepteur pédagogique peut indiquer les caractéristiques de la spécification actuellement construite. Les caractéristiques dépendront de l'implémentation de CD-SPECIES. Par exemple, on peut décider

d'indexer les spécifications uniquement en fonction de leur contexte pédagogique ou encore en fonction de leur contexte pédagogique et du type de diagnostic qu'elles définissent;

- (4) La consultation d'une spécification particulière se justifie dans la perspective de la réutilisation d'une spécification. Dans ce cas, il s'agit de fournir au concepteur pédagogique une interface pour l'indication des caractéristiques associées aux spécifications ciblées pour une réutilisation. Encore une fois, la nature des caractéristiques à indiquer dépendent de l'implémentation de CD-SPECIES en ce qui concerne le choix d'indexation des spécifications ;
- (5) La sauvegarde d'une spécification qui permettrait au concepteur pédagogique d'interrompre la définition d'une spécification ou de la sauvegarder définitivement afin qu'elle puisse être réutilisée ultérieurement. Lors de cette sauvegarde, CD-SPECIES devrait générer un fichier formaté et structuré, de manière à pouvoir être édité ultérieurement.

Lors de la création d'une spécification, le concepteur doit également pouvoir y ajouter des notes ou des détails propres qui ne peuvent pas être formulés à travers les éléments offerts par le CSDC à travers CD-SPECIES. Le concepteur doit disposer de la possibilité d'ajouter un élément à la structure du cadre, par exemple une nouvelle composante conceptuelle/opérationnelle du DC ou une nouvelle composante pour le contexte pédagogique. Dans le même sens, le concepteur doit aussi pouvoir ajouter de l'information aux instances des composantes du DC. Par exemple, un nouveau type de DC, une nouvelle approche d'acquisition d'indices, un nouveau type d'inférence, de nouvelles recommandations relatives à l'algorithme de DC. Cela signifie que CD-SPECIES doit offrir des outils d'édition, qu'il s'agisse d'édition libre pour annoter une spécification ou d'édition structurée pour effectuer des ajouts au contenu du CSDC.

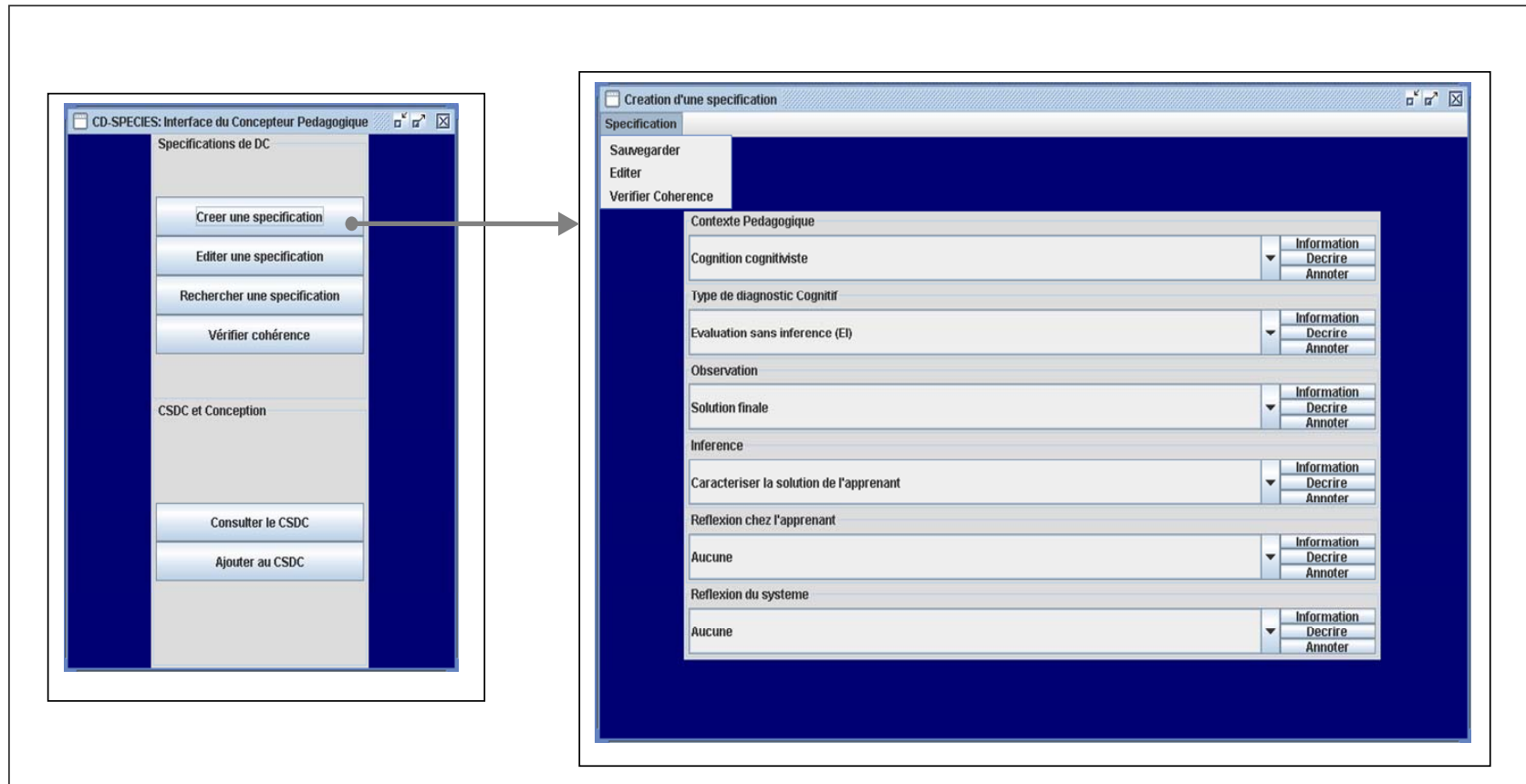


Figure III.3. CD-SPECIES: Interface de spécification d'une instance de DC

3.6.2.4. CD-SPECIES: Assistance à la conception pédagogique du DC

À travers les fonctionnalités précédemment évoquées, CD-SPECIES doit être perçu comme un conteneur structuré. En effet, les interfaces qui permettent au concepteur pédagogique d'effectuer chacune des opérations lui permettent aussi d'articuler ses connaissances relatives au DC. Au-delà de ce rôle de canevas, CD-SPECIES doit aussi être une source d'information et un outil d'aide au maintien de la cohérence d'une spécification.

Pour cela, on peut doter cet outil d'une interface d'assistance à la spécification d'un processus de DC. Les opérations suivantes constituent l'essentiel des fonctions d'assistance modélisées dans CD-SPECIES:

- (1) La sollicitation d'informations relatives à la nature d'une composante du DC. Par exemple, le concepteur pédagogique devrait pouvoir s'enquérir de la signification de "type de diagnostic", "contexte pédagogique", "acquisition d'indices", "inférence", "pensée réflexive", etc.
- (2) La sollicitation d'informations relatives à une instance possible de composante du DC. Pour la propriété "type de diagnostic" par exemple, le concepteur devrait pouvoir demander au système ce qui est entendu par "DC épistémique";
- (3) La vérification de la cohérence. Cette opération permettrait à CD-SPECIES de vérifier la cohérence de la spécification (Figure III.3, ci-dessus en page 101). La vérification de la cohérence doit s'élaborer à deux niveaux. Au premier niveau, cette vérification doit établir que les valeurs de chaque composante du DC *sont cohérentes* avec le contexte.

Dans ce cas, il s'agit du paradigme de cognition qui est considéré lors de la conception du STI. Au deuxième niveau, la vérification doit établir une cohérence interne entre les différentes valeurs des composantes du DC spécifié. Par exemple, il serait incohérent d'essayer d'arrimer un DC comportemental par interprétation reconstructive de la solution de l'apprenant avec une édition du modèle de l'apprenant comme exploitation pédagogique de ce DC.

Le concepteur pédagogique peut effectuer une requête d'information après qu'un contexte pédagogique a été défini. Dans ce cas, CD-SPECIES doit être assez flexible pour fournir l'information demandée soit de manière générale, soit de manière spécifique au contexte pédagogique. Cela dépend du choix indiqué par le concepteur pédagogique. Par exemple, ce dernier peut demander des informations à propos de la pensée réflexive après

avoir indiqué que sa spécification de DC correspond à un contexte qui adopte une perspective pragmatiste et socio-historique de la cognition.

Selon le choix du concepteur (demande d'informations générales ou demande d'informations spécifiques au contexte pédagogique) CD-SPECIES doit pouvoir fournir: soit une définition générale de la pensée réflexive, soit une définition générale accompagnée d'une explication spécifique de l'objet de ce phénomène dans une perspective pragmatiste de la cognition.

3.6.2.5. CD-SPECIES: Assistance à la programmation du module de DC

CD-SPECIES peut aussi être mis à contribution au début de la phase de programmation du module de DC d'un STI. Il ne faut pas oublier que l'un des principaux buts d'une spécification est de favoriser la communication entre le concepteur pédagogique et le programmeur de STI. Sachant que CD-SPECIES gère un ensemble de spécifications sauvegardées, celles-ci peuvent être soumises à certains traitements pour optimiser leur exploitation par le programmeur de STI. Une interface de consultation pour le programmeur a été modélisée dans l'architecture de CD-SPECIES. Cette interface permettrait la réalisation des opérations suivantes (Figure III.4):

- (1) La consultation d'une spécification (avec simple indication du nom du fichier correspondant)
- (2) La sollicitation d'informations à propos des recommandations sur la mise en oeuvre d'une spécification et sur les mécanismes associés de représentation des connaissances;
- (3) La production d'une version semi-computationnelle des algorithmes qui implémenteraient la spécification qui est actuellement consultée. Par "semi-computationnelle", l'on entend pseudo algorithme. Il s'agit d'une définition abstraite et indépendante du langage de programmation, des principales opérations qui matérialisent l'algorithme de DC correspondant à une spécification;
- (4) La production d'une architecture conceptuelle d'un module de DC dans un STI. Dans son expression la plus simple, cette architecture est entrevue comme une hiérarchie de répertoires ou de fichiers. Chaque répertoire ou fichier représenterait une fonction particulière ou une structure de données nécessaires à l'implémentation du DC correspondant à la spécification en main. L'intérêt de cette architecture conceptuelle est

qu'elle servirait de point de départ concret à un programmeur de STI, pour le module de DC.

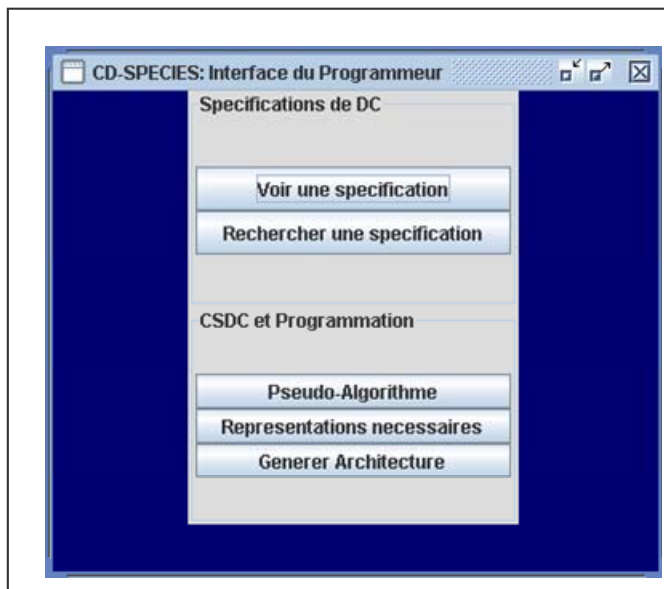


Figure III.4. CD-SPECIES: Interface du programmeur de STI

3.6.3. Architecture de CD-SPECIES

La réalisation des opérations précédemment proposées au sein de l'interface de CD-SPECIES doit pouvoir être assurée. Cette section présente les composantes et la dynamique interne de l'architecture de CD-SPECIES qui assurerait l'effectivité de ces opérations.

3.6.3.1. Les composantes de CD-SPECIES

Les opérations susmentionnées doivent se réaliser à partir d'une base de connaissances contenant trois composantes principales: (1) un ensemble regroupant les connaissances sur le DC sous forme déclarative, (2) un ensemble représentant les associations entre des spécifications et leurs caractéristiques respectives (par exemple le type de diagnostic qu'elles définissent, le contexte pédagogique auquel elles sont associées) et (3) une interface de communication entre l'utilisateur et CD-SPECIES.

3.6.3.1.1. Base de connaissances déclarative sur le DC

Les connaissances déclaratives à propos du DC sont de deux types: le vocabulaire des composantes du DC et les règles relatives à la cohérence entre des instances de composantes du DC.

Le vocabulaire du DC contient les dénominations des composantes (par exemple les composantes "type de diagnostic", "contexte pédagogique") du DC ainsi que leurs descriptions respectives (Tableau III.12).

Le vocabulaire comprend également les dénominations des possibles instances de composantes du DC ainsi que leurs descriptions respectives. Par exemple la dénomination "DC épistémique" est une instance d'une composante du DC dénommée "type de DC".

Tableau III.12. Vocabulaire dans la base de connaissances de CD-SPECIES

CD_SPECIES_Base_de_connaissances.Vocabulaire. Contexte_Pédagogique {Pragmatiste.ApprentissageConstructiviste}	
Composantes du DC	Vocabulaire
Inférences	<skill_inquiry> <skill_drawConclusion> <skill_changeOfUnderstanding> <skill_planning>, <skill_integrateCultureIntoLearning> <skill_knowledgeConstructionFromInteraction>
Observations	<skill_inquiry> <skill_informationCollection> [number_of_hypothesis_made] [number_of_conclusions_drawn] [number_consistent_conclusion_drawn]
Réflexion	<Tool_To_Relate_Activities_In_Inquiry_Process> <Tool_To_Trace_Interactions_Frequency>
Types d'algorithmes de DC	- <Relate_Resources_Inquired> <Infer_Inquiry_Method> <Human_Analysis>{Explanation_of_Drawn_Concept}

L'ensemble des règles relatives au DC permet de définir des contraintes entre les différentes valeurs qui peuvent être attribuées à ses composantes. En particulier lorsque les contraintes portent sur les instances des composantes du DC en relation avec le contexte pédagogique, on parle de "règles contextuelles". Au Tableau III.13 par exemple, la première règle contextuelle indique que CD-SPECIES devrait décourager l'utilisation de la

performance de l'apprenant comme moyen d'acquisition d'indices pour un DC qui se déroule dans un contexte d'apprentissage où la perspective de la cognition est pragmatiste et socio-historique.

La Figure III.5 illustre comment CD-SPECIES indique au concepteur pédagogique une incohérence identifiée au niveau de la nature de la phase d'acquisition d'indices, dans son contexte pédagogique.

Tableau III.13. Illustration: règles de la base de connaissances de CD-SPECIES

CD_SPECIES_Knowledge_Base.Rules	
RÈGLE CONTEXTUELLE	
1.	SI <type_of_observation> is {performance} AND <learning_context> contains {pragmatist:socio-constructivist} ⇒ inconsistency_message ("In Socio-constructivist contexts, learners actions and interactions are more relevant for Cognitive Diagnosis than performance")
2.	SI <learning_context> contains {pragmatist:socio-constructivist} ET SI <goal_of_observation> is {skill_knowledgeConstructInterActions} ET SI <type_of_observation> is {skill_informationCollection} ⇒ irrelevance_message ("Information collection will generally happen prior to the interaction with peers. It is more appropriate in the individual inquiry process")
3.	IF <learning_context> contains {pragmatist:.*} AND IF <goal_of_observation> is { skill_changeOfUnderstanding } ⇒ accept_values (<Observation> in {Association, Performance})
RÈGLE GÉNÉRALE	
1.	Si <contexte_d_apprentissage> contient {pragmatiste:.*} ET <but_spécifique_de_inference> est {evolution_de_compréhension_d_une_capacité} ⇒ Accepter_les_valeurs ({Association, Performance} pour_opération<Observation>)

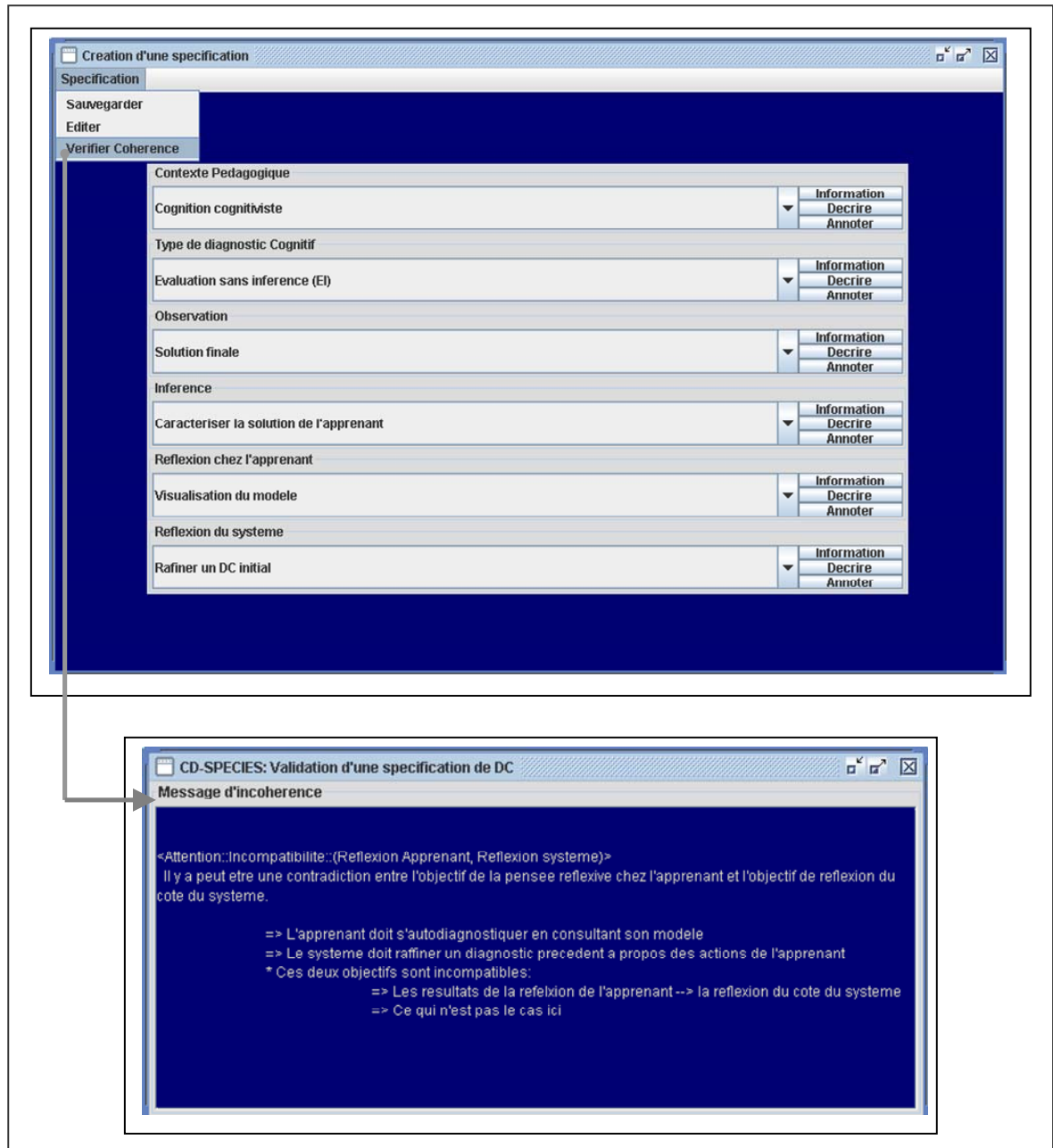


Figure III.5. CD-SPECIES: Interface de vérification de la cohérence

3.6.3.1.2. Base de spécifications

Enfin, CD-SPECIES doit pouvoir associer les spécifications qu'il sauvegarde aux caractéristiques qui permettent leur indexation. Cela peut se faire à travers une base de

données ou à travers un formatage particulier des fichiers où ces spécifications sont sauvegardées.

Dans le premier cas, les champs de cette base de données vont correspondre aux composantes de l'indexation. Si l'on assume par exemple que l'indexation correspond au contexte pédagogique du DC et à sa nature conceptuelle, les champs de la base de données seront: "Paradigme de cognition", "Théorie d'apprentissage", "Type de DC" (Figure III.6).

The figure consists of two screenshots of a software interface for saving and indexing a DC specification.

The top screenshot, titled "Creation d'une specification", shows a window with a menu bar containing "Specification", "Sauvegarder", "Editer", and "Verifier Coherence". The main area contains several sections, each with a dropdown menu and a set of buttons:

- Contexte Pédagogique**: "Cognition cognitiviste" (dropdown), buttons: Information, Decrire, Annoter.
- Type de diagnostic Cognitif**: "Evaluation sans inference (EI)" (dropdown), buttons: Information, Decrire, Annoter.
- Observation**: "Solution finale" (dropdown), buttons: Information, Decrire, Annoter.
- Inference**: "Caracteriser la solution de l'apprenant" (dropdown), buttons: Information, Decrire, Annoter.
- Reflexion chez l'apprenant**: "Aucune" (dropdown), buttons: Information, Decrire, Annoter.
- Reflexion du systeme**: "Aucune" (dropdown), buttons: Information, Decrire, Annoter.

The bottom screenshot, titled "CD-SPECIES: Sauvegarde d'une specification", shows a file save dialog with the filename "Nom du fichier" and extension ".dcspecies". Below it is a "Parametres d'indexation" section with the following checkboxes:

- Contexte pedagogique
- Type de DC
- Type d'observation
- Type d'inference
- Approche de reflexion

Buttons for "Sauvegarder" and "Annuler" are visible in the top right of the dialog.

Figure III.6. CD-SPECIES : Sauvegarde et Indexation d'une spécification de DC

Dans le second cas, il faudrait que les fichiers contenant les spécifications puissent en indiquer explicitement les caractéristiques. Une façon de procéder ainsi consiste à ajouter à chaque fichier de spécification un préambule décrivant les caractéristiques de la spécification qu'il contient.

Dans tous les cas, une base de données est nécessaire, ne serait ce que pour indiquer la localisation physique du fichier contenant une spécification. Par conséquent, CD-SPECIES doit disposer d'au moins une base de données associant soit le nom d'une spécification soit ses caractéristiques, à sa localisation physique (par exemple, le chemin d'un répertoire sur la machine locale de CD-SPECIES).

3.6.3.2. L'interface de CD-SPECIES

L'interface de CD-SPECIES permet un lien entre son utilisateur et la base de connaissances. C'est cette interface qui permettra à l'utilisateur d'effectuer des requêtes à CD-SPECIES. On distingue l'interface du concepteur pédagogique et l'interface du programmeur.

Le concepteur pédagogique dispose d'une interface où il pourra déclencher :

- La définition d'une spécification ;
- L'annotation d'une spécification ;
- La requête de vérification de la cohérence d'une spécification ;
- La requête d'informations à propos d'un élément du DC ;
- La requête de visualisation d'un ensemble de spécifications répondant à des caractéristiques qu'il aura spécifiées.
- La requête de vérification de la cohérence d'une spécification ;
- La requête d'informations à propos d'un élément du DC ;
- La requête de visualisation d'un ensemble de spécifications répondant à des caractéristiques qu'il aura spécifiées.
- La requête d'ajout d'informations dans la base de connaissances, soit au niveau de la structure du CSDC (composantes du DC), soit au niveau de son contenu (instances des composantes du DC).

Le programmeur dispose d'une interface d'où il pourra solliciter le système pour les principales opérations suivantes :

- (1) La visualisation d'une spécification en indiquant soit
 - a. le nom du fichier associé

- b. les caractéristiques associées à la spécification désirée
- (2) La génération de pseudo algorithmes associés à la spécification actuellement visualisée;
- (3) La génération de l'architecture conceptuelle du module de DC associé à la spécification actuellement visualisée.

3.6.4. La dynamique de CD-SPECIES

En plus de la base de connaissances, les réponses de CD-SPECIES à son utilisateur ainsi que sa gestion des spécifications s'appuie sur sa dynamique globale. Il existe un lien entre les composantes de CD-SPECIES, les requêtes de ses utilisateurs et les opérations nécessaires pour satisfaire ces requêtes. Trois grandes classes d'opérations peuvent assurer ce lien: les opérations permettant de présenter les spécifications de CD sur demande de l'utilisateur, les opérations permettant la création des spécifications et la vérification de leur cohérence, et les opérations permettant à CD-SPECIES de générer un pseudo algorithme et une architecture conceptuelle associés à une spécification.

3.6.4.1.1. *Lien entre les interfaces et la base de spécifications*

La Figure A. 1 (APPENDICE. A) illustre le lien entre l'interface et la base de connaissances de CD-SPECIES. Ce lien est une opération qui assure la réception et le traitement adéquat d'une requête de visualisation de spécification. L'opération s'élabore à travers trois commandes. Sur réception de la requête de l'utilisateur, CD-SPECIES:

- (1) Traite la requête afin d'en déterminer la nature.
- (2) Accède à la partie de la base de connaissances si nécessaire pour traiter la requête en question. Dans ce cas, il s'agit de la visualisation d'une spécification; CD-SPECIES doit donc accéder à la base de données qui associe des caractéristiques de spécification, à sa localisation physique;
- (3) Après traitement de la requête et obtention de la réponse appropriée, CD-SPECIES doit récupérer la spécification trouvée et l'afficher à l'interface de l'utilisateur.

Implications dans l'implémentation de CD-SPECIES

La première opération implique qu'un format de représentation de requête est nécessaire lors de l'implémentation de CD-SPECIES. Ce format doit permettre de différencier les types de requêtes (requête de visualisation, requête d'édition, requête de vérification de cohérence, requête de création d'une spécification) afin de déclencher l'algorithme de traitement

approprié. Les seconde et troisième opérations nécessitent la présence d'une bande de communication entre d'une part l'algorithme de traitement de requête de visualisation d'une spécification et, d'autre part la base de données qui associe les locations physiques des spécifications à leurs caractéristiques conceptuelles. La quatrième opération n'a pas d'implication additionnelle.

3.6.4.1.2. *Lien entre les connaissances sur le DC et les interfaces*

Les opérations permettant à CD-SPECIES de fournir des descriptions de composantes du DC et d'analyser la cohérence d'une spécification sont analogues à celles associées à la visualisation d'une spécification. La différence réside dans la partie de la base de connaissances à laquelle accèdent les opérations.

Pour une requête de description d'une composante ou d'une instance de composante, CD-SPECIES accède au vocabulaire du DC. Si la requête concerne une vérification de la cohérence d'une spécification, CD-SPECIES accède alors aux règles relatives au DC.

Dans tous les cas, des descriptions sont déjà associées à chaque élément du vocabulaire. Des messages d'erreurs sont également associés à toutes les règles qui sont violées. Ainsi, s'il est nécessaire de fournir une information à l'utilisateur, CD-SPECIES n'a plus qu'à récupérer ces informations et à les afficher.

3.6.4.1.3. *Autres liens avec l'interface de CD-SPECIES : génération d'interfaces secondaires*

Certaines requêtes nécessitent la génération d'interfaces secondaires par CD-SPECIES. Par exemple, lorsqu'un concepteur pédagogique désire créer une spécification, CD-SPECIES doit présenter une interface où il pourra inscrire les informations relatives au contexte pédagogique, aux caractéristiques conceptuelles de l'instance de CD en question, et à ses caractéristiques opérationnelles. Par contre si le concepteur désire simplement visualiser une spécification, CD-SPECIES devra présenter une interface où le concepteur peut indiquer soit les caractéristiques du DC associé à la spécification recherchée, soit le nom du fichier qui contient cette dernière. Cela implique que CD-SPECIES doit intégrer un ensemble d'interfaces typiques, de même qu'un processus de génération de l'interface appropriée selon chaque requête (APPENDICE. A: Figure A. 2).

3.7 **Conclusion**

L'apprentissage à travers un STI se caractérise par l'absence physique d'un tuteur humain *a priori*. La nécessité d'une attitude scientifique lors de leur conception est souhaitable et peut s'avérer bénéfique pour la stabilité à long terme du système informatique sous-jacent. En l'occurrence, la conception des fonctions pédagogiques d'un STI devrait se fonder sur les principes de théories de cognition, d'apprentissage, d'instruction ou tout au moins, sur un cadre de référence.

Ce chapitre a présenté un cadre de référence pour la conception ou la spécification du DC, une fonction pédagogique clé dans la mise en œuvre des STI. Les caractéristiques et opérations de base ont été introduites dans un cadre de base. Deux défis additionnels ont guidé le développement de ce CSDC, leur but étant d'y refléter explicitement les implications d'une conception "«pro-pédagogique»" du DC. Premièrement, nous avons jugé primordial de définir comment les résultats du DC peuvent être utilisés à des fins pédagogiques. L'intégration du phénomène de la pensée réflexive dans le processus de DC à travers une boucle diagnostic-réflexion a permis de montrer comment un apprentissage plus fondé peut être favorisé dans un STI. Deuxièmement, l'influence conceptuelle des paradigmes de cognition sur le processus de DC a été étudiée. Ces deux derniers facteurs constituent l'originalité de notre contribution dans le domaine AIED au niveau de la conception du processus de diagnostic cognitif pour un STI.

En outre, nous avons présenté l'architecture de CD-SPECIES. Pour les concepteurs de STI, il s'agit d'un outil de support à la spécification d'une instance de DC et son but est de véhiculer la perspective du DC préconisée à travers le CSDC. Pour le programmeur, il s'agit d'un outil de support à la traduction des conditions associées à une spécification de DC en termes informatiques. Il constitue un point de départ à l'implémentation d'un module de DC dans un STI: étant donné une spécification d'instance de DC, les recommandations sur les mécanismes de représentation des connaissances et sur les algorithmes de traitement peuvent être utilisées pour proposer une esquisse d'algorithme de DC dans un STI. Bien que CD-SPECIES soit une maquette et non un prototype de système informatique, il constitue une contribution originale dans le domaine AIED au niveau de la conception du module de diagnostic cognitif pour un STI.

Le CSDC proposé dans ce chapitre établit déjà un ensemble de balises permettant de communiquer notre perspective du DC. Dans la perspective de l'implémentation du DC dans

un STI, l'aspect le plus important (au sens opérationnel ou sur le plan de l'implémentation) de la perspective du DC préconisée dans ce cadre est la boucle diagnostic-remédiation. La formalisation de cette boucle en favoriserait l'implémentation effective lors de la programmation du module de DC dans un STI. Le prochain chapitre propose une description formelle de la boucle diagnostic remédiation à travers un algorithme générique de DC, basé sur l'inférence bayésienne.

Chapitre IV

FORMALISATION DE LA BOUCLE DIAGNOSTIC-REMÉDIATION: UNE APPROCHE BAYESIENNE

4.1 Introduction

Au chapitre précédent, l'originalité de la contribution de cette thèse a été mise en évidence au niveau de la conception du module de DC dans un STI, à travers un cadre de spécification reflétant une perspective pédagogique du DC. L'originalité de cette contribution s'inscrit également dans le domaine des applications de l'intelligence artificielle en éducation, ce en ce qui concerne l'implémentation du module de DC dans un STI.

Le but de ce chapitre est de favoriser l'implémentation de la perspective du DC préconisée dans le cadre de spécification défini au chapitre précédent. Nous proposons une formalisation de la boucle diagnostic-remédiation en s'appuyant sur une méthode d'inférence bayésienne. Cette approche a été choisie tant pour sa capacité à tenir compte des facteurs d'incertitude que pour le caractère fondé et théorique des algorithmes d'inférence qu'elle permet d'exploiter. La première section (section 4.2.) présente les fondements du raisonnement bayésien et son application dans les STI. La deuxième section (section 4.3.) a

pour objectif de justifier le choix de l'approche bayésien pour émuler le raisonnement incertain dans un système intelligent. La troisième section (section 4.4.) a pour but de mettre en évidence les implications d'une approche de formalisation bayésienne, sur la perspective du DC dans un STI présentée au chapitre précédent. La dernière section (section 4.5) présente la formalisation de cette perspective, ainsi que l'algorithme correspondant de DC.

4.2 Les inférences bayésiennes

Le raisonnement humain relève plus du sens commun que d'un déterminisme pur étant donné que les informations sur lesquelles il se base sont généralement incomplètes ou incertaines. Pour émuler ce raisonnement dans un système intelligent, on peut s'inspirer des principes issus de théories relatives au *raisonnement incertain*. L'une de ces théories est la théorie de Thomas Bayes (Bayes, 1763), basée sur les notions de causalité sémantique, de conditionnement et de vraisemblance.

La causalité sémantique permet de faire des déductions à partir d'informations partielles ou générales (contrairement à la causalité logique où ce n'est pas toujours les cas). Il existe une causalité sémantique de l'entité X sur l'entité Y si Y est une conséquence de X (Figure IV.1). Cette propriété est particulièrement importante dans un processus de raisonnement où l'on désire relier des effets à des causes (et vice-versa), ce qui est le cas du processus diagnostique en général (et du diagnostic cognitif en particulier). Si l'on considère le domaine de connaissances associées à la programmation logique par exemple; s'il existe une causalité sémantique de la capacité "Appliquer une Unification" sur la capacité "Appliquer une Résolution", alors une information sur l'état des connaissances de l'apprenant par rapport à l'une de ces deux capacités pourrait permettre de déduire ou d'estimer le niveau ou l'état d'acquisition de cet apprenant par rapport à l'autre capacité.

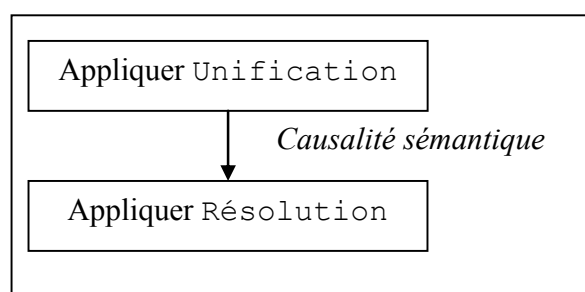


Figure IV.1. Causalité sémantique entre deux capacités dans un domaine

Le conditionnement permet d'estimer la probabilité d'un évènement X en fonction de l'occurrence certaine d'un autre évènement Y. Elle se traduit toujours par des affirmations de la forme "Sachant que X, alors, Y". Le conditionnement correspond à la caractéristique du raisonnement qui combine ou exploite une connaissance pour établir une opinion ou croyance.

La vraisemblance (encore appelée plausibilité) consiste à établir un ordre total ou partiel sur des évènements, ce à travers leur probabilité respective d'occurrence, étant donné la certitude autre d'un évènement. La vraisemblance est un thème important dans l'établissement d'un lien entre les probabilités et le raisonnement. Elle permet d'estimer la justesse d'une hypothèse en pesant les conséquences de son adoption sur la vraisemblance des autres hypothèses associées à un contexte de raisonnement incertain.

La causalité sémantique, le conditionnement et la vraisemblance constituent le fondement du *théorème de Bayes*, base théorique du raisonnement dans un réseau bayésien:

- Lorsqu'une entité C exerce une causalité sémantique sur une entité A, alors on peut exprimer la probabilité conditionnelle de l'entité A sachant l'entité C (Bayes, 1763; Figure IV.2):

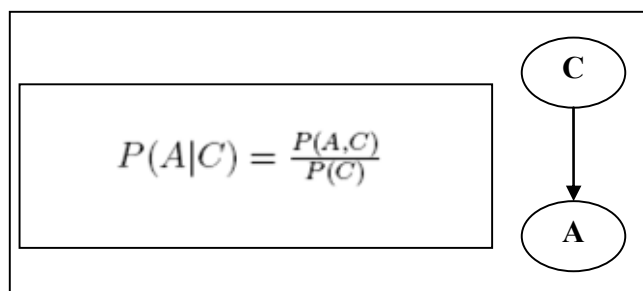


Figure IV.2. Déduction bayésienne Probabilité conditionnelle de A sachant B

Il s'agit de la déduction bayésienne qui établit la probabilité d'une conséquence étant donné l'occurrence de sa cause.

- La formule de Bayes donne la probabilité conditionnelle de A, sachant que C a été observé, ce lorsque *C est une cause de A* (Figure IV.2, ci-dessus). Lorsque c'est *plutôt A qui est observé, on recherche la vraisemblance de C* (quelle est la probabilité que l'observation de A est due à l'occurrence de C). Si l'on considère la: H est une cause de

e , mais c'est e qui est observée. Alors la probabilité de H sachant e est donnée par la formule de vraisemblance (Figure IV.3).

$$P(H|e) = \frac{P(e|H) \times P(H)}{P(e)}$$

Figure IV.3. Abduction bayésienne: vraisemblance de H sachant e

Il s'agit de l'abduction bayésienne qui établit la probabilité d'une cause (ou encore sa vraisemblance) étant donné l'observation de sa conséquence. Cette formule est une formalisation du raisonnement associé à l'activité de diagnostic.

Les deux approches d'inférence bayésienne présentées ci-dessus constituent le fondement du raisonnement incertain dans les systèmes intelligents.

4.2.1. Applications du raisonnement bayésien dans les STI

Dans un STI, le raisonnement bayésien peut être utilisé pour :

- Mettre à jour le modèle de l'apprenant (Nkambou et Tchétagni, 2002; Tchétagni et Nkambou, 2002);
- Diagnostiquer les causes des erreurs de l'apprenant (Tchétagni et Nkambou, 2004a);
- Prédire les actions de l'action de l'apprenant dans un processus de résolution de problème (Conati, Gertner, et VanLehn, 2002).

4.2.1.1. Les réseaux bayésien et le raisonnement bayésien

Pour que ces inférences soient possibles, on doit définir un ensemble de relations causales entre tous les éléments représentés dans le modèle des connaissances de l'apprenant ou dans le modèle de son raisonnement. Cet ensemble de relations causales est un réseau bayésien (Pearl, 1988).

Un réseau bayésien est un graphe dont les arcs sont dirigés et dont les nœuds représentent des variables aléatoires. Les arcs du réseau modélisent les relations causales entre ces variables, tandis que les valeurs numériques attribuées à chacun d'eux modélisent

l'intensité des probabilités conditionnelles associées à ces relations causales. À tout moment, l'état global d'un réseau bayésien est défini par l'union des états (des valeurs prises) par l'ensemble des variables qu'il contient. Dans ce cas, l'utilité d'un tel système est de pouvoir en tout temps :

- ⇒ calculer la probabilité de son nouvel état étant donné qu'un ensemble d'évènements a été observé (certaines variables ont été observées ou ont pris des valeurs). Il s'agit de l'inférence des probabilités postérieures des variables du réseau qui n'ont pas été observées (*Belief Updating*). Pour cela, Pearl (1988) propose l'application des formules présentées à la Figure IV.2 et à la Figure IV.3 (ci-dessus en page 117), pour chaque variable \mathcal{V}_i du réseau, selon que l'évidence observée corresponde à un nœud qui est une conséquence ou une cause de la valeur de \mathcal{V}_i .
- ⇒ trouver l'état du réseau qui explique le mieux un ensemble d'observations. Il s'agit de l'état du réseau qui maximise la probabilité des évènements observés ("*Belief Revision*"). Pour cela, Pearl (1988) utilise la formule de la règle en chaîne pour matérialiser ces algorithmes. Dans cette formule X_i est un nœud du réseau et Π_{X_i} est l'ensemble des nœuds parents de X_i . Ainsi, en tout temps, la probabilité de l'état d'un réseau bayésien de n variables est le produit des probabilités postérieures de chacune de ces variables (Figure IV.4).

$$P(X_1, X_2, \dots, X_n) = \prod_i^n (P(X_i | \Pi_{X_i}))$$

Figure IV.4. Probabilité de l'état global d'un réseau bayésien de n variables.

Les prochaines lignes expliquent plus en détails comment la mise à jour, le diagnostic des erreurs de l'apprenant peuvent être effectuées dans un STI, sur la base de la règle en chaîne.

4.2.1.2. Mise à jour du modèle de l'apprenant

Si l'on considère que les éléments du modèle de l'apprenant correspondent à l'ensemble des capacités visées à travers un STI (représentées dans un modèle des connaissances), alors un réseau bayésien devrait modéliser les relations causales entre l'état d'acquisition des capacités visées dans un STI. Dans ce cas particulier, l'état d'acquisition d'une capacité est une variable aléatoire binaire

- ⇒ Si sa valeur est 1, cela signifie que la capacité en question est acquise

⇒ Si sa valeur est 0, cela signifie que l'apprenant éprouve des difficultés ou n'a pas encore acquis la capacité en question.

Le modèle de l'apprenant représente la probabilité d'acquisition de l'apprenant par rapport à chaque capacité représentée dans le domaine de connaissances. La mise à jour du modèle de l'apprenant consiste à mettre à jour ces probabilités, en fonction des actions posées par l'apprenant lors de diverses activités d'apprentissage dans un STI. Pour faciliter la lecture, on adopte le vocabulaire suivant:

- "L'état d'un nœud du réseau bayésien est 0", signifie que la capacité représentée par ce nœud est non acquise ou encore que l'apprenant a posé une action qui démontre qu'il éprouve des difficultés avec cette capacité;
- "L'état d'un nœud du réseau bayésien est 1" signifie que la capacité représentée par ce nœud est acquise ou encore que l'apprenant a posé une action qui démontre qu'il possède cette capacité;
- L'état du réseau bayésien correspond à l'ensemble des états de tous ses nœuds.

Chaque fois que l'apprenant pose une action qui démontre qu'il a acquis une capacité, cette action est enregistrée par le réseau bayésien comme *évidence positive* (Pearl, 1988). S'il s'agissait d'une action qui démontre que l'apprenant éprouve des difficultés avec une capacité, elle aurait été enregistrée comme *évidence négative*. Sur la base des formules présentées à la Figure IV.2 et à la Figure IV.3 (ci-dessus en page 117), cette évidence est propagée à travers tout le réseau, avec comme point d'entrée la capacité à laquelle elle a été associée. Cette propagation encore appelée "*Belief Updating*" permet d'obtenir les probabilités d'acquisition *a posteriori* (étant donné l'évidence) de toutes les autres capacités représentées. La Figure IV.5 illustre cette propagation: l'apprenant réussit à résoudre un exercice qui sert d'évidence à l'acquisition de la capacité à *Appliquer une Unification en programmation logique*. L'illustration montre comment cette évidence est propagée, ce qui permet de calculer les nouvelles probabilités des autres capacités représentées dans le réseau bayésien.

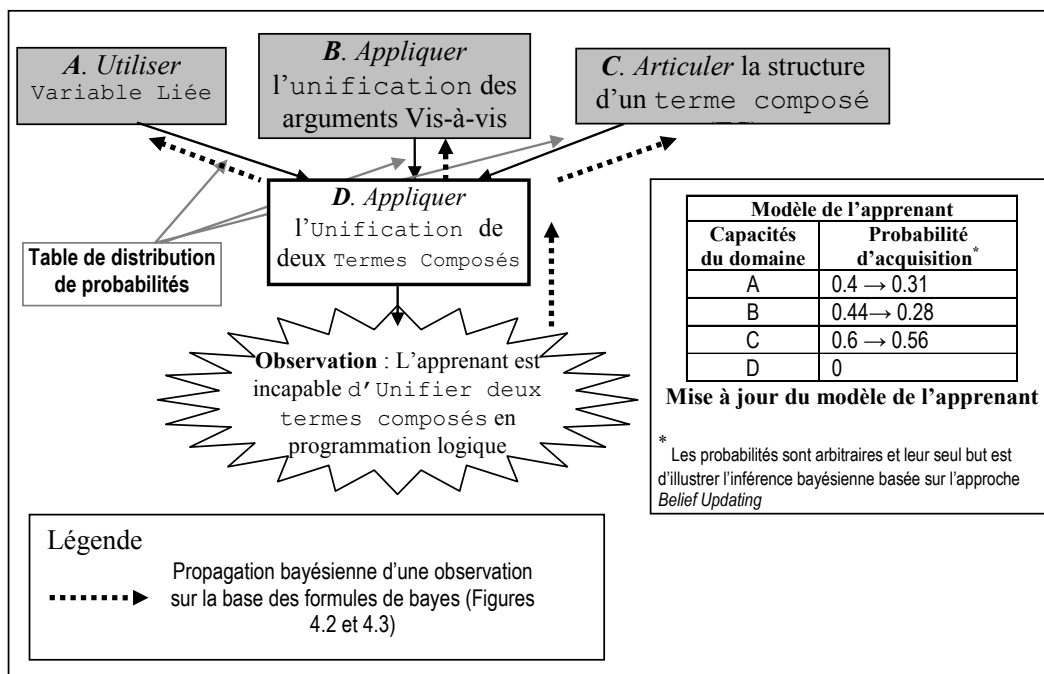


Figure IV.5. Déduction bayésienne pour la mise à jour du modèle de l'apprenant

4.2.1.3. Diagnostic des erreurs de l'apprenant

Dans ce même contexte, le diagnostic des erreurs de l'apprenant peut aussi être effectué. Lorsqu'une action démontrant que l'apprenant éprouve des difficultés avec une capacité C est enregistrée, on recherche les causes les plus probables de ces difficultés en recherchant l'état du réseau qui maximise la probabilité que C n'est pas acquise par cet apprenant (Figure IV.6). Cette recherche est assurée par un algorithme appelé *Belief Revision* ou *Most Probable Explanation* (Pearl, 1988). Dans l'exemple de la Figure IV.6, on a trois états possibles du réseau et l'état pour lequel la probabilité que la capacité à *Appliquer* une Unification de deux Termes Composés est non acquise implique que les nœuds : «*Identifier* une Variable Liée» et «*Unifier* les Arguments d'un Terme Composé» ont la valeur 0. Ces nœuds peuvent donc être considérés comme les causes les plus probables de l'observation selon laquelle la capacité à *Appliquer* une Unification de deux Termes Composés est non acquise.

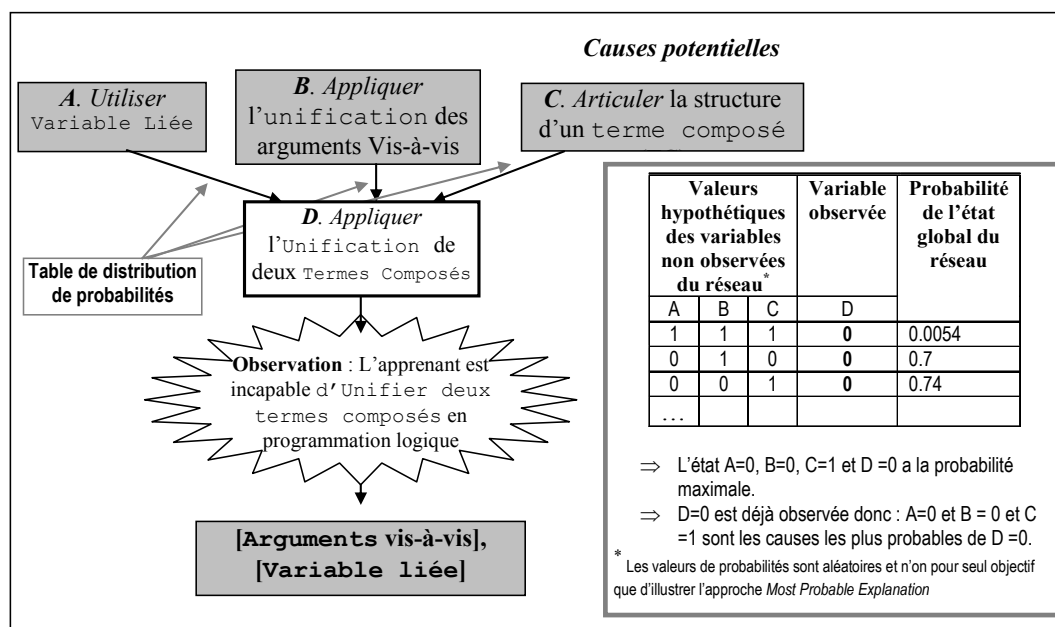


Figure IV.6. Abduction bayésienne pour le diagnostic des erreurs de l'apprenant

La prochaine section justifie le choix de l'approche bayésienne pour formaliser le DC dans cette thèse.

4.3 Choix d'une approche bayésienne: justification

Les réseaux bayésiens correspondent à une approche de représentation des connaissances permettant la mise en oeuvre du raisonnement incertain dans les systèmes intelligents. Ils favorisent donc une approche plus réaliste du raisonnement, étant donné que l'incertitude et les nuances caractérisent les situations auxquelles il s'applique. Le diagnostic est une forme de raisonnement au cours duquel le but est de relier un effet, un phénomène – en d'autres termes une observation – à une cause qui l'explique. Plusieurs causes peuvent concourir à l'explication d'un même effet et dans ce cas, il est nécessaire d'identifier les causes les plus probables pour favoriser une exploitation plus efficace des résultats d'un diagnostic. Les réseaux bayésiens constituent une approche indiquée puisqu'ils permettent de représenter les relations entre les informations associées à une situation, de manière à y appliquer un raisonnement incertain.

Outre les réseaux bayésiens, la logique floue (Zadeh, 1983), la théorie de Dempster-Shafer (Dempster, 1967; Shafer, 1976), les chaînes de Markov cachées (HMM, Hidden

Markov Chains) (Viterbi, 1967) sont les principales approches d'implémentation du raisonnement incertain dans les systèmes intelligents. Dans le cas du diagnostic cognitif appliqué dans un STI, la logique floue est un choix moins souhaitable car les inférences (le *modus ponens généralisé*) qu'elle permet d'établir ne reflètent pas toujours la réalité et surtout, entraînent l'élagage d'une partie importante de l'information. Les chaînes de Markov correspondent à une représentation de processus stochastiques séquentiels qui ne tiennent pas compte du passé lointain. L'état cognitif de l'apprenant tel que diagnostiqué au cours de l'apprentissage ne peut être associé à un processus séquentiel; par conséquent, les HMM ne peuvent pas être utilisées pour implémenter le DC basé sur un raisonnement incertain. La théorie de Dempster-Shafer est similaire aux réseaux bayésiens: elle permet de considérer toutes les hypothèses possibles pouvant expliquer une observation, et de les éliminer au fur et à mesure que de nouvelles observations les infirment et les confirment. Malgré la richesse intuitive de cette théorie, elle ne considère pas les ensembles d'hypothèses non disjoints: toutes les hypothèses considérées lors d'un diagnostic doivent être mutuellement exclusives. Dans le cas du DC dans un STI, on ne saurait imposer cette restriction puisque plusieurs ensembles d'hypothèses peuvent expliquer le même comportement de l'apprenant, plusieurs hypothèses pouvant se retrouver dans le même ensemble.

4.4 Implication de l'approche bayésienne sur l'implémentation du DC dans un STI: le modèle causal

La section 3.3.1.2. a porté sur la relation entre le DC et les principaux modules d'un STI qui contiennent des informations nécessaires à ce processus (le module du domaine et le module de l'apprenant). À un niveau conceptuel, la relation entre le DC et les modules d'un STI a été matérialisée à travers une structure que nous avons appelée *modèle causal* (Tchetagni et Nkambou, 2004b). Le modèle causal relie les actions possibles de l'apprenant à des composantes de son état cognitif. Dans un dialogue tutoriel par exemple, le modèle causal est la structure qui relie explicitement une question du dialogue à la capacité du domaine nécessaire pour y répondre correctement: en cas de réponse incorrecte de l'apprenant, on considère qu'on détient un indicateur que cette capacité n'est pas acquise. Des classes d'approches de représentation des connaissances nécessaires dans les modules du domaine et de l'apprenant ont été proposées, comme les mieux appropriées pour implémenter chaque type de DC.

Le but de cette section est de décrire les implications d'une approche bayésienne sur le modèle causal, le module du domaine et le module de l'apprenant pour chaque type de DC, ce en fonction des approches de représentation des connaissances qui y sont nécessaires: l'évaluation sans inférence, le DC comportemental par inférence du comportement non observable, le DC épistémique.

4.4.1. Modèle causal pour le DC par évaluation sans inférence

Les structures appelées *Frames* sont la principale approche de représentation des connaissances dans ce type de diagnostic. Ces structures permettent de caractériser les solutions possibles d'un exercice. Elles permettent aussi de caractériser les réponses possibles des apprenants à un exercice. Le but de l'évaluation sans inférence est de caractériser la réponse de l'apprenant à un exercice en l'associant à une classe de solutions, à un patron de solutions. En termes de diagnostic, il s'agit donc d'expliquer la solution de l'apprenant (observation, évidence ou effet dans le vocabulaire Bayésien) en termes de ses caractéristiques (les caractéristiques définissent une classe de solution ou un patron de solution). Le modèle causal qui supporte le DC par évaluation sans inférence doit donc permettre de relier ces deux informations. *Cette relation doit exprimer la probabilité qu'une réponse de l'apprenant appartienne à une classe ou à un patron de solution (défini dans le module du domaine), étant donné ses caractéristiques.* L'approche bayésienne n'a pas d'impact particulier sur le contenu du module de l'apprenant (une structure contenant pour chaque réponse fournie par l'apprenant, la classe de solutions ou le patron de solution associé).

4.4.2. Modèle causal pour le DC du comportement non observable

Outre les *Frames*, deux principales approches de représentation des connaissances sont nécessaires pour le DC du comportement mental non observable: les systèmes à base de règles de production (qui contribuent également à la construction dynamique d'un espace problème) et les bibliothèques de plans. Le but du diagnostic du comportement mental non observable est l'inférence d'un raisonnement que l'apprenant serait entrain de suivre (en terme d'une suite de règles de production ou de buts poursuivis), en fonction de sa réponse finale à un exercice ou de ses réponses partielles dans une construction progressive de réponse à un exercice.

Le modèle causal qui supporte le DC doit donc permettre de relier une réponse (finale ou partielle de l'apprenant) à l'utilisation d'une règle de production ou à la poursuite d'un but (ou d'un ensemble de buts). *Cette relation doit exprimer la probabilité qu'une règle de production est utilisée ou qu'un but est poursuivi par l'apprenant, étant donné sa réponse actuelle (et ses réponses passées si nécessaire).*

Pour faciliter le diagnostic dans ce cas, il serait préférable que le modèle de l'apprenant (intégré dans le module de l'apprenant du STI) exprime l'état cognitif de l'apprenant en termes probabilistes:

- Si ce modèle représente le comportement mental de l'apprenant: chaque règle de production ou chaque ligne de raisonnement qui y figure doit avoir une probabilité d'utilisation (ou chaque but – ensemble de buts – doit avoir une probabilité d'être poursuivi) étant donné un état de résolution d'un problème ou d'un exercice;
- Si ce modèle représente l'état d'acquisition des capacités de l'apprenant: chaque capacité du domaine visée dans le STI doit avoir une probabilité d'acquisition dans ce modèle.

En effet, les probabilités associées à chaque élément du modèle sont utilisées pour arbitrer le choix de l'hypothèse finale de diagnostic. Ce qui arrive lorsque plusieurs causes probables à niveaux égaux sont candidates à l'explication d'une observation.

Par ailleurs, les informations du modèle de l'apprenant représentent une *qualification de son état* cognitif relativement à une représentation du domaine appris (modèle du domaine). Par conséquent, une approche probabiliste de représentation de l'état cognitif de l'apprenant nécessite que le modèle du domaine soit conçu comme un réseau bayésien :

- Si le modèle du domaine représente le domaine à apprendre en termes d'ensembles de règles de production, chacun étant associé à une tâche ou à un exercice précis: dans ce cas, le réseau bayésien correspondant devrait définir la distribution des probabilités conditionnelles des règles de production. Il s'agit de l'ensemble des probabilités s'exprimant de la manière suivante pour chaque règle de production R_j : la probabilité que la prochaine action soit basée sur la règle de production R_j sachant que l'action actuelle est basée sur les règles de production R_1, R_2, \dots, R_n ;

- Si le modèle du domaine représente les capacités et leurs interrelations, le réseau bayésien correspondant devrait définir la distribution des probabilités conditionnelles de chaque capacité C_i : la probabilité que la capacité C_i est acquise, sachant les capacités C_1, C_2, \dots, C_n ;

4.4.3. Modèle causal pour le DC épistémique

Le diagnostic épistémique porte sur l'état des connaissances (proprement dit, l'état de ses capacités par rapport aux connaissances du domaine) de l'apprenant. Nous avons vu que les approches de représentation des connaissances nécessaires pour ce type de diagnostic dépendent du type de diagnostic comportemental sur lequel il se base. Le but du diagnostic épistémique est d'inférer les capacités acquises et les capacités non acquises par l'apprenant, à la lumière de ses réponses lors de la réalisation d'un exercice.

Le modèle causal qui supporte un DC devrait donc permettre de relier explicitement une réponse finale ou partielle de l'apprenant à la mise en oeuvre d'une capacité (ou au reflet d'une incapacité/difficulté). Cette relation doit exprimer la probabilité qu'une capacité est acquise par l'apprenant sachant qu'il fournit une bonne réponse à une question ou à un exercice.

Pour les mêmes raisons que dans le diagnostic comportemental, il serait préférable que les informations du modèle de l'apprenant indiquent l'état cognitif de l'apprenant en termes probabilistes. Dans ce cas, étant donné qu'on diagnostique les capacités acquises et non acquises par l'apprenant, ce modèle devrait représenter pour chaque capacité modélisée dans le domaine de connaissances, la probabilité qu'elle est acquise.

Par ailleurs, un réseau bayésien doit être calqué à partir du modèle du domaine, de manière à pouvoir effectuer des inférences qui se reflèteront dans le modèle de l'apprenant.

4.5 Représentation formelle du DC selon un raisonnement bayésien

La Figure IV.7 illustre l'algorithme générique associé au processus de DC tel que défini dans le CSDC. Après l'étape d'acquisition d'indices, deux phases importantes composent l'étape d'inférence d'hypothèses: (1) la génération des explications (ou des causes) les plus probables des indices observés et (2) l'identification des explications les plus plausibles.

Au terme de cette phase d'inférence, l'exploitation pédagogique du DC permet de le raffiner ou de le confirmer – cela dépend de la forme que prend cette exploitation -. Dans

l'optique du DC préconisée dans cette thèse, cette élaboration peut se faire à travers une boucle entre le diagnostic et la remédiation.

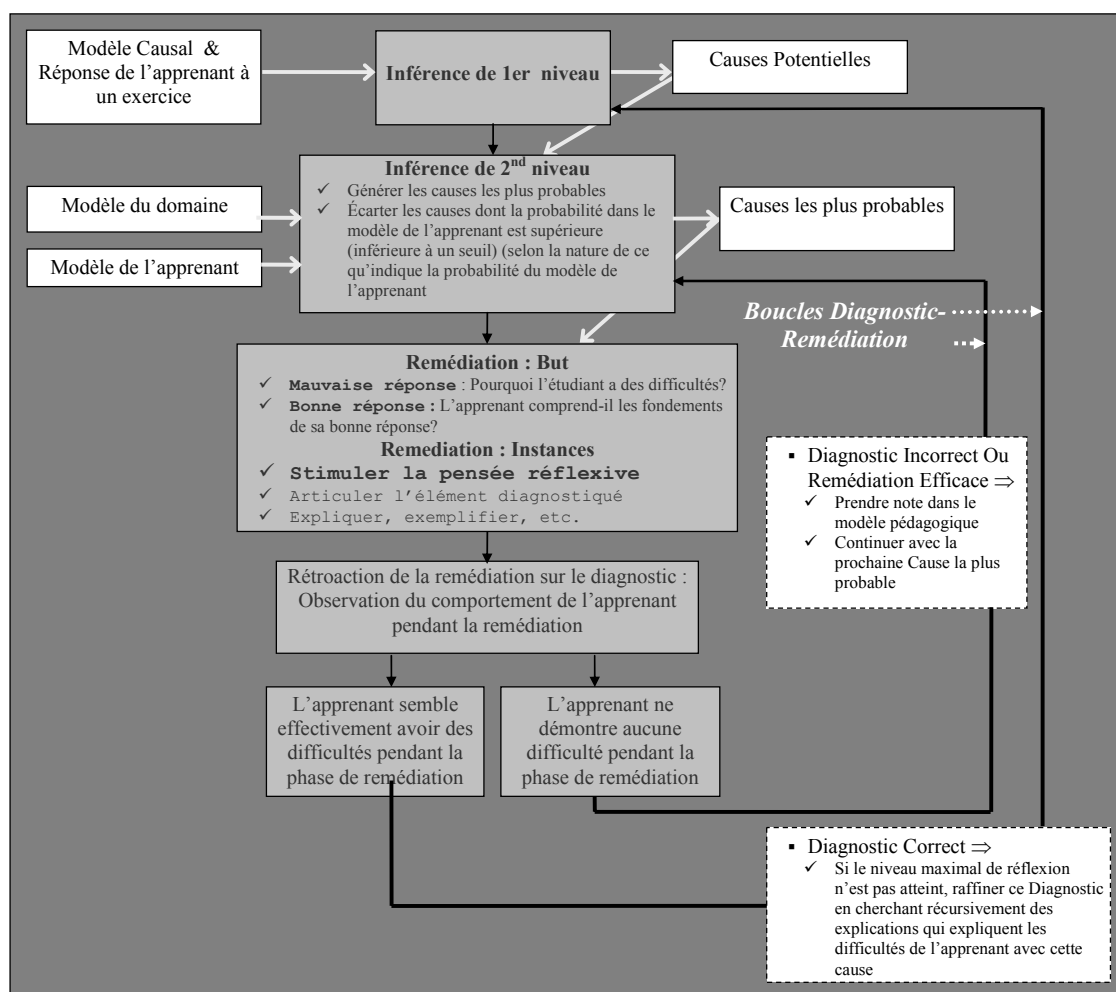


Figure IV.7. Formalisation de la boucle diagnostic-remédiation

4.5.1.1.1. La phase d'acquisition d'indices

La phase d'acquisition d'indices doit recueillir la réponse de l'apprenant (partielle ou finale). Le traitement de cette réponse est classique et n'a aucune particularité liée à l'approche bayésienne. La réponse est caractérisée (correcte, incorrecte ou en termes de caractéristiques prédéfinies et plus élaborées selon la nature de l'exercice).

4.5.1.1.2. *La phase d'inférence*

L'inférence s'échelonne sur deux niveaux dans cette approche de diagnostic. Au *premier niveau d'inférence*, la génération des causes utilise le modèle causal pour générer toutes les hypothèses potentielles qui pourraient avoir causé la réponse de l'apprenant. Au *second niveau d'inférence*, l'ensemble des hypothèses potentielles est raffiné pour ne retenir que les hypothèses les plus probables. L'algorithme d'inférence bayésienne appelé MPE (*Most Probable Explanation* ou *Belief Revision*) permet d'identifier ces hypothèses. L'interprétation de la MPE dépend de la signification des variables aléatoires associées au réseau bayésien formé par le modèle causal. Dans le cas des DC épistémiques en général, il s'agit de variables binaires associées à la capacité représentée par un noeud du réseau. À la Figure IV.6 par exemple (ci-dessus en page 121), lorsque les noeuds ont la valeur 1, cela signifie que la capacité associée est acquise, sinon, c'est la valeur 0. Par conséquent, les causes les plus probables sont les capacités dont les noeuds ont la valeur 0 dans l'état du réseau obtenu après avoir exécuté l'algorithme "*Belief Revision*". On peut davantage réduire l'ensemble des causes les plus probables en utilisant les informations du modèle de l'apprenant. Par exemple dans le cas d'un DC épistémique, on peut fixer un seuil δ : toutes les causes les plus probables C_p telles que $\text{Probabilité}[C_p] > \delta$ ne doivent pas être retenues.

4.5.1.1.3. *La boucle diagnostic-remédiation*

La remédiation porte sur les difficultés de l'apprenant qui ont été diagnostiquées. Le but de cette remédiation est d'aider l'apprenant à s'améliorer par rapport à ces difficultés. Mais le but est aussi de mieux comprendre les difficultés de l'apprenant en améliorant la qualité du DC initialement posé. Dans ce sens, l'apprenant doit être observé lors de cette remédiation afin de récupérer davantage d'information sur ce DC :

- Si l'apprenant éprouve effectivement des difficultés pendant la phase de remédiation, cela confirme le DC posé, tout en renforçant la nécessité de cette remédiation qui s'avère tout à fait appropriée
- Si l'apprenant n'éprouve aucune difficulté lors de la remédiation, cela indique que le DC initialement posé était incorrect ou encore que les difficultés de l'apprenant ont disparu entre temps. Dans ce cas, pour s'assurer d'examiner les difficultés de l'apprenant, le système peut sélectionner une autre explication candidate parmi les causes les plus probables qui avaient été identifiées dans le DC initial.

Exemple

Si l'on considère un exemple dans le domaine de la programmation logique: à la Figure IV.8, l'incapacité d'un apprenant à *appliquer* l'unification de deux termes composés pose le problème de diagnostic qui consiste à déterminer la cause de cette difficulté. Il s'agit d'un cas de diagnostic épistémique où le but est de déterminer les capacités du domaine avec lesquelles l'apprenant éprouve des difficultés. Selon l'illustration, trois causes sont potentielles au premier niveau d'inférence :

- l'incapacité d'unifier les arguments des vis-à-vis des deux termes
- l'incapacité de comprendre la structure d'un terme composé
- l'incapacité d'utiliser une variable liée

Au deuxième niveau d'inférence, si l'on applique l'inférence MPE au réseau bayésien formé par le noeud correspondant à la réponse de l'apprenant et les différentes causes susmentionnées. Par exemple, on peut considérer que la meilleure explication (obtenue par l'application de l'inférence MPE) de la réponse incorrecte de l'apprenant implique que:

- soit l'apprenant est incapable d'utiliser une variable liée
- soit l'apprenant est incapable de comprendre la structure d'un terme composé

Il s'agit donc des explications les plus probables. D'après l'algorithme de la Figure IV.7 (ci-dessus en page 126) on peut éliminer ces causes si la probabilité d'acquisition – indiquée dans le modèle de l'apprenant - des capacités correspondantes est supérieure à un seuil donné. Lorsqu'on obtient l'ensemble final des causes les plus probables de la réponse de l'apprenant, on peut engager une phase de remédiation spécifique à chacune d'elle.

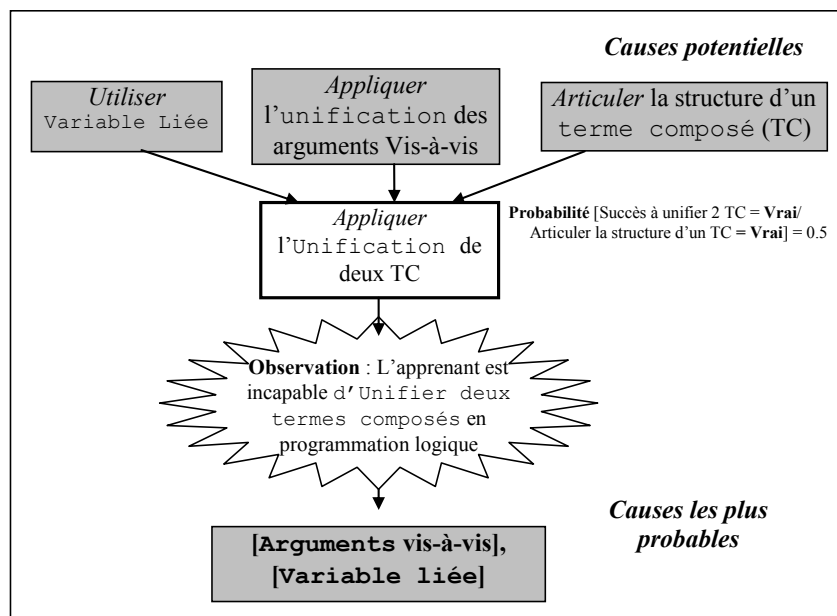


Figure IV.8. Diagnostic d'une incapacité à *appliquer* une unification en programmation logique

Dans les prochaines lignes, nous présentons un algorithme générique pour cette vision du processus de DC dans un STI.

4.5.2. Algorithme générique de DC

Le Tableau IV.1 présente l'algorithme générique de DC associé à l'approche bayésienne que nous venons de définir. Toutes les données utilisées dans ce tableau sont définies dans les tableaux de l'APPENDICE. B (Tableau B.1, Tableau B.2, Tableau B.3, Tableau B.4). Il s'agit d'une interprétation probabiliste de l'approche algorithmique "générer et tester" (Russel et Norvig, 2003). La génération des causes les plus probables et l'identification des causes les plus plausibles correspondent à la phase de génération des hypothèses. La boucle diagnostic-remédiation peut être interprétée comme la phase de test des hypothèses générées. Bien qu'une approche de DC similaire à la méthode "générer et tester" ait déjà été proposée (de Koning et al., 2000), l'originalité de notre contribution se reflète à deux niveaux dans la recherche AIED:

- Au niveau d'une perspective du DC à travers une boucle entre le diagnostic-remédiation;

- Au niveau de l'implémentation de cette perspective du DC à travers une librairie de programmes génériques et réutilisables, favorisant ainsi sa vulgarisation dans la mise en œuvre du DC dans les STI.

Tableau IV.1. Un algorithme générique de DC basé sur l'inférence bayésienne

Programmes supportant l'implémentation du DC	Pseudo-code en format XML
bayesian_generic_cognitive_diagnosis	<pre> <command_list> <If> (studentAnswer != null); <Then> evaluateAnswer; <If> (curentEvaluation); <Then> <If> (curentDiagnosisMode == examineDiagnosisAnyway); <Then> inference_level1; </Then> <Then> loop_diagnosis_reflexion; </Then> </If> </Then> </If> </Then> </If> <If> (!curentEvaluation); <Then> inference_level1; </Then> <Then> inference_level2; </Then> <Then> loop_diagnosis_reflexion; </Then> </If> <command>learnerModel.updateModel(curentEvaluation)</command> </command_list> </pre>
inference_level1	<i>generateCauses;</i>
inference_level2	<i>generateProbableCauses;</i>
generateCauses	<pre> <command_list> <command> curentCauses = causalModel.getCauses(curentEvaluation); </command> <For each> cause in curentCauses <command> domainModel.setObservedNodes(curentEvaluation, cause); </command> </For each> </command_list> </pre>
generateProbableCauses	<pre> <command_list> <command> mostProbableCauses = identifyMPES()</command> <For each> cause in mostProbableCauses; <If> (learnerModel.CognitiveState(cause) > Threshold) <Then> mostProbableCauses.remove(cause); </Then> </If> </For each> </command_list> </pre>

Programmes supportant l'implémentation du DC	Pseudo-code en format XML
identifyMPEs	<pre> <command_list> <command> domainModel.beliefRevise(); <For each> cause in curentCauses <If > (cause.getExplanatoryValue == 0) <Then> mostProbableCause.add(cause) </Then> </If> </For each> </command_list> </pre>
Loop_diagnosis_remediation	<pre> <command_list> <For each> cause in mostProbableCauses; <command> reflexionAspects = domainModel.BayesNet.getParents(cause); </command> <For each> aspect in reflexionAspects <If> <Else> If (diagnosisOn Cause is_not_Confirmed) pedagogicalNote(cause, notConfirmed); </Else> </pre>

4.5.2.1.1. Contribution d'une implémentation bayésienne de la boucle diagnostic-remédiation

Aucune recherche relative au DC dans le domaine AIED n'explique la relation entre le DC et toute forme de remédiation immédiate. Dans cette thèse, notre contribution sur ce chapitre découle du fait que notre considération explicite de cette relation est matérialisée à travers une boucle co-déterminante. Selon l'approche bayésienne, l'originalité de cette contribution est qu'elle s'appuie sur un background théorique fondé, en l'espèce la méthode d'inférence bayésienne «Belief Revision». Cette approche d'inférence permet d'identifier de manière théorique les explications les plus probables d'une observation.

4.5.2.1.2. Contribution d'une librairie de programmes réutilisables pour le DC

L'interprétation de la tâche de diagnostic à travers un algorithme générique tel que l'approche "générer et tester" fournit une base générale pour son implémentation dans un STI. Les différentes phases de cet algorithme fournissent une structure de base qu'on peut organiser dans un ensemble de classes génériques constituant un module de DC (approche valide dans une approche d'implémentation d'un STI orientée vers la programmation par objets). La structure de ces classes est la suivante:

- une classe pour le modèle du domaine, avec une représentation probabiliste des éléments de ce domaine (APPENDICE. B: Tableau B.1).

- une classe pour le modèle de l'apprenant avec une représentation probabiliste de son état cognitif (APPENDICE. B: Tableau B.2);
- une classe pour le modèle causal (APPENDICE. B: Tableau B.3);
- une classe pour l'algorithme générique de DC (APPENDICE. B: Tableau B.4).

L'originalité de cette contribution dans le domaine AIED découle du fait qu'une telle librairie de classes est réutilisable. En effet, pour chaque problème particulier dans STI particulier, chacun de ces modèles doit être instancié en y intégrant les données du problème. Par exemple pour un DC épistémique dans le domaine de la programmation logique, plus particulièrement, la tâche d'unifier deux termes composés, les observations sont des objets correspondant aux réponses de l'apprenant; la structure du modèle causal est un réseau bayésien où toutes les sous capacités de la capacité «unifier deux termes composés» sont associées à des noeuds parents du noeud correspondant à cette capacité.

4.6 Conclusion

Ce chapitre a permis de formaliser la boucle diagnostic-remédiation préconisée dans cette thèse au niveau de la conception et de la mise en oeuvre du DC dans un STI. Cette formalisation a permis de mettre en évidence les relations entre le module de DC et les autres modules d'un STI. Elle a surtout permis de représenter le processus de DC de manière formelle, en tenant compte des facteurs d'incertitude, des implications de la relation co-déterminante entre le diagnostic et la remédiation en général, ce à travers l'adoption d'une approche d'inférence bayésienne. Il s'agit d'une des principales contributions de cette thèse sur le plan informatique.

En effet, comme on le verra au prochain chapitre, cette formalisation facilite l'implémentation de la boucle diagnostic-remédiation à travers une librairie de programmes génériques, intégrables dans le module de DC d'un système tuteur. Le prochain chapitre présente la mise en oeuvre de la formalisation présentée dans ce chapitre, dans un prototype de système tuteur pour la programmation logique: Prolog-Tutor.

Chapitre V

INTÉGRATION D'UNE PERSPECTIVE PRO-PÉDAGOGIQUE DU DC DANS UN STI : PROLOG-TUTOR

5.1 Introduction

Cette thèse préconise une perspective du DC dans un STI axée sur la considération des paradigmes de cognition et sur l'intégration de la remédiation au DC à travers une boucle diagnostic-remédiation, matérialisée dans ce cas par une boucle entre le DC et la stimulation de la pensée réflexive (Chapitre III). Cette perspective a été formalisée à travers un algorithme générique de DC basé sur les principaux algorithmes d'inférences bayésienne (Chapitre IV).

Ce chapitre porte sur l'implémentation concrète de cet algorithme de DC (basée sur la technologie orientée objet avec le langage `Java`) dans un STI pour la programmation logique: Prolog-Tutor (Tchetagni, Nkambou, et Bourdeau, 2005a). L'implémentation du système informatique sous-jacent à un STI implique généralement l'élaboration de quatre composantes: le module du domaine incorporant une représentation des connaissances du domaine; le module de l'apprenant incorporant une représentation de l'état cognitif de ce dernier; le module pédagogique incorporant une représentation de l'expertise pédagogique, y

compris celle du diagnostic cognitif (DC); le module de communications incorporant les outils et les mécanismes qui permettent la communication entre le système informatique et l'apprenant humain. Dans Prolog-Tutor, l'accent a été mis sur le module pédagogique qui assure la mise en oeuvre de l'algorithme de DC lorsqu'un apprenant résout un problème. Étant donné que la dynamique de ce module implique les trois autres composantes d'un STI, la description de ces composantes (section 5.2) précède celle de l'algorithme de DC dans Prolog-Tutor (section 5.3) dans ce chapitre.

5.2 Implémentation générale de Prolog-Tutor

L'essentiel de l'effort d'implémentation de Prolog-Tutor s'est focalisé sur le module pédagogique, et plus particulièrement sur le DC. Les autres composantes ont cependant une certaine importance, étant donné que le DC s'appuie sur les informations qu'elles contiennent. Cette section décrit les composantes et la dynamique générale du système informatique sous-jacent à Prolog-Tutor.

5.2.1. Le module du domaine

Le module du domaine dans Prolog-Tutor incorpore un modèle des connaissances à proprement parler, une représentation des relations entre des objectifs d'apprentissage, des activités d'apprentissage et des ressources d'apprentissage et, une banque de ressources d'apprentissage.

5.2.1.1. Composantes du module du domaine

5.2.1.1.1. Le modèle des connaissances

Dans Prolog-Tutor les capacités associées aux connaissances sur la programmation logique sont modélisées à travers un réseau d'influence entre ses différents éléments, une forme particulière de représentation sémantique des connaissances (section 3.3.1.2.1.). Le graphe correspondant est interprété comme un réseau bayésien (Chapitre IV): chaque arc représente une relation causale entre le nœud correspondant à son origine et celui correspondant à sa destination; chaque arc est étiqueté avec un nombre qui exprime la probabilité du nœud destination, sachant qu'un événement au nœud source est survenu. Dans ce cas, l'évènement est l'"acquisition" de la capacité associée à un nœud. La Figure V.1 illustre un extrait de ce réseau bayésien: on peut voir que la probabilité d'acquisition de la capacité à appliquer

l'Unification sachant que la capacité d'identifier un Terme Composé est acquise est de 0.3. La classe générique `PrologCapabilities.java` a été utilisée pour implémenter ce réseau bayésien.

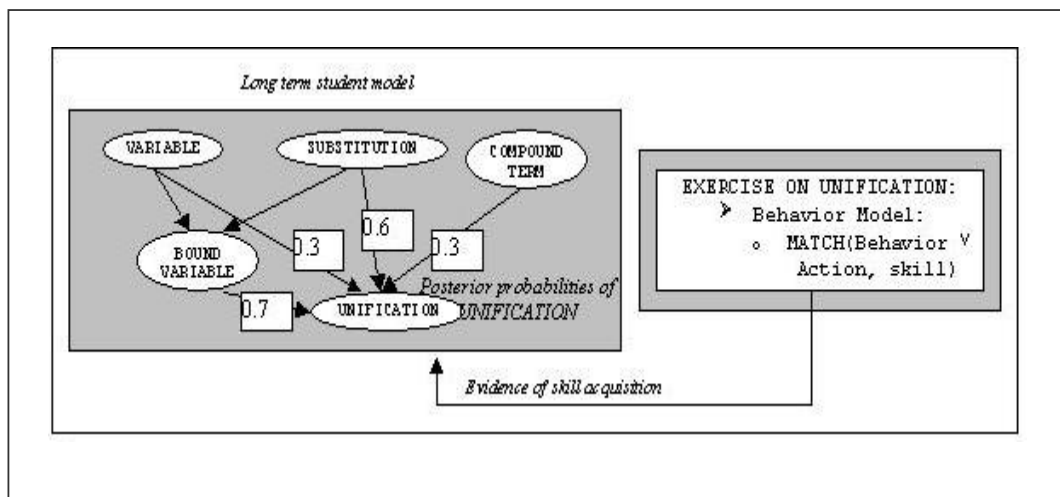


Figure V.1. Représentation bayésienne d'une partie des capacités de Prolog-Tutor

La représentation bayésienne présente deux attraits (Chapitre IV): l'inférence bayésienne (*Belief Updating*), l'abduction bayésienne (*Belief Revision* ou *Most Probable Explanation*). L'inférence bayésienne permet la propagation automatique de l'effet d'une action de l'apprenant sur l'interprétation de son niveau de maîtrise par rapport à l'ensemble des connaissances représentées (Nkambou et Tchétagni, 2002; Tchétagni et Nkambou, 2002). L'abduction bayésienne permet de poser des hypothèses sur l'état des connaissances de l'apprenant, qui expliquent le mieux une action de l'apprenant, en tant que réponse à une question relative à une connaissance représentée (Nkambou et Tchétagni, 2004). Les autres types de représentations de connaissances ne sont pas indiqués dans le contexte particulier de Prolog-Tutor. En effet, une représentation explicite ou ontologique du contenu de la programmation logique n'est pas appropriée dans la mesure où l'apprentissage se fait à travers des exercices (et non à travers l'exploration d'une représentation déclarative du contenu). De plus, une représentation à base de règles de production n'est pertinente que dans des situations de résolution de problèmes. La programmation logique est un domaine au contenu "logique" (bien structuré) et peu dense. La résolution de la plupart des exercices est

donc généralement directe dans ce sens qu'il est peu probable qu'un processus de "résolution de problème" - au sens cognitif – soit nécessaire à leur réalisation.

5.2.1.1.2. *Relations entre objectifs, activités et ressources d'apprentissage*

Au-delà d'une représentation formelle relative à certaines connaissances sur la programmation logique, le module du domaine intègre d'autres composantes reliées aux buts de l'enseignement. Il s'agit d'une représentation des objectifs d'apprentissage et d'une banque de ressources. Les objectifs d'apprentissage sont représentés à travers l'association entre une capacité selon la terminologie de Bloom (Bloom et al., 1956) et les éléments du modèle des connaissances (Nkambou, Frasson, et Gauthier, 1996). Les ressources représentent concrètement (mais non formellement) le contenu du domaine. Ces ressources correspondent à des exercices à résoudre, ou à d'autres contenus didactiques tels que: des définitions, des exemples, des explications.

Les ressources qui sont des exercices sont étiquetées avec deux informations: les capacités qui sont visées à travers leur résolution et leur niveau de difficulté. Le niveau de difficulté est une caractéristique qui permet de générer des exercices qui représentent le niveau d'acquisition de l'apprenant par rapport aux connaissances associées à l'exercice en question.

5.2.1.2. Dynamique du module du domaine

La dynamique du module du domaine est déterminée par l'exploitation de ses différentes composantes par d'autres modules dans un STI.

5.2.1.2.1. *Utilité du modèle des connaissances*

Le modèle des connaissances sert de point de repère pour la représentation des connaissances de l'apprenant puisque ce dernier couvre tous les éléments de ce modèle en cela qu'il leur associe une probabilité d'acquisition.

5.2.1.2.2. *Utilité d'une représentation XML des relations entre objectifs, activités et ressources d'apprentissage*

La représentation des objectifs d'apprentissage permet une organisation formelle des sessions d'apprentissage dans Prolog-Tutor. En effet, suivant toujours l'approche CREAM, un objectif d'apprentissage doit être atteint par l'exécution d'une ou plusieurs activités d'apprentissage. Chaque activité d'apprentissage se concrétise à travers la consultation et l'interaction de

l'apprenant avec des ressources d'apprentissage. Les ressources statiques concrétisent les activités d'apprentissage où l'apprenant absorbe un contenu relatif à la capacité ou à l'objectif visé à travers cette activité. Les ressources dynamiques, également appelées "ressources intelligentes" concrétisent les activités d'apprentissage "pratiques" où l'apprenant interagit avec le système de manière plus soutenue, pour résoudre un exercice (APPENDICE. C, Figure C. 1).

Dans CREAM, les sessions d'apprentissage sont dynamiquement planifiées sur la base d'un réseau. Ce réseau aussi appelé CKTN (*Curriculum Transition Network*) exprime les relations entre les connaissances, les objectifs d'apprentissages, et les ressources disponibles. Dans Prolog-Tutor, la planification est statique. Les relations entre les objectifs d'apprentissage, les activités d'apprentissage et les ressources sont spécifiées manuellement. Ces relations sont représentées dans un format XML qui est interprété par un planificateur statique (APPENDICE. C, Figure C. 1). Deux opérations principales correspondent à un algorithme de planification statique:

- (1) Il effectue une analyse syntaxique (*parsing*) de la représentation XML de la structure d'une session d'apprentissage;
- (2) Ensuite, il identifie chaque type d'élément et le représente de manière adéquate à l'interface du système;
- (3) Enfin, il associe chacune des ressources qui apparaissent dans la représentation XML à l'objectif d'apprentissage adéquat, afin qu'elle puisse être déclenchée lorsque cet objectif est abordé.

5.2.2. Le module de l'apprenant

Dans un STI, le module de l'apprenant s'intéresse à la représentation de l'état cognitif de l'apprenant: l'état de ses connaissances (Collins, Greer, et Huang, 1996; Martin et VanLehn, 1995), la nature ou la substance de son raisonnement ou la nature de ses émotions pendant une activité d'apprentissage (Faivre, Nkambou, et Frasson, 2002). Le module de l'apprenant intègre aussi la représentation d'informations plus générales à propos de l'apprenant. Il s'agit de ses préférences, de l'historique de ses interactions avec le système (les problèmes qu'il a résolus, la manière dont ils ont été résolus ainsi que les résultats de la performance à ces problèmes, les parties du curriculum qu'il a explorées, le temps passé dans chaque section,

etc). Le choix des techniques de représentation qui sont mises à contribution pour représenter les informations dans le module apprenant dépend également de l'utilisation qui en est faite par la suite. Prolog-Tutor s'intéresse particulièrement sur l'état des connaissances de l'apprenant à travers le diagnostic épistémique.

5.2.2.1. Les composantes du module de l'apprenant

Le module de l'apprenant dans Prolog-Tutor comporte trois composantes principales : une représentation de l'état des connaissances de l'apprenant à long terme – implémentée à travers la classe générique **LearnerModel_Generic_DC.java**; une représentation de l'état des connaissances de l'apprenant à court terme; une trace d'apprentissage.

5.2.2.1.1. L'état des connaissances à long terme

L'état des connaissances de l'apprenant à long terme représente son état de connaissances tel qu'interprété au fil de toutes ses interactions avec le système sous-jacent à Prolog-Tutor. Le modèle de l'apprenant dans Prolog-Tutor est un *overlay* du *modèle* des connaissances du domaine (Chapitre III). Ce modèle exprime pour chaque élément de connaissances représenté dans le modèle des connaissances du domaine, une probabilité d'acquisition (Figure V.2).

En fait, le choix de l'approche de représentation d'un modèle de l'apprenant est influencé par celle qui est utilisée pour représenter le domaine puisque l'apprentissage porte sur ce qui est représenté. De ce fait, représenter l'état des connaissances de l'apprenant comme une liste de probabilités découle de la représentation du domaine des connaissances comme un réseau bayésien (Chapitre IV). Ce choix a été renforcé par le fait qu'une représentation Bayésienne permet la mise à jour automatique du modèle de l'apprenant, grâce à l'algorithme de mise à jour bayésienne (*belief updating*). Pour chaque élément de connaissance du domaine, l'état des connaissances à long terme de l'apprenant indique également :

- (1) son pourcentage de succès chaque fois qu'il a été exposé à un exercice où cet élément de connaissance devait contribuer à la construction de la solution;
- (2) le pourcentage de fois où il a été directement exposé à l'élément de connaissance;

Ces informations permettent de relativiser la probabilité indiquée pour chaque élément de connaissance selon qu'elle provienne d'une observation directe des actions de l'apprenant ou d'une inférence bayésienne (ce qui équivaldrait à une observation indirecte).

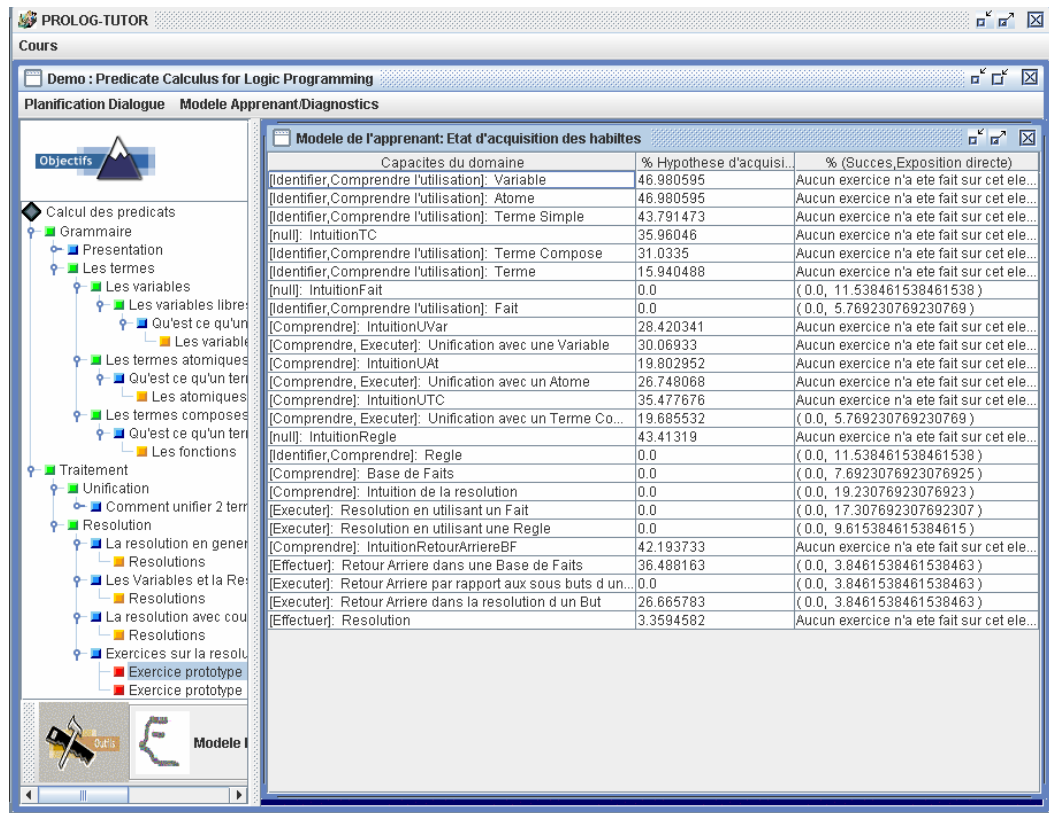


Figure V.2. Modèle de l'apprenant dans Prolog-Tutor

5.2.2.1.2. L'état des connaissances à court terme

L'état des connaissances à court terme est analogue à l'état des connaissances à long terme: il consigne les probabilités d'acquisition de l'apprenant par rapport à chaque élément de connaissance, à la lumière de la session d'apprentissage courante. À la Figure V.3 par exemple, la probabilité d'acquisition de la capacité "Exécuter l'Unification avec Terme Composé" est de 0% pour la session d'apprentissage *en cours*, alors que pour le long terme, elle atteint presque 17.5%. Cela est dû au fait que dans la session d'apprentissage courante, toutes les réponses de l'apprenant à des questions ou à des exercices dont la résolution fait intervenir cette capacité étaient incorrectes. Le modèle à court terme n'est pas actuellement exploité dans Prolog-Tutor, mais son but est de permettre de modéliser des phénomènes tels que l'oubli ou le fait de deviner les réponses (lorsque la différence entre la

probabilité à court terme et la probabilité à long terme sur un élément de connaissance est significative).

5.2.2.1.3. La trace d'apprentissage de l'apprenant

La trace d'apprentissage de l'apprenant consigne une représentation de toutes les actions qu'il a effectuées pendant son interaction avec le système, au cours d'une session d'apprentissage (Figure V.3). La trace d'apprentissage est sauvegardée dans un fichier texte. Son objectif est de permettre une sauvegarde physique des données d'interaction de l'apprenant, en dehors de l'état des connaissances de l'apprenant.

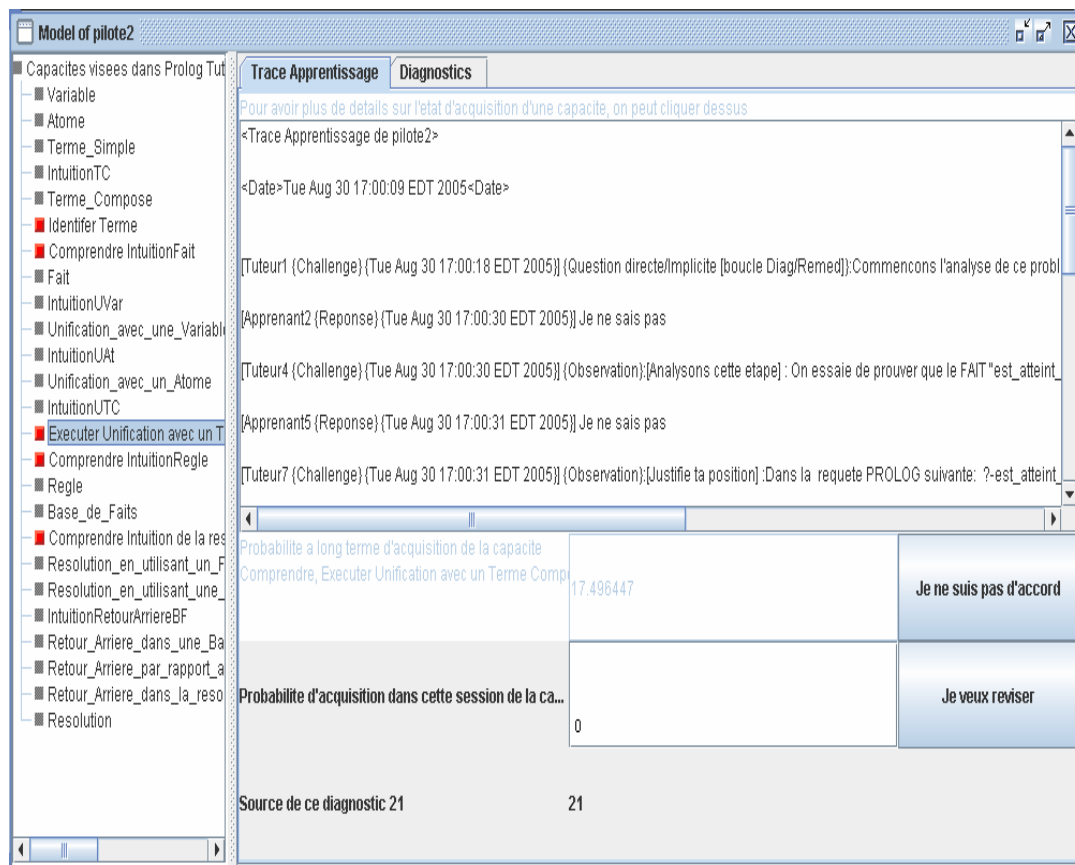


Figure V.3. Prolog-Tutor: État des connaissances de l'apprenant à court terme

5.2.2.2. La dynamique du module de l'apprenant

Contrairement au module du domaine, le module de l'apprenant n'a pas une dynamique interne significative dans Prolog-Tutor. Il est une source et un dépôt d'informations pour et par le module pédagogique (section 5.3). Le module pédagogique utilise les probabilités

d'acquisition d'un apprenant pour prendre des décisions pédagogiques dans le cours d'une session d'apprentissage, en l'occurrence, lors de l'identification des causes les plus probables pendant le diagnostic d'une réponse de l'apprenant (section 4.5). Le module pédagogique diagnostique les capacités acquises et non acquises par l'apprenant, pendant qu'il résout un exercice. Chaque exercice ou chaque action possible de l'apprenant lors de la réalisation d'un exercice est explicitement associée à une capacité (elle-même associée à un élément de connaissance du domaine) à travers un *modèle causal* (section 4.4). Dans Prolog-Tutor, le modèle causal est intégré au module pédagogique et grâce à ce modèle, chaque action/réponse de l'apprenant pendant un exercice peut être répercutée dans son modèle, à travers l'inférence bayésienne (section 4.2).

5.2.3. Le module de communication

Un STI est un artéfact d'IA, d'où l'importance des enjeux ergonomiques dans son implémentation. Le module de communication est la composante d'un STI qui assure la réalisation des échanges entre l'apprenant et le système informatique sous-jacent à cet STI. La principale composante du module de communication est une interface de communication. Lors de la conception d'une interface de STI, on ne peut considérer l'application des théories/modèles de communication humain/humain. En effet, les termes d'échanges qui sont créés entre l'humain et la machine sont fondamentalement différents de ceux créés entre deux personnes (Wenger, 1987a). Dans ce sens, des théories et des modèles adéquats de communication humain/machine ont émergé de la recherche en IA et en communication. Plusieurs travaux de recherche ont contribué à l'élaboration de tels modèles dans divers types de STI (Dufresne, 2001) et dans la perspective plus générale des interfaces humain/machine pour tout système informatique (Scapin et Bastien, 1997).

5.2.3.1. Composantes et dynamique du canal de communication

L'interface usager¹ d'un STI doit intégrer au minimum un modèle de discours sur lequel les échanges sont basés. Elle peut aussi offrir un environnement d'exploration des connaissances: un pôle où l'apprenant pourrait explorer les connaissances relatives au

¹ Il ne faut pas confondre l'interface usager de Prolog-Tutor avec le terme "interface" propre à la programmation orientée objet, utilisée pour implémenter ce système. Dans cette section, ce terme réfère à l'interface graphique de Prolog-Tutor, qui sert de medium entre l'apprenant et le système informatique sous-jacent.

domaine (APPENDICE. C, Figure C. 2). Dans ce cas, une bonne partie de la responsabilité de l'apprentissage est laissée à l'apprenant, favorisant par la même occasion la réflexion. Le discours dans Prolog-Tutor est modélisé à travers des interfaces graphiques structurées qui permettent à l'apprenant de répondre à des questions précises provenant du système (APPENDICE. C : Figure C. 3).

Prolog-Tutor communique avec l'apprenant seulement lorsque celui-ci résout un exercice. Pour cela, il doit pouvoir: poser des questions à l'apprenant et interpréter ses réponses. Prolog-Tutor dispose de deux types d'interfaces: une interface principale de l'apprenant (Figure V.4) – implémentée dans la classe `UserInterface.java` - et un ensemble d'interface secondaires qui ne sont visibles qu'en cas d'un besoin particulier (APPENDICE. C: Figure C. 6, Figure C. 7, Figure C. 8).

5.2.3.1.1. *L'interface principale : composantes et dynamique*

Les principaux exercices de Prolog-Tutor sont résolus à travers un dialogue tutoriel entre l'apprenant et le module pédagogique (section 5.3). Au tout début, un problème est exposé à l'apprenant et pour le résoudre, ce dernier devra répondre à un ensemble de sous-questions associées au problème. La communication dans ce dialogue tutoriel se déroule selon la structure suivante (Figure V.5):

- (1) **Le tuteur peut poser une question à l'apprenant.** Étant donné que Prolog-Tutor ne fait pas de traitement de langage naturel, les questions sont assez restreintes pour permettre une plage limitée de réponses. La provenance de ces questions est expliquée dans la description du module pédagogique (section 5.3);
- (2) **L'apprenant répond à la question en** utilisant les composantes de l'interface. Selon la question du tuteur, il se peut que l'apprenant utilise des interfaces secondaires;
- (3) En tout temps, l'apprenant peut poser des questions prédéfinies au tuteur.

The screenshot displays the Prolog-Tutor interface, which is divided into several panels:

- Top Panel:** Shows the application title "PROLOG-TUTOR" and the current course "Cours". Below it, a window titled "Demo : Predicate Calculus for Logic Programming" is open, showing "Planification Dialogue" and "Modele Apprenant/Diagnostics".
- Left Panel (Objectifs):** A tree view of learning objectives including "Calcul des predicats", "Grammaire", "Traitement", "Unification", "Resolution", and various sub-topics like "La resolution en gener", "Les Variables et la Res", "La resolution avec cou", and "Exercices sur la resolu".
- Main Window (Prolog-Tutor Resol):**
 - Visualisations:**
 - ENONCE:** A problem statement in French: "=> Supposons qu'on a la requete suivante dans un interpreteur SICSTUS PROLOG: ?- est_atteint_de(pierre,M). => Quel est le Resultat de la Preuve du BUT qui correspond cette requete?". It asks the user to construct a proof and respond to system questions.
 - BASE DE CONNAISSANCES:** A list of facts in Prolog: `est_atteint_de(X,M):-symptome(X,S),cause(S,M).` followed by several `cause` and `symptome` facts.
 - Outils:** A sidebar with buttons for "Constructeur", "Situation Dialogue/Arbre Resolution", and "Terminer Session".
 - Dialogue Box (M*):**
 - Recevoir Explication:** A red button to request an explanation.
 - Tuteur's Message:** "[Tuteur] [Analysons cette etape] : La REGLE: 'est_atteint_de(X,M):- symptome(X,S), cause(S,M)' DOIT etre utilisee pour resoudre le BUT correspondant a la requete ?- est_atteint_de(pierre,M). Cliquez sur le BOUTON selon ton avis." Below this are "D'accord" and "En desacord" buttons.
 - Learner's Response:** "[Learner] Je ne sais pas".
 - Feedback:** "[Feedback] Ok, tu vas expliquer sur quoi ta reponse se base et comment tu l'as trouvee".
 - Justification:** "[Tuteur] [Justifie ta position] :Ok, si tu n'est pas d'accord, cela signifie que les FAITS 'est_atteint_de(X,M)' et 'est_atteint_de(pierre, M)' ne sont pas UNIFIABLES. Pourquoi ne sont ils pas unifiables?". Below is a "Repondre" button.
 - Learner's Second Response:** "[Learner] Je ne sais pas".
 - Reponses/Questions. Usager:** A sidebar with buttons for "Voir Domaine", "Sais Pas", "Resultat =", "Reponse =", "Pourquoi? (Situation Dialogue/Question)", "Quoi?", and "Comment?".

Figure V.4. Prolog-Tutor: Interface Principale

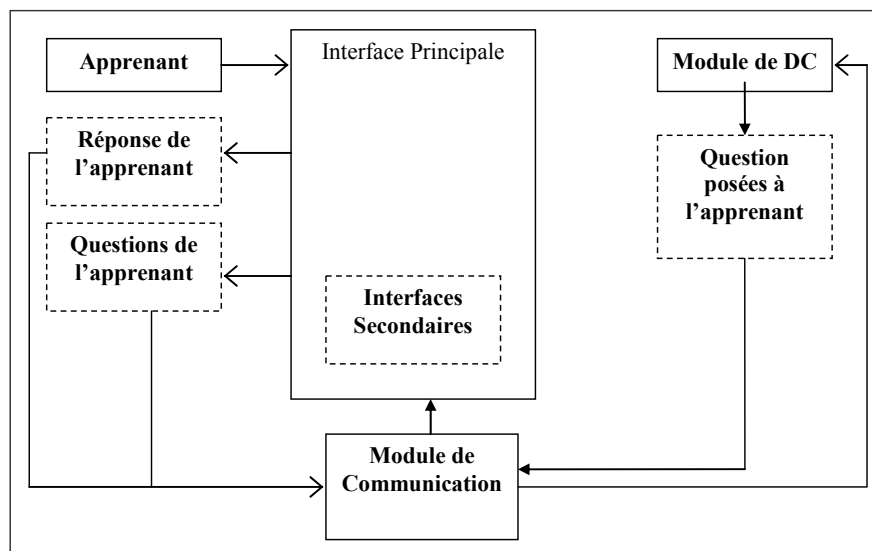


Figure V.5. Dynamique de communication dans Prolog-Tutor

Pour répondre à une question du tuteur, l'apprenant a trois possibilités par défaut :

- Il fournit une réponse directe en utilisant le bouton intitulé "Réponse" (APPENDICE. C: Figure C. 3, Figure C. 4) ; ce mode est utilisé pour permettre à l'apprenant de formuler une réponse par rapport à une question générale;
- Il fournit le résultat d'une unification en Prolog en utilisant le bouton intitulé "Résultat" (APPENDICE. C: Figure C. 4): les principaux exercices de Prolog-Tutor correspondent à la résolution d'un but, de nombreuses opérations d'unification sont nécessaires. Ce bouton permet de spécifier les résultats d'une unification;
- Il indique qu'il ignore la réponse à la question du tuteur en utilisant le bouton intitulé "Je ne sais pas".

En plus de ces trois modes par défaut, l'apprenant devra utiliser des interfaces secondaires pour répondre à certaines questions. En fait, chaque question posée par le module pédagogique est suivie d'instructions indiquant à l'apprenant quelle composante de l'interface principale il doit utiliser pour y répondre. Lorsque cette indication n'apparaît pas, c'est que l'apprenant doit utiliser le bouton bleu qui suit la question et qui est intitulé "Répondre".

L'apprenant peut poser trois types de questions au module pédagogique. Il peut demander la pertinence ou une justification par rapport à une question qui lui est posée et dans ce cas, le bouton de l'interface principal intitulé "Pourquoi?" doit être utilisé. Il peut indiquer qu'il ne comprend pas une question et dans ce cas, le bouton de l'interface principal intitulé "Quoi?" doit être utilisé. Il peut indiquer qu'il ne sait pas comment utiliser l'interface pour répondre à la question qui lui est actuellement posée avec le bouton "Comment?".

Donner la possibilité à l'apprenant de demander la justification (la raison d'être d'une question) est pertinent dans le cadre d'un dialogue tutoriel. En effet, ce mode de communication a la particularité d'entraîner dans certains cas l'éloignement de l'objectif ou du problème original. La réponse du tuteur dans ce cas a pour but de situer la question qui est actuellement posée par rapport aux implications ou aux enjeux de l'objectif original de l'exercice. L'importance des deux autres types de questions est évidente dans un contexte pédagogique.

Dans cette veine, l'apprenant peut consulter en tout temps un arbre de résolution en utilisant le bouton intitulé "Situation Dialogue/Arbre de résolution". Dans ce cas, le module de communication génère une représentation graphique du processus de résolution de but qui est entrain d'être effectué dans l'exercice en cours (APPENDICE. C: Figure C. 5). Cette représentation indique à l'apprenant les différentes phases de construction de la solution qu'il a réalisées jusqu'à ce moment. Il s'agit d'un moyen de communication dont l'objectif est de maintenir le lien entre l'état actuel du dialogue tutoriel et le problème original.

En plus de poser directement des questions au système, un apprenant peut examiner en tout temps les capacités du domaine dont il a besoin pour répondre à la question qui lui est actuellement posée (APPENDICE. C: Figure C. 2). Pour cela, il lui suffit d'utiliser le bouton intitulé "Voir Domaine" dans l'interface principale. Dans ce cas, le module de communication génère une représentation hiérarchique du domaine de Prolog. Les enfants d'un nœud doivent être interprétés comme des connaissances préalables ou composantes de celle correspondant à ce nœud. Dans cette représentation, le module de communication fait ressortir les capacités dont l'apprenant a besoin pour répondre à la question en les soulignant d'une coloration bleue. De plus, en cliquant sur cette l'étiquette de cette capacité, l'apprenant peut recevoir une explication sommaire sur la substance de cette dernière.

5.2.3.1.2. *Les interfaces secondaires*

Le module de communication de Prolog-Tutor a aussi la possibilité de générer des interfaces dites secondaires. Les interfaces secondaires sont celles qui permettent à l'apprenant de répondre à des questions spécifiques. Certaines questions peuvent d'ailleurs nécessiter l'utilisation combinée de plusieurs interfaces. Prolog-Tutor intègre de nombreuses interfaces secondaires dont les plus importantes permettent à l'apprenant de:

- (1) Répondre à une question à partir de plusieurs choix formulés en langage naturel (APPENDICE. C: Figure C. 6);
- (2) Prouver explicitement pourquoi une unification ne devrait pas réussir (APPENDICE. C: Figure C. 7);
- (3) Prouver explicitement un sous-but, y compris dans une situation où un retour-arrière doit être effectué (APPENDICE. C: Figure C. 8);

5.3 Le module pédagogique de Prolog-Tutor: le diagnostic cognitif

Le module pédagogique est la partie la plus développée de Prolog-Tutor et elle est centrée autour du DC. En s'appuyant sur l'algorithme générique de DC décrit au Chapitre IV, cette section présente les composantes et la dynamique du module pédagogique de Prolog-Tutor.

5.3.1. Les composantes du module pédagogique

Au Chapitre IV, un algorithme générique de DC a été proposé en harmonie avec la perspective de DC préconisée dans cette thèse et explicitée au Chapitre 3. Selon cet algorithme, le diagnostic dans un STI se base sur trois éléments principaux: le modèle du domaine, le modèle de l'apprenant et *un modèle causal*. Dans Prolog-Tutor, le modèle causal est intégré dans le module pédagogique et il est matérialisé par les dialogues tutoriels.

De plus, si l'on considère les principales phases du DC selon l'algorithme générique présenté au Chapitre 4 (Tableau IV.1, en page 130), on constate que certains objets seront nécessaires pour leur réalisation. En effet, la phase d'acquisition d'indices est supportée dans Prolog-Tutor par un *analyseur de réponses*. Les phases d'inférences et de boucle entre le diagnostic et la pensée réflexive (boucle diagnostic-réflexion) sont assurées dans Prolog-Tutor par un *Profiler*.

Outre les composantes liées au processus de DC proprement dit, le module pédagogique de Prolog-Tutor inclut également un *gestionnaire de dialogue* pour gérer le flux

des échanges entre le système sous-jacent et l'apprenant, et une *trace des actions et décisions pédagogiques* posées lors d'un dialogue tutoriel.

5.3.1.1. Le modèle causal dans Prolog-Tutor: les dialogues tutoriels

Comme on a pu le constater au chapitre précédent, le modèle causal associé à un exercice ou à une activité d'apprentissage dans un STI doit relier un comportement de l'apprenant à un élément du domaine de connaissances. Le comportement de l'apprenant est fortement relié au contenu de l'exercice (l'objectif de la question qui y est posée) et à la nature des capacités qui sont nécessaires à sa réalisation. Pour cette raison, le modèle causal est fortement relié au contenu du domaine associé à un STI. Par conséquent, son implémentation en tant que telle ne peut être générique. On peut tout de même préserver cette propriété en utilisant un objet de type *interface* dans une technologie de programmation orientée objet. Une interface énonce les propriétés et les fonctions d'un objet. Un objet implémente une *interface* lorsque sa définition est un programme correspondant aux attributs et aux fonctions de cette *interface*. Dans Prolog-Tutor, le modèle causal implémente une *interface* définie dans la classe **Context_Causal.java**. Cette interface inclut les attributs et fonctions génériques suivants: `current_learner_system_interaction` (attribut définissant la réponse de l'apprenant pendant l'actuelle interaction avec le système); `get_causes()` (permet de récupérer les causes ou explications potentielles de la réponse de l'apprenant) et `link_interaction_causes` (lors de la construction du modèle causal, permet de relier des réponses de l'apprenant - à des moments précis de l'exercice - avec des causes potentielles) (fonctions).

Les exercices implémentés dans Prolog-Tutor se réalisent à travers un dialogue tutoriel entre l'apprenant et le module pédagogique. Le choix de ce mode d'interaction a été influencé par le fait que sa nature interactive implique l'apprenant dans son propre diagnostic, ce qui favoriserait *a priori* la stimulation de la pensée réflexive (Dimitrova, 2003a). Deux types de *dialogues tutoriels* sont implémentés dans Prolog-Tutor: les dialogues tutoriels génériques et les dialogues tutoriels dont la structure a pour but de favoriser explicitement la pensée réflexive chez l'apprenant. Il est important de prendre connaissance de la structure et de l'implémentation de ces dialogues pour comprendre comment ils jouent le rôle de modèle causal dans le DC de Prolog-Tutor.

5.3.1.1.1. Structure des dialogues

Ces dialogues sont mis en œuvre sur la base d'une structure échelonnée sur deux niveaux – respectivement dénommés D1 et D1.2 - caractérisés par le type de questions qui y sont posées par le système (Figure V.6). Les questions du niveau D1 sont dites "questions diagnostic" car elles touchent un aspect particulier du processus de résolution de l'exercice et y répondre interpelle forcément l'usage d'une des capacités visées à travers cet exercice ou nécessaire pour le réaliser correctement. Cette association explicite entre une question et un ensemble de capacités fait de ces dialogues un modèle causal (Figure V.6).

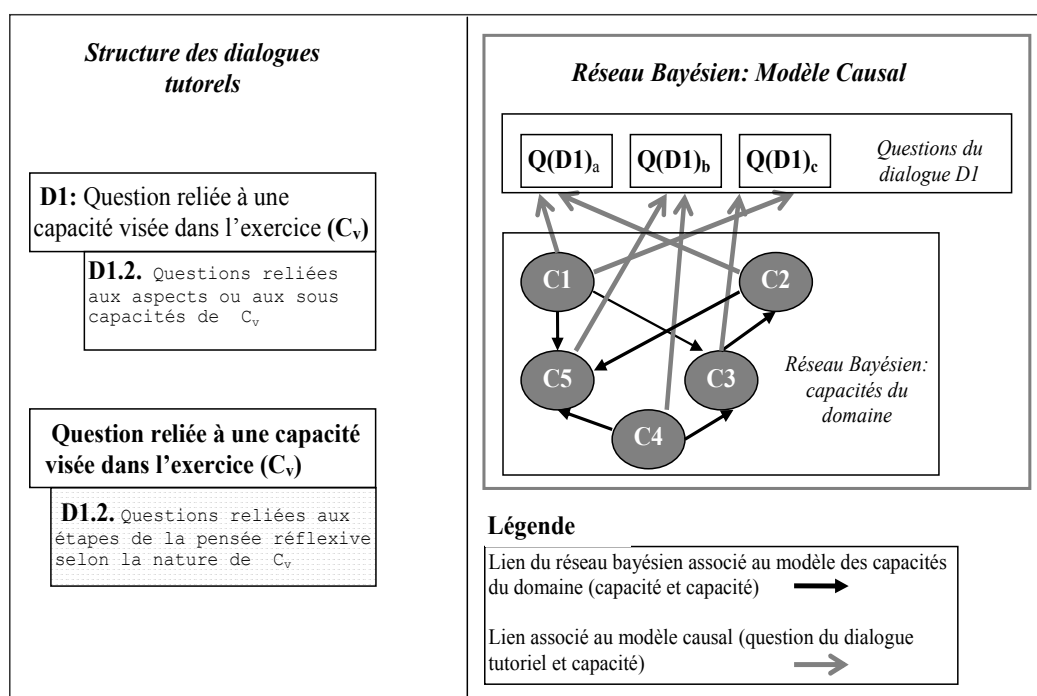


Figure V.6. Structure des dialogues tutoriels et modèle causal associé

Dans Prolog-Tutor, cette structure générique est préservée à travers l'utilisation d'une classe: **TutorialDialog.java**. Cela permet d'instancier plusieurs dialogues tutoriels à partir de cette seule structure générique.

5.3.1.1.2. Implémentation des dialogues: patrons ou questions préprogrammées?

Au début de l'implémentation de Prolog-Tutor, nous avons conçu différentes classes Java qui correspondaient à des éléments de la taxonomie des objectifs de Gagné (Gagné, 1992). On avait donc une classe pour des concepts, des principes, des règles et des procédures

(APPENDICE. C: Figure C. 9). La pertinence de ces classes est qu'elles peuvent servir de patrons (*frame* en anglais, dans le vocabulaire de la représentation des connaissances en IA) de définition des caractéristiques de chaque type d'objectif d'apprentissage. Le patron peut alors être instancié pour une instance d'objectif d'apprentissage, dans un domaine donné.

Par exemple, la classe "Concept" comporte les champs: `Attributs` (les attributs des concepts), `Valeurs` (les valeurs des attributs des concepts), `Exemple`, `ContreExemple`. Elle inclut également les fonctions "Identifier", "Classifier". Toutes ces fonctions permettent de définir des patrons de questions associés à la capacité dont elles portent le nom. Par exemple, la fonction "Identifier" retournerait un patron de question portant sur l'identification d'un concept. Si l'on considère le concept de `Variable` en Prolog. L'attribut principal de ce concept est sa première lettre. La valeur de cet attribut est que cette lettre doit toujours commencer par une majuscule. La question générique d'identification par rapport à une potentielle instance d'un concept est:

- Pour chaque attribut de ce concept:

⇒ Quel est `<Attribut>` de `<Instance de Concept>`?

S'il s'agit d'une instance du concept de `Variable` – que nous dénommons instance de `Variable` – cette question devient :

⇒ Quelle est la première lettre de `<instance de Variable>`?

L'avantage évident de cette approche est qu'il permet de définir un ensemble réduit de patrons qui caractérisent chacun des objectifs (capacités) d'apprentissage qui sont visés dans un STI. Ces patrons peuvent être instanciés – et donc réutilisés tant que nécessaire – pour les cas spécifiques auxquels ils sont applicables.

Néanmoins, nous avons rapidement réalisé que l'idéalisme de cette approche la rend inexploitable pour peu que le domaine étudié se complexifie. En effet, les manières d'identifier des concepts varient. Les conditions qui permettent l'application d'un principe peuvent être nuancées d'un contexte à l'autre, pour le même domaine étudié. Il aurait donc fallu faire plusieurs patrons pour le même type d'objectif d'apprentissage, ce qui rendrait leur pertinence caduque. De plus un effort minimal en traitement du langage naturel s'est avéré nécessaire, par exemple les marqueurs de discours pour référer aux attributs d'un concept.

Pour ces raisons, l'option des patrons associés aux capacités visées a été momentanément remplacée par l'option des questions prédéfinies (APPENDICE. C: Figure C. 10). Pour chaque exercice de Prolog-Tutor, les questions qui composent le dialogue tutoriel correspondant sont conçues et formulées à travers une base de données. Cette base de données indique principalement pour chaque question:

- ⇒ le niveau de dialogue auquel elle appartient;
- ⇒ le moment où elle doit être posée dans un dialogue tutoriel ;
- ⇒ la capacité qu'elle permet de vérifier (la capacité nécessaire pour y répondre correctement) et l'élément de connaissance qui y est associé;
- ⇒ la ou les réponses qui sont attendues. Ces réponses peuvent être complexes, d'où la nécessité de les formuler de manière codée. Lors de l'évaluation d'une réponse de l'apprenant, l'analyseur utilise cette information.

5.3.1.2. La phase d'acquisition d'indices: l'analyseur des réponses

L'*analyseur* des réponses de l'apprenant doit permettre de déterminer si une réponse de l'apprenant est correcte ou incorrecte. Cette caractérisation de la réponse est sauvegardée et peut être accédée plus tard par le *Profiler*, afin de diagnostiquer la réponse correspondante.

Prolog-Tutor ne permet pas encore de déterminer si une réponse est partiellement correcte. Dans la base de données des questions, chaque question d'un dialogue tutoriel est associée à la réponse (ou à l'ensemble des réponses) attendue. Une fois encore, l'évaluation des réponses est intrinsèquement *liée aux données d'un exercice* particulier et l'analyseur doit assurer tous les traitements où ces données sont indispensables. Cela permet de séparer les principales opérations du DC (les inférences et la boucle diagnostic-réflexion) des données spécifiques de l'exercice dans lequel il s'applique, préservant ainsi la généralité – et par conséquent la *réutilisabilité* - de l'algorithme de DC. L'analyseur implémente une interface définie dans la classe **Context_Indices.java** dans Prolog-Tutor. Cette interface nécessite l'implémentation de deux fonctions générique propres à la phase d'acquisition d'indices: la fonction `evaluate()` pour analyser et évaluer la réponse de l'apprenant et (2) la fonction `give_result()` pour récupérer le résultat de cette évaluation.

L'analyseur propre à Prolog-Tutor intègre en plus quatre algorithmes de traitement syntaxique qui correspondront à la fonction `evaluate()`, selon la nature de la question où l'apprenant est évalué:

- ⇒ l'algorithme pour le traitement des réponses génériques: il s'agit simplement d'effectuer une correspondance (*pattern matching*) entre la réponse de l'apprenant et l'une des réponses attendues;
- ⇒ l'algorithme de traitement des réponses sur une unification;
- ⇒ l'algorithme de traitement des réponses relatives à la résolution d'un sous-but;
- ⇒ l'algorithme de traitement des réponses relatives à l'échec d'une unification.

5.3.1.3. La phase d'inférence et la boucle diagnostic-réflexion: le *Profiler*

Le fonctionnement du *Profiler* reflète la dynamique du diagnostic. Cette composante du module pédagogique assure les phases d'inférence et la phase associée à la stimulation de la réflexion ainsi que sa rétroaction sur le DC.

5.3.1.3.1. Les phases d'inférence

L'algorithme générique décrit au Tableau IV.1 (en page 130) est matérialisé par la composante *Profiler* du module pédagogique implémentée à travers la classe **`Profiler_Generic_DC.java`**.

- *Premièrement*: la caractérisation de la dernière réponse fournie par l'apprenant dans le dialogue tutoriel est récupérée dans l'*analyseur de réponse*. Cette phase est assurée par la fonction générique `get_evaluation()`;
- *Deuxièmement*: en cas de réponse incorrecte (ou correcte: dans ce cas, on effectue quand même un diagnostic pour s'assurer que l'apprenant comprend bien les fondements sa réponse), le *Profiler* utilise le *modèle causal* correspondant au dialogue tutoriel pour générer les explications potentielles du caractère incorrect de cette réponse (dans Prolog-Tutor, il s'agit des capacités associées aux connaissances de la programmation logique avec Prolog). Le modèle causal est matérialisé par l'association de chaque question du dialogue tutoriel à un ensemble de capacités du domaine (Figure V.6, ci-dessus en page 148). Il s'agit du premier niveau d'inférence qui génère l'ensemble des capacités dont la "non acquisition" est potentiellement responsable de la

réponse incorrecte de l'apprenant. Cette phase est assurée par la fonction générique `diagnose_learner()` ;

- *Troisièmement*: le *Profiler* utilise le réseau bayésien associé au modèle causal (Figure V.6, ci-dessus en page 148), pour identifier les explications les plus probables du caractère incorrect de la réponse de l'apprenant. Ces explications correspondent aux capacités dont la "non acquisition" explique le mieux la réponse de l'apprenant (Figure V.7). Il s'agit du deuxième niveau d'inférence et la prochaine étape du diagnostic consiste à remédier aux difficultés ainsi identifiées, ce à travers la stimulation de la pensée réflexive sur ces capacités. Cette phase est assurée par la fonction générique `dialognis_remediation_loop()` ;

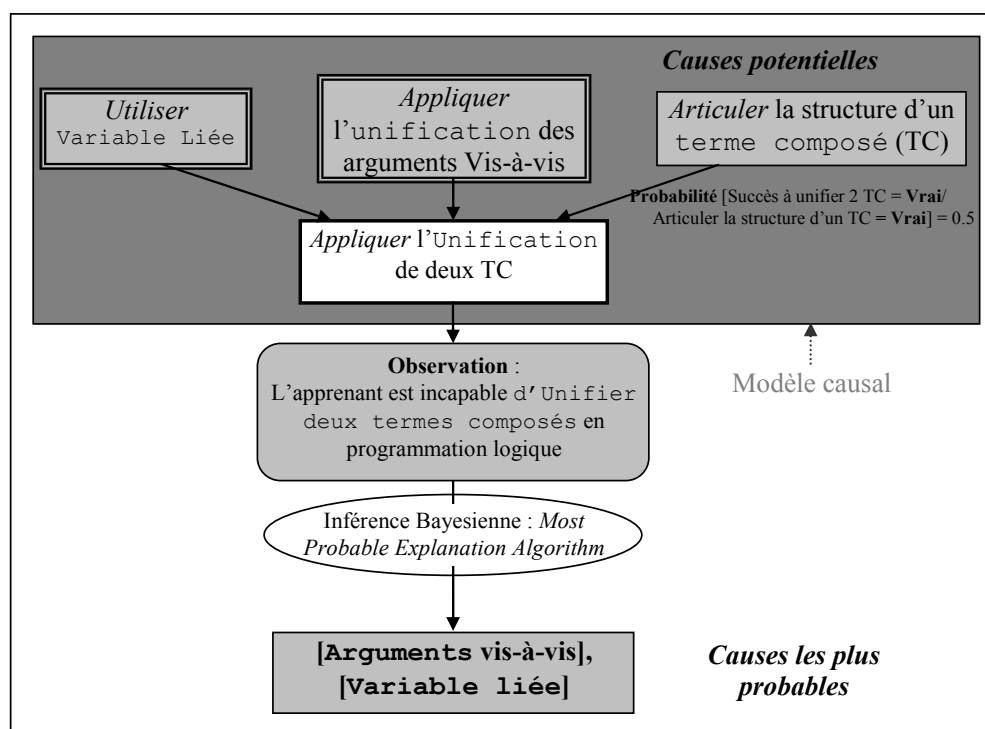


Figure V.7. Inférence des causes les plus probables d'une réponse incorrecte

5.3.1.3.2. Construction du réseau bayésien

Dans Prolog-Tutor, le réseau bayésien associé aux capacités du domaine est une *couverture* (*overlay*) du modèle du domaine (la classe `PrologCapabilities.java`). Cette classe est en fait une sous-classe de la classe `BayesNet.java` provenant de la librairie

JavaBayes (Cozman, 2001). Pour créer le modèle causal d'un exercice, on ajoute au réseau bayésien associé aux capacités du domaine, un ensemble de noeuds, chacun représentant une question du dialogue tutoriel (Figure V.6, ci-dessus en page 148). C'est ainsi que l'influence de chaque réponse de l'apprenant à une question du dialogue est refléter dans le modèle du domaine.

5.3.1.3.3. *La boucle diagnostic-réflexion : la stimulation de la pensée réflexive post-diagnostic*

Le concept de boucle entre le diagnostic et la remédiation introduit dans cette thèse est matérialisé avec le cas particulier d'une boucle entre le DC de la réponse de l'apprenant et la stimulation de la pensée réflexive (ou la réflexion) chez ce dernier. Dans le sens du diagnostic vers la réflexion, la relation est naturelle dans ce sens que le but pédagogique est de stimuler la pensée réflexive à propos des difficultés qui ont été diagnostiquées. Les dialogues tutoriels jouent un rôle important dans cette phase: les questions qui sont posées à l'apprenant au niveau D1.2 de la structure du dialogue, ont pour but de favoriser la pensée réflexive sur les difficultés diagnostiquées au niveau D1.

L'unique particularité des dialogues tutoriels génériques est qu'ils permettent au module pédagogique de poser des questions à l'apprenant dans le sens de le mener vers la solution d'un exercice. Pour chaque question du niveau D1, les questions du niveau D1.2 forment un sous dialogue qui articule les étapes de la phase correspondant à cette question. Dans le meilleur des cas, ces questions favorisent donc une pensée réflexive implicite chez l'apprenant, étant donné leur nature interactive (Dimitrova, 2003a).

Les dialogues tutoriels explicites sont construits de manière analogue aux dialogues génériques. Les principales questions qui seront abordées dans ce dialogue correspondront donc aux étapes de la pensée réflexive, appliquées selon les caractéristiques spécifiques de cette capacité (Tchetagni, Nkambou, et Kabanza, 2004). Au Chapitre VI par exemple, une illustration de ces étapes est présentée pour le cas de la capacité à *appliquer* un principe en général.

L'on doit noter que par défaut, le dialogue de niveau D1.2 est déclenché seulement lorsqu'une réponse incorrecte a été constatée pour la question correspondante, au niveau D1. Cependant, en utilisant le paramètre `defaultDiagnosisMode` (Tableau IV.1, en page 130) on peut déclencher ce dialogue en présence d'une bonne réponse. La pertinence de cette

éventualité découle du fait que certaines optiques pédagogiques préconisent la vérification de la compréhension de l'apprenant, au-delà du fait qu'il a fourni une bonne réponse à une question.

5.3.1.3.4. *La boucle diagnostic-réflexion : rétroaction des résultats de la pensée réflexive sur le DC*

Une fois qu'une réflexion a été stimulée auprès de l'apprenant sur les difficultés qui ont été diagnostiquées, il est question d'utiliser les résultats de cette réflexion pour améliorer le DC. Il s'agit de la rétroaction de la pensée réflexive sur le DC qui donne lieu à la relation circulaire entre le diagnostic et la réflexion. Dans cette thèse, deux formes principales d'amélioration du DC ont été identifiées: le raffinement du DC, la confirmation/infirmerie du DC.

Dans Prolog-Tutor, le DC porte sur les capacités de l'apprenant par rapport au domaine de la programmation logique. Le raffinement d'un DC consiste à déterminer les aspects d'une capacité (les sous capacités et leurs interactions pour former cette capacité) qui sont problématiques pour l'apprenant.

La confirmation d'un DC est une rétroaction plus générale dans la mesure où elle indique au module pédagogique si ses inférences sont correctes. Elle est considérée comme une amélioration du DC puisque lorsqu'un diagnostic n'est pas confirmé, cela indique au module pédagogique qu'il existe un problème au niveau de l'algorithme de DC ou au niveau de la représentation des capacités et de leurs interrelations. Actuellement, Prolog-Tutor réalise ces avertissements en sauvegardant dans une trace les informations relatives à toutes les capacités diagnostiquées à tort. Les vérifications qui doivent s'en suivre n'ont pas encore été explicitées et implémentées.

Dans les dialogues tutoriels génériques, les questions du niveau D1.2 sont dites "questions de raffinement" car elles interviennent lorsqu'un DC est posé au niveau D1. Ces questions permettent au système d'inspecter la difficulté de l'apprenant qui a été identifiée au niveau D1. Lorsque l'apprenant répond incorrectement à au moins une question de raffinement, cela est considéré comme une confirmation du DC posé au niveau D1. Dans le cas contraire, on infirme ce diagnostic en prenant note qu'une anomalie est apparue soit au niveau de l'algorithme correspondant, soit au niveau de la représentation des capacités du

domaine. À l'instar du niveau D1, chaque question du niveau D1.2 des dialogues tutoriels génériques est associée à une capacité du domaine.

Dans les dialogues explicites, les questions du niveau D1.2 ont pour but de provoquer une réflexion explicite par rapport à la capacité qui est associée à la question du niveau D1. Les principales questions qui seront abordées dans ce dialogue correspondent donc aux étapes de la pensée réflexive, appliquées selon les caractéristiques spécifiques de cette capacité. Dans ce sens, les questions du niveau D1.2 ne raffinent pas nécessairement un DC, mais au minimum, elles le confirment. En effet, chaque étape de la pensée réflexive à propos d'une capacité correspond à sa mise en œuvre explicite et fondée. Si l'apprenant est capable de les articuler, cela indique une anomalie au niveau de l'algorithme de DC ou de la modélisation explicite du processus de réflexion à propos de la capacité diagnostiquée en D1. Dans les dialogues tutoriels explicites, chaque question du niveau D1.2 est explicitement associée à l'étape de la pensée réflexive qu'elle est supposée stimuler. On verra que cette spécificité est nécessaire pour les besoins de l'évaluation de Prolog-Tutor

5.3.1.4. Le gestionnaire de flux du dialogue

Le flux d'un dialogue tutoriel peut suivre des cours différents selon un choix pédagogique. À travers une classe de gestion du flux de dialogue (**DialogueManager.java**) Prolog-Tutor implémente quatre algorithmes de gestion du flux du dialogue: l'algorithme pour le flux séquentiel, l'algorithme pour le flux adapté à court et à long terme, l'algorithme pour le flux adapté et révisé.

Dans l'optique séquentielle, toutes les questions qui sont posées suivent l'ordre préétabli dans la base de données des questions: elles sont présentées à l'apprenant les unes à la suite de l'autre.

On peut aussi adapter l'ordre des questions d'un dialogue tutoriel, en tenant compte des informations qui proviennent du modèle de l'apprenant. Dans Prolog-Tutor, l'adaptation à court terme du flux séquentiel consiste à ne pas poser les questions dont la capacité associée a été diagnostiquée comme acquise au cours de la session actuelle. Une capacité est considérée comme acquise au cours d'une session d'apprentissage si toutes les réponses aux questions où son usage est nécessaire sont correctes. L'adaptation du flux séquentiel à long terme est analogue : une question n'est pas posée si la probabilité d'acquisition de la capacité

correspondante dans le modèle de l'apprenant (modèle à long terme) est supérieure à un seuil δ prédéfini. Dans ce cas d'espèce, δ a été fixé à 0.8.

Nous avons envisagé la possibilité de réviser une adaptation qui a été faite dans un dialogue tutoriel. Le fait de sauter des questions peut occasionner une confusion ou un égarement chez l'apprenant. Dans ce cas, l'idée est d'effectuer un retour en arrière afin de recommencer le dialogue à un point où l'apprenant se retrouve (Tchetagni et Nkambou, 2005). Le retour en arrière consiste à identifier parmi les questions qui ont été sautées, celle qui correspond au meilleur compromis entre l'adaptabilité et la préservation de l'intelligibilité du dialogue. Dans Prolog-Tutor, ce compromis consiste à visiter chacune des questions sautées, en commençant par la dernière (Figure V.8). Dès lors que l'apprenant réussit à répondre à une de ces questions, le dialogue suit son cours normal.

L'adaptation du dialogue nécessite la prédéfinition d'un certain nombre de paramètres, tâche qui peut devenir ardue. Considérons que le dialogue tutoriel vient de poser la question numéro I et que selon les informations du modèle de l'apprenant, la prochaine question à aborder est la question $J = I+2$. Pour préserver l'intelligibilité des questions, le gestionnaire du dialogue doit pouvoir faire un pont entre les questions I et J (que s'est-il passé entre les deux questions?). Ce pont nécessite la prédéfinition d'expressions qui permettent de résumer ce qui s'est passé depuis la question I et d'introduire la question J . Ces liaisons contextuelles permettent de justifier la question J puisqu'elle ne suit pas directement la question I . Combien de liaisons sont nécessaires pour un dialogue de N questions ? Pour la question numéro K du dialogue tutoriel, il faut prévoir $(N-K)$ liaisons, ce qui nous mène à une complexité totale de l'ordre de $(N-1)!$. Pour cette raison, les algorithmes d'adaptation ont été implémentés dans Prolog-Tutor, mais leur exploitation demeure potentielle étant donné le caractère pléthorique des informations nécessaires.

The screenshot shows the PLEARN software interface for 'Le calcul des Predicats'. The main window is titled 'Cours' and contains a 'Demo : Predicate Calculus for Logic Programming' section. On the left, there is a navigation menu with 'Objectifs' and a tree view under 'Calcul des predicats' including 'Grammaire', 'Traitement', 'Unification', 'Resolution', and 'Exercices sur la resolution'. The central area is divided into three main sections: 'BASE DE CONNAISSANCES' (Knowledge Base), 'ENONCE DE L'EXERCICE' (Exercise Statement), and a dialogue log. The Knowledge Base contains several logical rules. The Exercise Statement asks for the result of a query. The dialogue log shows a sequence of interactions: a tutor providing feedback, a student asking for elaboration, the tutor providing two levels of elaboration, and a final negative feedback and confirmation of diagnosis. At the bottom, there are buttons for 'SUCCES', 'JE NE SAIS PAS', 'ECHEC', and 'Constructeur d'expression', along with a 'Reponse' input field.

Figure V.8. Adaptation et Révision du flux d'un dialogue tutoriel

5.3.1.5. La trace des actions du module pédagogique

Au cours de l'interaction avec l'apprenant à travers le dialogue tutoriel, les décisions pédagogiques sont invisibles. Une trace des actions du module pédagogique a été élaborée pour deux raisons (Figure V.9):

- Pour chaque question posée à l'apprenant, cette trace indique la raison de ce question (pour diagnostiquer, pour raffiner un diagnostic, pour provoquer une phase spécifique de la pensée réflexive);
- Pour chaque réponse de l'apprenant à une question, cette trace consigne toutes les actions pédagogiques qui en ont découlé:
 - ⇒ Le diagnostic qui a été posé
 - ⇒ L'ordre de mise à jour qui a été envoyé au modèle de l'apprenant (voir prochaine section)

⇒ Le cas échéant, un mémorandum rétroactif qui est sauvegardé dans un fichier à l'intention du concepteur ou du programmeur. Tel que discuté au Chapitre 3, il s'agit de divers types de remarques à propos de: la nécessité de réexaminer la représentation des connaissances du domaine, la conception du DC, les paramètres du DC, l'algorithme de DC. Ces remarques sont nécessaires lorsqu'un DC formellement posé au niveau D1 n'a pas été confirmé dans D1.2 (Chapitre III).

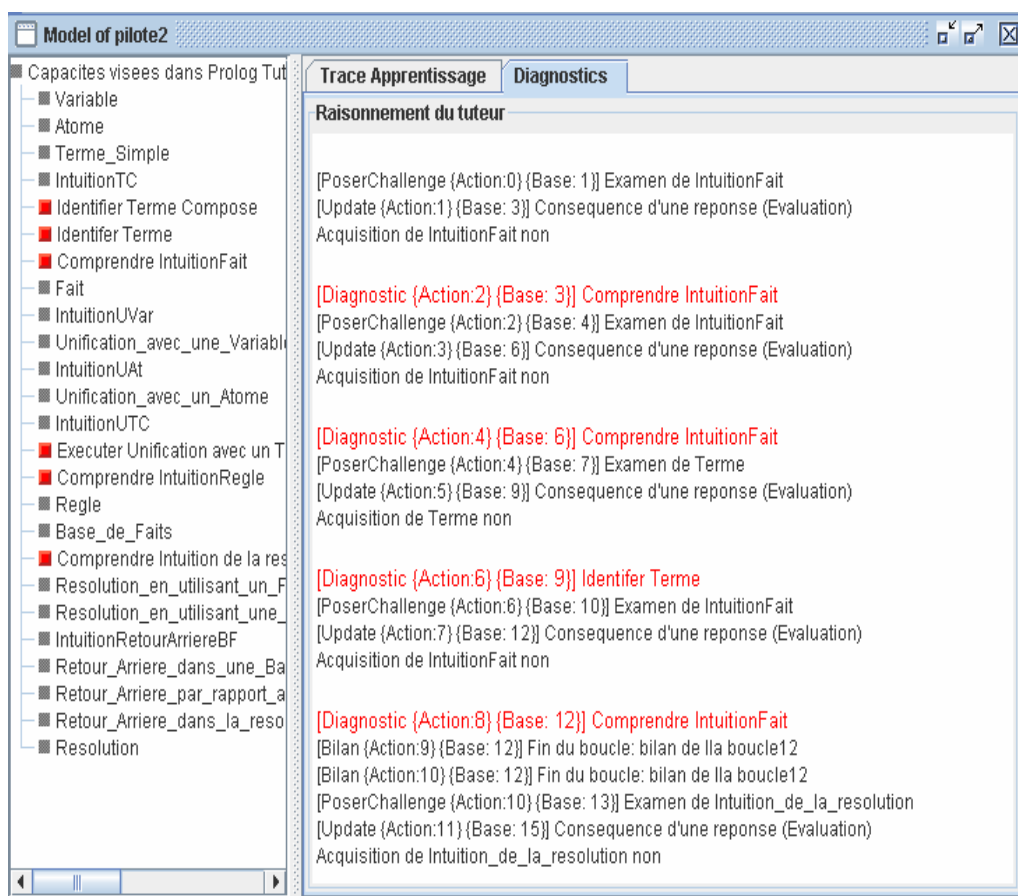


Figure V.9. Trace des actions dans le module pédagogique

5.3.2. Dynamique du module pédagogique

La Figure V.10 illustre la dynamique de fonctionnement du module pédagogique. Le module pédagogique fonctionne sur quatre macro opérations:

- Il communique avec le module du domaine pour charger l'exercice choisi par l'apprenant;
- Il communique avec la base de données des questions pour charger les questions associées à l'exercice actuel; l'identificateur de cette base de donnée est un attribut de l'exercice;
- Il communique au module de communication l'interface principal associé à l'exercice actuel et il engage le dialogue tutoriel;

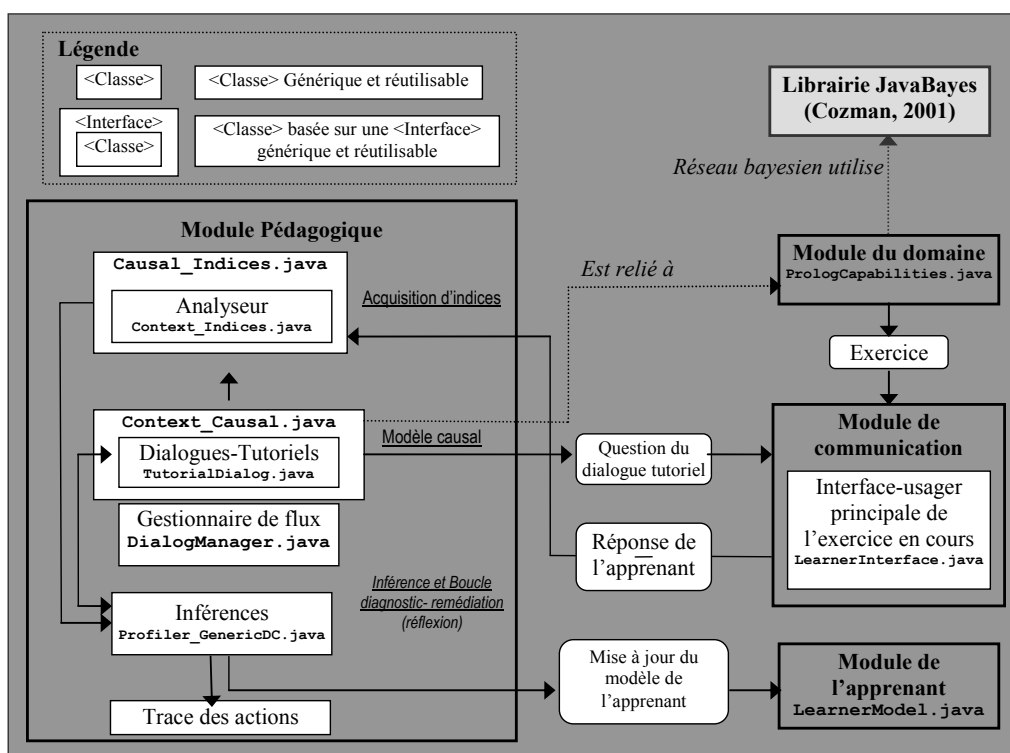


Figure V.10. Dynamique du module pédagogique dans Prolog-Tutor

- Pour chaque réponse de l'apprenant, le module pédagogique :
 - ⇒ Commande la mise à jour appropriée dans le modèle de l'apprenant. Le *Profiler* doit commander une mise à jour du modèle de l'apprenant en initiant des "observations" sur tous les noeuds du réseau bayésien associé au modèle causal correspondant aux capacités dont le DC a été confirmé lors de la phase de réflexion. Ces "observations" sont négatives lorsque le DC est celui d'une capacité non acquise sinon, elles sont positives (Chapitre IV). La propagation de l'effet de

ces observations dans le réseau donne lieu à des probabilités d'acquisition *a posteriori* pour chaque capacité du domaine. Ces probabilités correspondent au modèle *global* de l'apprenant à *court terme*: les capacités directement diagnostiquées - associées à une question posée à l'apprenant - auront la probabilité d'acquisition 1 ou 0 selon que la réponse de l'apprenant fût correcte ou incorrecte (respectivement); les autres capacités auront des probabilités *a posteriori* résultat de l'exécution de l'algorithme de mise à jour bayésienne (*Belief Updating*). Pour répercuter ces évidences dans le modèle de l'apprenant à *long terme*, la formule suivante est utilisée, à la $\mathcal{N}^{\text{ème}}$ mise à jour du modèle de l'apprenant, pour chaque capacité ζ représentée dans ce modèle:

$$\mathbf{Probabilité}_{\text{LTAc}}(\zeta)^{\mathcal{N}} = [\mathbf{Probabilité}_{\text{LTAc}}(\zeta)^{\mathcal{N}-1} * (\mathcal{N}-1) + \mathbf{Probabilité}_{\text{CTAc}}(\zeta)^{\mathcal{N}}] / \mathcal{N}$$

Où:

- $\mathbf{Probabilité}_{\text{LTAc}}(\zeta)^{\mathcal{N}}$ est la probabilité à long terme (dans le modèle de l'apprenant à long terme) que la capacité ζ est acquise à la $\mathcal{N}^{\text{ème}}$ mise à jour du modèle de l'apprenant.
- $\mathbf{Probabilité}_{\text{CTAc}}(\zeta)^{\mathcal{N}}$ est la probabilité à court terme que la capacité ζ est acquise à la $\mathcal{N}^{\text{ème}}$ mise à jour du modèle de l'apprenant.

Cette formule est nécessaire parce que le réseau bayésien qui est utilisé dans Prolog-Tutor n'est pas un réseau Bayésien temporel dynamique (Reye, 1998, 2004): la mise à jour bayésienne ne tient pas compte de la probabilité actuelle de chaque noeud du réseau, ni des mises à jours précédentes. Cette formule est une façon de considérer les inférences posées par le passé à travers d'autres interactions avec l'apprenant: à la $\mathcal{N}^{\text{ème}}$ mise à jour, le poids des inférences passées est pris en compte en multipliant la probabilité qui en résulte ($\mathbf{Probabilité}_{\text{LTAc}}(\zeta)^{\mathcal{N}-1}$) par le facteur $\mathcal{N}-1$. La moyenne de l'ensemble est effectuée en y ajoutant l'inférence actuelle ($\mathbf{Probabilité}_{\text{CTAc}}(\zeta)^{\mathcal{N}}$), avant de la pondérer par le facteur représentant l'ensemble des inférences qui ont été posées (\mathcal{N});

⇒ Si la question clôture un sous-dialogue D1.2, le module pédagogique doit également indiquer que le DC du niveau D1 n'a pas été confirmé si tel est le cas.

Dans Prolog-Tutor, cette information est sauvegardée dans la trace des décisions du tuteur.

5.4 Conclusion

Un STI intègre quatre composantes principales: un module du domaine, un module de l'apprenant, un module de communication et un module pédagogique. Pour décrire l'instanciation de chacun de ces modules dans Prolog-Tutor, nous avons d'abord rappelé les principaux enjeux d'IA reliés à leur implémentation dans un système informatique. Ensuite, nous avons expliqué pourquoi et comment:

- ⇒ Le module du domaine est principalement implémenté sur la base d'une représentation sémantique, interprétée comme un réseau bayésien;
- ⇒ Le module de l'apprenant est une représentation probabiliste de l'état des connaissances de l'apprenant;
- ⇒ Le module de communication est basé sur une gestion d'un ensemble varié d'interfaces usager;
- ⇒ Le module pédagogique est centré sur la représentation et la gestion de dialogues tutoriels qui supportent un DC interactif et qui tiennent lieu de modèle causal pour ce DC.

Concernant l'implémentation du DC dans les STI, l'originalité de notre contribution dans le domaine AIED est la proposition d'une librairie de fonctions génériques propres à ce processus (Figure V.10 ci-dessus, en page 159). De ce fait, l'usage d'une approche de programmation orientée objet et du langage java se justifie dans la mesure où elle permet de prédéfinir des objets et des fonctions standard qui y sont associées (à travers des objets de type `Interface`). La description du module pédagogique de Prolog-Tutor a permis de constater que l'algorithme formalisé au chapitre précédent a été matérialisé à travers une classe générique (`Profiler_Generic_DC.java`, `LearnerModel.java`, `BayesNet.java`) et un ensemble d'interfaces génériques (`Context_Indices.java`, `Context_Causal.java`). Ces objets constituent la base du DC et ils peuvent être regroupés dans une *librairie générique d'objets* pour le DC dans un STI. Notre contribution est due à cette généralité (et de la réutilisabilité qui en découle) puisque ces objets possèdent au moins une des deux propriétés suivantes: ils implémentent des fonctions associées au DC et ce de

manière générique, ils implémentent une interface (générique par nature) définissant des fonctions associées au DC (dans le cas où ces fonctions dépendent fortement du contenu associé à un STI et donc, ne peuvent être implémentées de manière générique).

Pour clôturer cet exposé sur une vision pédagogique du DC dans les STI, le prochain chapitre illustre la mise en oeuvre d'un ensemble de spécifications de DC, dans Prolog-Tutor.

Chapitre VI

ILLUSTRATIONS D'UNE PERSPECTIVE PÉDAGOGIQUE DU DC DANS UN STI: PROLOG-TUTOR

6.1 Introduction

Au cours des trois derniers chapitres, nous avons: (1) présenté la perspective de DC préconisée dans cette thèse à travers un cadre de spécification pour le diagnostic cognitif (CSDC) (Chapitre III); formalisé cette perspective à travers un algorithme générique de DC basé sur l'approche bayésienne (Chapitre IV); implémenté concrètement cet algorithme dans un STI pour la programmation logique (Prolog-Tutor: Chapitre IV, Chapitre V). D'un point de vue conceptuel l'originalité de notre contribution à travers le CSDC tient d'une part, de l'interprétation explicite de l'influence des paradigmes de cognition sur le DC et d'autre part, de l'intégration de la remédiation pédagogique dans le DC, à travers une boucle diagnostic-réflexion dans ce cas particulier.

Le but de ce chapitre est *d'illustrer* cette perspective du DC dans Prolog-Tutor (Tchetagni, Nkambou, et Bourdeau, 2005a), un STI pour les notions de base en Programmation Logique. Deux spécifications de DC sont instanciées à partir du CSDC. Leur *mise en oeuvre* dans Prolog-Tutor est illustrée et commentée en référence aux thèmes

discutés au Chapitre III, particulièrement en ce qui concerne: la correspondance entre la mise en œuvre de chaque spécification et l'intention pédagogique initiale associée à cette spécification. Une présentation du domaine de connaissances étudié dans Prolog-Tutor (section 6.2) précède l'illustration et l'analyse de chaque spécification (sections 6.3, 6.4, 6.5).

6.2 Prolog-Tutor: domaine étudié et capacités visées

Prolog (Colmerauer, 1990) est un langage de programmation utilisé pour formellement décrire une situation de la vie réelle à travers des axiomes logiques. Ce langage peut être considéré comme une traduction computationnelle du langage du calcul des prédicats ou de la logique du premier ordre (Boole, 1854; Frege, 1879; Russel et Whitehead, 1910-1913). Le but de cette conversion est de permettre à une machine – en l'occurrence l'ordinateur - d'effectuer diverses formes de traitement sur la base de ces axiomes (Turing, 1950). Au niveau pratique, Prolog est opérationnel grâce à des interpréteurs qui incorporent des algorithmes émulant un raisonnement par déduction: le chaînage avant et le chaînage arrière (SICSTUS-Prolog, 2005). L'apprentissage de Prolog commence typiquement par l'acquisition des notions de base, à savoir:

- (1) Le vocabulaire relatif aux structures de données en Prolog (quelles entités permettent de représenter l'information?);
- (2) Les principes qui régissent les processus de raisonnement dans un interpréteur Prolog (comment le raisonnement est-il simulé dans Prolog?).

Quatre structures de données de base composent le vocabulaire de Prolog: les constantes, les variables, les termes composés et les clauses de Horn. Des concepts d'ordre supérieur sont associés à ces structures de données: les concepts de fait et de but sont associés aux termes composés; le concept de règle (qui peut aussi être désigné par règle-Prolog) est associé au concept de clause de Horn; le concept de base de connaissances fait appel aux concepts de fait et de règle.

Dans un interpréteur de Prolog, l'émulation du raisonnement déductif s'appuie principalement sur deux algorithmes: l'unification et la résolution. L'unification peut être vue comme un processus de «*pattern matching*» qui permet d'établir que deux éléments du vocabulaire du langage du calcul des prédicats sont identiques. La résolution d'un but se base sur l'unification pour simuler le

raisonnement déductif. Étant donné une base de connaissances – qui peut être vue comme un monde miniature représentant une situation particulière –, la résolution consiste à prouver si un fait est Vrai ou Faux selon les informations de cette base. Cette émulation du raisonnement est aussi appelée: preuve de but ou résolution d'un but, le "but" en question étant le fait à prouver.

Dans Prolog-Tutor, ces structures de données et ces algorithmes sont considérés comme des éléments de connaissances du domaine de connaissances associées à Prolog (Figure VI.1). Chaque élément de connaissances intègre plusieurs aspects. Chaque aspect d'un élément de connaissance matérialise l'usage particulier de cet élément dans une circonstance précise. Les aspects d'une connaissance peuvent donc être considérés comme des capacités associées à cette connaissance ou encore comme des objectifs d'apprentissage.

6.2.1. Objectifs d'apprentissage: les capacités visées avec Prolog-Tutor

Plusieurs taxonomies relatives aux objectifs d'apprentissage permettent de clarifier cette notion dans le cadre d'un STI. Pour notre part, la taxonomie proposée par Nkambou, Frasson et Gauthier (2003a) a été adoptée. En effet, cette taxonomie est clairement articulée en ce qu'elle distingue clairement les rôles respectifs d'un élément de connaissance et d'une habileté, dans une capacité visée. De plus, dans cette taxonomie, les capacités sont des objectifs d'apprentissage. Une capacité est conçue comme une habileté appliquée à un élément de connaissance, les habiletés étant issues de la taxonomie de Bloom (Bloom et al., 1956). L'avantage majeur de cette optique réside dans le fait qu'une habileté peut être vue comme un processus générique, dont associable avec un modèle générique (Paquette, 2002). Les stratégies pédagogiques pour favoriser l'acquisition d'une capacité pourront donc être établies à partir du modèle générique de l'habileté correspondante. Selon le niveau de spécificité de la situation d'apprentissage à laquelle elles s'appliquent, ces stratégies pédagogiques peuvent donc aussi être génériques et réutilisables (section 0).

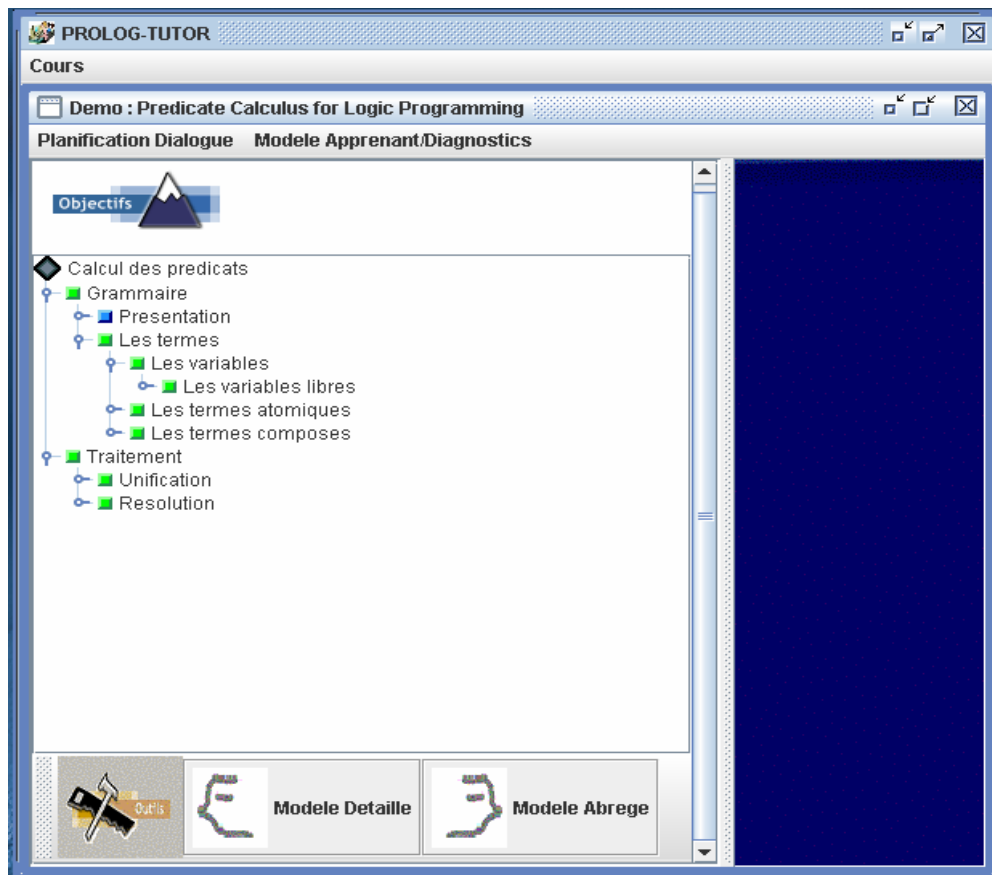


Figure VI.1. Éléments de connaissances dans Prolog-Tutor

L'ensemble des capacités visées dans Prolog-Tutor réunit des habiletés relatives au vocabulaire de Prolog et des habiletés relatives aux algorithmes d'unification et de résolution (Tableau VI.1).

Les illustrations qui font l'objet des prochaines sections concernent particulièrement la capacité à «Exécuter la résolution d'un but». Le Tableau VI.2 détaille les capacités sous-jacentes à la résolution d'un but. Ces sections présentent deux spécifications de DC - élaborées à partir du CSDC défini au Chapitre 3 – dans Prolog-Tutor.

Tableau VI.1. Un sous-ensemble des objectifs d'apprentissage dans Prolog-Tutor

Capacités visées dans Prolog-Tutor	Composantes de la capacité	
	Habilités	Élément de connaissance
<i>Identifier</i> une Variable	Identifier	Variable
<i>Identifier</i> une Constante	Identifier	Constante
<i>Comprendre</i> la notion de Variable Liée	Comprendre	Variable, Constante
<i>Utiliser</i> une Variable	Utiliser	Variable
<i>Identifier</i> un Terme Composé	Identifier	Terme Composé
<i>Comprendre</i> l'intuition d'un Fait	Comprendre	Terme Composé
<i>Identifier</i> une Clause de Horn	Identifier	Clause de Horn
<i>Comprendre</i> l'intuition d'une Règle	Comprendre	Clause de Horn
<i>Comprendre</i> le concept d'Unification	Comprendre	Unification
<i>Exécuter</i> une Unification	Exécuter/Appliquer	Unification
<i>Comprendre</i> le concept de Résolution	Comprendre	Résolution
<i>Exécuter</i> une Résolution	Exécuter/Application	Résolution

Premièrement, on adopte une optique rationaliste de la cognition où l'objectif pédagogique est d'estimer l'état des connaissances de l'apprenant. Pour chaque capacité C représentée dans le modèle du domaine, le DC doit établir: (1) la compréhension par l'apprenant, c'est-à-dire sa représentation correcte des interrelations entre les éléments (sous-capacités) de \mathcal{C} , (2) la représentation par l'apprenant du rôle (ou de la contribution) de chaque sous-capacité dans la mise en œuvre de \mathcal{C} . La première spécification doit permettre d'atteindre cet objectif pédagogique. Pour ce qui est de l'exploitation pédagogique de ce DC par la stimulation de la pensée réflexive, l'approche *Interactive Learner Modelling* (ILM) est utilisée. Cette spécification est concrétisée dans Prolog-Tutor par deux dialogues tutoriels: le premier dialogue a une structure générique et le second a une structure qui reflète les composantes de la pensée réflexive.

Deuxièmement, Prolog-Tutor adopte une optique behavioriste. L'objectif est de donner à l'apprenant l'occasion d'évaluer quantitativement son niveau d'acquisition par rapport aux capacités visées dans Prolog-Tutor. En entendant de réaliser cet objectif sous forme d'une réflexion "sur l'action", une deuxième spécification préconise un DC comportemental à travers l'évaluation sans inférence. On intègre au DC une réflexion implicite à travers l'approche *Open Learner Modelling* (OLM).

Tableau VI.2. Les capacités reliées à l'exécution de la résolution d'un but

Capacité principale: Exécuter (Résolution d'un But)		
Capacité	Élément de connaissance	Type de capacité selon la taxonomie de Bloom
Interpréter (Fait) comme un but à résoudre	Fait	Comprendre une proposition
Utiliser (Règle Prolog) lors la résolution d'un but	Règle Prolog	Appliquer des principes
Manipuler (Règle Prolog) lors la résolution d'un but	Règle Prolog	Appliquer des principes et des procédures
Utiliser (Base de connaissances) lors de la résolution d'un but	Base de connaissances	Appliquer des principes
Comprendre (Retour-arrière) lors de la résolution d'un but	Retour-arrière	Comprendre un concept
Utiliser (Retour-arrière) lors de la résolution d'un but	Retour-arrière	Appliquer des principes
Effectuer (Retour-arrière) lors de la résolution d'un but	Retour-arrière	Appliquer des principes et des procédures

6.3 DC avec réflexion basée sur un dialogue tutoriel générique

Cette section illustre la mise en œuvre d'une spécification de DC basée sur un dialogue tutoriel générique. Dans Prolog-Tutor, les dialogues tutoriels génériques sont composés de questions dont le but est d'amener progressivement l'apprenant vers la solution d'un exercice. Ils se distinguent des dialogues tutoriels dont les questions sont explicitement conçues de manière à stimuler la pensée réflexive selon la théorie de Dewey. La prochaine section décrit les caractéristiques de la spécification en tant que telle, en référence aux principaux éléments du CSDC défini au Chapitre 3. Cette description est suivie de l'illustration en tant que telle de la mise en œuvre de cette spécification dans Prolog-Tutor.

6.3.1. Spécification

Le Tableau VI.3 illustre la spécification décrite ci-dessous. Si l'on considère les principaux éléments du CSDC, on constate que chacun d'entre eux est représenté dans cette spécification : le contexte pédagogique, le type de diagnostic, la réflexion et les recommandations relatives à l'implémentation.

Tableau VI.3. Spécification d'une instance de DC épistémique dans Prolog-Tutor

Paramètre	Instance de paramètre	Détails
Contexte Pédagogique		
Perspective de la cognition	Rationaliste	✓ L'apprenant raisonne à travers des processus mentaux sur des représentations relatives à un problème posé
Perspective des mécanismes d'apprentissage	Cognitiviste	<ul style="list-style-type: none"> ✓ Représentation de concepts ✓ Représentation des principes ✓ Utilisation appropriée des principes ✓ Application des principes
Perspective de l'instruction et du design de l'instruction	Non spécifié dans ce cas	Aucune précision nécessaire
Propriétés du DC		
Type de diagnostic	Épistémique	✓ Basé sur l'évaluation des réponses de l'apprenant à des questions relatives aux connaissances visées
Type de réflexion	Approche <i>Interactive Learner Modelling</i>	✓ Basée sur les dialogues tutoriels
Usage pédagogique (Réflexion du système)	Ciblage des besoins d'apprentissage	✓ Utiliser les résultats de la réflexion de l'apprenant pour cibler les besoins d'apprentissage <i>en raffinant le DC</i> initialement posé
Implémentation		
Recommandations relatives aux représentations de connaissances nécessaires	- Représentation des connaissances du domaine	✓ Représentation bayésienne des influences épistémiques entre les capacités associées aux éléments de connaissances de la programmation logique avec Prolog
	- Représentation des dialogues tutoriaux	✓ Relier chaque question à une capacité visée comme objectif d'apprentissage et faire le lien avec le modèle de l'apprenant
	- Représentation du modèle de l'apprenant	✓ Couverture (<i>overlay</i>) de la représentation des capacités associées au domaine des connaissances
Recommandations relatives à l'algorithme correspondant au processus de DC	- Recueillir et évaluer les réponses de l'apprenant dans un dialogue tutoriel	✓ Pour le diagnostic épistémique
	- Raffiner les questions posées à l'apprenant ainsi que les diagnostics posés	<ul style="list-style-type: none"> ✓ Pour la réflexion de l'apprenant ✓ Pour la réflexion du côté du système

6.3.1.1. Contexte pédagogique: Paradigme de cognition

Dans Prolog-Tutor, l'un des objectifs pédagogiques est d'inférer la compréhension et la capacité d'appliquer la procédure de résolution d'un but par un apprenant. Il s'agit de diagnostiquer comment ce dernier représente les concepts et les principes clés de ce processus ainsi que leurs interrelations. Il s'agit également de déterminer si l'apprenant sait comment utiliser ces concepts et principes pour appliquer la résolution d'un but en Prolog (Tableau VI.2 ci-dessus, en page 168). Ces caractéristiques correspondent à une **vision rationaliste de la cognition et cognitiviste de l'apprentissage**. La réponse de l'apprenant constitue donc une donnée importante du DC, mais n'en représente pas la finalité. Selon notre CSDC, deux types de diagnostic sont possibles dans une telle situation.

6.3.1.2. Type de diagnostic

Premièrement, les réponses de l'apprenant peuvent être observées dans la mesure où les questions correspondantes du dialogue tutoriel portent explicitement sur des représentations conceptuelles, sur l'utilisation ou l'application d'un principe, dans le cadre de la mise en œuvre d'une capacité (Figure VI.2).

Dans cette optique, ces réponses tiennent lieu de données pour un diagnostic épistémique puisque selon le cas :

- elles permettent de diagnostiquer que l'apprenant possède une représentation correcte d'un concept;
- elles permettent de diagnostiquer que l'apprenant possède une représentation correcte d'un principe;
- elles permettent de diagnostiquer que l'apprenant sait quand utiliser un principe;
- elles permettent de diagnostiquer que l'apprenant sait appliquer un principe. Ce diagnostic épistémique s'élaborerait alors sur la base d'une évaluation sans inférence (diagnostic comportemental) des réponses de l'apprenant.

Prolog-Tutor Resol

Visualisations

ENONCE

=> Supposons qu'on a la requete suivante dans un interpreteur SICSTUS PROLOG:

```
?- est_atteint_de(pierre,M)
```

=> Quel est le Resultat de la Preuve du BUT qui correspond cette requete?

CONSTRUISEZ cette preuve en:

--> CONSIDERANT la Base de Connaissances specifiee dans la fenetre

"BASE DE CONNAISSANCES" (a droite) et en

--> REPONDANT aux questions du systeme par le biais des BOUTONS du panneau "Reponses/Questions. Usager"

BASE DE CONNAISSANCES

```
est_atteint_de(X,M):-symptome(X,S),cause(S,M).
cause(fievre,infection).
cause(taux_eleve_gb,pneumonie).
cause(calculs,insufisance_renale).
cause(insufisance_biliaire,jaunisse).
cause(obstruction_veine_hepatique,jaunisse).
cause(metastases,hepatiteB).
cause(vaumissement,diarree).
symptome(pierre,fatigue).
symptome(pierre,fievre).
symptome(marco,calculs).
symptome(georges,rhume).
symptome(georges,X).
est_atteint_de(marco,meningite).
est_atteint_de(charles,jaunisse).
```

Outils

Constructeur

Situation Dialogue/Arbre Resolution

Terminer Session

Reponses/Questions. Usager

Voir Domaine

Sais Pas

Resultat =

Reponse =

Pourquoi? (Situation Dialogue:Question)

Quoi?

Comment?

[Learner] Je ne sais pas

[Feedback] On veut prouver le fait `est_atteint_de(pierre,M)` **Recevoir Explication**

[Tuteur] Que signifie cette requete? **Repondre**

[Learner] Trouver que le FAIT `est_atteint_de(pierre, M)` est Vrai

[Feedback] Cette requete signifie simplement qu'on veut determiner les valeurs de `M` telles que le FAIT `est_atteint_de(pierre,M)` existe (est vrai) dans la BC

[Tuteur:Resume]Ok, Maintenant on sait ce qu'on cherche. On va utiliser la Base de Connaissances pour cela

[Tuteur] Indique le premier element de la BASE de CONNAISSANCES qui peut etre utilise pour cette preuve

[Learner] `est_atteint_de(X,M):-symptome(X,S),cause(S,M).`

[Feedback] =>Plutot: `est_atteint_de(X,Y):-symptome(X,S), cause(S, M)` **Recevoir Explication**

[Tuteur] Indique l'element "HEAD" de la clause de HORN `est_atteint_de(X,M):- ...`

Question permettant de diagnostiquer l'acquisition de la capacité "Identifier une règle-Prolog"

Figure VI.2. Construction dialogique d'une solution à l'interface de Prolog-Tutor

Deuxièmement, une liberté accrue peut être concédée à l'apprenant : sans lui poser des questions explicites, ses actions à l'interface du STI peuvent être observées dans le but d'inférer sa démarche (Figure VI.3). Cette démarche est alors interprétée en termes de l'usage qui est fait des concepts et des principes de la résolution d'un but. Il s'agit d'un diagnostic épistémique basé sur une évaluation reconstructive de la solution de l'apprenant.

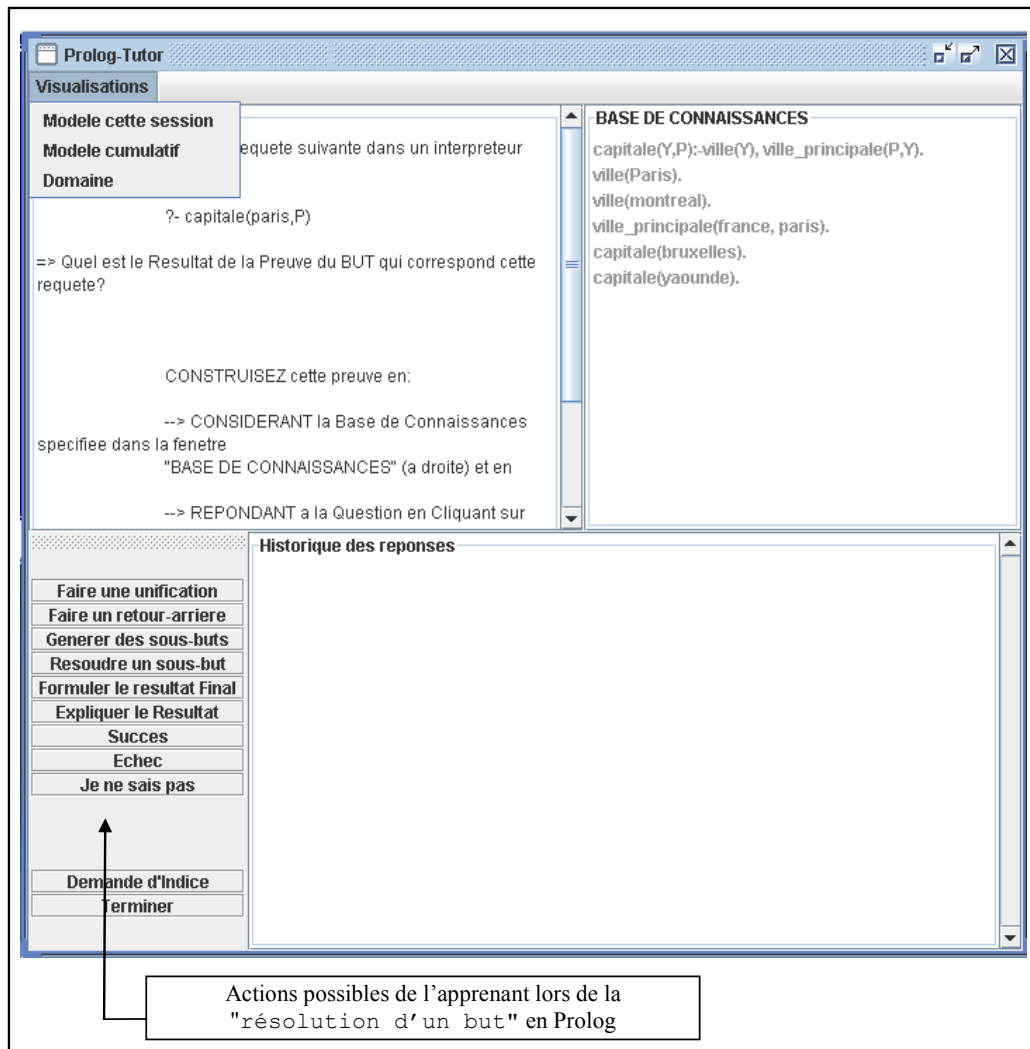


Figure VI.3. Construction libre d'une solution à l'interface de Prolog-Tutor

Le choix de l'approche de DC dépend surtout des ressources qu'on veut mobiliser au niveau de l'implémentation. L'évaluation permet d'associer directement les actions observables à des capacités. L'interprétation reconstructive nécessite une définition de l'espace problème associé à la résolution d'un but. Étant donné que l'application de la résolution d'un but ne génère pas un grand nombre d'actions possibles de la part de l'apprenant, sa modélisation comme un processus de résolution de problème (avec un espace problème) préconise des efforts inutiles. Pour cet exemple particulier, le DC

épistémique basé sur une évaluation sans inférence des réponses de l'apprenant a donc été choisi.

6.3.1.3. Approche de réflexion

Le cadre de spécification suggère que tous les types de réflexion sont applicables. La modélisation interactive de l'apprenant à travers les dialogues tutoriels a été choisie comme moyen de stimulation de la pensée réflexive. De fait, l'engagement d'utiliser une évaluation sans inférence implique que des réponses de l'apprenant à des questions spécifiques doivent être obtenues. Les dialogues tutoriels sont un moyen idéal de recueillir ces informations, d'autant plus qu'ils permettent également d'examiner les relations entre les sous-capacités d'une capacité diagnostiquée à un premier niveau d'inférence.

6.3.1.4. Implémentation

La section suivante présente l'implémentation de cette première spécification dans Prolog-Tutor. Au fil de cette présentation, il est important de relever comment les recommandations du CSDC ont été élaborées dans ce contexte particulier. Elles suggèrent d'associer explicitement une capacité à chaque question posée à l'apprenant. Elles suggèrent aussi d'élaborer autant que faire se peut chaque question posée, de manière à l'amener à s'articuler sur les différents éléments d'une capacité. La pertinence de ces recommandations tient du fait que dans ce contexte pédagogique particulier, l'un des objectifs est de diagnostiquer si l'apprenant comprend les interrelations entre les éléments d'une capacité.

6.3.2. Illustration dans Prolog-Tutor

Les recommandations du Tableau VI.3 (ci-dessus, en page 169) suggèrent une interaction au cours de laquelle les diagnostics doivent être posés à travers un dialogue tutoriel. L'approche préconisée dans Prolog-Tutor est présentée dans les prochaines lignes.

6.3.2.1. Comment le dialogue tutoriel permet-il un DC ?

L'apprenant est initialement confronté à une situation de *résolution de but* en Prolog. Si la solution proposée par ce dernier est incorrecte, le système décompose le problème en élaborant un ensemble de questions qui seront traitées par l'apprenant à travers un dialogue tutoriel. Chaque question correspond à une étape de la *résolution d'un but* en Prolog et incorpore deux objectifs pédagogiques : un DC épistémique et une réflexion implicite de

l'apprenant. En fait, on pense qu'une amélioration par rapport à la capacité associée à une question peut apparaître, comme un effet de bord de cette réflexion implicite (Dimitrova, 2002).

La Figure VI.2 (ci-dessus, en page 168) illustre le dialogue tutoriel qui a été implémenté dans Prolog-Tutor selon les recommandations de la spécification du Tableau VI.3. Ce dialogue possède une structure hiérarchique sur deux niveaux. Le premier niveau du dialogue correspond aux questions que le tuteur posera à l'apprenant en lien direct avec le processus de résolution d'un but- la notation D1 est dorénavant utilisée pour référer au contenu de ce niveau du dialogue -. Les questions suivantes ont un but spécifique de DC:

⇒ « Quel est le but à prouver? » est rattachée à la capacité « *Comprendre l'intuition de la résolution d'un but?* »

⇒ « Quel est le premier élément de la Base de Connaissances qui peut être utilisé pour résoudre ce but? » est rattachée aux capacités « *Comprendre l'intuition de la résolution d'un but* » et « *Comprendre l'intuition d'une règle-Prolog* ».

De cette façon, lorsque l'apprenant ne répond pas correctement à une de ces questions, les capacités qui y sont associées peuvent être directement diagnostiquées comme causant des difficultés à l'apprenant.

6.3.2.2. Pensée réflexive: comment le résultat du DC est-il exploité?

Pour répondre à cette question, on peut se référer à la situation du Tableau VI.4. L'apprenant doit formuler le résultat d'une opération en Prolog. Plus précisément, il doit formuler le résultat d'une première tentative de résolution du but `grand_father(X, joseph)` avec la règle `grand_father(X, y) :- father(X, Z), father(Z, y)`, étant donné la base de connaissances spécifiée dans l'énoncé. De prime abord, la question est suggestive en ce qu'elle indique à l'apprenant qu'une «règle Prolog» peut être utilisée. Il va de soi qu'idéalement, l'apprenant percevra l'intégralité de ces suggestions à travers une *réflexion implicite*, bien que cela ne soit pas toujours garanti puisque rien n'est observable dans ce cas.

En effet, pour comprendre la question qui lui est posée, les suggestions sous-jacentes et surtout, pour pouvoir y répondre, on peut considérer que l'apprenant doit accéder à une

représentation des principes et des concepts nécessaires. Dans ce cas, ces principes et concepts sont ceux qui contribuent au déploiement des capacités « *utiliser* une règle Prolog dans la résolution d'un but » et « *manipuler* une règle Prolog dans la résolution d'un but ». Un exemple d'un tel principe correspond à l'énoncé suivant :

[SI la tête de la REGLE-PROLOG EST unifiable avec le but à résoudre ALORS utiliser cette REGLE-PROLOG pour effectuer cette RESOLUTION]

Tableau VI.4. Exemple de dialogue tutoriel tel que conçu dans Prolog-Tutor

Énoncé de la question principale de l'exercice	
<p>On considère le fait <code>grand_father(X, joseph)</code>. Donner la solution de la résolution de ce fait, en utilisant la base de connaissances ci-dessous.</p> <p style="text-align: center;"><u>Base de connaissances</u></p> <pre> grand_father(X, Y) :- father(X, Z), father(Z, Y) grand_father(X, john) :- grand_son(john, X) . grand_father(X, P) :- father(X, P), father(Z, P) father(stepp, anton) . father(anton, ken) . father(anton, joseph) </pre>	
Réponse de l'apprenant	
<p>L'apprenant donne une réponse incorrecte pour cet exercice. Un dialogue tutoriel est déclenché afin d'effectuer une construction progressive de la solution. ⇒ Décomposition du problème à travers un dialogue tutoriel</p>	
Dialogue Tutoriel	
Réflexion <i>implicite</i> qu'on espère déclencher chez l'apprenant	Illustration
<p>⇒ [Intellectualisation] [On doit utiliser la règle <code>grand_father(X, P) :- father(X, P), father(Z, P)</code> pour résoudre ce but]</p> <p>⇒ [Raisonnement] [<code>grand_father(X, P) :- father(X, P), father(Z, P)</code> doit être utilisée parce que son élément <code>head</code> est unifiable avec le fait à résoudre]</p>	<p>1. Tuteur[Question] Quel est le résultat de la première tentative de résolution avec <code>grand_father(X, P) :- father(X, Z), father(Z, P)</code> ?</p> <p>2. Apprenant[Réponse] C'est un succès</p>

En d'autres termes, la réflexion à ce premier niveau du dialogue tutoriel peut se justifier en assumant que le contenu d'une question du dialogue tutoriel suggère : (1) les étapes nécessaires dans une démarche correcte de solution ; (2) les éléments pertinents pour la résolution de chaque étape, en l'espèce, les éléments conceptuels de même que les principes du domaine qui pourraient être applicables. Dans cette optique, le dialogue tutoriel

provoquerait une réflexion implicite (non observable) basée sur ces éléments et cette réflexion soutiendrait à son tour la réponse proposée par l'apprenant.

6.3.2.3. Rétroaction du résultat de la réflexion sur le DC

Pour chaque question du niveau D1, les capacités associées peuvent incorporer des sous capacités. Dans cette situation, la spécification indique que le dialogue doit être élaboré de manière à cibler le plus précisément possible quelle "sous capacité" est responsable des difficultés de l'apprenant. Ces difficultés sont identifiées lorsque ce dernier répond incorrectement à une question au niveau D1, c'est-à-dire au terme d'une présumée réflexion implicite.

Dans la structure de notre dialogue tutoriel, le deuxième niveau permet de réaliser cet objectif - la notation D1.2 est adoptée pour référer au contenu de ce sous dialogue - Les questions qui sont adressées à l'apprenant dans D1.2 abordent les différents aspects de la dernière capacité diagnostiquée au niveau D1. La Figure VI.4 illustre une telle situation dans Prolog-Tutor. Les deux questions du niveau D1.2 ont respectivement pour but de déterminer :

- (1) Si l'apprenant comprend qu'un but à résoudre correspond à un fait;
- (2) Si l'apprenant comprend qu'un fait n'est qu'une représentation formelle d'une assertion.

Ces questions déclinent deux aspects de la capacité "Comprendre l'intuition de la résolution d'un but" qui a été diagnostiquée au niveau D1. Les questions du niveau D1.2 n'auront pas nécessairement un lien direct avec l'exercice qui est actuellement résolu à travers le dialogue tutoriel. Le sous-dialogue D1.2. permet de raffiner le diagnostic posé au niveau D1, favorisant une délimitation de plus en plus précise des besoins d'apprentissage. Ce sous-dialogue permet aussi de confirmer le DC initial.

Question au niveau D1
 ✓ Diagnostiquer l'acquisition de la capacité:
 "Utiliser une règle-Prolog dans la résolution d'un But"

▪ Diagnostic au niveau D1:
 Capacité "Exécuter la résolution d'un But avec une règle-Prolog" non acquise
 ▪ Pourquoi? Causes possibles :
 ✓ Incapacité à Identifier une règle
 ✓ Incapacité à associer une règle et un but à résoudre
 ⇒ Raffiner avec le Niveau D1.2

Questions du niveau D1.2

Prolog-Tutor Resol

Visualisations

ENONCE
 => Supposons qu'on a la requete suivante dans un interpreteur SICSTUS PROLOG:
 ?- est_atteint_de(pierre,M)
 => Quel est le Resultat de la Preuve du BUT qui correspond cette requete?
 CONSTRUISEZ cette preuve en:
 --> CONSIDERANT la Base de Connaissances specifiee dans la fenetre "BASE DE CONNAISSANCES" (a droite) et en
 --> REpondant aux questions du systeme par le biais des BOUTONS du panneau Reponses/Questions. Usager

BASE DE CONNAISSANCES
 est_atteint_de(X,M):-symptome(X,S),cause(S,M),
 cause(fievre,infection),
 cause(taux_eleve_gb,pneumonie),
 cause(calculs,insufisance_renale),
 cause(insufisance_biliaire,jaunisse),
 cause(obstruction_veine_hepatique,jaunisse),
 cause(metastases,hepatiteB),
 cause(vaumissement,diarree),
 symptome(pierre,fatigue),
 symptome(pierre,fievre),
 symptome(marco,calculs),
 symptome(georges,rhume),
 symptome(georges,X),
 est_atteint_de(marco,meningite),
 est_atteint_de(charles,jaunisse).

Outils
 Constructeur
 Situation Dialogue/Arbre Resolution
 Terminer Session

Reponses/Questions. Usager
 Voir Domaine
 Sais Pas
 Resultat =
 Reponse =
 Pourquoi? (Situation Dialogue/Question)
 Quoi?
 Comment?

[Learner] Trouver que le FAIT est_atteint_de(pierre, M) est Vrai
 [Feedback] Cette requete signifie simplement qu'on veut determiner les valeurs de M telles que le FAIT est_atteint_de(pierre,M) existe (est vrai) dans la BC
 [Tuteur:Resume]Ok, Maintenant on sait ce qu'on cherche. On va utiliser la Base de Connaissances pour cela
 [Tuteur] Indique le premier element de la BASE de CONNAISSANCES qui peut etre utilise pour cette preuve
 [Learner] Je ne sais pas
 [Feedback] =>Plutot: est_atteint_de(X,Y):-symptome(X,S), cause(S, M) Recevoir Explication
 [Tuteur] Indique l'element "HEAD" de la clause de HORN est_atteint_de(X,M):- ...
 [Learner] Reponse: est_atteint_de(X,M)
 [Feedback]Bion.
 [Tuteur] Quel est le resultat de l'unification de est_atteint_de(X, M) avec le BUT a resoudre

Figure VI.4. Raffinement et confirmation d'un DC dans Prolog-Tutor

6.3.2.3.1. Ciblage des besoins d'apprentissage

Dans le cours de D1.2, si l'apprenant ne répond pas correctement à une des questions, la sous capacité qui y est associée est formellement diagnostiquée. Dès lors, on considère que le système effectue une réflexion au sens informatique. En effet, le diagnostic posé en D1 est remplacé par celui posé en D1.2. On peut donc dire que le système révisé et raffine son hypothèse initiale à propos de l'état cognitif de l'apprenant. Le besoin d'apprentissage ainsi localisé peut être satisfait à travers deux actions pédagogiques principales.

D'une part, le tuteur peut continuer de raffiner le diagnostic en générant récursivement d'autres sous dialogues (D1.n, $n > 2$).

D'autre part, lorsque le niveau de raffinement désiré est atteint, le tuteur peut cibler les mesures réparatrices sur la capacité correspondant au DC final. Il faut se rappeler que le niveau de raffinement est le niveau de descente dans la recherche des causes qui expliquent une réponse incorrecte de l'apprenant. Le premier niveau de raffinement est la génération des causes potentielles. Le deuxième niveau de raffinement est la sélection des causes les plus probables parmi ces causes potentielles. Si l'on avait eu un troisième niveau de raffinement, il aurait fallu les causes les plus probables de chacune des causes identifiées au deuxième niveau et, ainsi de suite de manière récursive. Dans Prolog-Tutor, le niveau de raffinement est fixé à 2, étant donné que la partie de ce domaine qui est abordée (le vocabulaire et les processus d'inférences) est très peu dense en termes de contenu. Une fois qu'une capacité a donc été diagnostiquée au niveau 2, le tuteur propose un feedback articulée sur deux explications. La première explication est générale et correspond à la nature de la capacité en question. La seconde explication est spécifique et correspond à la mise en œuvre de cette capacité dans le contexte particulier de l'exercice qu'on essaie de résoudre.

6.3.2.3.2. Confirmation du DC initial

Le raffinement du diagnostic effectué au niveau D1.2 permet également de confirmer l'hypothèse initialement posée au niveau D1. D1.2. permet de:

⇒ *confirmer le diagnostic posé au niveau D1* lorsqu'au moins une des réponses fournies par l'apprenant au niveau D1.2 est incorrecte. Non seulement, le DC posé au niveau D1 est confirmé (l'étudiant éprouve effectivement des difficultés avec la capacité associée à la question posée au niveau D1), mais, il est précisé (une ou plusieurs sous capacités de la capacité initialement diagnostiquée sont formellement identifiées comme problématiques). Le Tableau VI.5 illustre comment cela peut se réaliser dans un dialogue tutoriel. Après avoir initialement diagnostiqué que l'apprenant a de la difficulté à "utiliser une règle Prolog" dans l'"application de la résolution d'un but", le système désire inférer quel(s) aspect(s) de cette capacité est effectivement problématique. Trois aspects ont été identifiés dans

ce cas d'espèce : "Identifier une règle Prolog", "Comprendre la signification d'une règle Prolog" et "Utiliser une règle Prolog dans la résolution d'un but". Le prochain défi de l'apprenant consiste à répondre à des questions spécifiques sur chacune de ces trois capacités. Par exemple, si l'apprenant n'arrive pas à identifier les conditions et les conséquences représentées à travers une règle Prolog, on peut diagnostiquer que la capacité «Comprendre la signification d'une règle Prolog» est problématique.

⇒ *infirmer le diagnostic posé au niveau D1* lorsque l'apprenant répond correctement à toutes les questions du sous dialogue D1.2. Ce cas de figure peut être interprété comme une indication qu'un problème existe: soit dans la formulation de la question au niveau D1, soit dans la modélisation des sous-capacités au niveau D1.2, soit dans le mécanisme d'inférence qui a permis d'identifier quelles sous capacités seront examinées au niveau D1.2. Dans Prolog-Tutor, ce deuxième type de rétroaction demeure théorique puisqu'elle n'est pas vraiment exploitée à ce stade du développement du système. En fait, son implémentation est réduite à une indication dans un fichier correspondant à la trace des décisions du module pédagogique.

Il faut également relever le fait que le dialogue D1.2 permet de stimuler une réflexion implicite au même titre que le dialogue D1. Cette réflexion est quand même plus élaborée que celle provoquée en D1 puisque les questions déclinent les sous-capacités de la capacité diagnostiquée en D1.

Tableau VI.5. Illustration de l'usage pédagogique de l'élaboration du DC

Dialogue	Réflexion qu'on désire stimuler
<p>Module pédagogique du STI</p> <p>Quelle est la première règle de la base de connaissance qui correspond au but : grand_father(X, joseph?)</p>	<p>Réflexion implicite, anticipée ou attendue chez l'apprenant</p> <p>⇒ Quel est le rôle d'une règle-Prolog dans la résolution d'un but?</p> <p>⇒ Comment faut-il associer une règle-Prolog à un but à résoudre?</p>
<p>Apprenant: "Je ne sais pas"</p>	
<p>Module pédagogique du STI</p> <p>⇒ Observation: L'apprenant n'a pas pu répondre correctement à la dernière question</p> <p>⇒ Inference: Diagnostic Épistémique</p> <ul style="list-style-type: none"> ▪ Premier niveau d'inférence: <ul style="list-style-type: none"> → Incapacité à <Utiliser> une <règle-Prolog> dans la <résolution d'un But> ▪ Deuxième niveau d'inférence : <ul style="list-style-type: none"> → Incapacité à: <ul style="list-style-type: none"> <Identifier> une <regle-Prolog>? <Comprendre> la signification d'une <regle-Prolog>? <Associer> une <regle-Prolog> et un <But à résoudre>? <p>⇒ Dialogue de niveau D1.2. pour déterminer la réponse aux 3 questions</p>	<p>Réflexion du côté du système tuteur (Module pédagogique)</p> <p>⇒ Raffiner le DC à propos d'une capacité C (C = "Utiliser une règle-Prolog dans la résolution d'un But")</p> <p>⇒ Déterminer quels aspects de C (ou les sous capacités de C) sont responsables des difficultés de l'apprenant</p>

6.4 . DC avec réflexion basée sur un dialogue tutoriel à structure réflexive

Cette section illustre la mise en œuvre d'une spécification de DC basée sur un dialogue tutoriel dont les questions sont explicitement conçues de manière à stimuler la pensée réflexive selon la théorie de Dewey. À l'instar de la méthode de présentation adoptée dans la section précédente, la description de cette spécification précède son illustration dans Prolog-Tutor.

6.4.1. Spécification

Dans l'exemple précédent, on a considéré que le dialogue tutoriel provoque implicitement une réflexion chez l'apprenant. Cependant, cette position ne s'appuie que sur le caractère interactif d'un tel dialogue étant donné que cette réflexion demeure implicite et non observable. À présent, on aimerait favoriser une réflexion explicite au sens de Dewey. En utilisant le même contexte pédagogique et la même spécification que dans l'exemple précédent, il est maintenant question d'illustrer un dialogue tutoriel dont la structure a pour but d'extérioriser explicitement les étapes de la pensée réflexive chez l'apprenant.

6.4.2. Illustration dans Prolog-Tutor

Lorsque Prolog-Tutor diagnostique que l'apprenant éprouve des difficultés par rapport à une capacité, le sous-dialogue associé peut être organisé de manière à stimuler une réflexion explicite à propos de cette capacité. On aimerait structurer D1.2. de manière à favoriser l'occurrence des composantes de la pensée réflexive. Pour augmenter les chances de réutilisation d'un tel dialogue dans des contextes différents, il est important de séparer sa structure de son contenu. Si l'on considère la structure de dialogue réflexif proposée au Chapitre III (Tableau III.8, en page 89): l'examen de chaque sous-capacité à aborder doit s'amorcer en posant une question où l'apprenant est obligé de prendre une position. Dans Prolog-Tutor, cette question correspond à une assertion par rapport à laquelle l'apprenant doit signifier son accord ou son désaccord (Figure VI.5). La particularité de cette question est qu'au-delà de sa nature intrinsèquement interrogative, elle doit être assez problématique pour forcer l'apprenant à faire un examen poussé de la situation. Cette question correspond à la première phase de stimulation de la pensée réflexive. Lorsque l'apprenant prend position, deux objectifs sont poursuivis dans le reste du sous-dialogue D1.2. Si l'apprenant a adopté une position erronée, l'objectif du système est de l'amener à se contredire. Dans le cas contraire, l'objectif du système est de l'amener à se justifier.

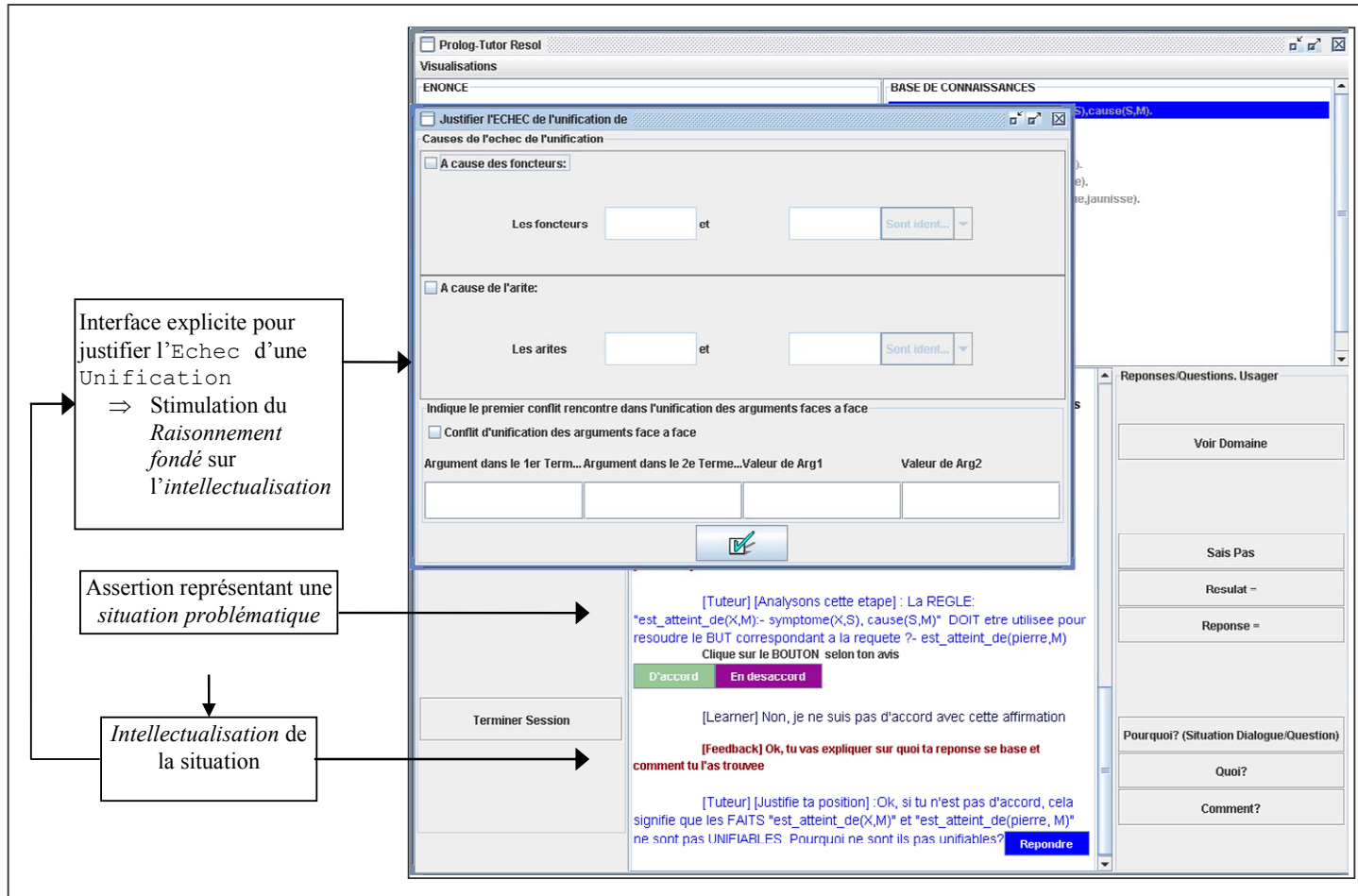


Figure VI.5. Prolog-Tutor: Dialogue dont la structure favorise la pensée réflexive

Dans tous les cas, la réflexion s'exprime explicitement en amenant l'apprenant à:

- (1) relever les informations pertinentes à la résolution du problème qui sous-tend l'assertion proposée (intellectualisation de la situation) ;
- (2) utiliser les concepts appropriés pour résoudre ce problème (raisonnement) ;
- (3) appliquer les principes appropriés pour résoudre ce problème (raisonnement) ;
- (4) tirer les conclusions qui s'imposent et à les arrimer avec la position de départ relativement à l'assertion posée (conclusion fondée sur des faits et un raisonnement correspondant).

Qu'il s'agisse pour l'apprenant de justifier sa réponse ou encore de constater qu'il se contredit, ces étapes correspondent aux phases de la pensée réflexive. À titre d'illustration, on peut considérer le sous-dialogue réflexif de la Figure VI.5 (ci-dessus en page 182). L'apprenant est invité à se prononcer par rapport à l'assertion selon laquelle la première tentative de résolution du but "est_atteint_de(pierre,M)" avec la règle Prolog "est_atteint_de(X,M) :- symptome(X,S), cause(S,M)" est un Echec (ce qui est une assertion exacte dans ce cas). Si l'apprenant indique qu'il est d'accord, alors l'objectif du système est d'amener l'apprenant à se justifier ce à travers une réflexion explicite qui dévoilerait le caractère fondé de sa conclusion. Avant d'examiner le sous-dialogue D1.2. qui suit cette assertion, analysons le raisonnement sur lequel elle se fonde. La résolution de est_atteint_de(pierre,M) avec la règle Prolog proposée passe par la résolution des sous-buts symptome(pierre,S) et cause(S,M). La preuve de est_atteint_de(pierre,M) est un Echec s'il est impossible de prouver au moins un de ces sous-buts. Dans ce cas particulier, la preuve de symptome(pierre,S) est un Succès avec S = fièvre. La preuve de cause(fièvre,M) est un Echec. Au-delà de l'exécution de ces sous-opérations, l'apprenant doit intégrer la logique sous-jacente à travers un certain nombre de capacités. Ainsi, pour prouver qu'au-delà de trouver la bonne réponse (être en accord avec l'assertion proposée par le système), il comprend le fondement (le raisonnement sous-jacent) à cette réponse, l'apprenant est invité à répondre aux questions suivantes:

- (1) "Quelle est la preuve de la première condition associée à cette règle, dans le contexte de la preuve est_atteint_de(pierre,M)?"
- (2) "Quelle est l'expression de la deuxième condition après cette preuve?"

(3) "Quel serait la preuve de cette condition ainsi exprimée?".

Dans Prolog-tutor, on ne se contente pas de la simple réponse de l'apprenant. Le système génère des interfaces qui permettent à l'apprenant d'exprimer avec le plus de détails possibles le raisonnement qui supporte sa réponse. Pour la première question de l'exemple énoncé ci-dessus, Prolog-Tutor génère une interface où l'apprenant doit non seulement formuler le sous-but qu'il essaie de prouver, mais indiquer quel élément de la base de connaissances il utilise à cette fin (Figure VI.6). De cette façon, il extériorise explicitement sa maîtrise de la capacité "Exécuter la résolution d'un but avec une règle-Prolog".

Finalement, l'on doit remarquer qu'on aurait pu obtenir les mêmes résultats sans passer par un dialogue tutoriel. Si l'on considère l'interface de la Figure VI.3 (ci-dessus, en page 172) l'apprenant peut indiquer directement sa démarche en choisissant l'ensemble des opérations qu'il effectue et en associant chaque opération à l'utilisation d'un concept ou à l'application d'un principe. Cette démarche correspondrait alors au processus communément dénommé *self-explanation* (Aleven, Popescu, et Koedinger, 2003; Chi et al., 1994). Le Tableau VI.6 décrit la correspondance entre chacune des questions posées dans le dialogue de la Figure VI.6 et les composantes de la pensée réflexive. Les points clés de la conception de ce type de dialogue portent sur le choix du contenu des questions posées pendant la phase de raisonnement: comment les présenter de manière à ce qu'elles incitent effectivement à utiliser la capacité qu'on désire examiner? Il s'agit en fait de la capacité nécessaire pour répondre correctement à l'assertion proposée au début de chaque sous dialogue D1.2. La nature d'une capacité informe la façon dont ces questions sont formulées. En effet, la nature d'une capacité indique l'objectif du processus de pensée réflexive dans un contexte particulier.

The screenshot shows the PROLOG-TUTOR interface with a 'Resolution d'un sous-but' dialog box. The dialog contains the following text:

Indique le sous-but à résoudre:

Indique l'élément correspon...

[Seulement pour les SOUS-BUTS] Aucun Terme de la Base de Connaissances ne peut être unifié avec ce sous-but

OK

Terminer Session

Recevoir Explication

[Tuteur] Donc si la preuve du BUT "est_atteint_de(pierre,M)" réussit tel que tu le soutiens, les preuves des FAITS "symptome(pierre,S)" ET "cause(S,M)" réussissent. Examinons cette preuve: Que donne la preuve du premier SOUS-BUT associée au BUT "est_atteint_de(pierre,M)" lorsqu'on utilise cette REGLE? Répondre

BASE DE CONNAISSANCES

- est_atteint_de(X,M): symptome(X,S), cause(S,M).
- cause(fievre, infection).
- cause(taux_eleve_gb, pneumonie).
- cause(calculs, insuffisance_renale).
- cause(insuffisance_biliaire, jaunisse).
- cause(obstruction_veine_hepatique, jaunisse).
- cause(metastases_hepatiteB).

Objectifs

- Calcul des prédicats
- Grammaire
- Traitement
- Unification
- Resolution
 - La resolution en gener
 - Les Variables et la Re
 - La resolution avec cou
 - Exercices sur la resolu
 - Exercice prototype
 - Exercice prototype

Questions. Usager

Voir Domaine

Sais Pas

Resultat =

Reponse =

Pour quoi? (Situation Dialogue/Question)

Quoi?

Comment?

✓ Sous-but à résoudre

✓ Élément de la Base de Connaissances utilise pour résoudre le sous-but

✓ Indiquer qu'aucun élément de la Base de Connaissance ne peut être utilisé pour résoudre ce sous-but

Figure VI.6. Explication de la démarche de résolution d'un sous-but

Tableau VI.6. Analyse d'un dialogue pour favoriser la réflexion explicite

Dialogue	Phase de réflexion explicite visée	Capacité associée
Système Tuteur La première tentative de preuve de <code>est_atteint_de(pierre,M)</code> avec la règle <code>est_atteint_de(X,M) :- symptome(X,S), cause(S,M)</code> est un Echec	Présentation d'une <i>situation problématique</i> à résoudre	⇒ Effectuer la résolution d'un but avec une règle-Prolog
Apprenant D'accord		
Système Tuteur Quelle est la preuve de la première condition associée à cette règle, dans le contexte de la preuve <code>est_atteint_de(pierre,M)</code> ?	Combinaison <i>Intellectualisation</i> [Quelle est la première condition ?] + <i>Raisonnement</i> [Si la preuve de cette condition réussit, alors il faut prouver la deuxième condition pour arriver au résultat]	⇒ Comprendre l'intuition de la résolution d'un But avec une règle-Prolog ⇒ Manipuler une règle Prolog dans la résolution d'un but ⇒ Effectuer la résolution d'un but avec un fait
Apprenant Succès. S = fièvre		
Système Tuteur Quelle est l'expression de la deuxième condition après cette preuve ?	Combinaison <i>Intellectualisation</i> [Quelle est la deuxième condition ? Quel est le résultat de la preuve de la condition précédente?] + <i>Raisonnement</i> [Comment propager les résultats de la preuve de la première condition]	⇒ Manipuler une règle Prolog dans la résolution d'un but
Apprenant cause(fièvre,M)		
Système Tuteur Quelle serait la preuve de cette condition ainsi exprimée ?	<i>Raisonnement</i> [Si la preuve de ce but est un échec alors la preuve de <code>est_atteint_de(pierre,M)</code> est un Échec]	⇒ Effectuer la résolution d'un but avec un fait

Dewey a souligné que les activités intellectuelles qui sous-tendent le processus de jugement sont analogues à la pensée réflexive. De ce fait, la description des objectifs du processus de jugement, telle que proposée par Drake (1976) peut être exploitée (Tableau

VI.7). Cette description explique en quoi le jugement consiste lorsqu'il concerne les capacités relatives aux concepts, aux principes et aux lois, et à l'argumentation. Les capacités visées dans Prolog-Tutor s'apparentent surtout aux concepts, aux principes (et à des règles) et à l'applications d'un ensemble de principes dans une procédure.

Tableau VI.7. Buts de la pensée réflexive selon Drake (1976)

Capacité	But de la pensée réflexive
<ul style="list-style-type: none"> ▪ Identifier un concept 	Déterminer la justesse d'une classification; Vérifier la signification d'un énoncé; Reconnaître un attribut ; Justifier l'inexactitude d'une classification
<ul style="list-style-type: none"> ▪ Utiliser un principe ▪ Appliquer un principe 	Identifier les prémisses et les conséquences d'une règle ou d'un principe; Appliquer les prémisses d'une règle aux faits d'une situation; Vérifier qu'une proposition ou qu'une assertion est supportée par un principe.
<ul style="list-style-type: none"> ▪ Vérifier un argument 	Déterminer une prémisse; Déterminer une conséquence d'une assertion; Arriver à une contradiction; Arriver à une confirmation.

Le Tableau VI.8 résume le déroulement de la pensée réflexive relativement à la capacité à "*appliquer*" un principe. Ce modèle a inspiré l'élaboration des dialogues tutoriels de Prolog-Tutor dont le but est de favoriser une pensée réflexive explicite.

Tableau VI.8. Réflexion explicite pour l'application d'un principe

Stimulation de la pensée réflexive dans sur la capacité " <i>Appliquer</i> le principe P1" ⇒ But de la pensée réflexive: Établir qu'une proposition S est supportée ou contredite par le principe P1	
Composante de la pensée réflexive	Interprétation dans le contexte de l'application d'un principe
Situation Problématique et Intellectualisation	1- Décrire les faits importants de la situation problématique 2- Mettre en exergue les faits, les conclusions ou les solutions de cette situation (y compris les solutions contradictoires ou non triviales, afin de stimuler une réelle réflexion).
Raisonnement à propos du principe P1	1- Proposer une solution à partir des faits relevés; 2- Vérifier si cette solution est contredite ou confirmée par P1.

6.5 DC avec réflexion basée sur l'édition du modèle de l'apprenant

Cette section illustre une deuxième spécification de DC dans Prolog-Tutor. Cette spécification est fondée sur l'approche *Open Learner Modelling* où le DC initial est effectué par l'apprenant lui-même.

6.5.1. Spécification

Comment peut-on envisager un scénario de réflexion sur l'action dans une perspective proactive? La spécification suivante (Tableau VI.9) a pour but de donner à l'apprenant l'occasion d'auto-évaluer quantitativement son niveau d'acquisition par rapport à différents objectifs d'apprentissage, tout en justifiant ses propositions.

Tableau VI.9. Spécification d'un processus de DC: contexte empirique/béhavioriste

Paramètre	Instance de paramètre	Précisions
Contexte Pédagogique		
Perspective de la cognition	Empiriste	✓ Les connaissances de l'apprenant sont considérées en termes de quantité d'information acquise
Perspective des mécanismes d'apprentissage	Béhavioriste	✓ Seules les réponses de l'apprenant en présence d'un exercice précis sont considérées
Perspective de l'instruction et du design de l'instruction	Non spécifié dans ce cas	✓ Aucune précision nécessaire
Propriétés du DC		
Type de diagnostic	Comportemental (Évaluation sans inférence)	✓ Basé sur l'évaluation des réponses de l'apprenant à des exercices spécifiques aux connaissances dont le niveau d'acquisition est négocié
Type de réflexion	Approche <i>Open Learner Modelling</i>	✓ Basée sur l'édition et la négociation des niveaux d'acquisition par rapport aux connaissances (ou aux capacités correspondant à des objectifs d'apprentissage)
Usage pédagogique (Réflexion du côté du système tuteur)	<ul style="list-style-type: none"> - Évaluation focalisée par connaissance ou par objectif d'apprentissage - Ajustement des paramètres de mise à jour du modèle de l'apprenant s'il y a lieu 	<ul style="list-style-type: none"> ✓ Utiliser les performances de l'apprenant aux exercices spécifiques à une connaissance ou à une capacité pour confirmer son autodiagnostic ; ✓ En cas de différence significative entre le niveau d'acquisition auto diagnostiqué (et confirmé) et le niveau d'acquisition inféré par le système, indiquer qu'un ajustement des paramètres de mise à jour du modèle de l'apprenant est nécessaire.

Paramètre	Instance de paramètre	Précisions
Implémentation		
Recommandations relatives aux représentations de connaissances nécessaires	- Banque d'exercices	✓ Banques d'exercices spécifiques à chaque connaissance dans Prolog-Tutor (ou capacité correspondant à un objectif d'apprentissage)
	- Représentation des dialogues tutoriaux	✓ Relier chaque question à une capacité visée comme objectif d'apprentissage et faire le lien avec le modèle de l'apprenant
	- Représentation du modèle de l'apprenant	✓ Réseau bayésien correspondant à une couverture (<i>overlay</i>) de la représentation du domaine des connaissances
Recommandations relatives à l'algorithme correspondant au processus de DC	- Recueillir et évaluer les réponses de l'apprenant dans un dialogue tutoriel	✓ Diagnostic épistémique
	- Raffiner les questions posées à l'apprenant ainsi que les diagnostics posés	✓ Réflexion de l'apprenant ✓ Réflexion du côté du système

6.5.1.1. Contexte pédagogique: paradigme de cognition

On désire permettre à l'apprenant de s'impliquer dans la construction de son modèle en indiquant le niveau d'acquisition qu'il estime posséder relativement à un ensemble de capacités. L'apprenant doit proposer un indicateur quantitatif de son niveau d'acquisition. Si ce niveau est supérieur à celui qui est estimé par le système, il doit se justifier en résolvant un exercice de performance. Le niveau de difficulté de l'exercice représente le niveau d'acquisition auquel l'apprenant prétend. Seule la réponse finale de ce dernier pour cet exercice est considérée par le système. Si elle est correcte, la modification demandée est apportée au modèle. Sinon, elle demeure telle quelle, jusqu'à ce que l'apprenant propose un niveau d'acquisition pour lequel il est capable de se justifier effectivement.

Ce contexte pédagogique est empiriste dans la mesure où les connaissances de l'apprenant sont considérées en termes de *niveau* d'acquisition. **L'apprentissage est appréhendé dans une perspective béhavioriste**: seule la réponse de l'apprenant est considérée, étant donné les stimuli représentés par les conditions des exercices de performance.

6.5.1.2. Type de DC

Selon le CSDC, on peut envisager un DC comportemental avec spécifiquement une évaluation sans inférence. On peut aussi envisager un DC épistémique sur la base d'une évaluation sans inférence ou d'une interprétation reconstructive de la solution de l'apprenant. Le choix dépend de la position du designer pédagogique quant aux nécessités de modélisation qui seront consenties dans le futur module de DC.

Les relations entre d'une part les différents buts que pourrait poursuivre l'apprenant lors de la construction d'une solution et d'autre part cette solution (ou les parties de cette solution) peuvent être représentées. Dans ce cas, un diagnostic comportemental par interprétation reconstructive de la solution de l'apprenant est possible. On peut en plus représenter des relations entre les buts de l'apprenant et les capacités du domaine qui soutiennent la poursuite de ces buts. Dans ce cas, un diagnostic épistémique serait possible suite au diagnostic comportemental des buts de l'apprenant.

Lorsque l'investissement consenti dans l'implémentation du futur module de DC est modeste (peu d'efforts de modélisation et de programmation), on peut envisager un diagnostic épistémique sur la base d'une évaluation sans inférence. Dans ce cas, il faudrait représenter des relations entre les propriétés des solutions de l'apprenant et les capacités du domaine. Dans cette circonstance particulière, l'apprenant s'implique dans la construction de son modèle en fournissant des solutions à des exercices de performance. Seule la nature de la solution finale est considérée comme garant du niveau d'acquisition par rapport à une capacité représentée dans le modèle de l'apprenant. Pour cela, une évaluation sans inférence est suffisante. Il faut noter que le système effectue un diagnostic comportemental par évaluation sans inférence, tandis que la finalité est un DC épistémique. En effet, l'autodiagnostic de l'apprenant concerne son niveau d'acquisition par rapport à un ensemble de connaissances (ou de capacités correspondant à des objectifs d'apprentissage), lequel est confirmé à travers le DC comportemental effectué par le système. Cet exemple illustre un DC épistémique basé sur une évaluation sans inférence.

6.5.1.3. Approche de réflexion

Les recommandations du CSDC sur l'intégration de la réflexion suggèrent une visualisation et une édition du modèle de l'apprenant dans ce contexte pédagogique. Dans ce cas, l'édition

négociée du contenu du modèle de l'apprenant a été choisie parce que Prolog-Tutor est perçu comme un outil de support à l'apprentissage en vue d'une performance. Le besoin de vérifier en tout temps la justesse des représentations de l'état cognitif de l'apprenant est donc indiqué. De plus, l'implication de l'apprenant dans le DC vise à stimuler sa réflexion.

6.5.1.4. Implémentation

Deux données essentielles doivent soutenir l'implémentation d'une telle spécification de DC. Premièrement, une banque d'exercices de performance doit être disponible. Deuxièmement, deux mécanismes de regroupement de ces exercices doivent être mis en place. Le premier mécanisme doit permettre le regroupement d'exercices dans des catégories correspondant à une capacité visée dans le STI. Le second mécanisme doit permettre pour chaque capacité, un regroupement d'exercices dans des sous-catégories correspondant à des plages de niveaux d'acquisition. Le premier regroupement permet de sélectionner les exercices appropriés lorsque l'apprenant indique un niveau d'acquisition pour une capacité donnée. Le second regroupement permet de sélectionner un exercice à un niveau de difficulté qui représente le niveau d'acquisition que l'apprenant prétend avoir pour une capacité donnée.

Concernant le DC en tant que tel, l'apprenant doit d'abord effectuer un autodiagnostic en indiquant un niveau d'acquisition qu'il estime plus représentatif de son réel état de connaissances. L'apprenant doit résoudre un exercice de performance lorsque le niveau auquel il aspire est supérieur au niveau actuel et un exercice de révision dans le cas contraire. Au terme de l'exercice, le système confirme ou infirme l'autodiagnostic de l'apprenant selon la nature de la solution qu'il a proposée (correcte, incorrecte). Dès lors, l'interaction recommence avec l'apprenant qui peut indiquer un niveau d'acquisition pour une autre capacité.

6.5.2. Illustration dans Prolog-Tutor

La particularité de cette spécification est que la stimulation de la pensée réflexive a lieu avant le processus de DC en tant que tel. En fait, ce sont les résultats du dernier DC effectué par le système qui sont exploités lors de la réflexion de l'apprenant qui aboutit à son autodiagnostic. Les recommandations relatives à l'implémentation du DC préconisent une interaction entre l'apprenant et le système basée sur deux grandes étapes: l'autodiagnostic de l'apprenant qui succède à sa réflexion implicite à propos de son état cognitif, et le diagnostic du système qui

précède la confirmation (ou l'infirmité) de l'autodiagnostic de l'apprenant: il s'agit d'une autre instance de la boucle diagnostic-remédiation à travers une boucle diagnostic-réflexion.

6.5.2.1. L'autodiagnostic de l'apprenant permet-il la réflexion?

À la suite d'une session d'apprentissage, l'apprenant peut accéder à son modèle pour examiner les inférences qui ont été effectuées par le système. Dans Prolog-Tutor, ce modèle correspond à une couverture des objectifs d'apprentissage (capacités visées) – encore appelée *overlay* (Carr et Goldstein, 1977)-. Dans l'optique qualitative, le modèle représente les relations causales entre les capacités visées dans Prolog-Tutor et pour cette raison, il est interprété comme un réseau Bayésien (Pearl, 1988). Dans l'optique quantitative, le modèle associe à chaque capacité visée une probabilité d'acquisition. Ces probabilités sont obtenues par inférence bayésienne à partir des résultats de l'apprenant aux exercices résolus dans le système (Tchetagni et Nkambou, 2002). Ce sont ces probabilités que l'apprenant est invité à modifier selon ses propres estimations. Il s'agit d'un autodiagnostic qui devrait succéder à une réflexion implicite de la part de ce dernier. En effet, tel que discuté précédemment, pour modifier la probabilité d'acquisition d'une capacité représentée dans son modèle, un apprenant devrait au moins:

- (1) Connaître la capacité (de quoi s'agit-il, particulièrement dans le domaine étudié?)
- (2) Comprendre en quoi consiste la capacité?
- (3) S'auto évaluer par rapport à l'articulation effectuée en (2)
- (4) Comparer les résultats de cette évaluation à celui correspondant au niveau d'acquisition inféré par le système
- (5) Proposer un autre niveau d'acquisition s'il y a lieu

Au Chapitre III, nous avons montré comment ces étapes peuvent être interprétées en termes de composantes de la pensée réflexive selon notre théorie de référence en cette matière – la théorie de Dewey.

6.5.2.2. Rétroaction du résultat de la réflexion sur le DC

Si l'on considère le cas où l'apprenant indique une probabilité d'acquisition supérieure à celle qui a été inférée par le système; ce dernier génère un exercice de performance dont la résolution avec succès est interprétée comme un justificatif de l'estimation de l'apprenant (Figure VI.7).

Les exercices de performance sont purement behavioristes et leur seul objectif est de faire une évaluation sans inférence: caractériser la réponse de l'apprenant afin de déterminer si elle est correcte. Le niveau de difficulté d'un exercice de performance est proportionnel au niveau d'acquisition indiqué par l'apprenant. De cette façon, une réponse correcte permettra au système de confirmer ce niveau et d'ajuster en conséquence le niveau des autres éléments de connaissance. Il s'agit là de la principale rétroaction de la réflexion de l'apprenant sur les DC antérieurement posés par le système (réflexion du côté du système). La **Erreur ! Source du renvoi introuvable.** illustre un épisode de négociation dans Prolog-Tutor dans le cas où l'apprenant désire augmenter sa probabilité d'acquisition par rapport à la capacité "*Utiliser une règle-Prolog dans la résolution d'un but*". Le système génère un exercice spécifique: la question posée dans l'énoncé oblige l'apprenant à manipuler la règle-Prolog `malaria(X):-fever(X), weakness(X)` pour résoudre le but `malaria(X)`.

Par ailleurs, pour améliorer le résultat de ses propres opérations (réflexion du côté du système), le système sous-jacent à Prolog-Tutor utilise un autodiagnostic confirmé pour ajuster ses paramètres de mise à jour du modèle de l'apprenant. En effet si une différence significative entre le niveau d'acquisition inféré et le niveau d'acquisition négocié et confirmé est constatée, on peut déduire qu'il y a un problème au niveau de la mise à jour du modèle de l'apprenant.

Cet état de fait est pris en compte dans le CSDC à travers une recommandation. Sa considération concrète au niveau de l'implémentation du DC est fortement dépendante du contexte et c'est pour cette raison qu'aucun détail additionnel ne figure dans le CSDC. Dans Prolog-Tutor, cette considération est statique. Tout au long d'une session d'apprentissage avec un apprenant particulier, le système gère plusieurs artéfacts mémoriels dont une trace des actions de l'apprenant et une trace des actions pédagogiques. C'est dans cette trace que le système indique qu'un examen de l'algorithme de mise à jour du modèle de l'apprenant est nécessaire.

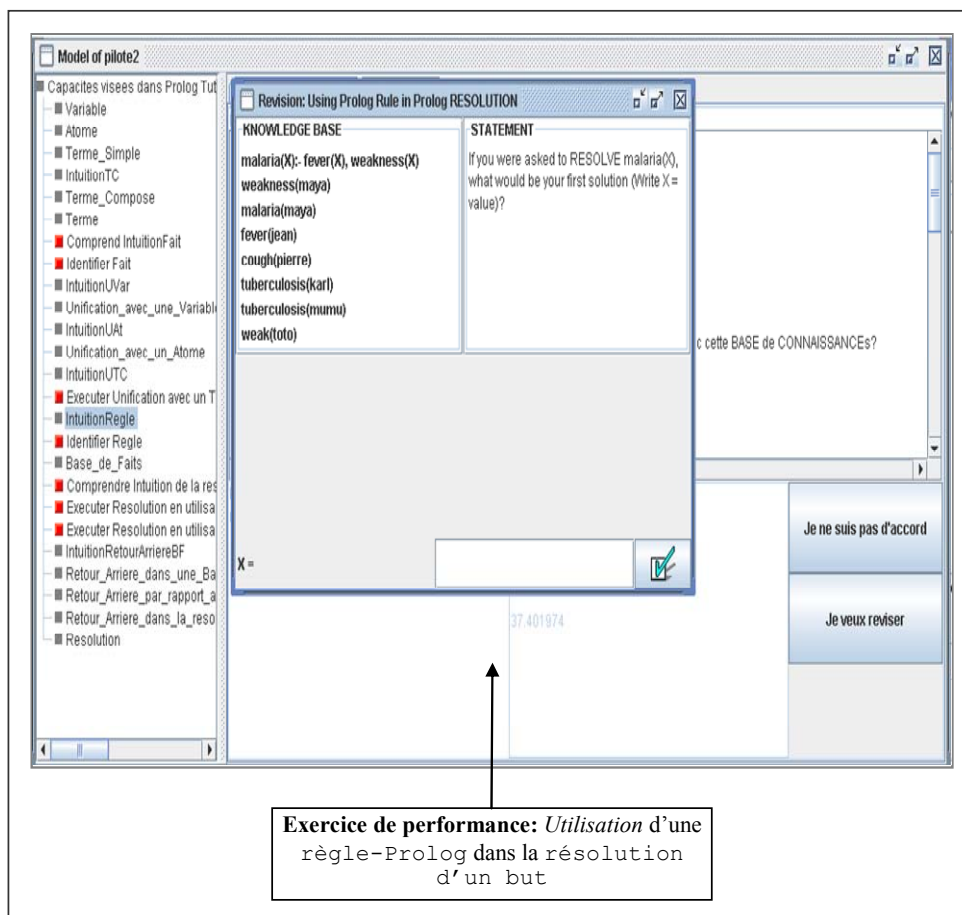


Figure VI.7. Implication de l'apprenant dans sa modélisation:exercice de performance

6.6 Discussion

Le but des exemples précédents était d'illustrer comment le CSDC peut être utilisé au cours de la conception et de l'implémentation du module de DC dans un STI. À la lumière de ces illustrations, trois questionnements suscitent un débat sur l'utilité de ce cadre.

Premièrement, peut-on dire que le DC déployé dans chaque exemple est effectivement implémenté tel que le recommande la spécification correspondante? Dans les deux premiers exemples, les questions qui déroulent le dialogue tutoriel sont explicitement associées à des capacités. De ce fait, les réponses de l'apprenant à ces questions constituent des indicateurs sur leur niveau d'acquisition par rapport à ces capacités. Dans le dernier exemple, l'apprenant exécute effectivement un autodiagnostic puisque l'essence de la session d'apprentissage consiste à changer le contenu de son modèle et à justifier tous les

changements désirés à travers des exercices behavioristes de performance. La réponse à ce premier questionnement est donc affirmative. Globalement, les mises en oeuvre du DC correspondent aux recommandations de leurs spécifications. Toutefois, de nombreux détails inhérents à l'implémentation informatique et au domaine appris ont imposé des ajouts.

Deuxièmement, peut-on parler d'exploitation pédagogique du DC à travers la réflexion tant du côté de l'apprenant que du côté du système? Dans le premier exemple, on peut avancer que par sa nature interactive, le dialogue tutoriel pourrait provoquer une forme de réflexion implicite. Dans le second exemple, la structure du dialogue tutoriel se veut explicitement pro réflexive. On a montré comment les composantes de ce dialogue correspondaient aux étapes de la pensée réflexive, telle que conçue par Dewey. On a également montré comment le contenu des questions de ce dialogue peut être organisé de manière à orienter la pensée réflexive en fonction de la nature de la capacité à propos de laquelle l'apprenant "réfléchit". Dans le dernier exemple, la réflexion est définitivement implicite: non observable, on assume qu'elle survient avant la décision de l'apprenant de changer son niveau d'acquisition par rapport à une capacité. L'exploitation pédagogique de l'évaluation de l'apprenant qui succède à cet autodiagnostic - consiste pour le système à ajuster le niveau d'acquisition de l'apprenant par rapport aux capacités qui sont reliées à celle dont le niveau a été négocié.

Troisièmement et en conclusion, est ce que ces états de fait auraient pu être réalisés en l'absence d'un cadre de spécification? Il va de soi qu'un concepteur de STI doit savoir *a priori* ce qu'il désire faire ressortir au fil des interactions entre l'apprenant et le STI. Cependant deux dimensions importantes sont à prendre en compte. D'abord, tout en maîtrisant la signification des objectifs pédagogiques généraux d'un STI, le concepteur pédagogique doit être conscient de leur relation avec les diverses formes de DC qui sont applicables, de manière à maintenir une cohérence pédagogique globale. Dans la dernière illustration, le cadre de référence a permis d'exclure l'interprétation reconstructive comme choix d'approche de DC comportemental, étant donné le contexte behavioriste qui caractérisait cette situation d'apprentissage. Dans les deux autres exemples, le contexte rationaliste cognitiviste implique l'éligibilité de toutes les approches de diagnostic selon le CSDC. Les recommandations du CSDC sur l'implémentation de chaque approche fournissent des informations qui permettent de peser le pour et le contre de chaque choix possible. Par

exemple, dans une optique où l'on préfère préserver le contrôle du diagnostic du côté du système, le DC épistémique basé sur la négociation du contenu du modèle de l'apprenant ou sur la modélisation interactive de l'apprenant s'avère préférable.

6.7 Conclusion

Dans l'apprentissage à travers un STI, les objectifs pédagogiques doivent être clairement arrêtés et les fonctions pédagogiques qui en découlent doivent suivre la logique de ces objectifs. Ce chapitre a illustré trois spécifications de DC, ainsi que leur mise en œuvre dans Prolog-Tutor, un STI pour la programmation logique.

L'analyse de ces illustrations a permis d'apprécier la portée et l'utilité du CSDC, précisant ainsi l'originalité de notre contribution dans le domaine AIED pour ce qui de la conception du DC dans les STI. En effet, on a pu constater à travers ces illustrations comment le CSDC :

- fournit des informations sur la nature des différents types de DC possibles;
- fournit des recommandations sur l'implémentation d'une spécification du DC, étant donné le contexte pédagogique caractéristique du STI auquel est destiné ce DC;
- fournit des recommandations sur les relations entre les types de DC, les contextes pédagogiques et les exploitations pédagogiques du DC;
- constitue un point de référence pour la vérification de la fidélité d'une implémentation d'un module de DC, par rapport à sa spécification conceptuelle.

Par ailleurs, l'on doit noter que ce chapitre a surtout porté sur la mise en œuvre – le déroulement – du DC dans Prolog-Tutor. Les enjeux relatifs à l'implémentation qui ont été traités dans les deux chapitres précédents ne sont donc pas visibles parce qu'ils agissent "en coulisse" et constituent le moteur de cette mise en œuvre du DC. Ce chapitre termine notre présentation de la perspective pédagogique du DC préconisée dans cette thèse. Deux arguments ont été avancés pour justifier la pertinence de cette perspective : *l'imbrication conceptuelle* du DC avec le paradigme de cognition qui est considéré lors de la conception d'un STI; *l'imbrication opérationnelle (algorithmique)* de l'algorithme de DC (dans un STI) avec l'exploitation pédagogique qui en est faite à des fins de remédiation – la stimulation de la pensée réflexive étant la forme de remédiation adoptée dans cette thèse -. Le prochain chapitre porte sur une évaluation du CSDC et de la mise en œuvre du DC dans Prolog-Tutor – selon la perspective préconisée dans le CSDC. Le but de cette évaluation est d'analyser la

portée de ces deux arguments est avérée pour les problématiques de la recherche AIED qui s'y rattachent.

Chapitre VII

ÉVALUATION QUALITATIVE DU CSDC ET DE LA PENSÉE RÉFLEXIVE DANS PROLOG-TUTOR

7.1 Introduction

Cette thèse souligne trois arguments principaux afin de conférer une dimension plus pédagogique à l'application du DC dans les STI: (1) la spécification d'une instance de DC à travers un cadre de référence (un CSDC); (2) la considération explicite de l'influence des paradigmes de cognition sur le DC dans les STI; (3) la considération explicite de l'influence de l'exploitation pédagogique qui est faite des résultats d'un DC sur ce DC, ce à travers une boucle diagnostic-remédiation, la stimulation de la pensée réflexive ayant été choisie comme approche de remédiation dans ce cas particulier. Ces trois arguments ont été intégrés dans un CSDC pour en favoriser la prise en compte lors de toute spécification d'instance de DC pour un STI. De plus, le troisième argument a été formalisé à travers un algorithme générique basé sur une approche bayésienne et un modèle de dialogue tutoriel dont le but est de stimuler explicitement la pensée réflexive lors du DC dans un STI. Cette formalisation a été implémentée et mise en œuvre dans Prolog-Tutor, un prototype de STI pour la programmation logique,.

Le but de ce chapitre est de présenter une évaluation qualitative du CSDC et une évaluation de l'implémentation de la pensée réflexive comme approche de remédiation dans Prolog-Tutor. Pour chacune de ces évaluations, la méthodologie d'évaluation, les données recueillies, les résultats obtenus ainsi qu'une analyse de ces résultats sont présentés. Mais auparavant, il convient de préciser les raisons du choix de l'approche qualitative d'évaluation.

7.2 Choix de l'approche qualitative d'évaluation

Dans un travail de recherche, l'on peut considérer une évaluation comme un processus systématique de collecte d'informations et d'analyse de ces informations, qui aboutit à un jugement de valeur. Le défi posé lors de l'évaluation d'un système informatique repose sur deux questions: sur quoi portera l'évaluation? Quelle en sera la méthode? La réponse à ces questions détermine le choix à faire entre une méthode quantitative et une méthode qualitative.

L'approche quantitative est appropriée lorsque l'on désire évaluer une hypothèse ou une théorie élaborée par l'observation de phénomènes. Elle consiste à colliger des données numériques relatives à un certain nombre de variables et à analyser ces données en vue d'établir une relation causale ou corrélationnelle et de vérifier ainsi des hypothèses. L'analyse statistique est l'approche typique d'analyse de données lors d'une évaluation quantitative. Les résultats doivent être reproductibles et généralisables; il est donc nécessaire de disposer du nombre requis de participants afin de contrôler l'impact du hasard et des autres perturbations.

L'approche qualitative vise plutôt à étudier en profondeur un objet ou un phénomène. Elle consiste à collecter des données qualitatives (paroles, comportement) sur l'objet ou le phénomène et à analyser ces données. La principale différence avec l'approche quantitative se situe au niveau de l'analyse et des conclusions tirées. Dans une approche qualitative, le chercheur observe afin d'inférer et de caractériser un phénomène. Bien que les résultats qui peuvent en découler ne soient pas nécessairement généralisables, ils sont explicatifs et servent souvent de point de départ pour l'évaluation quantitative (Miles et Huberman, 2003).

Si le but d'une évaluation est de prouver l'efficacité d'un système par rapport à sa tâche et à ses usagers, alors une évaluation quantitative est souhaitable. Toutefois, certaines

recherches s'intéressent à la description, la caractérisation et la justification de la nature profonde des artéfacts qu'elles produisent (Murray, 1993). Dans ces cas, les méthodes qualitatives doivent être mises à contribution pour collecter les données nécessaires. Pour comprendre le choix d'un devis qualitatif, il faut donc examiner les buts de l'évaluation présentée dans ce chapitre.

7.3 Buts de l'évaluation

Le but de l'évaluation présentée dans ce chapitre est de valider deux des principales contributions de cette thèse.

Dans la *perspective de la conception du DC pour les STI*, nous pensons que l'originalité d'un CSDC découle du fait qu'il favorise un pont sémantique entre les concepteurs pédagogiques et les programmeurs de STI – plus particulièrement le programmeur du module de DC -. De plus, le CSDC permet de véhiculer explicitement les relations innovatrices dans l'application du DC dans les STI : l'influence des paradigmes de cognition et la boucle diagnostic-remédiation. Le premier but de notre évaluation est de valider la pertinence et l'originalité de ces relations. Le deuxième but de notre évaluation est de valider la pertinence et l'originalité d'un outil tel que CD-SPECIES, pour exploiter le CSDC.

Dans la *perspective de l'implémentation du DC dans les STI*, l'originalité de cette contribution découle de l'implémentation formelle et générique de la dynamique de la boucle co-déterminante diagnostic-remédiation. Cette formalisation a pris la forme d'une approche bayésienne de diagnostic et d'un modèle formel et explicite de la stimulation de la pensée réflexive, basé sur la théorie de Dewey. Le troisième but de notre évaluation porte sur l'effectivité du modèle formel de stimulation de la pensée réflexive proposé dans cette thèse.

Pour réaliser ces buts, deux objectifs ont permis de préciser l'évaluation qui a été effectuée.

Le premier objectif de cette évaluation est une analyse de la pertinence du concept du CSDC, en sa nature même de cadre conceptuel pour le DC, mais aussi en termes du contenu qui a été proposé et en termes de son exploitation à travers CD-SPECIES. Cette analyse vise à caractériser les points forts du CSDC et à délinéer les améliorations indispensables pour son effectivité. Il ne s'agit donc pas de vérifier une hypothèse qui expliquerait un phénomène observable, d'où la justesse d'une approche qualitative d'évaluation. Dans cette première

évaluation, Un accent sera mis sur la pertinence de la boucle diagnostic. D'après les définitions proposées ci-dessus, la méthode qualitative semble la plus indiquée. Lors de cette première évaluation, l'une des principales questions qui est abordée concerne la pertinence de considérer explicitement la relation entre le diagnostic cognitif et la réflexion de l'apprenant, ce à travers une boucle diagnostic-réflexion. Si l'on considère que cette pertinence est avérée, il est possible d'établir que le phénomène de réflexion se manifeste effectivement chez l'apprenant, lorsque son comportement est soumis à un DC spécifié à partir du CSDC. Ceci nous mène au deuxième volet de notre évaluation.

Le deuxième objectif de cette évaluation est une analyse du phénomène de pensée réflexive chez un apprenant pendant la résolution d'un exercice à travers les dialogues tutoriels de Prolog-Tutor. L'un des arguments avancés en faveur du CSDC est qu'il constitue un pont sémantique entre la conception pédagogique du DC et sa programmation dans un STI. Ce pont étant une condition nécessaire au maintien de la fidélité entre les deux phases de construction de le STI (conception et programmation). Dans cette thèse en particulier, la fidélité doit être maintenue pour permettre une exploitation pédagogique des résultats du DC à travers la stimulation de la pensée réflexive chez l'apprenant. Une façon de favoriser cette fidélité est de concevoir l'algorithme de DC à travers des dialogues tutoriels dont la structure aurait le potentiel – en théorie – de stimuler la pensée réflexive. La théorie de la pensée réflexive selon Dewey constitue la référence fondamentale de l'approche de réflexion visée à travers le déploiement de dialogues tutoriel de Prolog-Tutor. Pour cette deuxième évaluation, il s'agit donc d'étudier dans quelle mesure les actions et les réactions d'un apprenant pendant son expérience avec un tel dialogue tutoriel, témoignent d'une occurrence de ce phénomène. Encore une fois, il ne s'agit pas d'expliquer un phénomène, mais de le décrire puis de le caractériser en référence à une théorie établie. Dans ce sens, le choix de la méthode qualitative est le plus approprié.

7.4 Évaluation qualitative du CSDC

Cette section présente le premier objectif de notre évaluation, à savoir l'analyse de la pertinence du concept de CSDC, de son contenu et de son exploitation à travers un outil comme CD-SPECIES. Après avoir présenté les participants et les données colligées, nous proposons une analyse basée sur une convergence des implications découlant de ces données.

7.4.1. Méthodologie

De nombreuses approches d'évaluation qualitative de systèmes informatiques sont proposées dans la littérature en intelligence artificielle appliquée à l'éducation (AIED): *wizard-of-Oz*, évaluation par des experts, preuve d'existence, observation et classification qualitative, analyses de tâches structurées et classification de phénomènes, études comparatives, évaluation interne (Iqbal et al., 1999; Mark et Greer, 1993; Murray, 1993).

Le CSDC est un outil conceptuel dont l'utilité opérationnelle peut se concrétiser à travers l'implémentation d'un système tel que CD-SPECIES. Il correspond donc au premier stade du cycle itératif de la construction d'un outil informatique. Par ailleurs, hormis l'évaluation par des experts, la majorité des méthodes précitées nécessitent au moins un prototype informatique. En fait dans une évaluation par des experts, ceux-ci peuvent contribuer à deux niveaux: (1) en observant les futurs usagers du système informatique pendant que ceux-ci interagissent avec le système ou (2) en répondant directement à des questions relatives à l'aspect du système qu'on désire évaluer. Cette deuxième option a été choisie dans le cas de la présente évaluation, étant donné que le CSDC est un cadre conceptuel.

D'après Murray (1993), les entrevues d'experts constituent un apport substantiel en ce qui concerne l'importance d'un concept et la faisabilité de ses applications. Cet apport est encore plus concluant lorsqu'un certain consensus (convergence) émerge des opinions recueillies.

7.4.2. Collecte des données

On peut étudier le CSDC en utilisant une approche d'analyse similaire à l'analyse d'une ontologie. En effet, à l'instar d'une ontologie, le CSDC décrit: un vocabulaire pour le DC, des relations conceptuelles (sémantiques) entre les éléments de ce vocabulaire et enfin des recommandations pour le maintien de la cohérence entre chaque instance de processus défini à partir de ce vocabulaire.

Étant donné que le CSDC n'est pas opérationnel, on peut adopter une analyse analogue à celle de l'analyse des compétences d'une ontologie. Ramené dans le contexte du CSDC, cette analyse devrait permettre de (Gruninger et Fox, 1995):

- (1) Juger la pertinence et l'importance du CSDC pour la conception pédagogique des STI;

- (2) Juger le niveau d'exhaustivité du contenu du CSDC et la pertinence de chacun de ses éléments;
- (3) Juger la justesse des relations proposées entre le DC, la pensée réflexive et les paradigmes de cognition.

Les questions qui sont abordées lors de cette évaluation ont pour but d'analyser ces trois points. Ces questions constituent le principal instrument d'une entrevue semi-structurée qui a été conduite auprès d'experts en conception pédagogique.

7.4.2.1. Les participants

Trois experts en conception pédagogique ont été contactés pour analyser le CSDC. Ces concepteurs avaient la particularité d'avoir déjà participé soit à la conception, soit à la construction d'un STI à l'aide d'outils auteurs ou en assistant un programmeur (Appendice D: Tableau D. 1). Ce choix garantissait donc une connaissance minimale de la notion de DC, des enjeux des paradigmes de cognition à tous les niveaux de conception des STI et enfin des exploitations pédagogiques du DC dans les STI.

7.4.2.2. Procédure

Avant de recueillir les avis des experts sur le CSDC, il était nécessaire qu'ils comprennent sa nature, son utilité et les manifestations concrètes de son utilité. Par conséquent, la procédure qui devait aboutir au recueil des avis des experts a incorporé trois étapes (Tableau VII.1):

- (1) **La prise de connaissance du CSDC.** Chacun des trois experts a été contacté par la chercheuse et deux documents leur ont été remis: un document en français – noté CSDC_1 - décrivant la pertinence du CSDC et le contenu du CSDC; un document *en anglais* (Tchetagni, Nkambou, et Bourdeau, 2005b) – noté CSDC_2 - décrivant des manifestations concrètes de l'utilité du CSDC, ce à travers des exemples de mise en œuvre d'instances de DC dans Prolog-Tutor;
- (2) **L'illustration de l'utilité du CSDC.** Deuxièmement, la chercheuse a rencontré séparément chacun des trois experts. La première partie de cette rencontre avait pour but de s'assurer que l'essence des documents avait été correctement saisie. Il s'agissait de montrer l'utilité du CSDC à travers un parcours de principales interfaces de CD-SPECIES, commentées par leurs fonctionnalités respectives. La seconde partie de cette rencontre avait pour but de mettre l'accent sur les manifestations

concrètes de l'utilité du CSDC. Une démonstration de Prolog-Tutor a été faite à travers la réalisation d'un exercice de résolution de but. Au cours de cette démonstration, il a été question de faire ressortir dans Prolog-Tutor les deux composantes principales du DC, qui constituent l'originalité de la contribution de cette thèse: la boucle entre le DC et la pensée réflexive de l'apprenant et le lien entre le DC et un paradigme de cognition.

- (3) **L'entrevue.** Au terme des explications et des illustrations concrètes relatives au CSDC, une entrevue semi structurée avec l'expert a été amorcée par la chercheure. Au cours de cette entrevue, six questions ont été posées à l'expert par la chercheure. L'entretien s'est présenté sous un format libre similaire à une conversation semi-contrôlée. Le dialogue a été enregistré sur des cassettes audio pour l'analyse subséquente.

Tableau VII.1. Procédure de collecte des données pour l'évaluation du CSDC

But	Prise de connaissances du CSDC		
Activités	Instruments		
<ul style="list-style-type: none"> Prise de contact avec les experts participants et communication de la documentation nécessaire pour comprendre le CSDC 	<ul style="list-style-type: none"> Document en Français décrivant le CSDC (CSDC_1) Document en Anglais décrivant le CSDC avec des exemples concrets (CSDC_2) 		
But	Illustration de l'utilité du CSDC		
Activités	Instrument		
<ul style="list-style-type: none"> Chercheure: Rencontre avec chacun experts participants et explication des interfaces de CD-SPECIES 	<ul style="list-style-type: none"> Présentation power point des interfaces de CD-SPECIES 		
<ul style="list-style-type: none"> Chercheure: Simule la résolution d'un exercice dans Prolog-Tutor et mettre en exergue les caractéristiques du DC Expert-participant: Observe, Effectue des tests à l'interface de Prolog-Tutor et surtout pose des questions 	<ul style="list-style-type: none"> Dialogue tutoriel dans Prolog-Tutor 		
But	Entrevue avec les experts		
Activités	Données et Instruments		
<ul style="list-style-type: none"> Chercheure : Poser des questions aux experts, élaborer des questions et discuter des réponses Expert participant : Répondre aux questions et discuter avec la chercheure 	Participant	Durée	Instrument de recueil de données
	E1	2 heures	Enregistrement Audio
	E2	1 heure	Enregistrement Audio
	E3	2 heures	Enregistrement Audio

7.4.2.3. Les instruments

Trois entrevues ont été conduites pour s'informer des avis des experts. Pour chacune de ces entrevues, les données ont pu être recueillies à partir de l'utilisation de deux instruments : un ensemble de six questions pour diriger l'entrevue (Tableau VII.2); un enregistreur audio pour enregistrer l'entrevue.

Dans l'introduction de la section 7.4.2. , trois points ont été mentionnés pour exprimer les principales interrogations qui feront l'objet de l'entrevue avec les experts participants. Le point (1) est analysé par les questions Q4, Q5, Q6. La question Q4 porte sur l'importance conceptuelle du CSDC tandis que les deux autres questions portent sur les applications informatiques qui en découlent : le lien entre une spécification issue du CSDC et sa mise en œuvre dans un STI; la pertinence d'utiliser le CSDC à travers un outil informatique.

Tableau VII.2. Instrument de collecte de données pour l'évaluation du CSDC

Question	Code	Formulation
Question1	Q1	L'étudiante-chercheur pose une question à l'évaluateur : « Qu'entendez-vous par diagnostic cognitif ? »
Question2	Q2	Comment pensez vous que l'intégration de la réflexion et du contexte pédagogique contribuerait à améliorer les résultats du processus de DC dans les STI?
Question3	Q3	Quels éléments de l'apprentissage jugez vous pertinents d'ajouter dans le CSDC?
Question4	Q4	Comment pensez vous que le CSDC pourrait améliorer: ⇒ la mise en œuvre du processus dans les STI ⇒ la fidélité du processus par rapport aux besoins spécifiés par un concepteur pédagogique ⇒ tout autre critère jugé pertinent
Question5	Q5	Quel lien établissez-vous entre l'exemple de spécification formulée dans le document CSDC_1 et sa mise en œuvre dans Prolog-Tutor?
Question6	Q6	Utiliseriez-vous le CSDC ou un outil comme CD-SPECIES?

Le point (2) est analysé à travers les questions Q1 et Q3. La question Q1 permet implicitement de prendre connaissance des aspects du DC qui le cas échéant, n'apparaissent pas dans le CSDC. La question Q3 permet de recueillir la même information de manière plus directe.

Le point (3) est analysé à travers la question Q2. En effet, cette question concerne le lien explicite entre le contexte pédagogique et la forme du DC, ainsi que le lien explicite entre l'exploitation pédagogique du DC et l'algorithme correspondant.

7.4.3. Résultats : présentation et analyse

L'analyse des données qualitatives s'élabore généralement à travers trois étapes principales (Miles & Huberman 2003): la transcription des données recueillies, le codage des données pour les catégoriser, les décrire ou les caractériser, et enfin l'énumération (représentation quantitative relative à l'occurrence des codes) ainsi qu'une analyse des relations entre les données. L'évaluation proposée dans ce chapitre s'est inspirée de cette méthodologie.

7.4.3.1. Transcription des données

Dans l'APPENDICE. E, le Tableau E. 1 et le Tableau E. 2 présentent une transcription complète de l'entretien avec l'expert E1 et l'expert E2 respectivement. Ces transcriptions correspondent aux enregistrements audio qui ont été effectués lors des entrevues semi structurées avec ces derniers. L'essentiel des propos recueillis a été extrait et est consigné au Tableau VII.4.

7.4.3.2. Codage des données

Le codage est une opération de marquage des segments de données avec des symboles (descriptifs d'une caractéristique), des noms de catégories. Le codage permet d'organiser les données brutes pour en faire émerger un sens. Les codes dépendent de l'objectif de l'analyse qualitative. Toujours selon Miles et Huberman (2003), il existe deux approches de codage de données: le codage *a priori* et le codage inductif. Dans le codage *a priori*, on décide à l'avance des différentes catégories ou caractéristiques qu'on cherche dans un corpus de données brutes. Dans le codage inductif, les codes sont établis pendant l'examen des données.

Dans ce travail, le codage *a priori* a été choisi parce que dès le départ, des interrogations précises ont été établies relativement au CSDC. À titre de rappel, ces questions se penchaient particulièrement sur les points suivants:

- (1) Juger la pertinence et l'importance du CSDC pour la conception pédagogique des STI;

- (2) Juger le niveau d'exhaustivité du contenu du CSDC et la pertinence de chaque élément de ce contenu;
- (3) Juger la justesse des relations proposées entre le DC, la pensée réflexive et les paradigmes de cognition.

Les avis des experts sont d'abord examinés et croisés sur la base d'un commentaire de leurs réponses respectives aux six questions de l'entrevue. Ce commentaire a pour but de juger s'il existe une *convergence entre les opinions* sur ces trois questions d'évaluation. C'est au niveau de ces croisements d'opinions qu'un codage de l'information recueillie s'est avéré nécessaire (APPENDICE. D: Tableau D. 2, Tableau D. 3).

7.4.3.3. Analyse des données

Pour répondre aux trois interrogations susmentionnées, nous avons procédé à une analyse des avis recueillis auprès des concepteurs. Le choix d'une telle approche argumentative se défend par le fait que les données recueillies à travers les entrevues sont purement qualitatives. Par conséquent, l'on ne saurait les catégoriser, les compiler statistiquement afin d'établir un accord inter juges sur la base d'une statistique Kappa (Cohen, 1960), tel que recommandé dans ce type d'évaluation.

Ainsi, dans les prochains paragraphes nous constatons la convergence (ou la non convergence) entre les avis des concepteurs, respectivement pour ce qui de:

- La pertinence et l'importance du CSDC pour la conception et la mise en oeuvre du DC dans un STI
- L'exhaustivité du contenu proposé pour ce CSDC
- La justesse des relations proposées dans ce CSDC.

Le Tableau VII.3 résume les résultats de notre analyse de la convergence des avis entre les experts. L'interprétation des signes qui caractérisent cette convergence doit se baser sur les codes expliqués à l'APPENDICE. D (Tableau D. 2 et Tableau D. 3).

Tableau VII.3. Croisement des arguments et des avis des experts concepteurs

Question d'évaluation	Questions correspondantes de l'entrevue	Croisement des arguments par question d'entrevue	Croisement des avis par question d'évaluation
(1) Le CSDC est pertinent et important pour la conception pédagogique des STI;	Q4	ACC _{nu}	(3 ₃ , 0)
	Q5	ACC	
	Q6	ACC _{nu}	
(2) Le contenu du CSDC est exhaustif	Q1	ACC _{nu}	(0, 3 ₃)
	Q3	DACC _{nu}	
(3) Les relations proposées entre le DC, la pensée réflexive et les paradigmes de cognition sont justes et importantes	Q2 Réflexion	ACC _{nu}	(3 ₃ , 0)
	Q2 Paradigmes de cognition	ACC	

Tableau VII.4. Résumés des entretiens avec les experts en conception pédagogique

Question d'entrevue	Expert1	Expert1	Expert3
Q1	<ul style="list-style-type: none"> ▪ Le diagnostic des connaissances et des compétences ▪ Le diagnostic de la motivation ▪ Mais beaucoup de connaissances sont difficiles à représenter parce que souvent les apprenants n'en sont même pas conscients. Par conséquent, il est plus important de leur apprendre à apprendre à se connaître. La vraie connaissance est compilée et c'est son diagnostic qui est compliqué. 	<ul style="list-style-type: none"> ▪ Détermination de l'état des connaissances de l'apprenant en début de formation, dans un domaine donné, par rapport à un objectif d'apprentissage donné pour ajuster le cours dès le début 	<ul style="list-style-type: none"> ▪ Détermination de l'état cognitif de l'apprenant au début de l'apprentissage ▪ Il y a trois niveaux d'observations de l'apprenant en fonction de leur position par rapport au processus d'apprentissage : avant (diagnostic), pendant (évaluation formative), après (évaluation sommative) ▪ La conception du DC présentée ici correspond surtout à l'évaluation formative
Q2 Réflexion	<ul style="list-style-type: none"> ▪ La réflexion permet de résoudre les situations où l'apprenant est bloqué. ▪ Elle permet de provoquer ou de démontrer une connaissance. ▪ Au pire des cas elle permet de diagnostiquer la méconnaissance. ▪ Par contre, la réflexion serait d'autant plus bénéfique (surtout dans un dialogue tutoriel tel que Prolog-Tutor) si l'on pouvait faire varier les questions de manière à faire varier la stimulation de la réflexion sur une capacité 	<ul style="list-style-type: none"> ▪ Pertinent dans Prolog-Tutor car ce lien permet à l'apprenant une appréhension profonde des capacités visées à travers les objectifs d'apprentissage ▪ Mais dans les domaines plus complexes que la logique, il sera difficile de relier la réflexion à une démarche de solution. Cette réflexion serait surtout utile pour préciser ou raffiner le modèle de l'apprenant 	<ul style="list-style-type: none"> ▪ La réflexion permet de confirmer et de raffiner un diagnostic, surtout dans les cas d'autodiagnostic; ▪ Sans la réflexion il ne peut y avoir d'apprentissage de haut niveau (On peut réciter une table de multiplication sans réflexion mais on ne peut pas appliquer une multiplication).

Question d'entrevue	Expert1	Expert1	Expert3
<p>Q2</p> <p>Paradigmes de la cognition</p>	<ul style="list-style-type: none"> ▪ Il faut surtout travailler sur le lien entre un paradigme de cognition et un type d'objectif d'apprentissage. ▪ Cet objectif suggère le paradigme de cognition et c'est aussi cet objectif qui suggère la manière dont le DC doit être effectué 	<ul style="list-style-type: none"> ▪ La différenciation en fonction des théories de la cognition est pertinente et même nécessaire, puisqu'elle détermine la nature des objectifs d'apprentissage et donc la nature de ce qui sera diagnostiqué. ▪ Cependant, l'apprentissage ne peut pas être enfermé dans une théorie. La manière d'apprendre peut varier à l'intérieur d'une même formation. On ne peut être behavioriste du début à la fin, ni cognitiviste du début à la fin. ▪ Il faut adopter une approche par objectifs d'apprentissage. Bien que ces objectifs puissent dépendre d'une théorie de cognition, ils déterminent surtout un type de capacité visée (déclaratif, procédural, etc.) et donc la manière dont le DC sera effectué dans les activités d'apprentissage correspondantes. 	<ul style="list-style-type: none"> ▪ Les positions quant aux théories d'apprentissage ne son pas tranchées et définitives ▪ Plutôt que de différencier le DC en fonction des théories, on pourrait le considérer en fonction de l'effort cognitif à appliquer à chacune des connaissances : les habiletés. ▪ Donc oui, le DC diffère selon la nature et le niveau de cognition, mais cela doit être défini par un niveau d'habileté et non par une théorie de cognition ou d'apprentissage
<p>Q3</p>	<ul style="list-style-type: none"> ▪ Il faut absolument ajouter un niveau spécifique et un contenu lors de la spécification d'une instance de DC. ▪ Les éléments offerts par le CSDC semblent très conceptuels et assez abstraits ▪ Après avoir défini de manière conceptuelle la nature du DC qu'il désire, le concepteur devrait pouvoir le spécifier davantage en indiquant le contenu sur lequel se basera effectivement le processus 	<ul style="list-style-type: none"> ▪ Inclure la culture, les autres acteurs, les outils ▪ Faire ressortir le fait que le contexte pédagogique ou les théories ne couvrent pas toute une formation, mais sont reliées à un épisode ou à un objectif d'apprentissage particulier 	<ul style="list-style-type: none"> ▪ Les caractéristiques personnelles des apprenants ▪ L'impact sur le DC de la différence entre le niveau de départ de l'apprenant et le niveau visé à travers une session d'apprentissage ▪ Pour le reste, je pense que c'est au fil des interactions entre le concepteur et le CSDC via CD-SPECIES que ce contenu s'améliorera ▪ <i>[À part cela, des commentaires sur la forme]</i>

Question d'entrevue	Expert1	Expert1	Expert3
<p>Q4</p>	<ul style="list-style-type: none"> ▪ Il faut créer une interface de communication avec des exemples. Une telle interface permettrait au concepteur ou à l'expert de vérifier la fidélité de manière plus raffinée et plus concrète 	<ul style="list-style-type: none"> ▪ L'articulation des connaissances sur le DC en réalisation certains aspects auxquels on ne pense pas toujours (le raisonnement de l'apprenant) ▪ L'articulation du lien entre le DC et les théories de la cognition renforce l'importance des objectifs pédagogiques formellement définis car c'est surtout eux finalement qui influencent la perspective du DC; ▪ L'articulation du lien entre les types de DC et les approches de représentation des connaissances ouvre une question de recherche sur la relation entre les objectifs pédagogiques et les représentations de connaissances associées dans les STI, plus particulièrement les systèmes tuteurs intelligents. 	<ul style="list-style-type: none"> ▪ Une fois qu'il est établi, le principal bénéfice du CSDC est facilite le dialogue entre le concepteur et le programmeur sans que l'un soit obligé de maîtriser le langage de l'autre ▪ Mais avant d'y arriver, il faut faire converger les conceptions des programmeurs et des concepteurs afin d'obtenir un langage commun. Cette convergence ne peut émerger que de l'expérience entre les deux parties ▪ Il faut également expliciter le rôle d'un instrument d'évaluation et d'ajustement de l'algorithme de DC (un humain dans la boucle par exemple)
<p>Q5</p>	<ul style="list-style-type: none"> ▪ J'aurais préféré voir les recommandations sur les approches de représentations de connaissances; ▪ C'est vrai que tel que je vois le Réseau Bayésien et le dialogue tutoriel, cela semble correspondre à des cas particuliers de Overlay et de modélisation interactive de l'apprenant, tel que recommandé d'après ton cadre. 	<ul style="list-style-type: none"> ▪ Le dialogue tutoriel semble pousser effectivement à la réflexion puisque je vois qu'il est très élaboré et qu'il creuse beaucoup avec l'apprenant; ▪ Mais il faut faire attention, ça ne correspond pas à un apprentissage constructiviste mais à une instruction constructiviste; ▪ La spécification impose au concepteur de formuler clairement les questions du dialogue tutoriel. Elle impose également au programmeur de définir un lien explicite entre les questions du dialogue, le modèle de l'apprenant et le modèle des connaissances du domaine. C'est intéressant mais c'est plus pertinent dans les systèmes tuteurs intelligents. 	<ul style="list-style-type: none"> ▪ Le dialogue tutoriel et le modèle de l'apprenant visualisés correspondent à la spécification; ▪ Par contre, la théorie constructiviste ne ressort pas de l'exemple que j'ai vu dans Prolog-Tutor. Cela prouve d'ailleurs qu'il vaut mieux fonctionner par habileté.

Question d'entrevue	Expert1	Expert1	Expert3
<p>Q6</p>	<ul style="list-style-type: none"> ▪ Il s'agit d'un bon premier pas, très prometteur. Pour le moment, les spécifications sont génériques et il serait encore mieux d'en avoir une version plus spécifique avant de passer à l'implémentation ▪ Il faudrait aussi tenir compte des façons de réfléchir des apprenants, de leur niveau d'habileté par rapport au niveau visé 	<ul style="list-style-type: none"> ▪ Les concepteurs et les programmeurs ont des bagages différents. Le CSDC peut faciliter la communication et augmenter l'efficacité de la conception et de la programmation; ▪ Le programmeur peut être plus motivé lorsqu'il comprend les retombées et les applications du programme qu'on lui demande de développer. 	<ul style="list-style-type: none"> ▪ Oui, à condition de minimiser le temps d'apprentissage de l'interface de CD-SPECIES. Cela nécessite une étude des besoins des concepteurs. À condition de proposer un contenu essentiel et concis qui ne perdra pas le concepteur ▪ En général, cet outil faciliterait aussi le fameux ajustement donc ont a parlé à propos de la sémantique ou du langage commun entre concepteurs et programmeurs

Question d'évaluation 1: la pertinence et l'importance du CSDC pour la conception et la mise en oeuvre du DC dans un STI

Lors de notre réflexion sur l'idée d'un CSDC, nous avons été guidés par des hypothèses d'ordre conceptuel et d'ordre pratique. Au niveau conceptuel, nous avons imaginé qu'un CSDC favoriserait l'établissement, le maintien et l'évolution (conceptuelle) d'un pont sémantique entre les concepteurs pédagogiques et les programmeurs de STI. Au niveau pratique, nous avons imaginé qu'un CSDC favoriserait la préservation de la fidélité pédagogique de la mise en oeuvre du DC dans un STI, ce à travers les recommandations sur l'implémentation du module de DC, étant donné une spécification conceptuelle de DC. Les questions Q4, Q5 et Q6 ont permis d'examiner la justesse de ces hypothèses.

- **Au niveau conceptuel**

La question Q4 a permis d'aborder la pertinence et l'importance du CSDC sur un plan conceptuel. Les trois experts s'accordent sur le fait que l'idée du CSDC est pertinente et tout en reconnaissant la pertinence, un certain nombre de conditions préalables a été évoqué.

Pour l'expert E1, le CSDC proposé dans cette thèse constitue un premier pas vers l'harmonisation entre la conception et l'implémentation du DC dans un STI. Il serait primordial d'y annexer une interface de communication à un niveau plus concret, qui permettrait au concepteur pédagogique de *spécifier un contenu pour chaque instance particulière de DC* (par exemple, les questions qui vont être posées à l'apprenant pour diagnostiquer une capacité donnée).

Selon l'expert E2, le CSDC ouvre plusieurs questions de recherche. Ces questions portent principalement sur: l'articulation du corpus des connaissances relatives au DC dans un STI, la relation entre le DC et les théories de la cognition *mais surtout* en référence aux objectifs d'apprentissage et aux habiletés visées dans un STI. En d'autres termes, il serait plus utile de se pencher sur le lien explicite entre un objectif d'apprentissage défini en termes de niveau d'habileté et le DC d'un apprenant dans un STI. Selon notre perspective, loin d'être une contre indication, cette remarque élabore la vision proposée dans cette thèse. En effet, la considération des théories d'apprentissage dans le CSDC a été justifiée par leur relation avec les objectifs d'apprentissage (Chapitre III): l'atteinte des objectifs d'apprentissage est favorisée par des théories d'apprentissage, elle mêmes inspirées des théories de cognition. Si l'on prend la taxonomie des habiletés de Bloom, l'apprentissage behavioriste suffit pour atteindre la connaissance d'un objet, l'apprentissage cognitiviste (et dans certains cas pragmatiste) est nécessaire pour l'analyse d'un phénomène. Ceci étant, cette remarque du concepteur revêt une certaine importance dans la mesure où une considération du CSDC en fonction des objectifs d'apprentissage plutôt que des théories de la cognition nécessite la définition d'un plus grand nombre de relations.

L'expert E3 est du même avis pour des raisons différentes. Le CSDC est un concept incontestablement bénéfique dans la mesure où il constitue une plateforme de dialogue entre le

concepteur et le programmeur en minimisant les efforts de traduction d'une part et d'autre. Cependant, une convergence sémantique préalable pour un tel statut du CSDC et seule l'expérience émergeant des interactions entre les deux parties mènera à cette convergence. Cette remarque corrobore ce que nous avons déjà anticipé: le CSDC proposé dans cette thèse jette les fondations d'un pont sémantique entre concepteurs et programmeurs. L'évolution conceptuelle de son contenu est nécessaire comme produit d'une considération des besoins et des intérêts des concepteurs pédagogiques et des programmeurs d'un STI. C'est d'ailleurs dans cette perspective que l'exploitation du CSDC à travers un outil informatique tel que CD-SPECIES a été suggérée dans cette thèse (section 3.6, en page 94).

- **Au niveau pratique**

Les questions Q5 et Q6 permettent d'aborder la pertinence et l'importance du CSDC sur le plan pratique. La question Q5 porte sur l'implémentation d'une spécification de DC dans Prolog-Tutor. La question Q6 porte sur la possibilité d'utiliser et d'exploiter le CSDC à travers CD-SPECIES. *Concernant Q5, les experts s'accordent sur le fait que le dialogue tutoriel de Prolog-Tutor élabore dans le sens d'entretenir l'apprenant dans un mode permanent de questionnement et de réflexion. Concernant Q6, les experts s'accordent sur le fait que CD-SPECIES représente un potentiel pour une conception pédagogique articulée du DC.*

Toutefois, sur la question de l'implémentation de CD-SPECIES, les experts E2 et E3 s'accordent aussi sur le fait qu'il y a une discordance entre la théorie de cognition (pragmatique avec apprentissage constructiviste) spécifiée et la mise en œuvre de l'apprentissage dans Prolog-Tutor. Selon eux, c'est l'enseignement qui est constructiviste dans ce prototype de STI, en lieu de l'apprentissage. Cette remarque illustre bien l'usage pratique du CSDC et de la relation entre une spécification et son implémentation. Dans ce cas d'espèce, le concepteur pédagogique a évalué l'implémentation du DC dans un STI en référence à une spécification de DC. Il a pu établir quelles composantes du DC étaient conformes (la pensée réflexive stimulée à travers un dialogue tutoriel) et quelles composantes ne le sont pas (l'apprentissage qui n'est pas nécessairement constructiviste). Il s'agit là d'un exemple d'interaction.

De même, sur la question de l'utilisation de CD-SPECIES, l'expert E3 a reporté un certain nombre de conditions désirables pour une utilisation effective (Tableau VII.4, ci-dessus, en page 209): une étude des besoins des concepteurs par rapport à l'implémentation du DC, la concision des informations offertes dans CD-SPECIES, la simplicité de l'interface de CD-SPECIES. Sans se démarquer de nos hypothèses de recherche, ces suggestions montrent que l'implémentation de CD-SPECIES constitue à elle seule un sujet de recherche. En effet, l'étude des besoins des concepteurs nécessite à elle seule une consultation et une étude exhaustives des intéressés dans leur contexte de travail. Dans le même sens, l'expert E1 a signalé qu'il serait encore plus intéressant de permettre au

concepteur pédagogique dans CD-SPECIES de spécifier le contenu d'une instance de DC (par exemple la manière de formuler une question de diagnostic) en fonction du niveau d'habileté de départ et des préférences propres à chaque potentiel apprenant.

Question d'évaluation 2: Exhaustivité du contenu du CSDC

Le terme STI désigne un grand nombre d'artéfacts du domaine de l'intelligence artificielle en éducation (AIED). Ces artéfacts partagent une caractéristique: ils favorisent l'apprentissage à travers des systèmes informatisés. Ainsi, depuis les environnements pour l'instruction individuel (*one-to-one tutoring*) jusqu'aux environnements pour la formation à distance et le *e-learning*, en passant par les environnements de simulations, le DC adopte diverses orientations. Un CSDC devrait être assez exhaustif pour couvrir toutes ces orientations ou au moins leurs composantes principales. Les experts ont reconnu que les éléments du CSDC tel qu'actuellement défini correspondent aux éléments principaux de base du DC. Cependant, de nombreux ajouts demeurent désirables pour une utilité accrue. Les deux prochains paragraphes discutent respectivement des avis des experts sur l'exhaustivité du contenu du CSDC et de leurs recommandations sur les améliorations possibles afin de renforcer cette propriété dans le CSDC.

- **Sur l'exhaustivité du contenu du CSDC**

Les commentaires des experts E2 et E3 relatifs à la question Q1 reflètent une vision du DC fortement reliée à la formation à distance ou au *e-learning*. En effet pour eux, le DC se résume à l'évaluation de l'état des connaissances de l'apprenant au début de la formation ou de l'apprentissage. Au terme d'une discussion soutenue (APPENDICE. E, Tableau E. 1 et Tableau E. 2), il a été convenu que le DC peut effectivement prendre une forme plus élaborée en essayant par exemple de comprendre le raisonnement de l'apprenant lorsqu'il interagit avec un STI. Cette forme de DC se retrouve surtout dans les STI adaptatifs, à l'instar des STIs. Cette première conclusion est favorable à la question sur l'exhaustivité du CSDC. Sans nécessairement être complet, le contenu du CSDC considère des aspects variés du DC, correspondants à une majorité des types de STI: l'état des connaissances, les représentations mentales, les processus mentaux correspondant au raisonnement (y compris la capacité d'apprendre à apprendre - tel que suggéré par l'expert E1 - qui serait encouragé dans une conception pragmatiste d'un STI). La réponse de l'expert E1 rappelle cependant une limite de notre travail en ce que les aspects personnels des apprenants – la motivation par exemple – ne sont pas considérés.

- **Sur l'amélioration de l'exhaustivité : les ajouts possibles au contenu du CSDC**

Un fait important à relever est que pour les experts, le DC tel que présenté dans cette thèse correspond à l'évaluation formative dans le vocabulaire de la conception pédagogique. Cela corrobore la problématique du pont sémantique entre les concepteurs pédagogiques et les programmeurs. La définition du DC dans le CSDC devrait mentionner ce type de relations terminologiques.

Par ailleurs, tous les éléments proposés jusqu'ici dans le CSDC ont été jugés pertinents. Toutefois, certaines améliorations sont désirables au niveau de leur présentation dans le cadre de spécification (question Q3). Premièrement, il faudrait faire ressortir le fait que les paradigmes de cognition ne réfèrent pas à un STI, mais à une habileté ou à un objectif d'apprentissage (expert E3). En effet, CD-SPECIES serait d'autant plus apprécié par un concepteur pédagogique dans la mesure où les recommandations ou les conseils proposés ne contraignent pas à la considération d'un paradigme de cognition tout au long de l'apprentissage (expert E2). Deuxièmement, le contenu d'une référence comme le CSDC devrait aussi relier les facteurs personnels (émotions, attitudes, la culture, les autres acteurs intervenant dans l'utilisation du STI, les outils) à la nature du DC et de l'algorithme correspondant dans un STI (expert E2). Troisièmement, il faudrait aussi définir la relation entre le différentiel cognitif à combler (différence entre le niveau d'habileté de départ de l'apprenant et le niveau d'habileté visé dans un objectif d'apprentissage dans un STI) et la manière dont la pensée réflexive doit être stimulée chez l'apprenant, définir les représentations de connaissances et les algorithmes nécessaires en conséquence (expert E3). Enfin, il faudrait absolument augmenter le CSDC d'une composante spécifique, qui permettrait au concepteur de préciser le contenu de l'instance de DC qu'il est en train de définir et de fournir le plus d'exemples possibles au programmeur (expert E1).

Tout en concédant l'importance de ces paramètres, nous considérons qu'étudier leurs relations respectives avec le DC constitue un sujet de recherche en soi. Dans cette thèse, nous nous sommes particulièrement penchés sur la relation entre le DC dans un STI, les théories de la cognition, et surtout le phénomène de la pensée réflexive.

Question 3 d'évaluation: Justesse des relations proposées dans le CSDC

L'originalité de notre contribution dans la définition du CSDC porte sur deux relations principales : la relation entre le DC dans un STI et la pensée réflexive à travers une boucle diagnostic-remédiation; la relation conceptuelle entre le DC et les paradigmes de cognition adoptés dans un STI. Les deux prochains paragraphes présentent une discussion relative à la justesse de ces relations, selon l'avis des experts consultés.

- **Sur la relation entre le DC et la pensée réflexive**

La relation entre le type de DC, l'algorithme de DC et son exploitation pédagogique est définitivement primordiale d'après les trois experts. En l'espèce, la pensée réflexive en tant qu'exploitation pédagogique de ce processus permet :

- d'une part de confirmer et de raffiner le DC (réflexion au sens informatique du côté du système);
- d'autre part, d'appréhender de manière plus précise les fondements d'une habileté, d'autant plus lorsqu'il s'agit d'une habileté de haut niveau (appliquer, exécuter, analyser, résoudre un problème).

Toutefois, une certaine réserve demeure sur la portée de la stimulation de la pensée réflexive par une machine d'après l'expert E2. Il a effectivement relevé le fait que dans des domaines plus complexes, la solution à un exercice s'élabore à travers des raisonnements différents, des nuances, des retours en arrière et même des compromis dans certains cas. Il devient alors difficile pour une machine de stimuler la pensée réflexive sur ces habiletés de haut niveau. Premièrement, il faut se rappeler que Prolog-Tutor a été utilisé pour illustrer des mises en œuvre de la conception du DC proposée dans cette thèse. Il ne s'agit en aucun cas d'une référence unique pour l'application de la pensée réflexive et c'est un choix circonstanciel à l'élaboration de notre travail. Deuxièmement, selon notre approche de la stimulation de la pensée réflexive n'est pas *ad hoc*, elle s'appuie sur la nature de la capacité à laquelle s'applique ce phénomène (section 6.4, ci-dessus en page 181). Plus particulièrement, nous avons exposé en quoi consiste la pensée réflexive à propos de *l'utilisation et de l'application d'un principe*. Ainsi, en analysant la nature d'une habileté de niveau supérieur, on pourrait déduire la manière dont la pensée réflexive sur une capacité correspondante devrait se réaliser. Cette analyse n'a pas été jugée nécessaire dans ce travail, du fait de la nature du domaine étudié dans Prolog-Tutor. Toutefois, nous anticipons qu'une recherche future aboutirait à une fin favorable sur la question de la stimulation de la pensée réflexive dans un STI, sur des habiletés de haut niveau. En effet, certaines recherches ont montré que toute habileté peut être représentée par un processus générique (Paquette, 2002). Par conséquent, une telle représentation formelle (idéalement systématique) serait facilement traduisible dans un langage compréhensible par la machine.

- **Sur la relation entre le DC et les paradigmes de cognition**

En ce qui concerne la relation entre le DC et les paradigmes de cognition, les experts sont tous du même avis : on ne saurait enfermer la conception pédagogique dans un paradigme de cognition, une théorie d'apprentissage ou une théorie d'instruction. Encore une fois, les objectifs pédagogiques doivent être formellement définis en termes d'habileté applicables à des éléments d'un domaine de connaissances. Seule la nature d'une habileté devrait influencer la nature du DC et l'algorithme correspondant dans un STI. C'est seulement à ce niveau que les paradigmes de cognition et les théories d'apprentissage interviennent. En effet, la nature d'une habileté peut être reliée à un paradigme de cognition et les processus qui en favorisent l'acquisition peuvent être reliés à une théorie d'apprentissage. Selon ces concepteurs, une telle considération permettrait de raffiner les relations entre le DC et ces paramètres (paradigmes de cognition et théories d'apprentissage), étant paramétrées par les habiletés associées aux capacités visées dans un STI. Ce raffinement favoriserait la définition de recommandations plus précises au niveau du CSDC.

À notre avis, cette position est pertinente et adéquate dans une certaine mesure, mais elle mérite fortement d'être nuancée. Une fois encore, nous maintenons que le DC implémenté dans un STI dépend de la position du concepteur par rapport à ce qu'il considère comme étant la cognition.

Une habileté d'application (niveau 3 dans la taxonomie de Bloom) peut être mesurée ou évaluée de manière behavioriste alors qu'elle correspond manifestement à un phénomène rationaliste où des causes et des effets sont exploités pour résoudre un problème. Par exemple, pour savoir si un apprenant sait appliquer la multiplication de deux nombres contenant plus de deux chiffres, on peut décider de ne considérer que les réponses qu'il fournit à des exercices de multiplication. Il s'agit là d'un diagnostic par évaluation sans inférence (recommandé dans un contexte empiriste behavioriste) appliqué à une habileté correspondant à un contexte d'apprentissage rationaliste cognitiviste.

7.5 Évaluation qualitative de Prolog-Tutor

Les aspects de la pertinence de la pensée réflexive pour le DC à travers une boucle diagnostic-réflexion ont été discutés lors de l'évaluation du CSDC par les experts. Plus particulièrement, l'originalité d'une boucle diagnostic-réflexion a été relevée et reconnue sur un plan conceptuel (Tableau VII.4, ci dessus, en page 209): le diagnostic d'une réponse de l'apprenant entraîne un STI à stimuler la réflexion de l'apprenant sur l'objet de ce diagnostic; le résultat de cette réflexion sont exploités à leur tour par le système pour appliquer divers ajustements au diagnostic initial (confirmation et raffinement en l'occurrence). Étant donné que Prolog-Tutor est un STI qui concrétise une telle vision de la relation DC-réflexion à travers les dialogues tutoriels, il est à présent question de confirmer la présence du phénomène de réflexion lors du DC qui se déroule dans ce système. En effet, de par la nature du dialogue tutoriel, la présence de la boucle diagnostic-réflexion en découlerait automatiquement.

7.5.1. Méthodologie : protocole de "penser à voix haute"

L'évaluation de Prolog-Tutor vise l'observation du phénomène de la pensée réflexive chez un apprenant, pendant qu'il résout un exercice à travers un dialogue tutoriel. Si l'on s'en tient à la classification des méthodes d'évaluation qualitative des STI de Murray (1993), l'on peut associer cette évaluation à une observation et une classification qualitative de phénomènes. L'objectif de cette méthode est d'observer des sujets pendant qu'ils effectuent une tâche reliée au phénomène ciblé et d'enregistrer la substance de leurs interactions. La tactique du protocole de "penser à voix haute" a été choisie dans ce sens. Le protocole de "penser à voix haute" est communément utilisé pour extérioriser les processus mentaux qui surviennent lors de l'exécution d'une tâche (Van Lehn, 1988). Le Tableau VII.5 résume les informations reliées à cette évaluation.

Tableau VII.5. Résumé des données de l'évaluation de Prolog-Tutor

Type d'information	Information
Objectif de l'évaluation	Détecter la pensée réflexive pour valider le dialogue tutoriel de Prolog-Tutor
Participants	6 étudiants expérimentés en Prolog
Données	Six enregistrements Audio
Durée des enregistrements	Entre 45 minutes et 1 heure 30 minutes

7.5.2. Collecte des données

De manière analogue à l'évaluation du CSDC, la collecte de données pour l'évaluation de Prolog-Tutor a commencé par le choix des participants. Les prochaines lignes décrivent les caractéristiques de ces participants, ainsi que la procédure et les instruments qui ont supporté leur participation respective.

7.5.2.1. Les participants

Pour valider la présence de la pensée réflexive chez un apprenant à travers l'interaction avec Prolog-Tutor, notre critère par rapport à cet apprenant est qu'il connaisse suffisamment le langage Prolog pour effectuer un exercice de *résolution d'un but*. Six étudiants dont le domaine d'intérêt variait entre l'informatique et la linguistique computationnelle ont participé à l'expérimentation avec Prolog-Tutor sur une base volontaire. Malgré qu'ils ne fussent pas tous des étudiants à temps plein en programmation logique et à l'université, ces participants ont été considérés comme représentatifs étant donné qu'ils présentaient chacun l'expérience minimale requise par rapport au langage Prolog. Leur point commun est qu'ils possédaient une connaissance suffisante de ce langage pour comprendre la *résolution d'un but* dans Prolog (APPENDICE. D: Tableau D. 4 et Tableau D. 5). La participation s'est faite sur une base volontaire après publication d'un avis d'expérimentation (courrier électronique, babillard universitaire, présentation de Prolog-Tutor lors des cours d'informatique dans des universités montréalaises) au cours du mois de Mai 2005. Six personnes ont répondu à cet avis de publication et ont toutes été retenues. Ces personnes ont été gratifiées d'une somme de 50 dollars canadiens au terme de l'expérimentation, ce après avoir signé une entente d'agrément issue d'un dossier de déontologie sur l'éthique de recherche que nous avons préalablement mis sur pied. Ce dossier a été soumis et approuvé par le comité compétent du département d'Informatique de l'Université du Québec à Montréal.

7.5.2.2. Procédure

Les participants ont utilisé Prolog-Tutor dans le cadre d'un exercice de *résolution de but*. Les expérimentations ont eu lieu entre le 08 Juillet et le 25 Juillet 2005 au Laboratoire GDAC (Laboratoire de Gestion et d'Acquisition des Connaissances) du département d'Informatique de l'Université du Québec à Montréal.

L'expérimentation s'est déroulée en deux étapes. Chaque participant était tenu d'effectuer: un pré test et un test avec Prolog-Tutor. Le pré test avait pour but d'apprécier le niveau de maîtrise du participant. Le test avait pour but d'étudier l'extériorisation de la pensée réflexive chez le participant. Chaque participant était tenu de penser à voix haute pendant la phase de test afin que son processus de réflexion soit enregistré.

7.5.2.3. Les instruments

L'instrument de cette évaluation est un dialogue tutoriel relatif à un exercice dans Prolog-Tutor. Au cours de cet exercice, les participants doivent effectuer la résolution d'un but en Prolog. La solution de cette résolution de but est produite de manière incrémentielle à travers un dialogue tutoriel. Tel que mentionné au Chapitre III et au Chapitre V, la structure de ce dialogue a pour but de stimuler la pensée réflexive. Le Tableau D. 6 (APPENDICE. D) décrit toutes les questions associées à ce dialogue tutoriel. L'on peut constater que certaines questions (celles des sous dialogues) ont pour but de stimuler une composante de la pensée réflexive.

7.5.3. Résultats : présentation et analyse

Dans cette section, nous expliquons d'abord comment nous avons codé les informations recueillies auprès des participants à travers un protocole de pensée à voix haute. Ensuite, nous analysons les résultats obtenus après avoir codé ces informations.

7.5.3.1. Codage

Prolog-Tutor implémente une structure de dialogue tutoriel dont le but est de stimuler explicitement ce phénomène chez l'apprenant. À titre de rappel, ces dialogues ont une structure à deux niveaux. Au premier niveau (dénomé D1), on retrouve des questions représentant une étape de construction de solution; chaque question est conçue de manière que la mise en œuvre d'une capacité est nécessaire pour y répondre. Chaque question du niveau D1 est associée à un ensemble de questions dont le but est de stimuler la pensée réflexive à propos de la capacité correspondante. Cet ensemble correspond au second niveau (niveau D1.2.) du dialogue tutoriel dans Prolog-Tutor. Chaque dialogue de niveau D1.2. est considéré comme un sous dialogue associé à la question correspondante du niveau D1. Les codes de cette analyse ont permis de regrouper et d'interpréter les enregistrements des protocoles de pensée à voix haute de chaque participant. La nature de ces codes est justifiée par les objectifs de cette analyse que nous commençons par préciser.

7.5.3.1.1. Objectif de l'analyse des protocoles de pensée à voix haute

Pour identifier la présence d'éventuels patrons de pensée réflexive pendant que les participants répondent aux questions du niveau D1.2, il faut déjà définir quelle étape de la pensée réflexive

chacune de ces questions est censée stimuler. L'analyse en tant que telle du protocole de pensée à haute voix a quatre buts principaux.

1. Premièrement, on aimerait établir dans quelle mesure le dialogue tutoriel a effectivement provoqué la présence ou la mise en œuvre de la pensée réflexive chez l'apprenant (section 7.5.3.2.1. et 7.5.3.2.2.).
2. Deuxièmement, on aimerait établir dans quelles mesures les participants sont conscients de cette pensée réflexive et en tirent profit (section 7.5.3.2.3.).
3. Troisièmement, on aimerait discuter d'une caractéristique imprévue de la pensée réflexive qui a émergé de l'interaction avec Prolog-Tutor (section 7.5.3.2.4.).
4. Enfin, une discussion relative à l'impact des facteurs de l'implémentation de Prolog-Tutor sur les perceptions des participants et sur l'objectif de provoquer une pensée réflexive (section 7.5.3.2.5.).

L'atteinte de ces buts nécessite la recherche de l'occurrence de certains événements dans les protocoles enregistrés. Les prochaines lignes décrivent les codes d'analyse qui ont été définis pour mener formellement cette analyse (Miles et Huberman, 2003).

7.5.3.1.2. Codes pour la structure du dialogue tutoriel

Le Tableau D. 6 (APPENDICE. D) présente des codes qui permettront d'identifier:

- les quatre sous dialogues utilisés dans cette expérimentation avec Prolog-Tutor;
- les questions associées à chacun de ces sous dialogues; on peut remarquer que les codes de ces questions indiquent implicitement la composante de la pensée réflexive qu'elle vise à stimuler

7.5.3.1.3. Codes pour l'analyse du dialogue tutoriel

Concernant la présence de la pensée réflexive en tant que telle, on cherche dans les enregistrements de chaque participant, à identifier pour chaque question, un événement pouvant être interprété comme une occurrence de la composante associée de pensée réflexive. Concernant la question d'être conscient de la démarche pédagogique de provoquer la pensée réflexive, on cherche dans les enregistrements à identifier pour chaque sous dialogue, une réaction qui indiquerait que le participant comprend pourquoi le tuteur lui pose les questions correspondantes.

Les codes définis au Tableau D. 7 (APPENDICE. D) ont été utilisés pour analyser les enregistrements de chaque participant (APPENDICE. E: Tableau E. 3, Tableau E. 4, Tableau E. 5, Tableau E. 6, Tableau E. 7 et Tableau E. 8); les codes définis au Tableau D. 8 (APPENDICE. D) ont été utilisés pour désigner chacun des six participants ainsi que leurs enregistrements respectifs. Notre démarche s'intéresse à trois étapes principales de la pensée réflexive: l'exposition à une situation problématique, l'intellectualisation de la situation et le raisonnement sur les faits intellectualisés. On remarque qu'un ensemble de codes a été défini pour chacune de ces phases. Ces codes sont également utilisés pour étudier les caractéristiques imprévues de la réflexion. Il est important de noter que pour

chaque composante de la pensée réflexive, les codes ont été énumérés dans un ordre préférentiel d'occurrence. Ainsi, lorsqu'on désire reconnaître qu'un participant a effectivement été soumis à une situation problématique dans le dialogue tutoriel (à travers une question numéro j dont le code serait alors Q_j_S), il serait préférable d'observer une réaction perplexe plutôt qu'une réponse directe. De même dans les phases de raisonnement, on préfère voir le participant énoncer des principes, des lois avant de les appliquer à la situation particulière de l'exercice, plutôt que d'observer une application directe – à l'instar d'une connaissance compilée- d'une loi ou d'un principe.

Concernant l'impact des facteurs d'implémentation de Prolog-Tutor sur la perception par les apprenants de leur expérience, une étape de débriefing a été réalisée pour chaque participant (APPENDICE. E: Tableau E. 3, Tableau E. 4, Tableau E. 5, Tableau E. 6, Tableau E. 7 et Tableau E. 8). Au cours de ce débriefing, ces participants ont eu l'occasion d'exprimer leur appréciation personnelle de l'expérience avec Prolog-Tutor, tant sur la forme (la facilité d'utilisation de l'interface de Prolog-Tutor) que sur le fond (la perception de la démarche pédagogique déployée dans Prolog-Tutor).

7.5.3.2. Analyse

Pour compiler et analyser les résultats de l'évaluation de la pensée réflexive dans Prolog-Tutor, nous avons effectué un examen approfondi d'environ 6 heures d'enregistrement des protocoles de pensée à haute voix des participants (APPENDICE. D: Tableau D. 9).

7.5.3.2.1. Présence et caractéristiques de la pensée réflexive

Les informations du Tableau VII.6 ont été extraites des transcriptions détaillées des protocoles de verbalisation de la pensée des participants (APPENDICE. E: Tableau E. 3, Tableau E. 4, Tableau E. 5, Tableau E. 6, Tableau E. 7 et Tableau E. 8). Ils présentent une compilation codée de la pensée extériorisée par chacun des participants au cours de leur expérience respective avec Prolog-Tutor. Un examen en perspective de ces tableaux permet d'estimer que les sous dialogues tutoriaux de Prolog-Tutor provoquent effectivement la pensée réflexive. En effet, pour chacun des participants, on peut constater que chacun des sous dialogues donne lieu à des réactions d'où émerge un patron correspondant aux trois composantes principales de la pensée réflexive: la prise de connaissance d'une situation problématique, l'intellectualisation de cette situation en y relevant les faits qui lui confèrent ce caractère problématique et enfin la formulation d'une solution à la situation sur la base d'un raisonnement fondé sur un lien entre des principes, des lois et les faits intellectualisés.

Malgré une confirmation claire de l'occurrence du phénomène de pensée réflexive sur une capacité diagnostiquée (comme acquise ou non acquise) au niveau D1, l'on se doit d'examiner plus en détails la portée de cette occurrence sur la qualité de l'expérience d'apprentissage. Deux critères nous permettent d'examiner cette qualité.

Tableau VII.6. Pensée extériorisée dans les sous dialogues de Prolog-Tutor

Participant	Numéro de sous dialogue	Séquence codée de la pensée extériorisée
P_1	SD0	RDC → [OBS_Q, OBS_E] → [R_Arg]
	SD1	RDC → [OBS_Q] → [R_Pr, OBS_E] → [R_App]
	SD2	RDC → [R_E, R_App] → [R_Pr, R_E, R_App]
	SD3	DAT → [R_Pr, R_App] → [R_E, R_App]
P_2	SD0	RDC → [OBS_E] → [R_Arg]
	SD1	[RDC, ANP] → [OBS_E] → [PX, OBS_E] → [OBS_E, OBS_Q, R_App, R_E]
	SD2	HT → [R_E, R_App] → [R_E, R_App]
	SD3	RDC → [R_App] → [R_App]
P_3	SD0	RDC → OBS_E → [R_0]
	SD1	RDC → [OBS_E] → [PX, OBS_E] → [R_App]
	SD2	HT → [OBS_R, OBS_Q, R_App] → [OBS_E, R_E, R_App]
	SD3	RDC → [R_E, R_App] → [R_App]
P_4	SD0	RDC → OBS_E → R_Arg
	SD1	RDC → [OBS_Q, ANP] → [OBS_R] → [R_App]
	SD2	HT → [OBS_Q, R_App] → [OBS_E, R_E, R_App]
	SD3	RDC → [R_Pr, OBS_R, R_App] → [R_App]
P_5	SD0	[HT, DAT] → OBS_E → R_0
	SD1	RDC → [OBS_Q, OBS_R] → [OBS_E] → [R_App]
	SD2	RDC → R_App → [R_E, R_App]
	SD3	RDC → R_App → [OBS_R, OBS_E, R_App]
P_6	SD0	[HT, OBS_Q] → OBS_E → [OBS_E, R_Arg, R_Pr]
	SD1	RDC → [OBS_E] → [HT, OBS_R, OBS_E] → [R_App]
	SD2	RDC → [OBS_Q, OBS_E, R_App] → [R_E, R_App]
	SD3	RDC → [R_E, R_App] → [OBS_E, R_App]

Premièrement, une lecture de la nature spécifique des occurrences représentant chaque composante de la pensée réflexive doit être effectuée en vue de la caractériser. Deuxièmement, la stimulation de la pensée réflexive n'est d'intérêt que si un apprenant est conscient d'effectuer cette démarche. Dans ce sens, nous avons essayé d'identifier des réactions qui peuvent révéler d'une

manière ou d'une autre que les participants ont compris que le but des sous dialogues était de stimuler la pensée réflexive.

7.5.3.2.2. *Caractéristiques de la pensée réflexive*

L'analyse précédente nous a permis d'établir la présence de la pensée réflexive à travers l'interaction avec Prolog-Tutor. Ce paragraphe vise à caractériser l'occurrence de ce phénomène dans ce contexte.

Le Tableau VII.7 énumère les occurrences des codes associés à chaque étape de la pensée réflexive, pour chaque participant. Dans le dernier groupe d'énumération, on peut voir les codes co-occurents relatifs à la phase de raisonnement. L'énumération de ces codes co-occurents est importante. En effet, cela permet une différenciation entre l'application directe d'un principe (ou d'une loi) et l'application directe d'un principe après l'avoir clairement énoncé. Du point de vue de la pensée réflexive, cette différenciation est importante dans la mesure où il est question de stimuler un raisonnement fondé. Dans ce sens, l'application d'un principe ou d'une loi après son énonciation montre une connaissance du principe, une capacité à l'utiliser adéquatement (étant donné les faits associés à une question, reconnaître quel principe, quelle loi permet de répondre à la question) et une capacité à l'appliquer adéquatement. L'application directe d'un principe ou d'une loi a contrario ne permet pas de tirer une telle conclusion, puisqu'elle peut simplement découler d'une connaissance compilée.

Les codes qui ont été utilisés pour analyser les enregistrements sont soumis à une échelle préférentielle. En effet, l'on doit se rappeler que ces codes sont associés à différentes manifestations de chaque composante de la pensée réflexive. Certaines manifestations représentent mieux chacune de ces composantes; par exemple, une énonciation claire des faits relatifs à une situation problématique représente une intellectualisation plus authentique qu'un simple questionnement sur ces faits.

On constate premièrement que la majorité (80%) des situations problématiques proposées pour stimuler la pensée réflexive lors de la mise en oeuvre d'une capacité a suscité une réponse directe des participants (Tableau VII.7, ligne du code RDC). Les situations problématiques ont très peu provoqué une confusion (4%) ou une polémique (1%), tel qu'il aurait été souhaitable dans une situation de stimulation de pensée réflexive plus authentique, selon les prescriptions de la théorie de Dewey. Deux explications potentielles permettent de comprendre cet état de faits. La première explication est reliée au niveau de complexité du domaine étudié dans Prolog-Tutor, plus particulièrement, lorsqu'il s'agit d'effectuer la résolution d'un but en Prolog. Il s'agit d'une tâche procédurale, bien définie et bien cadrée et dont la résolution est directe étant donné un contexte. Il serait donc difficile de définir une situation réellement problématique et non triviale dans des exercices relatifs à cette tâche. La seconde explication est reliée au niveau de maîtrise des participants à cette expérimentation. En effet,

si l'on reconsidère les informations du Tableau D. 4 (APPENDICE. D), la majorité des participants présentaient un niveau de maîtrise de Prolog plutôt élevé, ce qui les prédispose à résoudre le problème posé dans Prolog-Tutor assez facilement. Pour palier à cette faiblesse de Prolog-Tutor, une attention devrait être portée sur deux questions principales:

- (1) Les implications inhérentes à la stimulation de la pensée réflexive après le diagnostic d'une capacité relative aux connaissances d'un domaine plus complexe que la logique et surtout, plus riche en contenu et en situations d'application (suggestion de la majorité des participants; voir APPENDICE. E);
- (2) La variation des présentations des questions. Il s'agit de poser des questions de manière différente, par exemple en présentant la solution ou une partie de solution à un problème et en demandant à l'apprenant de construire le processus de raisonnement qui y a mené (suggestion du participant P_4; APPENDICE. E: Tableau E. 6).

Deuxièmement, la majorité des questions visant à amener le participant à observer les faits saillants et les informations clés d'une situation à résoudre ont atteint leur but. En effet, 67% des réactions des participants correspondent à une identification effective de ces informations (Tableau VII.7, ligne du code OBS_E), 19% d'entre elles correspondent au moins à des interrogations. Bien que cela indique clairement que les questions correspondantes du sous dialogue ont provoqué une réelle intellectualisation de la situation, il ne faut pas oublier que cette intellectualisation n'est utile que si un apprenant est effectivement conscient qu'il est entrain de relever les informations clés qui lui permettront de résoudre la situation en main. Fort à propos, de notre analyse de 5h et 20 minutes d'enregistrement, il est ressorti une impression selon laquelle les questions d'intellectualisation étaient justement trop explicites. L'on entend par là que ces questions étaient tellement directes (elles interrogent effectivement sur les faits importants du problème) que dans certains cas, les participant n'arrivaient pas à les relier à la situation à résoudre. Une discussion relative à ce constat fait l'objet de la prochaine section sur la "conscience de la pensée réflexive".

Troisièmement, l'on s'est penché sur l'effet des questions visant à amener le participant à formuler une solution ou une réponse à une situation problématique, en se fondant sur un principe ou une loi du domaine (1) et en appliquant ce principe ou ces lois aux faits intellectualisés (2). Il ressort que la majorité du temps (40%), les participants ont effectué une application directe d'un principe (Tableau VII.7, ligne du code R_App) pour formuler la solution à une situation problématique; on remarque tout de même que 31% du temps, l'énonciation de l'application d'un principe en précédait l'application directe lors de la formulation d'une réponse. Du reste, l'énonciation formelle d'un principe a été très peu observée puisqu'elle apparaît dans seulement 9% des réponses (Tableau VII.7, lignes des patrons de codes [R_Pr, R_E, R_App] et [R_Pr, R_App]). Au tout début de notre analyse, il nous a semblé primordial que les apprenant indiquent clairement et formellement sur quoi ils se

basent pour formuler une réponse, d'où l'importance accordée a priori aux patrons de raisonnement de la forme (par ordre d'importance): [R_Pr, *], [R_E,*]. Toutefois, l'analyse des protocoles de pensée à voix haute nous a permis d'établir que dès lors qu'on est capable d'identifier les points clés d'un problème et de formuler une solution en énonçant comment s'appliquent ces points à des principes, on peut inférer que la personne en question comprend de quoi il est question. Du reste, la majorité relative des applications directes des principes peut s'expliquer par le fait que les participants présentaient un très bon niveau de maîtrise de Prolog. Par conséquent, il est tout à fait possible que leurs capacités aient atteint un certain niveau de compilation (en ce qu'elles sont mises en oeuvre et utilisées automatiquement, à l'instar d'un expert).

7.5.3.2.3. *Conscience de la pensée réflexive*

Pour analyser la manière dont les participants ont compris la démarche pédagogique sous-jacente au dialogue tutoriel dans Prolog-Tutor, l'on s'est penché sur deux questions. La première d'ordre général s'interroge sur la nature même de cette compréhension en examinant les faits qui l'attestent. La seconde question est plus spécifique et s'interroge sur le moment auquel les participants ont manifesté des réactions qui témoignent de cette compréhension. Cette deuxième question est importante pour estimer l'impact de la pensée réflexive sur l'expérience d'apprentissage. En effet, il ne faut pas oublier que la pensée réflexive est "provoquée" dans Prolog-Tutor, elle ne découle donc pas d'une volonté propre de l'apprenant. Nous pensons que la mise en oeuvre de ce processus n'est bénéfique que lorsqu'il est réalisé consciemment, l'apprenant étant conscient de la pertinence de chacune des actions ou des activités associées, par rapport à l'objectif global d'un exercice. Par conséquent, il est souhaitable qu'un apprenant soit conscient de la démarche de pensée réflexive et ce le plus tôt possible dans les sous dialogues.

Le Tableau VII.8 indique respectivement pour chaque participant: (1) le nombre de fois qu'une réaction indiquant une compréhension de la démarche pédagogique sous jacente au dialogue tutoriel (sans l'intervention de l'expérimentateur et avec l'intervention de l'expérimentateur) a été observée pour chaque participant; (2) le nombre de fois qu'une réaction indiquant une incompréhension de cette démarche pédagogique a été observée, pour chaque participant; (3) le nombre minimum de fois qu'une réaction de compréhension de la démarche pédagogique de Prolog-Tutor a été observée pour chaque sous dialogue du dialogue tutoriel, pour chaque participant.

Tableau VII.7. Fréquences d'occurrence des omposantes de la pensée réflexive

Nombre total de stimulation	Codes des composantes de la pensée réflexive	# Occurrences observées par participant						# Total d'occurrences observées
		P_1	P_2	P_3	P_4	P_5	P_6	
		Phase de présentation d'une situation problématique						
4	PX	0	0	0	0	0	0	0
	HT	0	1	1	0	1	1	4 (16%)
	DAT	0	0	0	0	1	0	1 (1%)
	ANP	0	1	0	0	0	0	0
	IS	0	0	0	0	0	0	0
	RDC	4	3	3	4	3	3	20 (80%)
	IB	0	0	0	0	0	0	0
Total d'occurrences								25
		Phase d'intellectualisation						
3	OBS_Q	2	0	0	1	1	0	4 (19%)
	OBS_E	2	3	3	1	2	3	14 (67%)
	OBS_R	0	0	0	1	1	1	3 (14%)
Total d'occurrences								21
		Phase de raisonnement et d'inférence correspondante d'une réponse						
6	R_Pr	1	0	0	1	0	1	3
	R_E	2	3	2	1	1	2	11
	R_LC	0	0	0	0	0	0	0
	R_App	3	5	5	5	5	5	28
	R_V	0	0	0	0	0	0	1
	R_Arg	1	1	0	1	0	1	4
	R_0	0	0	1	0	1	0	2
Total d'occurrences								49
		patrons de raisonnement (Codes co-occurents)						
	[R_E, R_App]	2	3	2	1	1	2	11 (31%)
	[R_Pr, R_App]	1	0	0	1	0	0	2 (6%)
	[R_Pr, R_E, R_App]	1	0	0	0	0	0	1 (3%)
	R_App							14 (40%)
	Autres							7 (20%)
Total d'occurrences								35

Selon les informations du Tableau VII.8, les participants sont presque toujours conscients (4.16 fois pour 4 sous dialogues visant à provoquer la pensée réflexive ou encore 104% du temps) que le but de Prolog-Tutor est de les amener à effectuer une pensée réflexive sur les capacités associées à l'exercice qu'ils étaient en train de résoudre. La majorité du temps (5 fois sur 6 ou 83% du temps), une réaction indiquant cet état de faits a été identifiée au moins une fois dans chaque sous dialogue visant à

provoquer la pensée réflexive (Tableau VII.6, ci-dessus en page 223). Enfin, on peut estimer que ce discernement de la démarche pédagogique qui se cache derrière les dialogues tutoriel émerge naturellement des interactions avec Prolog-Tutor. En effet, on constate que seulement 0.83 fois sur 4, une intervention explicite de l'expérimentateur a été nécessaire pour permettre au participant de comprendre la démarche pédagogique sous jacente au dialogue tutoriel. Si l'on considère les participants qui ont eu des difficultés à comprendre cette démarche (P_3 et P_6 en l'occurrence), on constate qu'ils finissent par comprendre la démarche pédagogique après l'intervention du tuteur (Tableau VII.6, groupe de lignes associé à P_3) ou plus tard dans le dialogue tutoriel (APPENDICE E: Tableau E. 8). Ceci nous amène à l'examen des moments auxquels les participants ont manifesté une certaine compréhension de cette démarche lors de leur expérience avec le dialogue tutoriel de Prolog-Tutor.

Tableau VII.8. Conscience de la démarche de pensée réflexive

Nombre minimum d'observations souhaitables	Réactions des participants indiquant s'il comprend (ou pas) la démarche de pensée réflexive	Nombre de réactions observées par participant						Moyenne et pourcentages
		P_1	P_2	P_3	P_4	P_5	P_6	
4 (1 par sous dialogue)	#R_A_BUT sans intervention du tuteur	4	4	4	5	5	3	4.16/4 (104%)
	# R_A_BUT avec intervention du tuteur	0	2	1	1	1	0	0.83/4 (21%)
	# R_NA_BUT	0	0	2	1	0	1	0.67/4 (17%)
	# minimum de R_A_BUT par sous dialogue	1	1	1	1	1	0	0.83/sous-dialogue

Le Tableau VII.9 indique le moment auquel chaque participant a réalisé - ou a compris- la nature de la démarche pédagogique sous jacente au dialogue tutoriel. En d'autres termes, il s'agit du moment où les participants ont pris conscience du fait que le but du dialogue était de les amener à effectuer une pensée réflexive. Dans ce cas, nous avons retenu pour chaque sous dialogue – dont le but est de stimuler la pensée réflexive sur la capacité associée à la question correspondante dans le dialogue de niveau supérieur (D1) -, le (les) numéro(s) de la question au cours de laquelle une réaction de compréhension de la démarche pédagogique a été enregistrée. En examinant les données du Tableau VII.9, on peut considérer que les participants prennent conscience de la démarche pédagogique assez tard dans les sous dialogues tutoriaux. Au minimum, les participants passent plus de la moitié d'un sous dialogue sans comprendre ce qui se passe, bien que cela finisse par arriver au

plus tard à fin de chaque sous dialogue. Trois caractéristiques du dialogue tutoriel de Prolog-Tutor peuvent justifier ce constat.

Tableau VII.9. Étapes auxquelles la démarche de pensée réflexive est conscientisée

Participant	Étape où la démarche pédagogique a été comprise (Numéro de la question/ # total de questions dans le sous-dialogue)			
	P_1	Q0.Assert (3/3)	Q1.Assert1 (4/4)	Q2.Assert1 (2/2)
P_2	Q0.Assert (3/3)	Q1.Intellect (2/4) Q1.Intellect2 (3/4)	Q2.Assert1 (2/2)	Q3.Assert1 (3/3)
P_3	Q0.Assert (3/3)	Q1.Intellect (2/4)	Q2.Assert0 (1/2) Q2.Assert1 (2/2)	Q3.Assert1 (3/3)
P_4	Q0.Assert (3/3)	Q1.Intellect (2/4)	Q2.Assert1 (2/2)	Q3.Assert0 (2/3)
P_5	Q0.Assert (3/3)	Q1.Intellect (2/4)	Q2.Assert1 (2/2)	Q3.Assert1 (3/3)
P_6	Q0.Assert (3/3)	Q1.Intellect (2/4)	Q2 (0/2 : dialogue de niveau supérieur) Q2.Assert1 (2/2)	Q3.S (1/3)
Sous-Dialogue	SD0	SD1	SD2	SD3
Étape Moyenne où la démarche pédagogique a été comprise	3/3	2.43/4	1.62/2	2.5/3

Premièrement, au niveau de la structure du dialogue tutoriel, les commentaires des participants recueillis dans la phase de débriefing de l'expérimentation indiquent qu'il y aurait un intérêt à introduire explicitement chaque sous dialogue avec une description de son but pédagogique par rapport au but global de l'exercice actuellement résolu.

Deuxièmement, au niveau du contenu du dialogue tutoriel, nous nous sommes demandés si certaines questions n'étaient pas trop explicites. En fait, si l'on réexamine le Tableau VII.7 (ci-dessus, en page 230), on constate par exemple que les participants intellectualisent effectivement la situation problématique qui leur est présentée dans chaque sous dialogue (67% du temps, les informations clés du problème sont effectivement identifiées). Cependant, la phase d'intellectualisation (du processus de pensée réflexive) survient généralement au milieu du sous dialogue. Par conséquent, certains participants intellectualisent une situation problématique sans nécessairement savoir à quoi cela sert dans l'exercice qu'ils essaient d'effectuer. Nous pensons qu'à l'instar des questions visant à provoquer une intellectualisation de la situation, certaines questions des sous dialogues sont tellement explicites

qu'elles provoquent effectivement le résultat visé, sans qu'un lien soit nécessairement effectué avec le but global de l'exercice. Une façon de remédier à cette faiblesse serait d'encourager les participants à utiliser la commande "Pourquoi" de l'interface de Prolog-Tutor (Tableau VII.10). En effet, lorsque sollicitée, cette commande permet à Prolog-Tutor d'expliquer à son utilisateur la pertinence d'une question par rapport au but global de l'exercice.

Troisièmement, une considération du contenu du dialogue tutoriel sous un angle différent indique qu'on peut annexer directement chaque question du sous dialogue avec une justification de sa pertinence par rapport au but pédagogique global de l'exercice qui est actuellement résolu.

Tableau VII.10. Fréquence de demande d'explication sur les questions des sous dialogues

	Participants					
	P_1	P_2	P_3	P_4	P_5	P_6
# Sollicitations de la commande "Pourquoi" de Prolog-Tutor	0	0	0	1	1	1

7.5.3.2.4. Caractéristiques imprévues de la pensée réflexive

Un examen attentif du Tableau VII.6 (ci-dessus en page 223) révèle que certaines manifestations de différentes composantes de la pensée réflexive apparaissent à des moments inattendus. Dans ce cas, les codes correspondants à ces manifestations sont indiqués en gras (pour signifier que ces manifestations n'avaient pas été prévues à cet endroit dans la conception du dialogue tutoriel de Prolog-Tutor). Par exemple, au Tableau E. 4, sous dialogue SD1: le participant P_2 essaie de répondre à une question dont le but est de provoquer un raisonnement fondé sur des faits probablement identifiés dans le sous dialogue précédent (SD0). Tout en répondant, il formule des faits observés et se pose des questions sur la situation en question. À notre avis, cette remarque n'a pas vraiment d'impact sur le dialogue implémenté dans Prolog-Tutor.

Cependant, elle suggère que la nature interrogative d'une approche pédagogique de réalisation d'exercice dans un STI pourrait stimuler des composantes de réflexion. En effet, si l'on considère les informations compilées au Tableau VII.11, on constate que des réactions indiquant la présence de la réflexion ont été observées dans des questions qui ne correspondaient pas à un sous dialogue visant à provoquer ce phénomène (questions Q1, Q2, Q3, Q4, Q5). On peut aussi voir que les réactions les plus fréquentes de la pensée réflexive correspondent à l'intellectualisation et au raisonnement (respectivement 18 et 19 occurrences, contre 2 occurrences seulement de réaction associables à une confusion émanant d'une situation problématique) sur une situation. On peut interpréter cela en suggérant que la pensée réflexive peut spontanément apparaître dans un contexte pédagogique centré sur l'interrogation de l'apprenant, même si la situation qu'il doit étudier n'est pas particulièrement problématique.

7.5.3.2.5. Impact de l'implémentation de Prolog-Tutor sur la pensée réflexive

Dans les phases de débriefing réalisées auprès des participants, on a pu constater que certaines caractéristiques de l'implémentation de Prolog-Tutor ont fortement influencé leur expérience et par conséquent, leur perception de la pédagogie déployée dans le système. Deux caractéristiques ont révélé les impacts les plus importants. Il s'agit de :

- (1) l'interface principale de Prolog-Tutor : la majorité des participants a émis des recommandations pour la rendre plus explicite de la pédagogie. En l'occurrence, on a suggéré de définir une numérotation hiérarchique des questions du dialogue, de manière à faire ressortir les questions du niveau D1 par rapport à celles du niveau D2. On également suggéré de fournir des feedback plus explicites, à travers un agent émotif par exemple. Cet agent adopterait une attitude selon la nature de chaque réponse fournie par l'apprenant. L'importance du feedback tient du fait qu'il permet de comprendre *le but des sous dialogues* (réponse incorrecte: raffiner le diagnostic; réponse correcte: renforcer ou vérifier le diagnostic de l'acquisition d'une capacité. Des feedbacks plus explicites permettraient de faire ces justifications distinctives pour chaque sous dialogue déclenché;
- (2) le niveau de complexité de l'exercice étudié : la majorité des participants a relevé le fait que les activités d'apprentissage actuellement implémentées dans Prolog-Tutor seraient d'un apport essentiel pour un débutant en Programmation Logique. Étant des experts ou des personnes maîtrisant bien ce domaine de connaissances, ces participants ont reconnu que le bénéfice qu'ils ont tiré de leur expérience est que la pédagogie de Prolog-Tutor les a forcé à articuler leur connaissance, dévoilant des éléments importants qu'ils tenaient pour acquis ou pour implicites, alors qu'il s'agissait de capacité à part entière. Dans ce sens, deux suggestions ont été retenues. Premièrement, il serait intéressant de tester Prolog-Tutor avec des débutants en Programmation Logique. Deuxièmement, il serait intéressant d'implémenter des activités plus complexes dans ce domaine; par exemple une telle activité aurait pour but d'amener les apprenants à construire une base de connaissances en Prolog, étant donnée une situation décrite en langage naturel (APPENDICE. E: Tableau E. 8).

7.6 Conclusion

Dans le but d'analyser la portée des hypothèses avancées dans cette thèse, deux évaluations qualitatives ont été présentées dans ce chapitre. La première évaluation a été réalisée à travers trois entrevues avec trois experts en conception pédagogique. La deuxième évaluation a été réalisée par le biais d'une expérimentation où six participants ont interagi avec Prolog-Tutor. La mise en oeuvre d'un protocole de pensée à haute voix a permis d'enregistrer les réactions de ces participants lors de cette interaction, permettant ainsi une analyse de l'effet du dialogue tutoriel de Prolog-Tutor relativement à la stimulation de la pensée réflexive sur une capacité diagnostique comme acquise ou non acquise.

La première évaluation a porté sur trois points: la pertinence du CSDC, la pertinence du contenu du CSDC, l'utilité effective du CSDC. Des entrevues qui ont été effectuées, il est ressorti un avis favorable par rapport à ces points, qui nous permet de confirmer nos hypothèses de départ jusqu'à un certain point. Étant donné le caractère fondateur du CSDC, de nombreuses améliorations sont possibles et dans ce sens deux suggestions principales ont été retenues de cette évaluation. Premièrement, en plus de la considération explicite des paradigmes, de considérer la relation entre un niveau d'habileté visé dans une activité d'apprentissage dans un STI et le DC correspondant. Deuxièmement, bien que les illustrations de CD-SPECIES semblent prometteuses pour la conception du DC dans un STI, son utilité effective serait tributaire d'une interface concise et d'une possibilité pour le concepteur pédagogique de spécifier plus de contenu associé à l'instance de DC qu'il est entrain de spécifier.

La deuxième évaluation a également porté sur trois points: l'occurrence de la réflexion dans Prolog-Tutor, la nature de cette occurrence et sa portée sur l'expérience d'apprentissage. L'analyse des enregistrements des protocoles de pensée à haute voix nous a permis de confirmer la présence de patrons de réactions associées aux composantes de la pensée réflexive selon Dewey. Ces patrons révèlent que les phases d'intellectualisation et de raisonnement sont effectivement provoquées, tandis qu'en général les phases de présentation d'une situation problématique n'ont pas vraiment provoqué de confusion chez les participants. Nous avons pu expliquer cela en référence à leur niveau de maîtrise assez élevé du domaine de la programmation logique. Pour ce qui est de la portée de la réflexion sur

l'expérience d'apprentissage, nous avons pu établir que les participants finissent par comprendre que le but des dialogues tutoriels est de provoquer en eux une réflexion, bien que cela arrivât généralement tard dans le cours de leur interaction avec Prolog-Tutor.

Ces constats reflètent la portée de la contribution de cette thèse, tout en évoquant indirectement les futures pistes de recherche qui demeurent à explorer.

CONCLUSION

Dans le tutorat humain, le diagnostic du comportement d'un apprenant par l'enseignant survient naturellement et intuitivement, à l'instar de la majorité des autres mécanismes pédagogiques. Nonobstant une absence de formalisme dans sa définition, il est mieux appréhendé du fait des moyens de communications privilégiés qui caractérisent ce contexte de tutorat.

Le domaine de l'intelligence artificielle en éducation a pour but de produire des artefacts qui favorisent l'apprentissage basé sur une interaction Humain-Machine, à travers une instruction adaptative et ciblée. Les STI constituent une catégorie de ces artefacts et le diagnostic cognitif du comportement de l'apprenant lors de ses interactions avec un STI est indispensable à l'adaptabilité de l'instruction. Contrairement à l'instruction classique, l'application du DC dans un STI nécessite une compréhension et une formalisation claires de ce processus, afin d'en assurer une conception adéquate, eu égard aux moyens relativement limités qui caractérisent la communication Humain-Machine.

Le travail élaboré dans cette thèse porte sur une considération explicite de la dimension pédagogique du DC, lors de son application dans un STI. Partant du constat qu'une spécification à part entière de chaque instance de processus de DC devrait en précéder l'implémentation dans un STI pour favoriser la fidélité pédagogique de sa mise en oeuvre, nous avons proposé un cadre de spécification du DC pour un STI. Deux dimensions pédagogiques du DC ont particulièrement été soulignées dans ce cadre: (1) la considération explicite de l'influence d'un paradigme de cognition sur la nature du DC dans un STI et, (2) la considération explicite du lien entre le DC et l'exploitation pédagogique qui en est faite pour favoriser l'apprentissage, ce à travers une boucle diagnostic-remédiation. Dans cette

thèse en particulier, la stimulation de la pensée réflexive telle que définie dans la théorie de John Dewey a été adoptée comme approche de remédiation.

Pour dégager l'utilité pratique de la conception du DC à travers un CSDC, l'architecture et les fonctions de CD-SPECIES - un système d'assistance à la spécification d'une instance de DC - ont été présentées.

Pour dégager l'utilité pratique de la perspective du DC sous l'angle de son exploitation pédagogique par la stimulation de la pensée réflexive, elle a été implémentée et mise en œuvre dans Prolog-Tutor – un prototype de STI pour l'apprentissage de la programmation logique.

Deux évaluations qualitatives ont clôturé cet exposé sur la dimension pédagogique du DC dans les STI. La première évaluation a été réalisée à travers trois entrevues avec trois experts en conception pédagogique. Le but de ces entrevues était une évaluation conceptuelle des hypothèses de notre travail:

- pour favoriser une fidélité de l'implémentation du DC par rapport à ses objectifs pédagogique: la nécessité d'un CSDC comme intermédiaire sémantique entre un concepteur de STI et un programmeur de STI;
- pour favoriser une cohérence conceptuelle du DC: la considération explicite de l'influence des paradigmes de cognition sur la nature du DC dans un STI;
- pour favoriser la continuité dans l'application du DC: la considération de l'influence de la stimulation de la pensée réflexive comme exploitation pédagogique du DC (ou comme remédiation pendant un DC), sur la manière dont il doit être implémenté dans un STI, ce à travers une boucle diagnostic-remédiation.

La deuxième évaluation a été réalisée à travers une interaction entre le prototype de STI Prolog-Tutor et six participants ayant un niveau de maîtrise relativement élevé de la programmation logique. Le but de cette évaluation était d'observer et de caractériser les occurrences de pensée réflexive provoquée au cours de cette interaction caractérisée par la mise en œuvre de la boucle diagnostic-remédiation dans Prolog-Tutor. Ces évaluations nous ont permis de valider les hypothèses de ce travail et surtout, de mettre en exergue les pistes de recherches qui en ont émergé.

Contribution à la recherche AIED

L'originalité de la contribution de cette thèse s'aligne d'abord sur l'**aspect conceptuel de la recherche sur les STI dans le domaine AIED**. En essence, l'importance de considérer l'application du DC dans un STI en tenant compte de trois facteurs pédagogiques a été soulignée. Il s'agit principalement de:

1. Favoriser la fidélité de l'implémentation du DC par rapport à l'intention pédagogique sous-jacente. Il s'agit surtout de choisir les mécanismes de représentations de connaissances les mieux appropriés pour effectuer un type d'inférence à propos de l'état cognitif de l'apprenant.
2. Refléter l'influence des paradigmes de cognition sur le DC dans un STI.
3. Refléter l'influence de l'exploitation pédagogique du DC dans un STI à travers une boucle diagnostic-remédiation.

Nous pensons que ces facteurs confèrent une dimension plus pédagogique à un processus qui a tendance à se limiter à l'application d'un algorithme, étant donné le milieu informatique auquel il est confiné. Cette perspective est d'autant plus importante qu'il s'agit d'émuler une des fonctions pédagogiques les plus importantes du tutorat humain. La première évaluation qui a été effectuée au terme de notre travail corrobore la réalité de cette contribution. En effet, tous les évaluateurs se sont accordés sur le fait que DC dans les STI est généralement compris sous un angle unique. Le CSDC qui a été présenté définit les principales perspectives de considérations du DC tout en précisant clairement les conditions de chacune de ces perspectives. En outre, pour être adéquatement adaptable, le tutorat doit cibler les besoins réels des apprenants à travers un DC, c'est-à-dire des besoins qui reflètent ses faiblesses en relation aux objectifs d'apprentissage visés. Ces objectifs étant définis entre autres par un paradigme de cognition, d'où l'importance d'effectuer un DC en fonction des implications découlant d'un tel paradigme. Enfin, le DC ne doit pas seulement identifier les faiblesses ou les difficultés de l'apprenant; il doit pouvoir les caractériser, détecter leur cause et surtout il doit pouvoir s'auto évaluer. Notre argumentaire sur la stimulation de la pensée réflexive comme exploitation pédagogique du DC a été approuvé dans le sens qu'elle favoriserait effectivement la présence de ces propriétés. Telles sont les principales remarques positives qui ressortent de l'évaluation conceptuelle qui a été menée dans notre travail.

Sur l'aspect informatique des STI dans le domaine AIED, plus particulièrement, leur implémentation, notre défi a été de traduire la perspective pédagogique du DC préconisée dans cette thèse en des termes formels. Cette formalisation est une étape incontournable vers la mise en œuvre de la perspective pédagogique du DC dans un programme informatique, notamment l'ensemble des programmes représentant le module de DC d'un STI. L'originalité de notre contribution dans ce sens a été développée à travers trois niveaux de formulation de cette perspective pédagogique du DC, chacun de ces niveaux étant utile à une étape particulière de la mise en œuvre du DC dans un STI. Il s'agit des niveaux de: (1) conception du DC pour un STI ; (2) implémentation du DC pour un STI et (3) intégration du DC dans un STI.

1. **Au niveau de la conception du DC dans un STI:** l'originalité de notre contribution tient de l'explicitation de la perspective pédagogique du DC à travers un cadre de spécification du DC, potentiellement exploitable par un concepteur pédagogique et un programmeur. Cette exploitation potentielle a été illustrée en définissant l'architecture de CD-SPECIES, un outil auteur - ou un outil conseil – dont la base de connaissances correspond au CSDC et dont les fonctionnalités permettent essentiellement :
 - ⇒ À un concepteur pédagogique de spécifier les caractéristiques conceptuelles d'une instance de DC qu'il désire voir implémentée dans un STI, tout en maintenant une cohérence conceptuelle entre les valeurs de ces caractéristiques.
 - ⇒ Au programmeur : de savoir quels mécanismes de représentation des connaissances au niveau du module du DC et des autres modules d'un STI sont les mieux appropriés pour implémenter effectivement toute instance de DC spécifiée par un concepteur.
 - Dans cette optique, en plus de ces fonctionnalités, l'originalité de CD-SPECIES représente une philosophie où le DC dans un STI a plus de chances de retrouver l'importance pédagogique qui est sienne dans le tutorat humain. En effet, la spécification à part entière de cette fonction favorise une articulation ciblée du processus (ou de l'algorithme) correspondant, en fonction des objectifs pédagogiques qu'on désire atteindre.

2. **Au niveau de l'implémentation du DC dans un STI** (1) formalisation de la boucle diagnostic-remédiation. *L'originalité de notre contribution tient de la formalisation de la dynamique d'une boucle diagnostic-remédiation (une composante de la perspective du DC) à travers un algorithme générique de DC basé sur une approche de raisonnement bayésien.* L'approche bayésienne a joué un rôle important dans la mesure où elle a permis d'exprimer de manière systématique et surtout générique:
- ⇒ Les facteurs d'incertitude intrinsèques aux principales situations de DC dans un STI. Chaque situation étant définie par un type de DC (d'après le cadre de spécification) dont il est question. Les relations entre le module de DC et les autres modules d'un STI ont été reformulées pour tenir compte de ces facteurs d'incertitude.
 - ⇒ La dynamique de la boucle diagnostic-remédiation en tenant compte des facteurs d'incertitude. Cette dynamique est basée sur deux méthodes formelles d'inférence ayant un support théorique fondé: (1) la mise à jour des probabilités bayésienne qui peut être assimilée à l'inférence par déduction (*belief updating*) et (2) la recherche de l'explication la plus probable, étant donnée une observation de comportement de l'apprenant, qui peut être assimilée à l'inférence par abduction (*belief revision ou most probable explanation*)
3. **Au niveau de l'implémentation du DC dans un STI** (2): formalisation de la pensée réflexive selon Dewey en tant que méthode de remédiation lors du DC. *Un autre aspect de l'originalité de notre contribution est la formalisation d'une approche particulière de remédiation dans le cadre de la boucle diagnostic-remédiation.* Il s'agit de la stimulation de la pensée réflexive – telle que définie par Dewey – à travers un dialogue tutoriel pour le DC. Cette contribution est importante dans la mesure où cette formalisation avait pour but de refléter les principales composantes de la pensée réflexive selon Dewey, favorisant ainsi l'occurrence de la pensée réflexive lors d'un dialogue tutoriel avec l'apprenant dans un STI, tout en raffinant ou en confirmant le DC posé à travers ce dialogue.
4. **Au niveau de l'intégration du module de DC dans un STI**: une librairie de programmes réutilisables pour le DC. L'algorithme générique de DC et la formalisation de la stimulation de la pensée réflexive ont été implémentés sous forme d'une librairie de programmes génériques. *L'originalité de notre contribution dans ce sens découle d'une*

part du fait que ces programmes reflètent la perspective du DC préconisée dans cette thèse et d'autre part du fait qu'ils sont réutilisables dans un STI où:

- ⇒ Une séparation claire entre le modèle des connaissances du domaine, le modèle de l'apprenant et le module pédagogique est effectuée ;
- ⇒ Les objectifs d'apprentissage sont exprimés en termes de capacités visées
- ⇒ Les capacités visées sont modélisées à travers un réseau bayésien
- ⇒ Les exercices sont réalisés à travers un dialogue tutoriel. Cette dernière condition est souhaitable mais non indispensable pour la réutilisabilité des programmes. En effet, la formalisation de l'approche de remédiation utilisée dans ces programmes (la stimulation de la pensée réflexive) est intrinsèquement reliée aux dialogues tutoriels.

La deuxième évaluation de ce travail a confirmé notre hypothèse selon laquelle la pensée réflexive intégrée au DC dans un STI favorise un apprentissage fondé. En effet, malgré l'expertise des participants à cette évaluation, il s'est avéré que les actions de Prolog-Tutor ayant pour but de provoquer la pensée réflexive les ont forcés à articuler des capacités dont ils n'étaient pas conscients ou qui leur semblaient acquises alors qu'elles ne l'étaient pas. Cette mise en oeuvre a donc contribué à analyser la portée d'un stimulant d'un apprentissage effectif dans un contexte de Programmation Logique.

Fondements cognitifs de la contribution

La conception de l'instruction dans les STI a un caractère scientifique du fait de l'émulation du tuteur humain par un système tutoriel. Ce caractère est assuré par une modélisation formelle et systématique des fonctions pédagogiques qui composent l'instruction, modélisation qui s'appuie sur des paradigmes de cognition, ainsi que sur les théories correspondantes d'apprentissage et d'instruction. Le DC est une de ces fonctions pédagogiques et nous considérons que sa mise en oeuvre dans un STI est fortement influencée par le paradigme de cognition considéré. La pensée réflexive quant à elle a souvent été avancée comme un moyen sûr de favoriser un apprentissage fondé et donc effectif. Tout au long de cette thèse, nous avons tenté de montrer que ces deux paramètres influencent le diagnostic cognitif dans les STI (représentation des connaissances, algorithme) et nous avons défini la nature de cette influence.

Du point de vue des sciences cognitives, notre contribution s'est basée sur l'analyse des relations conceptuelles qui existent :

- d'une part entre le DC du comportement de l'apprenant dans un STI et le paradigme de cognition qui caractérise les prémisses de ce STI
- d'autre part entre ce DC et la stimulation de la pensée réflexive en tant qu'exploitation pédagogique des résultats de ce DC (ou remédiation pédagogique associée à ce DC).

Cette analyse s'est matérialisée par un examen marqué des principaux paradigmes de cognition en psychologie éducationnelle, ainsi que des principales théories de la pensée réflexive (Chapitre I).

Leçons tirées et pistes de recherche

Deux aspects des contributions de cette thèse ont été évalués. Au niveau de la conception du DC dans un STI, l'on s'est interrogé en général sur la pertinence d'un CSDC et en particulier sur la pertinence des relations paradigmes de cognition-DC et de la relation stimulation de la pensée réflexive-DC à travers une boucle diagnostic-remédiation. Au niveau de l'implémentation du DC dans un STI, l'on a observé si l'implémentation du modèle formel de stimulation de la pensée réflexive dans une boucle diagnostic-remédiation, en favorise effectivement l'occurrence chez l'apprenant. Les résultats de ces évaluations ont permis de dégager des pistes de recherche.

Leçons tirées et pistes de recherche sur le CSDC

Le CSDC proposé dans cette thèse peut être considéré comme une idée fondatrice sur le sujet qu'il aborde. Dans ce sens, de nombreuses améliorations y demeurent possibles, tel que suggéré dans la première évaluation que nous avons effectuée.

Principalement nous avons appris d'une part que le DC ne saurait être limité à une seule théorie de la cognition. Le processus d'apprentissage est complexe, variable et continu. Les approches peuvent changer selon les préférences des apprenants, leur vitesse d'apprentissage, mais surtout selon les objectifs d'apprentissage. Plus qu'un paradigme de cognition, le niveau d'habileté correspondant à un objectif d'apprentissage visé dans un STI en influence – en détermine presque – la forme de DC qui y sera effectué. Sans exclure l'importance des paradigmes de cognition, il est tout aussi important de considérer ces

paramètres. Ces constats donnent lieu à deux pistes de recherche qui peuvent être exprimées à travers les questions suivantes:

- (1) Outre les paradigmes de cognition et la stimulation de la pensée réflexive, quelle est la relation du DC avec les facteurs suivants du contexte d'apprentissage dans un STI: les émotions, les préférences, l'écart cognitif à combler dans une formation, la bagage culturel, etc.?
- (2) En lieu de considérer uniquement la relation entre le DC et les paradigmes de cognition, quelle serait la substance (conceptuelle et en termes de représentation des connaissances et d'algorithme de DC) de la relation triangulaire : Paradigme de cognition, Objectif d'apprentissage, DC dans un STI ?

D'autre part, nous avons appris que pour être effectif, CD-SPECIES doit être doté d'une interface concise qui n'impose pas au concepteur pédagogique d'effectuer une consultation exhaustive avant de trouver ce qu'il recherche. Plus important, CD-SPECIES doit favoriser une communication soutenue entre le concepteur d'un STI et le programmeur: non seulement pour établir un langage commun, mais surtout pour permettre au concepteur de spécifier le contenu qui supportera une instance de DC spécifié, permettant ainsi au programmeur d'avoir une perspective plus concrète de ce qui est attendu de lui. Ces constats donnent lieu à une piste de recherche:

- Quelles fonctionnalités faudrait il ajouter à CD-PECIES pour favoriser une plateforme de communication entre un concepteur de STI et un programmeur de STI afin de favoriser la réalisation des deux buts suivants: atteindre progressivement un langage et une compréhension commune des termes informatiques relatifs à l'implémentation du module de DC dans un STI et des termes de psychologie de l'apprentissage relatif au DC; spécifier et s'informer sur le contenu qui concrétisera une spécification formelle et abstraite d'une instance de DC.

Leçons tirées et pistes de recherche sur l'implémentation du DC dans Prolog-Tutor

Prolog-Tutor a permis de matérialiser une instance de la boucle diagnostic-remédiation à travers la relation entre le DC et la stimulation de la pensée réflexive. Les résultats de l'évaluation de ce système sont prometteurs mais dans le même temps, ils ont révélé quelques nuances relatives à l'application de ce processus. Premièrement, la pensée réflexive telle que défini par Dewey ne se manifeste pas toujours selon ce schéma prédéfini. Cette manifestation

est tributaire du sujet qui réfléchit et surtout de son niveau de maîtrise par rapport à la capacité impliquée dans sa réflexion. Deuxièmement, quand bien même des faits tangibles correspondent à des patrons de pensée réflexive, ils n'en témoignent pas pour autant de sa présence effective. Pour que sa réflexion lui soit utile, le sujet qui l'effectue doit en être conscient et surtout, il doit comprendre son but et le gain qu'il en tire. Troisièmement, la pensée réflexive n'apparaît pas seulement lorsqu'elle est explicitement stimulée. La nature simplement interrogative d'une interaction pédagogique peut en provoquer des manifestations. Enfin, la stimulation de la pensée réflexive dépend fortement de la complexité de la capacité sur laquelle elle porte, de même que de la complexité de la situation d'apprentissage dans laquelle elle se déroule. Ces constats ont fait l'objet des principales nuances apportées relativement au succès de la stimulation de la pensée réflexive dans Prolog-Tutor. Ces nuances introduisent les questions de recherche suivantes:

- (1) Comment pourrait-on améliorer les dialogues tutoriels de Prolog-Tutor de manière à favoriser chez les apprenants une prise de conscience précoce du processus qu'on essaie de stimuler chez lui et surtout de la pertinence de ce processus?
- (2) Comment devrait se déployer la stimulation de la pensée réflexive dans des tâches plus complexes dans Prolog-Tutor? Dans des tâches encore plus complexes dans d'autres domaines?

Ces interrogations constituent l'essentiel des pistes que cette thèse a contribué à dégager, et qui seront examinées dans une recherche future.

APPENDICE. A

ILLUSTRATIONS DE CD-SPECIES

Cet appendice présente des illustrations sur l'architecture de CD-SPECIES (Chapitre III).

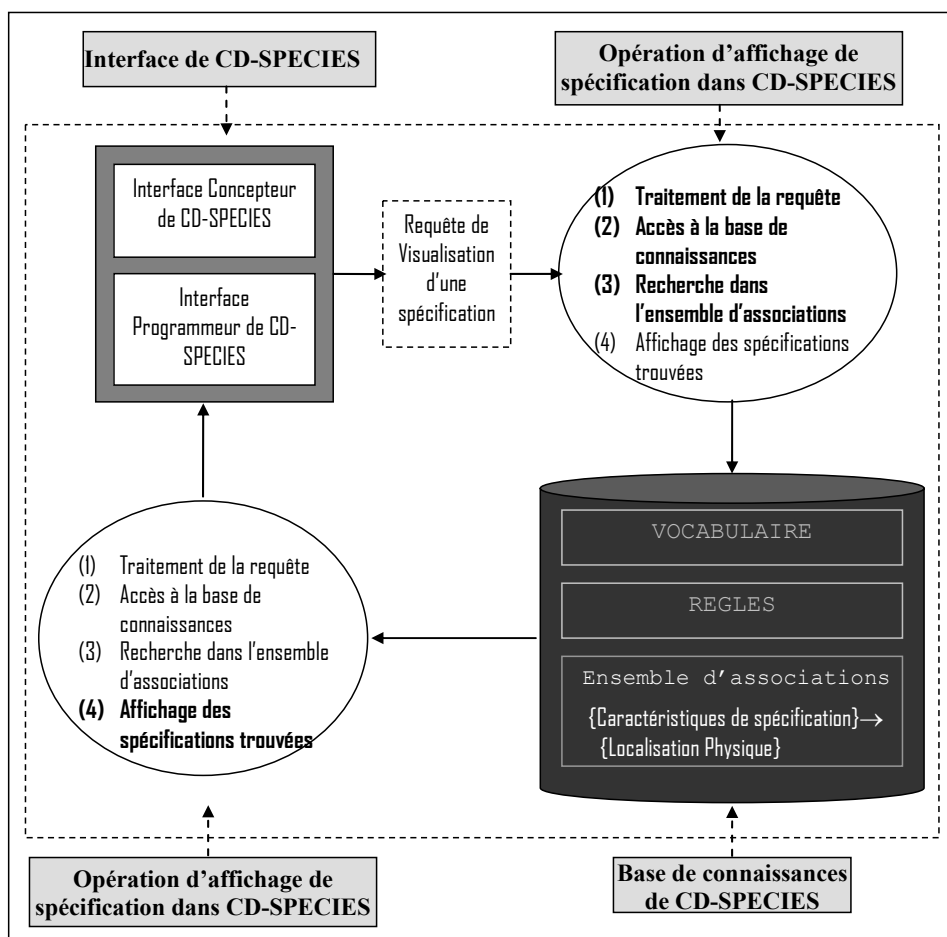


Figure A. 1. Dynamique de CD-SPECIES pour la visualisation d'une spécification

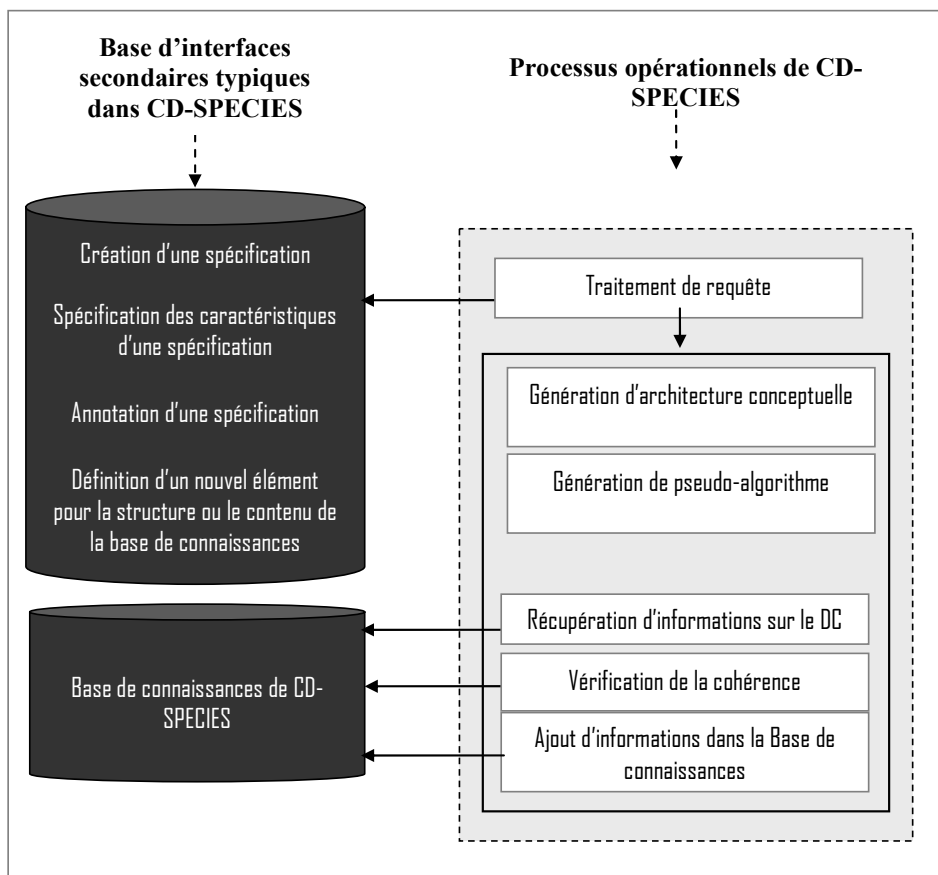


Figure A. 2. Processus opérationnels de CD-SPECIES et données associées

APPENDICE. B

ILLUSTRATION DE LA FORMALISATION DE LA BOUCLE DIAGNOSTIC-
REMÉDIATION

Tableau B.1. Approche bayésienne générique: attributs et fonctions de l'objet «modèle du domaine»

<DomainModel-Object-for-Bayesian-CD>	Description
<Attributes> <ul style="list-style-type: none"> ▪ <BayesNet/> 	Réseau Bayésien exprimant les relations causales entre les éléments représentés dans le modèle du domaine
</Attributes>	
<Methods>	
<ul style="list-style-type: none"> ▪ <getElement/> 	Récupérer un élément du réseau
<ul style="list-style-type: none"> ▪ <getParent(Element e)/> 	Récupérer les parents (les causes) d'un élément du réseau
<ul style="list-style-type: none"> ▪ <setObservedNodes(Observation o, Element e) /> 	Indiquer qu'une Observation a été effectué au niveau d'un élément e
APPENDICE B<beliefUpdate(); beliefRevise()/>	Commander les algorithmes de mise à jour et de recherche de l'explication la plus probable sur le réseau Bayésien
<ul style="list-style-type: none"> ▪ <getDistribution(Element e)/> 	Extraire la probabilité a posteriori d'un élément (après la mise à jour bayésienne)
<ul style="list-style-type: none"> ▪ <getExplanatoryValue(Element e)/> 	Extraire la valeur explicative d'un élément (après la recherche de l'explication la plus probable)
</Methods>	
</DomainModel-Object-for-Bayesian-CD>	

Tableau B.2. Approche bayésienne générique: attributs et fonctions de l'objet «modèle de l'apprenant»

<LearnerModel-Object-for-Bayesian-CD>	Description
<Attributes> <ul style="list-style-type: none"> ▪ <cognitiveState/> ▪ <myDomainModel/> 	Structure représentant l'état cognitif de l'apprenant. Modèle du domaine sur lequel se base la structure cognitiveState
</Attributes>	
<Methods>	
<ul style="list-style-type: none"> ▪ <updateModel(Observation o)> <command_list> <pre> myDomainModel.BayesNet.beliefUpdate(o); For each element in CognitiveState probability(element) = meanan(Prob(element), myDomainModel.getDistribution(element) </pre> </command_list> </updateModel(Observation o)> 	Commander la mise à jour de la représentation de l'état cognitif de l'apprenant, étant donné une observation de son comportement lors d'un exercice
</Methods>	
</LearnerModel-Object-for-Bayesian-CD>	

Tableau B.3. Approche bayésienne générique: attributs et fonctions de l'objet «modèle causal»

<CausalModel-Object-for-Bayesian-CD>	Description
<Attributes> <ul style="list-style-type: none"> ▪ causalStructure 	Structure représentant les relations entre le comportement de l'apprenant (observations) et ses possibles causes (elements du modèle du domaine)
</Attributes>	
<Methods> <ul style="list-style-type: none"> ▪ getCauses(Observation o) 	Extraire à partir de l'objet causalStructure l'ensemble des causes probables d'une observation
</Methods>	
</CausalModel-Object-for-Bayesian-CD>	

**Tableau B.4. Approche bayésienne générique: attributs et fonctions de l'objet
«diagnostic cognitif»**

<Bayesian-CognitiveDiagnosis-Object> <Attributes>	Description
<ul style="list-style-type: none"> ▪ <studentAnswer/> ▪ <curentEvaluation/> ▪ <domainModel/> ▪ <learnerModel/> ▪ <causalModel/> ▪ <curentCauses/> ▪ <mostProbableCauses/> ▪ <causeAspects/> ▪ <curentDiagnosisMode/> ▪ <Treshold/> ▪ <Reflection_level/> ▪ <Reflection_level_subjects/> 	<p>Réponse de l'apprenant actuellement diagnostiquée et évaluation (caractérisation) correspondante</p> <p>Modèles du domaine, de l'apprenant et modèle causal associés à l'exercice au cours duquel le comportement de l'apprenant est diagnostiqué</p> <p>Mode de diagnostic : Permet d'indiquer si la remédiation (la pensée réflexive dans ce cas) doit être effectuée quand bien même le comportement de l'apprenant a un caractère correct (par exemple, pour s'assurer qu'il comprend bien, au lieu de se contenter de sa bonne réponse).</p> <p>Valeur de probabilité d'une cause potentielle (explication) du comportement de l'apprenant au delà de laquelle elle est exclue comme cause plausible.</p> <p>Niveau d'élaboration de la réflexion : niveau 1, seules les causes diagnostiquées feront l'objet de réflexion;</p> <p>Niveau 2 : les éléments du domaines dont les causes diagnostiquées sont dépendantes feront l'objet de réflexion;</p> <p>Niveau 3 : Idem. Pour les éléments qui ont gait l'objet de réflexion au Niveau 2, etc.</p>
</Attributes>	
<Methods>	
<ul style="list-style-type: none"> ▪ bayesian_generic_cognitive_diagnosis ▪ <evaluateAnswer/> ▪ <generatesProbablesCauses/> ▪ <identifyMPEs/> ▪ <generatePlausibleCauses/> ▪ <loop_diagnosis_reflexion/> 	<p>Algorithme generique de DC base sur l'approche bayésienne et sur la perspective de DC preconisee dans le CSDC</p> <p>Caractériser la réponse de l'apprenant (partielle ou finale) lors d'un exercice</p> <p>Générer les causes potentielles d'une réponse</p> <p>Identifier et générer les causes les plus probables d'une réponse</p> <p>Stimuler la réflexion de l'apprenant sur les causes diagnostiquer;</p> <p>Confirmer leur diagnostic en conséquence du résultat de cette réflexion</p>
</Methods>	
</ Bayesian-CognitiveDiagnosis-Object>	

APPENDICE. C

ILLUSTRATION DE L'IMPLÉMENTATION DE PROLOG-TUTOR

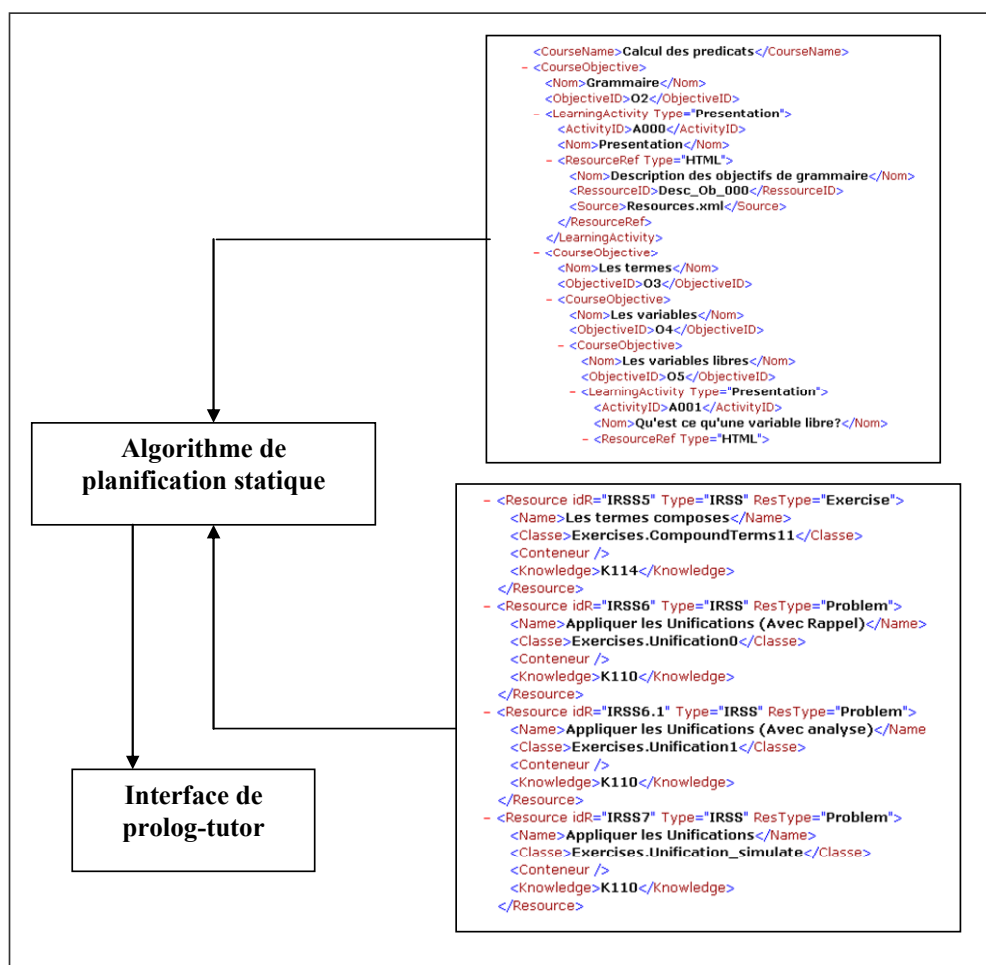


Figure C. 1. Prolog-Tutor: Représentation des objectifs et des ressources

PROLOG-TUTOR

Cours

Demo : Predicate Calculus for Logic Programming

Planification Dialogue Modele Apprenant/Diagnostics

Prolog-Tutor Resol

Explication de l'intuition-fait

En cliquant sur ce bouton, vous devez obtenir une explication theorique et generale relative a l'element de connaissances correspondant. Par exemple si cet element est un "FAIT" en Prolog, l'explication suivante app:

Un FAIT exprime une proposition, une verite dans un monde;
Prolog permet d'exprimer en langage Informatique l'etat d'un monde sur lequel raisonner

Tache a completer: Definir les explications pour tous les elements du domaine

requete?

CONSTRUISEZ cette preuve en:

--> CONSIDERANT la Base de Connaissances specifiee dans la fenetre "BASE DE CONNAISSANCEE" (a droite) et en

--> REpondANT aux questions du systeme par le biais des BOUTONS du panneau "Reponses/Questions. Usager"

cause(obstruction_veine_hepatique,jaunisse).
cause(metastases,hepatiteB).
cause(vaumissement,diarree).
symptome(pierre, fatigue).
symptome(pierre, fièvre).
symptome(marco, calculs).
symptome(georges, rhume).
symptome(georges,X).
est_atteint_de(marco, meningite).
est_atteint_de(charles,jaunisse).
est_atteint_de(pierre,malaria).

Capacites visees dans PROLOG-Tut...

- Capacites visees dans Prolog Tutor
 - Variable
 - Atome
 - Terme Simple
 - IntuitionTC
 - Terme Compose
 - Terme
 - Comprend Intuition-fait
 - Fait
 - Comprend IntuitionFait
 - Terme
 - IntuitionUVar
 - Unification avec une Variable
 - IntuitionUAt
 - Unification avec un Atome
 - IntuitionUTC
 - Unification avec un Terme Compose
 - IntuitionRegle
 - Regle
 - Base de Faits
 - Intuition de la resolution
 - Resolution en utilisant un Fait
 - Resolution en utilisant une Regle
 - IntuitionRetourArriereBF
 - Retour Arriere dans une Base de Faits
 - Retour Arriere par rapport aux sous buts d'une regle

Outils

Constructeur

Situation Dialogue/Arbre Resolution

Terminer Session

[Tuteur] Quel est le but qu'on desire prouver

[Learner] Je ne sais pas

[Feedback] C'est est atteint_de(pierre,M) Recevoir Explication

[Tuteur] Quel est le FAIT qu'on veut prouver avec cette BASE de CONNAISSANCES?

[Learner] Reponse: est_atteint_de(pierre,M)

[Feedback] Parfait.

[Tuteur] Que signifie cette requete?

Repondre

Reponses/Questions. Usager

Voir Domaine

Sais Pas

Resultat =

Reponse =

Pourquoi? (Situation Dialogue/Question)

Quoi?

Comment?

Figure C. 2. Visualisation des capacités nécessaires pour répondre à une question

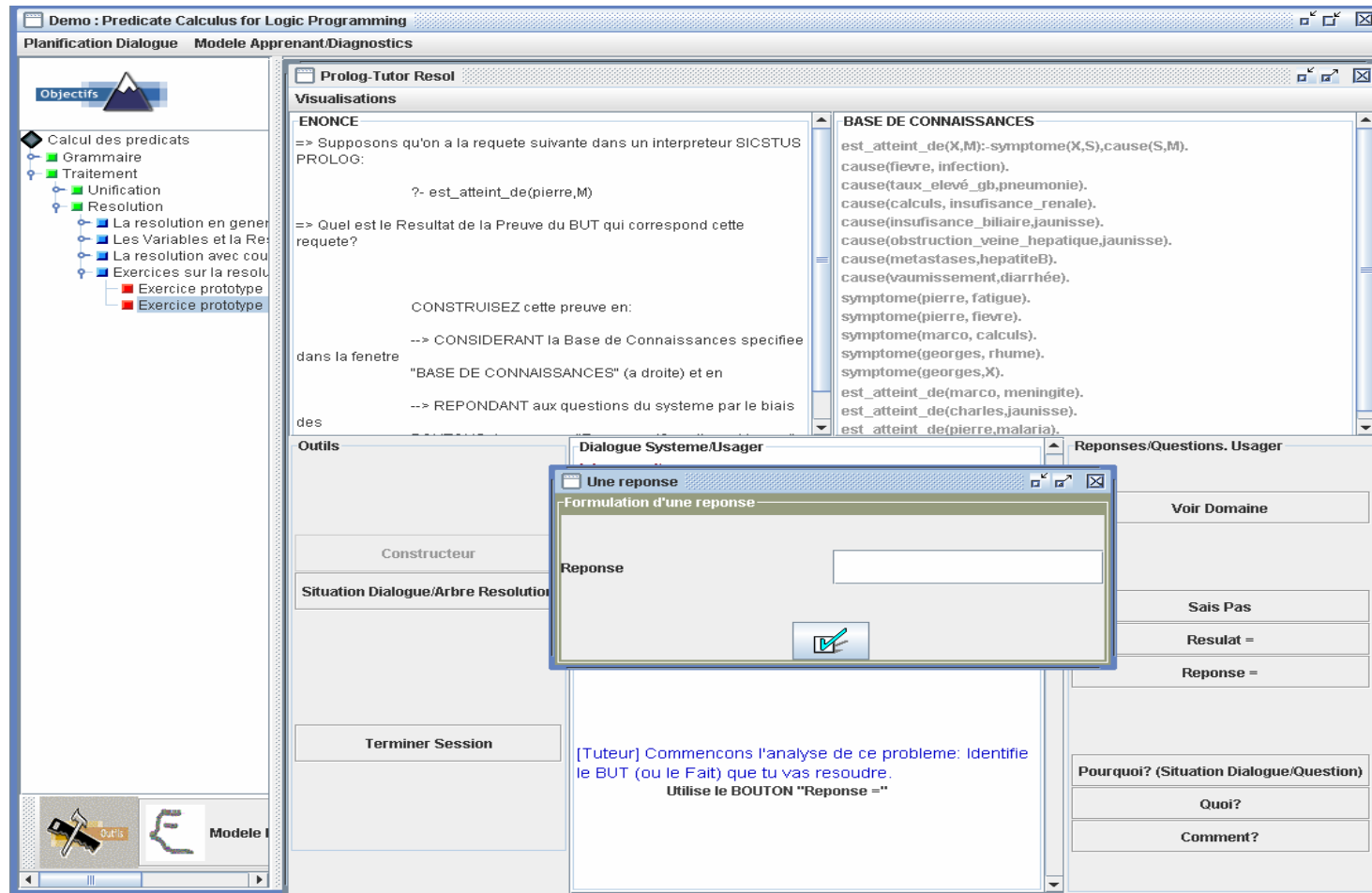


Figure C. 3. Réponse directe de l'apprenant à une question du module pédagogique

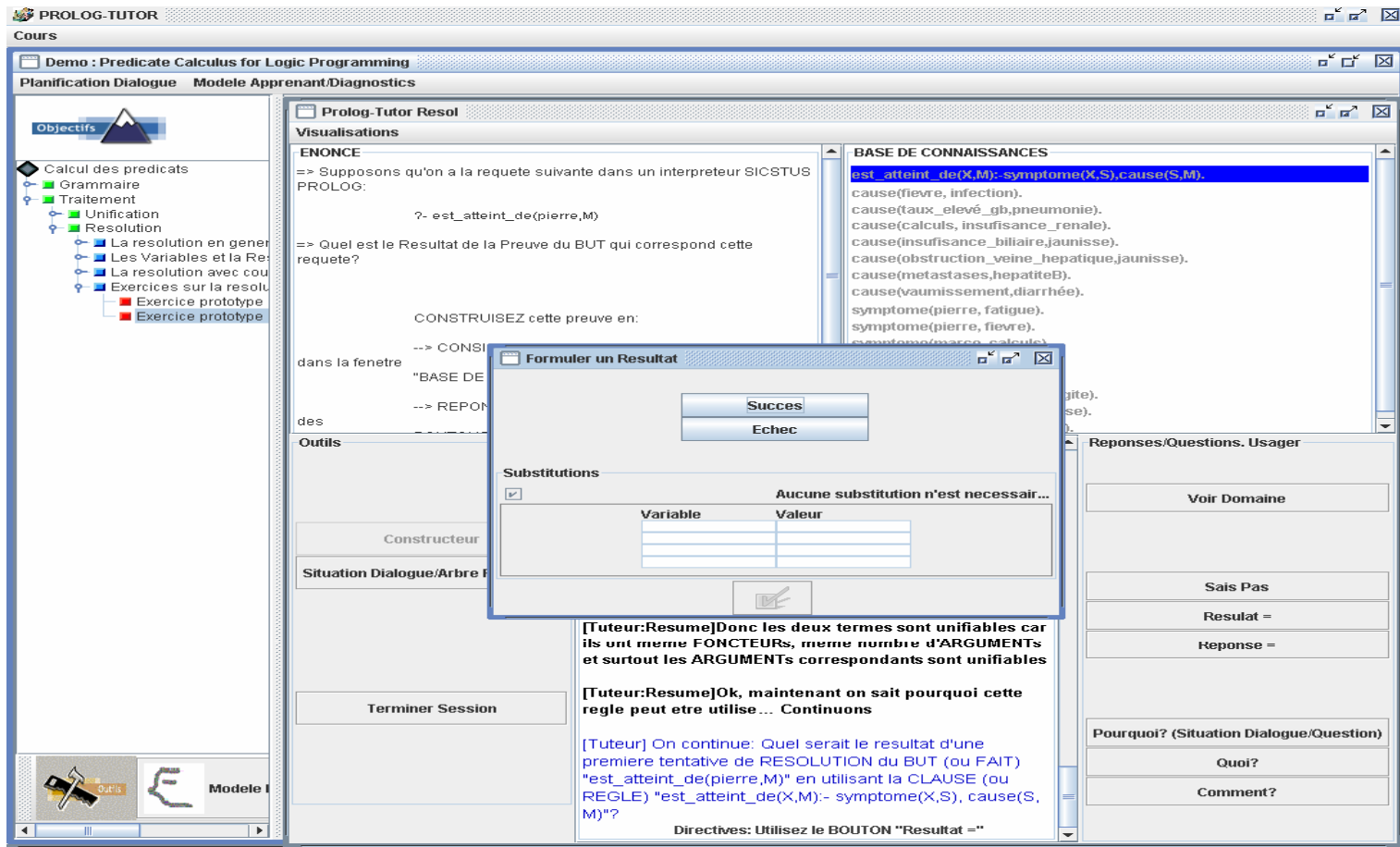


Figure C. 4. Interface de formulation des résultats d'une unification

PROLOG-TUTOR

Cours

Demo : Predicate Calculus for Logic Programming

Planification Dialogue Modele Apprenant/Diagnostics

Objectifs

- Calcul des predicats
 - Grammaire
 - Traitement
 - Unification
 - Resolution
 - La resolution en gener
 - Les Variables et la Re
 - La resolution avec cou
 - Exercices sur la resolu
 - Exercice prototype
 - Exercice prototype

Etat de la resolution jusque dans l'etape actuelle

```

est_atteint_de(X,M):-symptome(X,S),cause(S,M).
cause(fievre, infection).
cause(taux_eleve_gb, pneumonie).
cause(calculs, insuffisance_renale).
cause(insuffisance_biliaire, jaunisse).
cause(obstruction_veine_hepatique, jaunisse).
cause(metastases, hepatiteB).
cause(vaurissement, diarrhee).

symptome(pierre, fatigue).
symptome(pierre, fievre).
symptome(marco, calculs).
symptome(georges, rhume).
symptome(georges, X).

atteint_de(marco, meningite).
atteint_de(charles, jaunisse).
atteint_de(pierre, malaria).

```

atteint_de(Pierre,M) :- symptome(pierre,S), cause(S,M)

atteint_de(pierre,malaria)

symptome(pierre,S) ?

M=malaria

Constructeur

Situation Dialogue/Arbre Resolution

Terminer Session

[Feedback] Dans l'UNIFICATION DE : "est_atteint_de(pierre,M)" AVEC "est_atteint_de(X,M)": L'ARGUMENT "pierre" peut etre UNIFIE a l'ARGUMENT "X" (qui devient une VARIABLE LIEE) et, L'ARGUMENT "M" peut etre unifie a l'ARGUMENT "M" (qui demeure des VARIABLES LIBRES) puisque ce sont deux variables

[Tuteur:Resume]Donc les deux termes sont unifiables car ils ont meme FONCTEURS, meme nombre d'ARGUMENTS et surtout les ARGUMENTS correspondants sont unifiables

[Tuteur:Resume]Ok, maintenant on sait pourquoi cette regle peut etre utilise... Continuons

[Tuteur] On continue: Quel serait le resultat d'une premiere tentative de RESOLUTION du BUT (ou FAIT) "est_atteint_de(pierre,M)" en utilisant la CLAUSE (ou REGLE) "est_atteint_de(X,M):- symptome(X,S), cause(S,M)"?

Directives: Utilisez le BOUTON "Resultat ="

Reponses/Questions. Usager

Voir Domaine

Sais Pas

Resultat =

Reponse =

Pourquoi? (Situation Dialogue/Question)

Quoi?

Comment?

Figure C. 5. Prolog-Tutor : Representation graphique de la résolution d'un but

Demo : Predicate Calculus for Logic Programming
Planification Dialogue Modele Apprenant/Diagnostics

Objectifs

- Calcul des predicats
 - Grammaire
 - Traitement
 - Unification
 - Resolution
 - La resolution en gener
 - Les Variables et la Re
 - La resolution avec cou
 - Exercices sur la resolu
 - Exercice prototype
 - Exercice prototype

Prolog-Tutor Resol

Visualisations

ENONCE
=> Supposons qu'on a la requete suivante dans un interpreteur SICSTUS
PROLOG:

?- est_atteint_de(pierre,M)

=> Quel est le Resultat de la Preuve du BUT qui correspond cette
requete?

CONSTRUISEZ cette preuve en:

--> CONSIDERANT la Base de Connaissances specifiee
dans la fenetre
"BASE DE CONNAISSANCES" (a droite) et en

--> REPONDANT aux questions du systeme par le biais
des

BASE DE CONNAISSANCES

```
est_atteint_de(X,M):-symptome(X,S),cause(S,M),
cause(fievre, infection).
cause(taux_eleve_gb,pneumonie).
cause(calculs, insuffisance_renale).
cause(insuffisance_biliaire,jaunisse).
cause(obstruction_veine_hepatique,jaunisse).
cause(metastases,hepatiteB).
cause(vaumissement,diarree).
symptome(pierre, fatigue).
symptome(pierre, fievre).
symptome(marco, calculs).
symptome(georges, rhume).
symptome(georges,X).
est_atteint_de(marco, meningite).
est_atteint_de(charles,jaunisse).
est_atteint_de(pierre,malaria).
```

Outils

vrai, d'accord?
Clique sur le BOUTON selon ton avis
D'accord En desaccord

[Learner] Non, je ne suis pas d'accord avec
cette affirmation

**[Feedback] Ok, examinons
"est_atteint_de(pierre,M)"**

[Tuteur] [Justifie ta position] :Dans la requete
PROLOG suivante: ?-est_atteint_de(pierre,M), quelle
est la VARIABLE?
Clique sur le BOUTON "Reponse ="

[Learner] Reponse: M

[Feedback]OK.

[Tuteur] Que signifie cette
requete?
Rep

Reponses/Questions. Usager

Voir Domaine

Sais Pas

Resulat =

Reponse =

Pourquoi? (Situation Dialogue/Question)

Quoi?

Prouver que le FAIT "est_atteint_de(pierre,M)" est VRAI pour au moins une valeur de M
Prouver qu'il Existe au moins une valeur de M Telle que "est_atteint_de(pierre,M)" est VRAI
Trouver les valeurs de M telles que le FAIT "est_atteint_de(pierre, M)" est VRAI

Figure C. 6. Interface secondaire pour les questions à choix multiples

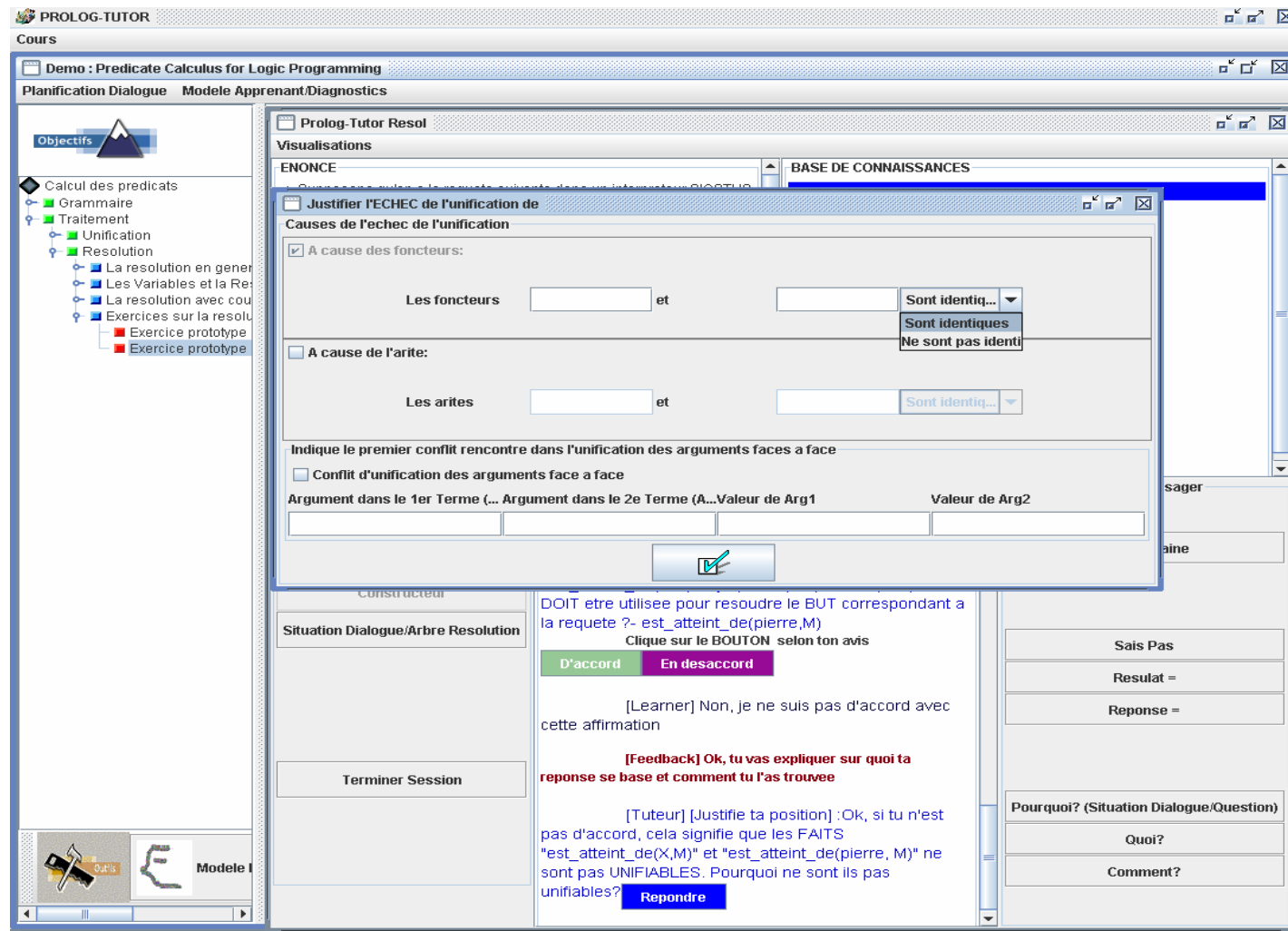


Figure C. 7. Interface secondaire pour prouver l'Échec d'une unification

The screenshot displays the PROLOG-TUTOR application interface. The main window is titled "Cours" and contains a "Planification Dialogue" and "Modele Apprenant/Diagnostics" section. A sidebar on the left lists "Objectifs" such as "Calcul des predicats", "Grammaire", "Traitement", "Unification", and "Resolution".

The central area shows a "Prolog-Tutor Resol" window with the following content:

Visualisations

ENONCE

=> Supposons qu'on a la requete suivante dans un interpreteur SICSTUS
 PROLOG:

```
?- est_atteint_de(pierre,M)
```

=> Quel est le Resultat de la Preuve du BUT qui correspond cette requete?

BASE DE CONNAISSANCES

```
est_atteint_de(X,M): symptome(X,S),cause(S,M).
cause(fievre, infection).
cause(taux_eleve_gb,pneumonie).
cause(calculs, insufisance_renale).
cause(insufisance_biliaire,jaunisse).
cause(obstruction_veine_hepatique,jaunisse).
cause(metastases,hepatiteB).
```

A secondary window titled "Resolution d'un sous-but" is overlaid on the main interface. It contains the text "Specifie les termes qui vont etre unifies" and two input fields:

Indique le sous-but a resoudre:

Indique l'element correspon...:

At the bottom of this window, there is a checkbox: [Seulement pour les SOUS-BUTS] Aucun Terme de la Base de Connaissances ne peut etre unifie avec ce sous-but, followed by an "OK" button.

Below the secondary window, a "Recevoir Explication" button is visible. A text box contains the following explanation:

[Tuteur] Donc si la preuve du BUT "est_atteint_de(pierre,M)" reussit tel que tu le soutiens, les preuves des FAITS "symptome(pierre,S)" ET "cause(S,M)" reussissent. Examinons cette preuve: Que donne la preuve du premier SOUS-BUT associe au BUT "est_atteint_de(pierre,M)" lorsqu'on utilise cette REGLE?

Below the text box is a "Repondre" button.

On the right side of the interface, there is a "Questions. Usager" section with buttons for "Voir Domaine", "Sais Pas", "Resultat =", and "Reponse =". Below this, there is a "Pourquoi? (Situation Dialogue/Question)" section with buttons for "Quoi?" and "Comment?".

At the bottom left, there is a "Terminer Session" button and a "Modele" icon.

Figure C. 8. Interface secondaire pour la résolution d'un sous-but

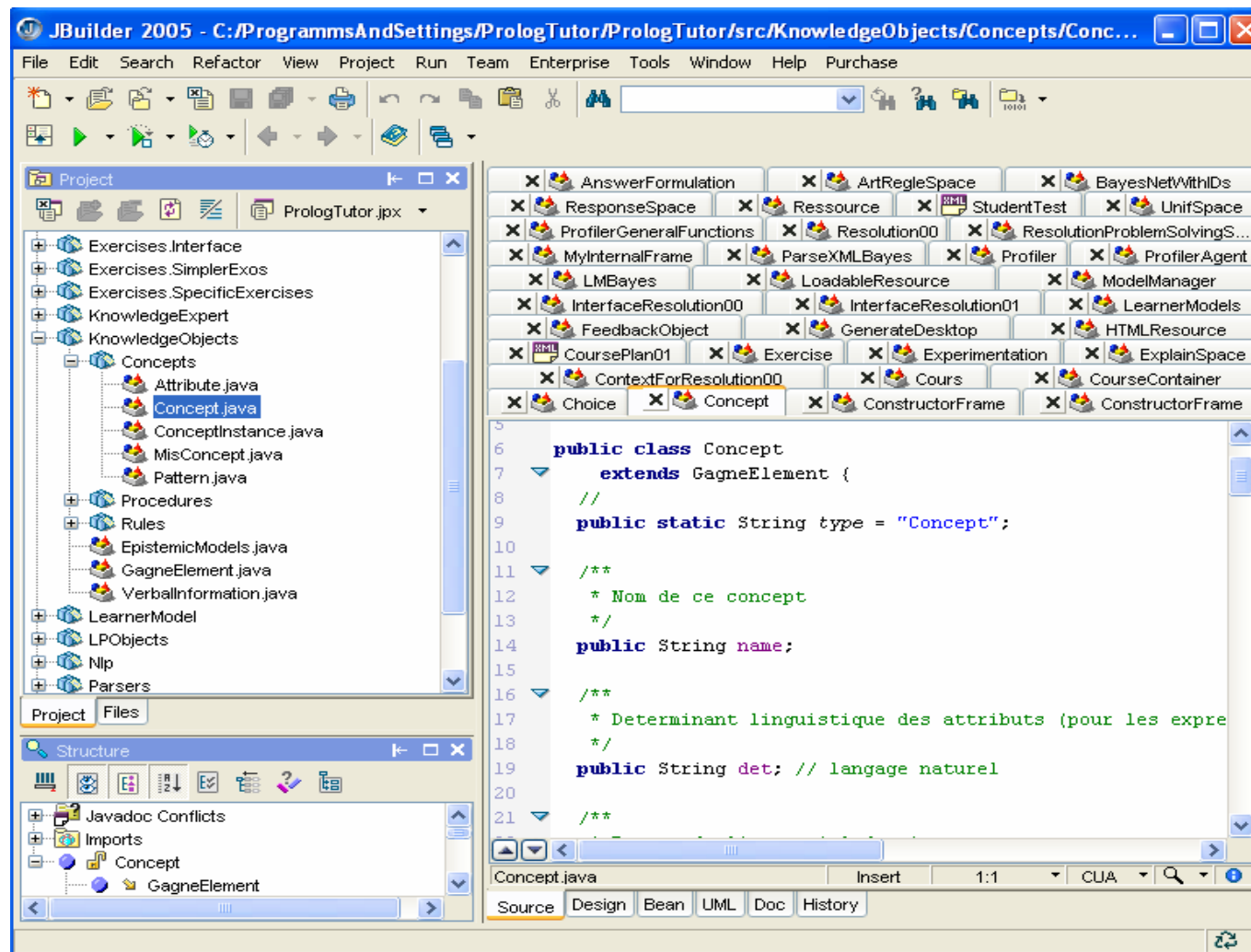


Figure C. 9. Classes définissant les patrons de questions selon le type de capacité dans la taxonomie de Gagné

Resolution00_sub_skills0.3				
	A	B	C	E
1	Reasoning	SkillName	Question	ExpliQ Answer
2	MainQuest	Executer;Resolution	null	null RES_:M=infection, M=malaria
3	.0	Comprendre;IntuitionFait	Quel est le but qu'on désire prouver	As tu com est_atteint_de(pierre,M)
4	.1	Comprendre;Intuition_de_la_re	Indique le premier element de la BASE c	On doit pa est_atteint_de(X,M):- symptom
5	.2	Executer;Resolution_en_utilis	Quel serait le resultat d'une tentative de	La REGLE ECHEC
6	.3	Executer;Retour_Arriere_dans	Quel serait le resultat de la 2ème tentati	On a essa RES_:M=infection
7	.4	Utiliser;Base_de_Faits	Y a t il encore des solutions possibles a	On a pu et Alternatives_:NON_Ait_non
8	.5	Executer;Retour_Arriere_dans	Il y a encore une solution possible, laque	Oui, il faut RES_:M=malaria
9	.0.0	Identifier;Fait	Quel est le FAIT qu'on veut prouver avec	Avant de c est_atteint_de(pierre,M)
10	.0.1	Comprend;IntuitionFait	Que signifie cette requete? PossibleChc	On essaie Trouver les valeurs de M telles
11	.1.0	Identifier;Regle	Indique l'élément "HEAD" de la clause d	Il faut que est_atteint_de(X,M)
12	.1.1	Executer;Unification_avec_un	Quel est le resultat de l'unification de es	Il faut savo RES_:X=pierre, M=M
13	.2.0	Identifier;Regle, Comprendre;I	Quels sont les deux sous-buts qu'il faut	Le but de (List_symptome(pierre,S),caus
14	.2.1	Executer;Resolution_en_utilis	Quel est le resultat de la preuve de sym	Le but de (RES_:S=fatigue
15	.2.2	Comprendre;Intuition_de_la_re	Quel est le resultat de la prochaine oper	Il faut cont ECHEC
16	.3.0	Comprendre;Retour_Arriere_p	Quel est le dernier SOUS-BUT pour leq	On veut dé symptome(pierre,S)
17	.3.1	Utiliser;Base_de_Faits	Quel est le dernier element de la Base d	On veut dé symptome(pierre,fatigue)
18	.3.2	Executer;Resolution_en_utilis	Quel est le prochain resultat de la preuve	On veut de RES_:S=fievre
19	.3.3	Comprendre;Intuition_de_la_re	Donc quel est le resultat pour cause(fiev	On veut de RES_:M=infection

Resolution00_sub_skills0.3_ER			
	A	B	D
1	ReasoningS	Dewey	SkillName Question
2	.0	Question directe/Implicite [bo	Comprendre;IntuitionFait
3	.1	Question directe/Implicite [bo	Comprendre;Intuition_de_la_resolution
4	.2	Question directe/Implicite [bo	Executer;Resolution_en_utilisant_une_Re
5	.3	Question directe/Implicite [bo	Executer;Retour_Arriere_dans_la_resolut
6	.4	Question directe/Implicite [bo	Utiliser;Base_de_Faits
7	.5	Question directe/Implicite [bo	Executer;Retour_Arriere_dans_une_Base
8	.0.Assertion	Observation	Comprendre;IntuitionFait
9	.0.Assert.0	Observation	Identifier;Terme
10	.0.Assert.1	Inference	Comprendre; IntuitionFait
11	.0.Contradiction	Observation	Identifier;Terme
12	.0.Contradiction	Inference	Comprendre; IntuitionFait
13	.1.Assertion	Assertion	Comprendre;IntuitionRegle
14	.1.Assert.0	Observation	Identifier;Regle
15	.1.Assert.1	Reasoning	Comprendre;Resolution_en_utilisant_une
16	.1.Assert.2	Reasoning	Comprendre;Unification_avec_un_Terme

Figure C. 10. Questions prédéfinies dans une base de données

APPENDICE. D

CARACTÉRISTIQUES DES PARTICIPANTS À L'ÉVALUATION ET CODES
D'ANALYSE DES DONNÉES QUALITATIVES D'ÉVALUATION

Tableau D. 1. Caractéristiques des participants à l'évaluation du CSDC

Participant	Background en conception pédagogique
Expert 1 (E1)	Professeur en sciences cognitives et en IHM
Expert 2 (E2)	Concepteur pédagogique professionnel
Expert 3 (E3)	Chercheur en conception pédagogique

Tableau D. 2. Codes de convergence des experts sur les questions d'entrevue

Code	Description
ACC	Accord Total entre les avis
ACC_{nu}	<ul style="list-style-type: none"> ▪ Accord entre les avis, avec des nuances d'une part ou d'une autre; ▪ Accord pour des motifs différents; Positions disjointes et/ou complémentaires (aucun désaccord) ▪ Avis complémentaires
DACC_{nu}	Désaccord entre les avis avec des nuances d'une part ou d'une autre qui pourraient réconcilier les avis vers un accord
DACC	Désaccord total entre les avis

Tableau D. 3. Codes de convergence des experts les questions d'évaluation

Code	Description
(p,n)	Un nombre "p" de réponses positives, et un nombre "n" réponses négatives
m_a	Un nombre "m" de réponses positives (ou négatives) parmi lesquelles un nombre "a" de réponses sont nuancées dans le sens positif

Tableau D. 4. Expérience des participants à l'évaluation de Prolog-Tutor avec Prolog

Participants	Expérience avec Prolog	Niveau de maîtrise de Prolog
Participant 1	Étudiant de premier cycle en Informatique. A suivi un cours d'introduction à Prolog	Très bon
Participant 2	Étudiant de premier cycle en Linguistique Computationnelle. A utilisé Prolog pour ses travaux	Expert
Participant 3	Professionnel informaticien. A suivi un cours d'introduction à Prolog	Bon
Participant 4	Étudiant de troisième cycle en Informatique Cognitive. A suivi un cours d'introduction à Prolog	Expert
Participant 5	Étudiante de troisième cycle en Sciences de l'Éducation. A utilisé Prolog pour ses travaux	Très bon
Participant 6	Étudiante de premier cycle en Informatique. A suivi un cours de Prolog	Très bon

Tableau D. 5. Échelle de maîtrise de Prolog considérée

Échelle de capacité en Prolog	Description
Expert	Une personne capable d'utiliser Prolog comme outil de programmation pour une application dans un autre domaine : linguistique, système expert
Très bon	Une personne qui comprend les concepts et les principes de Prolog et qui les applique correctement
Bon	Une personne qui connaît les concepts et les principes de Prolog et qui peut les appliquer

Tableau D. 6. Codes associées aux sous dialogues et aux questions du dialogue étudié

Numéro de sous dialogue	Étapes de la pensée réflexive associées à chaque question des sous dialogues	Questions du niveau D1	Code
		Commençons l'analyse de ce problème: Identifie le but (ou le fait) que tu vas résoudre.	□□
Questions pour la pensée réflexive au niveau D1.2			
SD0	Situation	On essaie de prouver que le fait "est_atteint_de (pierre, M) " est vrai, d'accord?	□□.S
	Intellectualisation	Dans la requête PROLOG: ?- est_atteint_de (pierre, M), quelle est la variable?	□□.Intellect
	Raisonnement	Que signifie cette requête?	□□.Assert

Numéro de sous dialogue	Étapes de la pensée réflexive associées à chaque question des sous dialogues	Questions du niveau D1	Code
		On continue: Indique le premier élément de la base de connaissances qui peut être utilisé pour cette preuve	Q1
Questions pour la pensée réflexive au niveau D1.2			
SD1	Situation	La règle "est_atteint_de(X,M) :- symptome(X,S), cause(S,M)" doit être utilisée pour résoudre le but correspondant à la requête ?-est_atteint_de(pierre,M)	Q1.S
	Intellectualisation	La clause de horn "est_atteint_de(X,M) :- symptome(X,S), cause(S,M) exprime une règle. Pourquoi? Quelle est la "conséquence" qui correspond à cette règle et quelles sont les "conditions" qui y correspondent?	Q1.Intellect
	Intellectualisation	Observons la "conséquence" de la clause "est_atteint_de(X,M) :- symptome(X,S), cause(S,M)". Est elle unifiable avec le but à résoudre?	Q1.Intellect2

Numéro de sous dialogue	Étapes de la pensée réflexive associées à chaque question des sous dialogues	Questions du niveau D1	Code
	Raisonnement	Démontre comment les deux termes ont été unifiés	Q1.Assert1
SD1	Raisonnement	Ok, si tu n'est pas d'accord, cela signifie que les faits "est_atteint_de(X,M)" et "est_atteint_de(pierre, M)" ne sont pas unifiables. Pourquoi serait ce le cas?	Q1.Contradict1
	Raisonnement	Ok, les faits "est atteint de(X,M)" et "est_atteint_de(pierre, M)" ne seraient pas unifiables si: leurs foncteurs sont différents. Est ce le cas?	Q1.Contradict2
	Raisonnement	Ont ils le même nombre d'arguments?	Q1.Contradict3
	Raisonnement	Y a t-il un conflit dans l'unification des arguments vis à vis?	Q1.Contradict4
		On continue: Quel serait le résultat d'une première tentative de résolution du but "est_atteint_de(pierre,M)" en utilisant la clause "est_atteint_de(X,M) :- symptome(X,S), cause(S,M)"?	Q2
Questions pour la pensée réflexive au niveau D2			

Numéro de sous dialogue	Étapes de la pensée réflexive associées à chaque question des sous dialogues	Questions du niveau D1	Code
SD2	Situation	La première tentative de prouver "est_atteint_de(pierre,M)" avec "est_atteint_de(X,M) :-symptome(X,S), cause(S,M)" échoue	Q2.S
	Raisonnement	Comment fais tu la preuve du premier sous-but (ou encore de la première "condition" correspondant a cette règle)?	Q2.Assert0
	Raisonnement	Ensuite? Que donne la preuve du prochain sous-but?	Q2.Assert1
SD2	Intellectualisation	Justifie ta position: Que signifie cette règle?	Q2.Intellect
	Raisonnement	Donc si la preuve du but "est_atteint_de(pierre,M)" réussit tel que tu le soutiens, les preuves des faits "symptome(pierre,S)" et "cause(S,M)" réussissent. Examinons cette preuve: Que donne la preuve du premier sous-but associé à "est_atteint_de(pierre,M)" lorsqu'on utilise cette règle?	Q2.Contradict1
	Raisonnement	Quelle est l'expression de l'autre condition (sous-but) après cette première preuve?	Q2.Contradict2
	Raisonnement	Que donne donc la preuve de ce sous-but?	Q2.Contradict3
		On continue: Quel serait le résultat de la 2ème tentative avec cette	Q3

Numéro de sous dialogue	Étapes de la pensée réflexive associées à chaque question des sous dialogues	Questions du niveau D1	Code
		règle (s'il y en a)?	
Questions pour la pensée réflexive au niveau D2			
SD3	Situation	La 2ème tentative de prouver "est_atteint_de(pierre,M)" avec la règle "est_atteint_de(X,M) :-symptome(X,S)", "cause(S,M)" réussit.	Q3.S
	Raisonnement	Comment se fait le balayage de la Base de Connaissances pour effectuer une autre tentative de solution? A partir de quel élément de la base de connaissances se fera ce balayage pour la preuve de "symptome(pierre,S)"?	Q3.Assert.0
	Raisonnement	Donc on a la preuve "symptome(pierre,S)" avec S = fièvre. Ensuite? La preuve de la deuxième condition donne?	Q3.Assert.1
SD3	Raisonnement	Supposons que cette 2ème tentative échoue: cela signifie qu'après un Retour Arrière, on n'arrive toujours pas à prouver que les conditions pour "est_atteint_de(pierre,M)" sont vraies. Quelle est la preuve de "symptome(pierre,S)" après le Retour-Arrière qui suit l'échec de la première tentative?	Q3.Contradict.0
	Raisonnement	Alors, selon ton hypothèse, le résultat de prouver "cause(fievre,M)" est un Échec. Est il vrai qu'il n'existe aucun élément de la Base de connaissances qui peut être unifié à ce fait?	Q3.Contradict.1
		On continue: Y a t il encore des solutions possibles avec cette	Q4

Numéro de sous dialogue	Étapes de la pensée réflexive associées à chaque question des sous dialogues	Questions du niveau D1	Code
		règle?	
		On continue: Y a t-il encore une solution possible en utilisant un autre élément de la Base de connaissances?	Q5

Tableau D. 7. Codes associés aux composantes de la pensée réflexive

Type de réaction associé aux codes	Codes
Phase de présentation d'une situation problématique	
Perplexité et surprise	PX
Hésitation	HT
Désaccord avec une assertion du tuteur	DAT
Avis différent sur la nature d'un problème ou sur la manière de poser un problème	ANP
Incompréhension de la situation ou de la question	IC
Interrogation sur le but d'une question	IB
Réponse directe par rapport à une compréhension directe de la question	RDC
Phase d'intellectualisation	
Énonciation des informations pertinentes de la question	OBS_E
Questionnement sur les informations pertinentes (faits, conditions, préalables d'une loi ou d'un principe à appliquer pour répondre à la question) de la question	OBS_Q
Répétition de la question	OBS_R
Phase de raisonnement et d'inférence correspondante d'une réponse	
Mention d'un principe pour répondre à une question	R_Pr
Énonciation de l'application d'un principe ou d'une loi pour répondre : Énonciation du lien entre les faits intellectualisés (explicitement lors d'une étape d'intellectualisation ou implicitement) et les prémisses ou les implications d'un principe (ou d'une loi)	R_E
Mention d'une loi commune pour répondre	R_LC
Application d'un principe pour répondre : formulation de la solution ou d'une réponse en la reliant aux faits intellectualisés précédemment	R_APP
Mention de la violation du principe ou de la loi pour répondre	R_V
Mention d'un argument général pour répondre	R_Arg
Formulation directe de la réponse sans aucune référence aux faits intellectualisés	R_D
Intervention de l'expérimentateur et réaction correspondante du participant	
Question du tuteur sur le but d'une question posée dans le dialogue tutoriel	INT_BUT
Intervention générale du tuteur	INT_GEN

Type de réaction associé aux codes	Codes
Réponse ou Interprétation appropriée sur le but d'une question ou sur la démarche pédagogique qui correspond aux sous dialogues.	R_A_BUT
Réponse ou Interprétation appropriée sur le but d'une question ou sur la démarche pédagogique qui correspond aux sous dialogues, après une indication explicite de l'expérimentateur	R_A_BUT _{Exp}
Réponse/Interprétation inappropriée sur le but d'une question ou sur la démarche pédagogique	R_NA_BUT
Réactions du participant au feedback ou aux explications du tuteur	
Réaction positive à la solution ou à l'explication fournie par le système à la suite d'une réponse incorrecte	RP_E
Réaction négative ou confusion devant un feedback ou une explication fournis par le système à la suite d'une réponse incorrecte	RN_E
Réaction neutre ou sceptique devant une solution ou une explication fournies par le système à la suite d'une réponse incorrecte	RNS_E
Performance des participants	
Bonne réponse	B_R
Réponses incorrectes	I_R
Aucune réponse	A_R

Tableau D. 8. Codes associés aux participants de l'expérimentation avec Prolog-Tutor

Participant	Code de dénomination	Code d'enregistrement
Participant 1	P_1	audio_1
Participant 2	P_2	audio_2
Participant 3	P_3	audio_3
Participant 4	P_4	audio_4
Participant 5	P_5	audio_5
Participant 6	P_6	audio_6

Tableau D. 9. Durée des enregistrements par participant

audio_1	Durée : 45 minutes
audio_2	Durée : 48 minutes
audio_3	Durée : 75 minutes
audio_4	Durée : 45 minutes
audio_5	Durée : 47 minutes
audio_6	Durée : 60 minutes

APPENDICE. E

TRANSCRIPTION DES ENREGISTREMENTS EFFECTUÉS LORS DES
ÉVALUATIONS

Tableau E. 1. Transcription des données recueillies auprès de l'expert E2

Questions	Transcription des réponses
<p>Q1</p>	<p>Pour moi le diagnostic cognitif probablement de voir l'état des connaissances de l'apprenant à un moment X de l'apprentissage au moyen de pré-test. Mais, là je réalise que c'est un processus un peu plus global qui permet de savoir l'état des connaissances de l'apprenant en général et en tout temps.Sa façon de raisonner, ce qu'il a acquis, ses lacunes. Donc le DC permettrait au système de faire un enseignement plus spécialisé.</p> <p>Ma conception ? C'était plus l'état des connaissances dans un domaine donné, dans un contexte donné. Par exemple, le DC D'un apprenant En début de formation pour voir qu'est ce qu'il connaît, qu'est ce qu'il connaît moins, qu'est ce qu'il faut travailler pendant la formation. Il peut avoir un DC après la formation. C'était beaucoup plus sur l'état des connaissances de l'apprenant que sur ses démarches pour les acquérir. Mais ça dépend toujours des objectifs pédagogiques parce que si l'objectif pédagogique d'un cours c'est d'amener à être capable l'apprenant à apprendre à apprendre, de développer des stratégies d'apprentissage ou de l'amener à réfléchir sur ces stratégies. Alors un test de diagnostic portera sur ces habiletés mais il demeure que pour moi le DC c'était à utiliser pour ajuster un cours dès le début.</p> <p>Ça dépend de l'état des connaissances toujours dans un contexte donné. Le contexte correspond surtout au niveau de connaissances. Et à travers le concept que j'ai vu, j'ai constaté qu'on ne parlait pas de la même chose, que le diagnostic cognitif allait au-delà de l'état des connaissances</p> <p>[Intervention Expérimentateur] : Es tu d'accord avec cela?</p> <p>Oui puisque le raisonnement peut être vu comme une connaissance. En fait, ce la dépend du niveau d'habileté (intellectuelle,)</p>
<p>Q2 Réflexion</p>	<p>C'est d'autant plus pertinent que c'est une démarche de pensée vers l'apprentissage.</p> <p>[Intervention Expérimentateur] Je voulais savoir la pertinence du lien entre la réflexion de l'apprenant et le diagnostic cognitif</p> <p>Dans le contexte de Prolog-Tutor où il y a une démarche de réflexion vraiment profonde de l'apprenant et étant donnée la nature du domaine (logique et bien structurée), c'est facile de lier la réflexion aux capacités du domaine (et par conséquent à un diagnostic cognitif). Mais dans un autre domaine, plus dense et moins structuré où les apprenants n'ont pas nécessairement une façon de penser déterminée d'avance comme dans la pensée logique, ça pourrait difficilement être lié à une façon de pensée ou aux objectifs d'apprentissage.</p>

Questions	Transcription des réponses
	<p>Par contre, on pourrait surtout utiliser la réflexion pour raffiner le modèle de l'apprenant pour lui fournir des interactions qui correspondent plus à son style d'apprentissage à lui. En apprenant, il y a des apprenants qui ont beaucoup plus besoins de stimulus – réponses, et il y a des apprenants plus constructivistes qui ne veulent pas être dérangés. Il faut donc diagnostiquer son style. Dans Prolog-Tutor, ça aide à préciser le modèle apprenant et aussi à acquérir le contenu mais il n'y a pas une seule façon de penser pour acquérir les connaissances dans certains domaines et dans ce cas, la réflexion fournirait au système les informations sur le type d'apprenant par rapport au type d'interaction qu'il préfère</p>
<p style="text-align: center;">Q2 Théories de la cognition</p>	<p>Oui, c'est pertinent mais moi je le ferais dans une granularité plus fine parce que dans une formation entre le début et la fin d'une formation, les contextes peuvent changer. Il ne faut pas se limiter à un contexte possible du début à la fin d'une formation. C'est très lié aux théories.</p> <p>[Intervention Expérimentateur] Rappel du but du DC selon la théorie de la cognition.</p> <p>Parce que moi je pense que dans une situation d'apprentissage d'une langue par exemple, on peut apprendre les mots avec une approche stimulus réponse (behavioriste) alors que l'approche de compréhension de la langue nécessite une pratique en situation. Donc pour le même objectif d'apprentissage, on aura différents types de diagnostics cognitifs. On aura besoin d'information aux différents niveaux de cognition. Je pense donc qu'il ne faut pas se limiter. Un apprentissage peut changer au fur et à mesure que l'apprentissage évolue (le niveau d'acquisition visé change). Dans une formation complète, le diagnostic cognitif peut prendre différents niveaux, différents degrés de profondeurs, dépendamment des objectifs pédagogiques. Et dans une formation, les objectifs changent toujours. Au début de la formation: des connaissances plus déclaratives et ça va toujours en se complexifiant. Pour ce qui est de l'impact des théories d'apprentissage, je pense que l'apprentissage ne peut pas se faire d'une seule façon</p> <p>[Intervention Expérimentateur] C'est vrai en théorie mais dans l'instruction, on nous force presque à adopter un certain style d'apprentissage</p> <p>Disons que cela va dépendre du niveau des objectifs d'apprentissage; il faut surtout relier les objectifs d'apprentissage, les théories de cognition, et le diagnostic cognitif. Comme cela, ça permet de ne pas s'enfermer définitivement dans une théorie de cognition et d'apprentissage, sans tenir compte des variations propres au processus d'apprentissage réel. En fait l'approche par objectif est très importante dans ce sens car c'est elle qui permet vraiment d'orienter comment l'instruction, l'apprentissage et le diagnostic se déroulent.</p>
<p>Q3</p>	<p>Pour moi le contexte va être beaucoup plus large: la culture, les autres acteurs, les outils. Dans un contexte d'apprentissage, il faut tenir compte de tout ce qui est autour. Si je reprends l'exemple de l'apprentissage de la langue, la compréhension de certains aspects d'une langue nécessite dans une certaine mesure la connaissance et la compréhension de la culture et de</p>

Questions	Transcription des réponses
	<p>l'histoire reliées à cette langue. Il faut aussi bien respecter la variété de l'apprentissage à l'échelle microscopique, Dans une même session d'apprentissage, l'apprenant peut avoir différents types de démarches d'apprentissage. Il faut donc bien préciser qu'en fonction de l'objectif pédagogique du moment, il est possible d'embrasser divers styles de diagnostic cognitif dans la même activité d'apprentissage.</p>
Q4	<p>Dans mon cas particulier, ce cadre m'a fait réaliser ou articuler des connaissances à propos du diagnostic cognitif dont je n'étais pas consciente. Suite à notre discussion, il a deux points clés que j'ai vus :</p> <ul style="list-style-type: none"> le fait d'amener les théories de cognition au premier plan fait ressortir la nécessité d'un lien entre les objectifs pédagogiques (leur niveau) et le diagnostic cognitif, ce dont on n'est pas toujours conscient le fait d'essayer de combler le fossé entre le concepteur pédagogique et le programmeur à travers les mécanismes de représentations de connaissances ouvre un aspect intéressant de la recherche qui ferait un lien entre les objectifs d'apprentissage, la représentation des connaissances et l'utilisation de ces représentations par l'algorithme de diagnostic cognitif fait par un programmeur
Q5	<p>Ce que je c'est que le dialogue tutoriel est très élaboré et c'est sûr qu'il va au fond des choses ... peut être trop d'ailleurs. Cela dit, si je vois le mode d'interaction, il ne correspond pas à la théorie d'apprentissage constructiviste. L'instruction est constructiviste mais pas l'apprentissage dans ce cas.</p> <p>C'est vrai que dans la spécification d'attacher explicitement les connaissances ou les capacités à chaque question impose deux choses que je trouve intéressantes: le concepteur est obligé de vraiment bien réfléchir à ses questions de sorte qu'elles soient formulées pour stimuler une capacité précise; le programmeur doit explicitement relier un modèle des connaissances du domaine à des questions et à une représentation des connaissances de l'apprenant. Oui, c'est pas mal comme concept. Mais c'est sûr que c'est beaucoup plus pertinent dans les systèmes tuteur intelligents.</p>
Q6	<p>Oui, je crois qu'un tel cadre pourrait faciliter les communications entre un concepteur et un programmeur. Il est parfois difficile pour le concepteur d'expliquer ses objectifs au programmeur pour qu'il comprenne bien le pourquoi de ce qui lui est demandé. Je crois aussi qu'un programmeur qui comprend bien le but pédagogique est plus motivé</p> <p>Concepteurs et programmeurs ont des bagages de connaissances différents et le fait d'utiliser un cadre de référence peut faciliter la</p>

Questions	Transcription des réponses
	communication et augmenter l'efficacité...

Tableau E. 2. Transcription des données recueillies après de l'expert E3

Questions	Transcription des réponses
<p style="text-align: center;">Q1</p>	<p>Le diagnostic cognitif est l'identification de l'état cognitif de l'apprenant avant l'apprentissage. Ceci afin de connaître ses caractéristiques et ainsi déterminer les priorités d'instruction et d'apprentissage. En fait l'apprenant est observé avant, pendant et après l'apprentissage. Pour nous concepteurs pédagogique, chacune de ces observations correspond respectivement au diagnostic cognitif (les acquis de l'apprenant), à l'évaluation formative (progression pendant l'apprentissage) et à l'évaluation sommative (quel est l'état après l'apprentissage). C'est vrai que la considération du DC dans cette recherche embrasse ces trois observations et au-delà, ainsi qu'elle précise l'usage pédagogique dans diverses dimensions, Disons que j'admets que c'est une question de terminologie. Si je comprends bien, ce que tu appelles diagnostic cognitif correspond à ce que nous appelons évaluation sommative.</p>
<p style="text-align: center;">Q2 Réflexion</p>	<p>Il faut bien faire la différence entre la réflexion dans le but du DC (réflexion du côté du système) et la réflexion dans le but de l'apprentissage (pensée réflexive chez l'apprenant). Pour le système, la "pensée réflexive chez l'apprenant" permet d'analyser et de confirmer le DC dans le sens que:</p> <ul style="list-style-type: none"> - la question est posée à l'apprenant - le système analyse le résultat - le système élabore le DC en raffinant la question et surtout en encourageant la justification (caractéristique principale de la pensée réflexive) <p>Par exemple, lorsque l'étudiant s'auto diagnostique, il donne des réponses sans trop savoir (Connais tu la soustraction? Où est ce que tu l'as appris? Peux tu soustraire un nombre à 2 chiffres?). Par la réflexion, le système est capable de confirmer les dires de l'apprenant. La réflexion peut aussi avoir un effet sur l'apprentissage. Sans réflexion, il n'y a pas d'apprentissage de haut niveau. On peut réciter une multiplication sans être capable d'effectuer une multiplication. On peut réciter un poème en langue étrangère (connaître la phonétique de la langue) sans nécessaire la comprendre (comprendre la sémantique de la langue)</p>
<p style="text-align: center;">Q2 Théories de la cognition</p>	<p>Il faut faire attention: les positions sur les théories de la cognition ne sont pas tranchées (l'exemple béhavioriste: Skinner a posé une hypothèse pour expliquer certains phénomènes cognitifs, sans pour autant admettre que cette hypothèse éliminait les autres). Il est difficile de tracer une ligne dure dans les théories de la cognition, c'est-à-dire des différences marquées et définitives. Au lieu de différencier les diagnostics cognitifs en fonction des théories de la cognition, il faut le regarder sous l'angle de l'habileté visée à travers une activité d'apprentissage où l'apprenant est observé. Par exemple, si on considère Gagné, on a le niveau le niveau application (3) , le niveau d'analyse (5) , et le niveau d'évaluation (6). Ces</p>

Questions	Transcription des réponses
	<p>habiletés permettent d'aller chercher un certain niveau chez l'apprenant par rapport à une connaissance. Il serait plus intéressant de faire le diagnostic cognitif en fonction de l'effort cognitif à appliquer à chacune des connaissances, par exemple on a les niveaux suivants :</p> <p>Awareness (connais-tu?) Identification (peux tu différencier?) Application (peux tu faire?) Évaluation, métacognition (peux tu évaluer?)</p> <p>En autant que le niveau d'habileté est associé au DC. Cependant, l'argument de départ demeure : le DC diffèrera selon le type d'habileté auquel il s'applique. Par exemple, si je veux déterminer si un étudiant en droit sait défendre une cause (application), je m'y prendrai différemment que si je veux déterminer s'il connaît la loi (awareness). De même, si je veux déterminer qu'un étudiant en médecine sait faire une injection (application), je m'y prendrai différemment que si je veux savoir s'il connaît les symptômes de certaines maladies (Awareness)</p>
Q3	<p>Dans l'utilisation du CSDC, il faut marquer la différence entre la réalisation d'un exercice qui est une activité d'apprentissage générique et la résolution d'un problème qui est une habileté de haut niveau.</p> <p>Pourquoi y a-t-il 3 types d'inférences dans le DC épistémique et 2 seulement dans le DC comportemental?</p> <p>À part cela, je ne vois rien d'autre et je pense de toutes façons que c'est avec l'expérience avec le CSDC que les ajouts viendront, selon les besoins des concepteurs. C'est sûr qu'il y a les caractéristiques telles que la personnalité, la différence entre le niveau de départ et le niveau ciblé par rapport à chaque capacité.</p>
Q4	<p>Si je comprends bien, le CSDC doit être vu comme un outil qui supporte le concepteur pédagogique dans la spécification du diagnostic cognitif. Cette spécification est implémentée sous forme computationnelle par un programmeur, le but étant de préserver les intentions pédagogiques du concepteur, c'est-à-dire: la cohérence entre le diagnostic et la théorie de cognition choisie et la stimulation de la pensée réflexive sur les capacités diagnostiquées comme manquantes à l'apprenant.</p> <p>Je pense que c'est un schéma logique, cohérent et utile dans le cadre général de la conception pédagogique. Cependant, il ne faut pas oublier que les résultats du diagnostic doivent être évalués justement pour vérifier ces propriétés. Il faudrait inclure explicitement l'intervention d'un tel instrument de mesure ou tout au moins un humain dans la boucle (avec leur rôle, les ressources qu'ils utilisent, et l'impact qu'ils produisent dans le processus de conception du diagnostic).</p> <p>L'utilité de ce cadre dépendra aussi du fait qu'il devienne une référence dans le langage entre le concepteur et le programmeur. Il faut absolument faire converger les conceptions entre les concepteurs et les programmeurs. Cela ne peut se faire que par approximation, à travers l'expérience</p>

Questions	Transcription des réponses
	<p>entre les deux parties. La difficulté est reliée au langage utilisé de part et d'autre. Oui, un cadre serait effectivement un point de départ pour ces ajustements.</p> <p>Il y a beaucoup de difficultés pour cette convergence mais une certitude demeure, un CSDC serait utile dans la mesure où il permet de faciliter le dialogue entre le concepteur et le programmeur sans que l'un soit obligé de maîtriser le langage de l'autre,</p>
Q5	<p>Le dialogue tutoriel remplit bien son rôle de stimulateur de la pensée réflexive. Cependant la théorie constructiviste ne ressort pas de ce dialogue. Cela confirme le fait qu'il vaut peut être mieux fonctionner par habileté. Le modèle de l'apprenant aussi correspond bien à un diagnostic épistémique.</p>
Q6	<p>Le concepteur ne veut pas investir du temps à comprendre un système, même si ce système est supposé l'aider dans sa tâche. Ainsi, l'utilité de CD-SPECIES dépendra de son interface. Cependant, je peux déjà dire que la principale difficulté est d'assurer une certaine concision. Par exemple, le concepteur n'a pas besoin de se voir présenter une liste de 50 types de CD. Il faudrait mettre les types les plus fondamentaux ou les plus couramment utilisés ou les plus connus. Cela nécessite une analyse des besoins du concepteur.</p> <p>J'admets cependant qu'une bonne raison d'être de CD-SPECIES est qu'il pourrait supporter d'une manière ou d'une autre le fameux ajustement entre les concepteurs et les programmeurs.</p>

Tableau E. 3. Transcription de l'enregistrement audio_1 avec Prolog-Tutor

Question	Transcription des réponses du participant	Interprétation
Q0	[App] Le but? Mais c'est est_atteint_de(pierre,M)	BR >
Q0.S	[App] D'accord	RDC →
Q0.Intellect	[App] La variable? C'est grand M	OBS_Q, OBS_E
Q0.Assert	[App] Que signifie cette requête? Si je veux tous les résultats possibles c'est que je cherche toutes les valeurs de M.	OBS_R, R_Arg, R_A_BUT
Q1	[App] Le premier c'est est_atteint_de(X,M) :- symptôme(X,S), cause(S,M)	BR >

Q1.S	[App] Oui, je suis d'accord car est_atteint_de(X,M), on le retrouve dans la base de données.	RDC →
Q1.Intellect	[App] La conséquence de cette règle et les conditions? La conséquence? Les conditions ce sont symptôme(X,S) et cause(S,M), La conséquence c'est vrai, on doit vérifier que symptôme(X,S) et cause(S,M) est vrai. [Lecture du feedback] [App] Je ne comprends pas. [Demande d'explication et Lecture de l'explication] [App] Ahh! Ok, je vois maintenant qui est la conséquence	OBS_Q, RP_E
Q1.INTELLECT2	[App] Il me pose cette question parce que l'unification est une opération très importante en Prolog. Pour que est_atteint_de(X,M) soit valide, il faut prouver symptôme et cause. Il faut que la règle s'unifie avec le but. Oui ils sont unifiables.	R_Pr, OBS_E
Q1.Assert1	[App] Pour les foncteurs : Le foncteur c'est est_atteint_de et est_atteint_de sont identiques Les arguments, ah! On a 2 arguments dans chaque terme dont ça va. Les variables X va être unifiée avec pierre. M c'est M, ça va rester tel quel.	R_App, R_A_BUT
Q2	[App] Ok. Non, ça va échouer.	BR >
Q2.S	[App] Oui je suis d'accord	RDC →
Q2.Assert0	[Participant Répète la question] [App] Le premier sous but ce sera symptôme(X,S), c'est ça le premier sous but. Oui je vais utiliser symptôme(pierre, fatigue). Ça réussit, X = pierre et M = fatigue	R_E, R_App
Q2.Assert1	[App] Le prochain sous-but c'était cause(S,M). Si on remplace S, on cherche cause(fatigue,M) L'élément correspondant c'est cause....Oh! Il n'y a aucun élément correspondant	R_Pr, R_E, R_App, R_A_BUT
Q3	[App] On va utiliser notre deuxième symptôme. On avait utilisé symptôme(pierre,	IR ➡

	fatigue), maintenant on va utiliser symptôme(pierre,fièvre). Résultat euuuuuh .. succès. Mais toutes les autres causes seront un Échec, don résultat, c'est un Échec.	
Q3.S	[App] Je ne suis pas d'accord avec cela.	DAT
Q3.Contradict	[App] Ça veut dire que cherche la preuve de symptôme(pierre,S) après le retour arrière. Ah, il me demande de chercher une alternative puisque symptôme(pierre,fatigue) n'a pas marché. La deuxième tentative ici c'est symptôme(pierre, fièvre)	R_A_BUT, R_Pr, R_App
Q3.Contradictl	[App] Non, puisque j'ai un élément cause(fièvre, infection) donc M = infection. Ah oui, à cause de ce détail, je me rends compte effectivement ... oui je m'étais trompé à la dernière question	R_E, R_App, RP_E
Q4	[App] Non, il n'y a plus de solution possible.	R_□
Q5	[App] Oui. est_atteint(pierre,malaria). Oui on peut utiliser ça en faisant un retour arrière, Mais il faut parcourir toute la BC Et la variable M est égale à malaria.	R_Pr, R_App
Débriefing		
SG_I	<ul style="list-style-type: none"> - Mettre un feedback qui ressort plus; - Mettre des indicateurs qui signalent le déclenchement du sous-dialogue ainsi que le retour au dialogue de niveau supérieur - Il faudrait qu'il y ait une structure et que le système dise explicitement à l'apprenant qu'il a divisé le problème en étapes. Et il doit indiquer à chaque moment dans quelle étape on est et à quel niveau on se trouve. 	

SG_P	<ul style="list-style-type: none">- Je ne sais pas souvent à quel niveau du raisonnement il se trouve. Ce serait bien s'il pouvait dire à quel niveau du raisonnement ou de la construction de la solution on se trouve par rapport à la question générale;- Oui je comprends la démarche du tuteur. Vers la fin, j'ai compris le pattern du dialogue; que le but était de décortiquer la question pour mieux comprendre comment y répondre; <p>Mon problème est que je me suis perdu par rapport au but principal. Le système se concentrait sur des détails comme symptôme, cause alors que le problème était de résoudre est_atteint_de. J'avais de la difficulté à faire le lien entre les deux. À part le bouton «Pourquoi», il faudrait peut être un schéma qui indique où est ce que la question actuelle se trouve par rapport à l'objectif initial ou au problème initial, pour mieux comprendre où le tuteur veut en venir. Mettre des couleurs pour indiquer le rôle de chaque intervenant: voilà où on se situe, voilà la question, voilà tes réponses, etc.</p>
------	---

Tableau E. 4. Transcription de l'enregistrement audio 2 avec Prolog-Tutor

Question	Transcription des réponses du participant	Interprétation
Q0	[App] est_atteint_de (pierre,...)	BR ➤
Q0.S	[App] Oui	RDC ➡
Q0.Intellect	[App] M	OBS_E □
Q0.Assert	[App] Hummmm... Moi je vois les 2 premières réponses. Dépend de l'utilisation qu'on veut en faire	R_Arg
	[Intervention Expérimentateur] Est-ce que ça te permet de faire un lien entre la requête et ce que tu dois faire?	☆ INT_BUT
	[App] Prouver le fait est_atteint_de(pierre,M) dépendamment de : pierre, M, de la base de connaissances. À ce moment la ... hummmmm! Les 2 premières sont possibles.	R_A_BUT, R_Arg
Q1	[App] est_atteint_de(X,M) :-	BR ➤
Q1.S	[App] Oui ... Ça pourrait ne pas avoir être utilisé dépendamment de l'ordre de parcours de la base de connaissances. Je suis d'accord avec cette affirmation	(RDC ➡, ANP)

Q1.Intellect	<p>[Intervention Expérimentateur] Comment tu la redirais en langage naturel?</p> <p>[App] Un individu est atteint d'une maladie M si cet individu a cette maladie et s'il présente les causes</p> <p>[Question de l'apprenant] Je ne comprends pas la question</p> <p>[Clarification du système]</p>	
	<p>[Intervention Expérimentateur] Pourquoi cette question t'est – elle posée? Quel est le lien entre la conséquence de la règle et ses conditions?</p>	INT_BUT
	<p>[App] Hummm. Ok. La conséquence de cette règle correspond au même prédicat que le but à prouver Et les conditions ... cause(X,S), symptôme(S,M)</p> <p>Ahhhh! Oui, ok. Maintenant je vois le lien entre la règle et le but à prouver. Pour prouver est_atteint_de, il me faut prouver symptôme et cause</p>	R_A_BUT, OBS_E, R_A_BUT
Q1.INTELLECT2	<p>[App] Pourquoi il me pose cette question ...?</p> <p>Ah. Pour pouvoir utiliser la règle, il faut unifier sa conséquence au but. Oui</p>	PX, R_A_BUT
Q1.Assert1	<p>[App] Probablement ils sont exactement les mêmes foncteurs. Puis heuuuuu...</p> <p>Ah! Dans ce cas les arités? ok c'est 2, l'arité est 2 et Puis la valeur de X est pierre Et M reste = à M. Donc ils sont unifiables.</p>	OBS_E, OBS_Q, R_App(Unification arguments), R_E (Unification termes)
Q2	<p>Ah!</p> <p>[Erreur de frappe]</p>	
Q2.S	<p>[App] Beeeeeeeeeen! Je ne suis pas sur, c'est possible qu'il y ait des échecs. Je suis d'accord.</p>	HT
Q2.Assert0	<p>[App] Le X ici est devenu Pierre, donc j'essaie probablement de prouver que pierre euhhhh symptôme S. Oui je vais essayer de prouver que pierre a un symptôme S. Le premier avec lequel il va être prouvé est symptôme(pierre,</p>	R_A_BUT, R_E, R_App

	fatigue). Euhmmmm. Donc ça marche parce la valeur S devient S = fatigue	
Q2.Assert1	[App] Le prochain sous but étant «la cause de la fatigue est M». Il n'y a aucune «cause de la fatigue» donc ça va échouer [Utilisation de l'interface pour indiquer ce résultat].	R_A_BUT, R_E, R_App
Q3	[Erreur de frappe]	
Q3.S	[App] Oui je suis d'accord	RDC:→
Q3.Assert0	[App] En fait, la première condition peut être prouvée avec symptôme(fièvre,pierre) et cause(fièvre, infection)	R_App
Q3.Assert1	[App] cause(fièvre, infection) donc c'est ça, on a un succès	R_App
Q4	[App] Non, parce que je n'ai plus de symptôme(pierre,...) ... A moins qu'on ne trouve d'autres causes de fièvre?	R_App, OBS_Q
Q5	[App] Ah! C'est ce qu'on avait au début, pierre est atteint de Malaria. Alors la variable M prendra la valeur malaria.	R_D
Débriefing		
SG_I	<ul style="list-style-type: none"> - La fenêtre déroulante doit être plus pratique. On ne doit pas avoir besoin à chaque fois de la dérouler pour voir où est ce qu'on en est avec le dialogue; - Corriger les fautes de français voir accès à la base de connaissances; - Surligner de couleur les éléments de la base de connaissances qui sont impliqués dans une explication; 	

SG_P	<ul style="list-style-type: none">- Au niveau de certaines questions, je n'avais pas nécessairement appris les mêmes termes que ceux correspondant à la terminologie utilisée dans le dialogue. J'avais des fois des termes. Une fois que je comprenais la question, je comprenais toujours pourquoi elle apparaissait dans le dialogue,- Les sous questions ont effectivement une valeur pédagogique dans ce sens qu'elles mettent sur une piste de réponse ou tout au moins, elles permettent de comprendre les fondements profonds d'une réponse, fondements dont on n'est pas toujours conscients. Ceci dit, cette valeur n'est réelle que si le langage, la terminologie et la sémantique du tuteur sont compréhensibles et interprétables. Ce qui n'était pas toujours le cas dans ce cas. Il faudrait inclure une composante de langage naturel.
------	--

Tableau E. 5. Transcription de l'enregistrement audio 3 avec Prolog-Tutor

Question	Transcription des réponses du participant	Interprétation
Q0	[App] Chercher la maladie dont pierre est atteint. Alors, il faut que je donne la solution? [Intervention Expérimentateur] Non, il faut identifier le but tel qu'il est dans le problème [App] Je ne vous suis pas là M?	IR ➡
Q0.S	[App] D'accord	RDC ➡
Q0.Intellect	[App] La variable? C'est M	OBS_E ☐, R_NA_BUT
Q0.Assert	[App] Trouver les valeurs de M telles que est_atteint_de() est vraie. Oui je dois trouver les valeurs de M telles que ce fait est vrai.	R_0, R_A_BUT
Q1	[App] est_atteint_de(X,M) :-	BR ➤
Q1.S	[App] Oui je suis d'accord.	(RDC ➡)
Q1.Intellect	[Intervention Expérimentateur] À ton avis, c'est quoi le but de cette question? [App] Le but de cette question, c'est de prouver pourquoi cette règle doit être utilisée. La conséquence, c'est "trouver M" [Intervention Expérimentateur] Pourquoi tu penses que tu dois utiliser cette règle? [App] je vais d'abord prouver symptôme() et ensuite cause(), c'est pour cela que cette règle doit être utilisée [Intervention Expérimentateur] Comment tu exprimes cette règle en langage naturel? [App] Cette règle veut dire, X est atteint de M [Indication d'une réponse à l'interface, M] [Lecture du feedback] [App] Il va falloir que je prouve... Donc la conséquence est_atteint_de() et les causes sont symptôme() et cause(). Ahhhhh!!! D'accord je comprends. cette règle signifie Si	INT_BUT, R_A_BUT, R_PE

	symptôme() et cause(), Alors est_atteint_de()	
Q1.INTELLECT2	[App] Pourquoi il me pose cette question ...? Est-ce que je peux prouver mon but avec ces règles là? est_atteint_de_(X,M). Oui tout à fait, elle est unifiable	PX
Q1.Assert1	<p>[Intervention Expérimentateur] Pourquoi tu penses qu'on te pose cette question dans le contexte de cet exercice?</p> <p>[App] Cela veut dire que j'ai déjà repéré Pierre parmi les X</p> <p>[Intervention Expérimentateur] est_atteint_de(X,M) foncteur est est_atteint_de,</p> <p>[App] Ce sont les prédicats. Je veux prouver que symptôme(pierre,) et cause(pierre).</p> <p>[Intervention Expérimentateur] Pourquoi est_atteint_de(X,M) est unifiable avec est_atteint_de(pierre,M)</p> <p>[App] Ok. La variable X = pierre et M=M</p>	R_App, R_NA_BUT
Q2	<p>[App] Je cherche symptome(pierre,S). J'ai symptôme(pierre,fièvre) donc selon la base de connaissances, pierre est atteint de fièvre. Je vais donc chercher cause(fièvre, infection). J'ai donc M = infection. Ah! J'ai aussi le fait est_atteint_de(pierre, malaria)</p> <p>[Intervention Expérimentateur] Tu peux relire la question?</p> <p>[Relecture de la question]</p> <p>[App] Ok, Pour cette réponse j'ai «infection». J'ai identifié «infection»</p>	IR ➡
Q2.S	<p>[App] Je ne suis pas d'accord</p> <p>[Apprenant réexamine la réponse à la dernière question]</p> <p>[App] Ah, c'est vrai que j'avais symptôme(pierre, fatigue)... Ok, je suis d'accord avec cette affirmation</p>	HT
Q2.Assert0	<p>[Participant Répète la question].</p> <p>[App] Ok. Donc c'est symptôme(X,S) je pense? Ah! Non c'est vrai que je cherche symptôme(pierre,S).</p>	OBS_R, OBS_Q, R_App, RP_E.

	<p>[Utilisation de l'interface pour spécifier le sous but] [App] Indiquer l'élément de la base de connaissances que vous utilisez?! Ok, je vais utiliser symptôme(pierre, fatigue). Aucune substitution nécessaire [Utilisation de l'interface des substitutions] [Lecture du Feedback] [App] Ah!! Mais oui, Bien sûr que j'ai substitué S à fatigue. Non, je pensais déjà à l'étape suivante. Effectivement j'ai substitué S à fatigue</p>	R_A_BUT
Q2.Assert1	<p>[Participant répète la question] [Intervention Expérimentateur] Est-ce que tu comprends la question? [App] Ok, le prochain sous-but c'est cause(S,M)!! On vient de trouver S = fatigue, donc on cherche cause(fatigue,M) et donc là la preuve de cause(fatigue,M) va donner M. [Participant indique qu'il cherche cause(fatigue,M)] [App] Hummm!! Indiquer l'élément correspondant de la base de connaissances???? Ah ok,... cause(fievre, infection)?? non... Non!! Il n'y a pas d'élément correspondant donc c'est un échec [Intervention Expérimentateur] Est-ce que tu as compris ce que tu viens de faire. Quel est le but de toutes ces questions? [App] Ah! Le but c'est de rechercher la maladie. là le tuteur me montre pas à pas à repérer ma preuve jusqu'à la fin. Il m'emmène vers la solution</p>	OBS_E, R_E, R_App, R_A_BUT
Q3	<p>[App] Ok Là je dois revenir avec symptome(pierre, fatigue).. J'ai symptome(pierre,S). Donc maintenant, je remonte et je crois que je dois utiliser symptome(pierre, fievre).</p>	BR > R_E, R_App, R_A_BUT
Q3.S	<p>[App] Oui je suis d'accord</p>	RDC
Q3.Assert0	<p>[App] Si je comprends bien la question, on veut faire une deuxième tentative en ignorant ce qui a déjà été utilisé. C'est-à-dire au-delà de symptôme(pierre, fatigue) ... Hummm! Ok c'est donc S = fièvre maintenant avec symptôme(pierre,S) [Participant Utilise l'interface pour sa BC]</p>	R_E, R_App, R_A_BUT
Q3.Assert1	<p>[App] Ok. Et on a cause(fievre, infection) M= infection.... Aaaaah ... je vois. Il m'amène</p>	R_App, R_A_BUT

	progressivement à réaliser mon erreur. Il m'amène à reconstituer mon raisonnement. En revenant sur mes réponses étape par étape	
Q4	[App] Non. Parcequ'on a épuisé l'utilisation de tous les faits symptôme(pierre,S).	R_E, R_App
Q5	[App] Oui. est_atteint(pierre,malaria). M = malaria. Oui c'est un succès	R_App
Débriefing		
SG_I	Mettre un feedback qui ressort plus	
SG_P	<p>J'ai bien compris que le système à pour but d'amener l'apprenant à justifier ses solutions et à les construire Par contre, lorsque le système commence un sous dialogue, ce serait intéressant de l'indiquer clairement au début. Par exemple : "Ok, maintenant on va reconstituer le raisonnement".</p> <p>Ce que j'ai bien aimé, c'est cette manière progressive de m'amener vers mes erreurs. Mais ce serait bien qu'il donne des explications pour m'éviter de faire des erreurs. Ah non! C'est vrai que le but est de me laisser commettre l'erreur. Oui, c'est encore mieux comme cela effectivement. En comprenant mon erreur, je comprends mieux</p>	

Tableau E. 6. Transcription de l'enregistrement audio 4 dans Prolog-Tutor

Question	Transcription des réponses du participant	Interprétation
Q0	[App] Bon, c'est est_atteint_de(pierre,M)	BR >
Q0.S	[App] D'accord	RDC →
Q0.Intellect	[App] C'est M	OBS_E □
Q0.Assert	[App] Les choix possibles ... D'après moi c'est En fait j'ai une ambiguïté entre deux alternatives. Je pense que toutes sont possibles, mais je pense aussi qu'en Prolog, on veut toutes les valeurs de M [Lecture du Feedback] [App] Il faudrait qu'il me prouve que ma réponse est vraie, j'ai besoin de savoir si mon cheminement est correct	R_Arg, R_A_BUT
Q1	[App] est_atteint_de(X,M) :- symptôme(X,S), cause(S,M)	BR >
Q1.S	[App] Oui, je suis d'accord	RDC →
Q1.Intellect	[App] Qu'est ce que je dois répondre? Je pense qu'on a un problème de vocabulaire. [Lecture de l'explication de la question] [App] Quelle est la conséquence de cette règle et les conditions? La conséquence? [Répétition de la question] [App] Examinons la forme: Si conditions alors conséquence. S'il y a des symptômes alors cause. Donc cause(S,M) est donc la conséquence et la condition est cause [Lecture du feedback] [Demande d'explication] [App] J'essaie de valider si j'ai réussi à bien répondre pour pouvoir passer à autre chose et là le feedback ne me le dit pas. Ah oui, je viens de comprendre que la clause de Horn est comme l'expression d'une règle en français mais à l'envers. Ok. Je viens de faire le lien entre les 2.	ANP, OBS_Q, R_PE, R_A_BUT

Q1.INTELLECT2	Répétition de la question. App : Oui elle est unifiable avec le but à résoudre.	RDC ➡
Q1.Assert1	[App] Pour les foncteurs : le foncteur c'est est_atteint_de et est_atteint_de et ils sont identiques. Et on a la substitution X = pierre. Ok. Ouais je vois pourquoi cette question. Là on a prouvé qu'on va pouvoir résoudre cette requête avec cette règle. Si les deux éléments n'avaient pas été unifiables, cela n'aurait pas pu être possible.	R_App, R_A_BUT
Q2	[App] En Prolog, ça fonctionne de gauche vers la droite. Donc je vais essayer de résoudre symptôme(X,S) en remplaçant X par la substitution X = pierre, S = fatigue. S = fièvre et on trouve M qui devient infection et près cela il n'y a pas d'autre symptôme pour pierre [Lecture du feedback] [Demande d'explication]	IR ➡ Accidentellement dû à une inattention
Q2.S	[App] Oui je suis d'accord	RDC ➡
Q2.Assert0	[Répète la question]. [App] symptome?Ok. symptome(pierre, S). Succes avec S = fatigue	OBS_Q, R_App
Q2.Assert1	[App] Le prochain étant cause(S,M). Le sous-but c'est cause(S,M). D'après le feedback que je viens de recevoir, je devrais indiquer cause avec fatigue et M. Puis ensuite, l'élément correspondant de la BC, on va chercher cause(fatigue,M) hummm!! Je ne comprends pas qu'est ce qu'il veut que je fasse? [Intervention de l'expérimentateur] [App] Ok, aucun élément de la BC ne correspond	OBS_E, R_E, R_App, R_NA_BUT, R_A_BUT _{Exp}
Q3	[App] C'est un succès avec M = infection	BR ➤
Q3.S	[App] Je suis d'accord avec ça	RDC ➡
Q3.Assert0	[App] Ah! C'est du <i>backtracking</i> en Prolog. J'imagine que ce sera à partir de ... [Répétition de la question] [App] Là comme on est rendu à la 2ème tentative, ça va être symptôme(pierre, fièvre).	R_P, OBS_R, R_App, R_A_BUT

	Ouais je comprends maintenant pourquoi il me demande de donner la valeur de S	
Q3.Assertf	[App] Le sous-but à résoudre est maintenant cause(fièvre,M) et si je vais regarder dans ma base de connaissances, j'ai un succès avec M = infection	R_App
Q4	[App] Non, il n'y a pas d'autres symptômes qui fonctionne avec pierre, donc il n'y a plus de solution possible.	R_E_R_App
Q5	[App] Non, je n'en vois as d'autre. [Pause] [App] Ah. Si je vois le fait est_atteint_de(pierre,malaria). Donc M = malaria est aussi une solution	R_App
Débriefing		
SG_I	<p>Les interfaces secondaires peuvent porter à confusion. Par exemple, j'ai été perdu par la résolution d'un sous-but à travers l'utilisation de deux interfaces en séquence.</p> <p>Les feedback doivent explicitement indiquer si la réponse est correcte ou incorrecte puisque cela indique si on raisonne bien ou pas. Un feedback devrait pouvoir clairement indiquer si on a la bonne réponse, et justifier que le but du sous-dialogue est de s'assurer que j'ai bien compris malgré ma bonne réponse. Dans le cas d'une mauvaise réponse, le feedback devrait indiquer que le sous-dialogue a pour but d'analyser la question.</p> <p>Le contenu des interfaces: il faut les placer côte à côte car souvent on a besoin des informations dans des interfaces en arrière qui sont cachées par les interfaces secondaires.</p>	
SG_P	<ul style="list-style-type: none"> - Les explications aident donc il faudrait les améliorer d'autant plus qu'elles sont intéressantes étant basées sur des exemples - La structure du dialogue: On a un problème global et on le décortique jusqu'à sa plus simple impression. Cette approche est plus utile pour un débutant ou au début d'un cours. Mais pour des experts, ils passent surtout leur temps à essayer de comprendre pourquoi le tuteur pose toutes ces questions. - L'articulation du système m'amène même en temps qu'expert à articuler des choses qui étaient implicites (par exemple, les sous-buts associés à une règle). Ça nous oblige à articuler nos connaissances. - [Intervention de l'expérimentateur] Le participant a souvent la bonne réponse dans la tête mais dans certains cas le système n'arrive même pas à identifier cette bonne réponse. Vous avez 	

entrée cause(S,M) qui a été diagnostiqué comme une mauvaise réponse alors qu'en fait l'apprenant avait la bonne réponse (donc problème de formulation de la question ou problème de compréhension)

- Le bouton désaccord n'a jamais été utilisé. Mais tous les désaccords correspondent à la mauvaise réponse. Un expert ne va donc jamais cliquer dessus. En fait il faut proposer des situations plus complexes de manière qu'il y ait possibilité de se tromper même pour l'Expert.
- Le but des sous-dialogues était de se justifier effectivement, comme si c'était des sous questions. Mais, il vaut mieux aller droit au but et éliminer D1 pour ne pas nous obliger à réfléchir 2 fois. Il serait peut être plus intéressant de commencer par la réponse finale et amener l'apprenant à déterminer comment cette réponse est construite.

Tableau E. 7. Transcription de l'enregistrement audio 5 avec Prolog-Tutor

Question	Transcription des réponses du participant	Interprétation
Q0	[App] De quoi est atteint pierre. Ok. est_atteint_de(pierre,M)	BR >
Q0.S	[App] Non je ne suis pas d'accord	HT, DAT
Q0.Intellect	[App] La variable? La variable c'est M	OBS_E ☐
Q0.Assert	[App] Trouver que le fait est_atteint_de(pierre,M) est vrai pour au moins une valeur de M. Ah!! C'est parce que la question ne disait pas pour au moins une valeur de M	R_0, R_A_BUT
Q1	[App] Je vais chercher un fait qui pourrait être unifié à la requête est_atteint_de(pierre,M)	R_E, R_App, BR >
Q1.S	[App] Oui, je suis d'accord	RDC ➡
Q1.Intellect	[App] Est-ce que c'est toujours en rapport avec la requête est_atteint_de(pierre,M)? En Prolog, la conséquence est symptôme(X,S). [Demande d'explication] [App] Ah!!!! Ok. Je vois. La manière que j'avais compris que les règles sont comme des Si alors. Maintenant je vois : la conséquence c'est est_atteint_de ...	R_NA_BUT, RP_E, R_A_BUT
Q1.INTELLECT2	[App] Oui.	RDC ➡
Q1.Assert1	[App] Ce que je vois, c'est que les foncteurs sont pareils, les arités sont pareilles (arité 2 partout). C'est pour ça qu'elles sont unifiables. Ah. Je comprends le lien entre la règle Prolog, la condition et la conséquence. [Intervention de l'expérimentateur] Est-ce que tu as compris pourquoi on te pose les questions? [App] J'ai l'impression que les sous-questions ont pour but de mieux clarifier la	R_App, R_A_BUT

	question du niveau supérieur. C'est mon impression.	
Q2	[App] Premièrement j'aurai est_atteint_de(pierre,M) ce qui me donne symptôme(pierre,S) et cause(S,M) comme première tentative. On a symptôme(pierre,S) et cause(S,M). J'essaie d'unifier symptôme. On a symptôme(pierre,S) avec symptôme(pierre,fatigue) donc S = fatigue. Ensuite on aura cause(fatigue,M). cause(fatigue,M) hummmmm! Ok. Je pense que cette tentative a échoué	R_App
Q2.S	[App] Oui je suis d'accord	RDC →
Q2.Assert0	[App] Le sous-but c'est symptôme(pierre,S) et l'élément de la BC c'est symptôme(X,S). Celui là va être un succès avec la variable S = fatigue	R_App
Q2.Assert1	[App] Le prochain sous-but c'est cause(S,M) et donc cause(fatigue,M). Et il n'y a aucun élément de la BC qui corresponde à ce fait	R_E, R_App
Q3	[App] On a essayé symptome(pierre, fatigue), maintenant on essaie symptome(pierre,fièvre) ce qui donne un Succès avec la variable M = infection	R_App, BR >
Q3.S	[App] Je suis d'accord	RDC →
Q3.Assert0	[App] C'est donc S = fièvre	R_App
Q3.Assert1	[App] C'est quoi la deuxième condition? Ah oui, la première c'était symptôme, donc c'est cause. La deuxième condition est cause(fièvre,M) et donc M = infection	OBS_R, OBS_E, R_App, R_A_BUT
Q4	[App] Si on fait un retour ..., alors on aura encore symptôme de pierre ... hum Non, il n'y a plus de fait symptôme de pierre donc il n'y a plus de solution pour cette règle.	R_P, R_App
Q5	[Répétition de la question] [App] Que ca soit un échec ou un succès c'est considéré comme une solution non? Ok, Si on prend est_atteint_de(pierre,M), on a un succès avec M = malaria	R_App
Débriefing		

SG_I	<p>Pour les outils de l'interface ... surtout les commandes ça va. J'ai surtout eu de la difficulté avec la taille des fenêtres: il faut toujours agrandir une fenêtre pour bien lire son contenu</p> <p>Il faudrait peut être donner un visage au tuteur, des émotions, un visage souriant, ou autre selon la nature de ma réponse. Surtout pour me donner aussi des directives sur les actions à poser.</p> <p>J'ai trouvé que les questions étaient clairement formulées. Le seul problème avec le dialogue est qu'on repose la même question deux fois. Ce serait bien si on pouvait directement aborder les sous questions. Il faudrait aussi numéroter les questions et les sous questions par rapport aux grandes questions pour aider à garder le fil.</p>
SG_P	<ul style="list-style-type: none">- J'ai trouvé que le système va vraiment jusqu'au fond, un peu trop même. Il y a avait des termes et des parties que je n'ai pas maîtrisées dans le cours et là je fais la différence sur des choses subtiles.- Je comprends que les sous questions me permettent de me justifier ou de m'expliquer. Il faudrait aussi considérer une utilisation par des personnes qui ne connaissent pas du tout Prolog. Ajouter des définitions : c'est quoi un fait, une règle, une unification, etc. <p>C'est intéressant parce que malgré le fait que j'ai déjà pris le cours, je me demandais à tout moment si je vais dans la bonne direction à cause de toutes ces questions.</p>

Tableau E. 8. Transcription de l'enregistrement audio 6 avec Prolog-Tutor

Question	Transcription des réponses du participant	Interprétation
Q0	[App] est_atteint_de(pierre,M)	BR ➤
Q0.S	[App] Euhhhhh. Est ce qu'on essaie de prouver ça? Ou on essaie de prouver quelle est la variable M? Hummmm!! Je suis d'accord avec ça	HT, OBS_Q
Q0.Intellect	[App] M. La variable est M	OBS_E ☐
Q0.Assert	[App] Quelle requête? Hummm... Je me demande laquelle des affirmations est la meilleure ... Je me demande si la requête est_atteint est prouvée à partir du moment où on instancie M? Mais peut être qu'on veut aussi savoir toutes les valeurs de M qui permettent de répondre à la requête? Bof, je ne sais pas. Disons les 2 premières? En fait dans Prolog, aussitôt que tu tombes sur une valeur qui instancie M, il arrête. C'est toi qui faut qu'il lui demande si tu veux d'autres valeurs en utilisant le "point virgule". Donc tout ce qu'on veut c'est au moins une valeur de M.	OBS_Q, R_Arg, R_P
Q1	[App] J'aimerais que la fenêtre soit un peu plus grande. Qu'est ce que je dois regarder? Je dois regarder est_atteint_de(pierre,M). Dans la BC, la première information que j'ai correspond justement à cette règle.	OBS_Q, BR ➤
Q1.S	[App] Oui, je suis d'accord. C'est sûr.	RDC ➡
Q1.Intellect	[App] Je vais relire la question dans ma tête pour comprendre ce qu'on me demande. Je me demande pourquoi on me demande ça. Étudiant demande une justification de la question [App] Ben oui, je le vois bien. OK? Ahhh!!! Ok je comprends maintenant la forme d'une règle. Donc la conséquence c'est : est_atteint_de(X.,M) et les conditions : cause, symptôme(X,S)	R_NA_BUT, RP_E, R_A_BUT
Q1.INTELLECT2	[App] Euhhhh.	HT, OBS_R, RDC ➡

	<p>[Répète la question] [App] Ben oui!!</p>	
Q1.Assertf	<p>[App] Les foncteurs qui sont est_atteint_de Les arités, les arguments ... Bon ben ok, ça va. (bonne réponse) [Intervention de l'expérimentateur] T'as bien interagit. Est ce que tu as compris pourquoi le tuteur te posait toutes ces questions? [App] Ben il me montrait comment faire. Comment résoudre le but.... Ben, parce que c'est la première étape pour arriver à la réponse ... Ben parce que c'est ce que ferait un interpréteur. J'ai aussi compris qu'il a généré le sous-dialogue en réponse à mon accord avec son affirmation.</p>	R_App, R_NA_BUT
Q2	<p>[App] Ben, en fait il aurait pu commencer par me demander ça! [Intervention de l'expérimentateur] Et pourquoi à ton avis il t'a demandé toutes les questions précédentes? [App] Ben, pour être sur que je comprends bien pourquoi on utilise cette règle!! OK. J'ai mes deux variables, donc X est remplacée par pierre et M reste M. Mais en fait, c'est M que je cherche. J'ai instancié X donc je fais symptôme(pierre,S). Je tombe en premier sur symptôme(pierre, fatigue) ce qui me donne cause(fatigue, M) et cause(fatigue, M) n'est pas là.</p>	R_A_BUT, R_App
Q2.S	[App] D'accord	RDC
Q2.Assertf	<p>[Répétition de la question] [App] Le premier sous-but? Euhhhh. C'est celui-là? C'est est_atteint_de(pierre,M)? J'imagine qu'il faut que j'aille chercher symptôme(X,S). Ok, donc je fais ça avec symptôme de pierre fatigue. J'ai trouvé que S dans un premier temps avait la valeur fatigue</p>	OBS_Q, OBS_E, R_App
Q2.Assert1	<p>[App] Le sous-but maintenant c'est cause(S,M), ce qui devient cause(fatigue,M). Ah mais oui, je devrais le matcher avec des faits de la base de connaissances Malheureusement il n'y en a pas [Intervention de l'expérimentateur] [App] Ben je ne sais pas, je ne comprends pas où est ce qu'il veut en venir. J'ai</p>	R_E, R_App, R_A_BUT

	l'impression qu'il essaie juste de me donner la méthode pour trouver la bonne réponse ... Ah !!! mais c'est ça!! Il veut qu'on passe à travers le raisonnement pour voir comment ça s'est passé ... Mais pourquoi il m'a fait refaire tout ça au lieu de passer à la 2ème tentative justement? Peut être qu'il voulait savoir comment j'ai fait?	
Q3	[App] Eummmm!! symptôme(fièvre,), cause(fièvre .. Donc là j'ai juste à ... Donc M égale ... [silence] M = infection. Ahhhh! Donc je comprends que là il me fait faire le raisonnement qui m'a permis de construire ma réponse.	R_App, BR ➤
Q3.S	[App] D'accord. Ahhh! Je comprends maintenant. C'est pour être sûr que je n'ai pas dit ça comme ça par hasard ou en devinant ... Ah je comprends	RDC➡, R_A_BUT
Q3.Assert0	[App] EuuuuH. Il le fait à partir de symptôme(pierre,fièvre). Donc S j'avais dit est fièvre.	R_E, R_App
Q3.Assert1	[App] Là je vais à cause, je fais comme tantôt. Là j'ai cause(fièvre, M). Puis là c'est un succès. M =infection	DBS_E, R_App
Q4	Ben non, parce qu'il n'y a pas d'autre symptôme. Oh à moins que cause(fièvre ait une autre valeur. Ben non! Il n'y a pas d'autre valeur.	R_E, R_App
[Répétition de la question] Ah mais je n'avais pas vu le fait avec malaria, je n'ai pas pensé à agrandir la fenêtre.. Mais c'est super bien. J'avais beaucoup de difficultés à construire les règles. Mais je pense que de nous obliger à tout refaire, à justifier notre raisonnement étape par étape aide beaucoup à comprendre la démarche et surtout les fondements de Prolog : les bases de connaissances, le parcours des bases de connaissances, justement, la clé de prolog est d'arriver à décomposer ces éléments. C'est un bon exercice en tous cas...		R_App
SG_I	La base de connaissances doit être vue au complet, la fenêtre de gauche doit être moins grande car on n'y revient pas souvent contrairement à celle de la base de connaissances	

SG_P	<p>La démarche oblige à faire le raisonnement que tu effectues avant de formuler une réponse. C'est sûr qu'en Prolog il n'y a pas trente six façons de résoudre un but. Le fait de retracer les étapes va servir pour quelqu'un qui ne comprend pas très bien mais quelqu'un qui maîtrise le logiciel trouvera cela inutile. Il faudrait que ce soit un peu plus complexe pour supporter les non débutants. Par exemple : construire une base de connaissances, construire un fait, à partir de la description d'une situation en langage naturel.</p> <p>Le feedback doit permettre de faire la différence quand on a fait une erreur. Il faut qu'il pointe bien où est ce que l'erreur a lieu. Et que plutôt de donner la réponse, qu'il donne à l'apprenant la chance de réessayer.</p>
------	--

BIBLIOGRAPHIE

- Aleven, V., Popescu, O., etKoedinger, K. (2003). *Towards tutorial dialog to support self-explanation: adding natural language understanding to a cognitive tutor*. Paper presented at the Proceedings of AIED 2003.
- Anderson, J. (1983). *The Architecture of Cognition*. . Cambridge, MA: Harvard University Press.
- Anderson, J. (1993). *Rules of the Mind*. Hillsdale, NJ: Erlbaum.
- Anderson, J., Boyle, C., Farrell, R., etReiser, B. (1987). Cognitive principles in the design of computer tutors. In P. Morris (Ed.), *Modeling Cognition*. New York: John Wiley.
- Anderson, J., Corbett, A., Koedinger, K., etPelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences* 4(2), 167-207.
- Arruarte, A., Fernandez-Castro, I., etGreer, J. (1996). The CLAI Model: A Cognitive Theory of Instruction to Guide ITS Development. *Journal of artificial intelligence and education*, 7(3/4), 277-314.
- Arruarte, A., Ferrero, F., Fernández-Castro, I., Urretavizcaya, A., Álvarez, A., etGreer, J. (2003). The IRIS Authoring Tool. In T. Murray, S. Blessing & S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software*. Dordrecht: Kluwer Academic Publisher.
- Ashley, K., Desai, R., etLevine, R. (2002). *Teaching Case-Based Argumentation Concepts using Dialectic Arguments vs. Didactic Explanations*. Paper presented at the Proceedings of ITS2002.
- Baker, J. (2000). The roles of models in Artificial Intelligence and Education research: a prospective view. *International Journal of Artificial Intelligence in Education*, 11, 122-143.
- Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions*, 53, 370-418.
- Blessing, S. (2003). A Programming by Demonstration Authoring Tool for Model-Tracing Tutors. In T. Murray, S. Blessing & S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive,*

interactive, and intelligent educational software. Dordrecht.: Kluwer Academic Publishers.

- Bloom, B., S. (1968). Learning for Mastery'. In B. S. Bloom (Ed.), *All Our Children Learning*. London: McGraw-Hill.
- Bloom, B. S., Englehart, M. D., Furst, E. J., Hill, W. H., etKrathwohl, D. R. (1956). *A Taxonomy of Educational Objectives: Handbook I. Cognitive Domain*. New York: David McKay.
- Boole, G. (1854). *An Investigation of the Laws of Thought on which are founded the Mathematical Theories of Logic and Probabilities*. London: Walton and Maberley.
- Brown, J., etBurton, R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, 2, 155-191.
- Brown, J., S., Collins, A., etDuguid, P. (1988). Situated Cognition and the Culture of Learning. *Educational Researcher*, 18(1), 32-42.
- Brusilovsky, P., etPeylo, C. (2003). Adaptive and intelligent Web-based educational systems. *International Journal of Artificial Intelligence in Education*, 13(2-4), 159-172.
- Bull, S., McEvoy, T., etReid, E. (2003). *Learner Models to Promote Reflection in Combined Desktop PC / Mobile Intelligent Learning Environments*. Paper presented at the Proceedings of AIED2003, Australia.
- Bull, S., etNghiem, T. (2002). *Helping Learners to Understand Themselves with a Learner Model Open to Students, Peers and Instructors*. Paper presented at the Proceeding of the International Conference on Intelligent Tutoring Systems, ITS2002.
- Capuano, N., Marsella, M., etSalerno, S. (2000). ABITS: An Agent Based Intelligent Tutoring System for Distance Learning. In G. Gauthier, C. Frasson & K. VanLehn (Eds.), *International Conference on Intelligent Tutoring Systems* (pp. 31-38). Amsterdam: Springer.
- Carbonell, J., R. (1970). AI in CAI: an artificial intelligence approach to computer assisted instruction. *IEEE Trans. on Man-Machine*, 11, 190-202.
- Carr, B., etGoldstein, I. (1977). *Overlays: A theory of modelling for computer aided instruction* (No. Technical Report AI Memo 406). Cambridge, Mass.: MIT.
- Chi, M. T., De Leeuy, N., Chiu, M., H., etLavancher, C. (1994). Eliciting Self explanations improves understanding. *Cognitive Sciences*, 18(3), 439-477.
- Chomsky, N. (1959). A review of Skinner's. *Language*(25), 26-58.

- Cimolino, L., Kay, J., etMiller, A. (2003). *Incremental Student Modelling and Reflection by Verified Concept Mapping*. Paper presented at the Artificial Intelligence in Education, Australia.
- Cohen, J., A. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 196037(46).
- Collins, J., A., Greer, J., E., etHuang, S., X. (1996). *Adaptive Assessment Using Granularity Hierarchies and Bayesian Nets*. Paper presented at the Proceedings of the International Conference on Intelligent Tutoring Systems, Montreal, Canada.
- Colmerauer, A. (1990). An Introduction to Prolog III. *Communications of the ACM*, 33(7), 66-90.
- Conati, C., Gertner, A., etVanLehn, K. (2002). Using Bayesian Networks to Manage Uncertainty in Student Modeling. *Journal of User Modeling and User-Adapted Interaction*, 12(4), 371-417.
- Conati, C., Gertner, A., VanLehn, K., etDruzdzal, M. (1997). *On-Line Student Modeling for Coached Problem Solving Using Bayesian Networks*. Paper presented at the Proceedings of the International Conference on User Modeling, Sardinia, Italy.
- Conati, C., etVan Lehn, K. (1996). *POLA: a student modeling framework for Probabilistic On-Line Assessment of problem solving performance*. Paper presented at the Proceedings of the International Conference on User Modeling, Kailua-Kona, Hawaii.
- Conati, C., etVan Lehn, K. (2000). Toward Computer-Based Support of Meta-Cognitive Skills: a Computational Framework to Coach Self-Explanation. *International Journal of Artificial Intelligence in Education*, 11, 349-415.
- Corréard, M., Grundy, V., Ormal-Grenon, J., etPomier, N. (2004). *Le dictionnaire Hachette-Oxford compact*. Paris: Hachette.
- Cozman, F., G. (2001). The JavaBayes system. *The ISBA Bulletin*, 7(4), 16-21.
- de Koning, K., Bredeweg, B., Breuker, J., etWielinga, B. (2000). Model-Based Reasoning about Learner Behaviour. *Artificial Intelligence in Education*, 117(2), 173-229.
- Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38, 325-339.
- Descartes, R. (1637). *Discours de la méthode*. Paris: GF-Flammarion, Édition de 1966.
- Dewey, J. (1933). *How we think: A restatement of the relation of reflective thinking to the educative process*. Boston: Heath & Company.

- Dimitrova, V. (2002). STyLE-OLM: Interactive open learner modelling. *International Journal of Artificial Intelligence in Education*, 13, 37-58.
- Dimitrova, V. (2003a). *Diagnostic interactions that promote learner reflection*. Paper presented at the Proceedings of Workshop on Learner Modelling for Reflection in the 11th International Conference on Artificial Intelligence in Education, Sidney.
- Dimitrova, V. (2003b). *Diagnostic Interactions that Promote Learner's Reflection*. Paper presented at the Proceedings of AIED 2003. Workshop on learner modelling for reflection, Australia.
- Dimitrova, V., Self, J. A., etBrna, P. (2001). *Applying interactive open learner models to learning technical terminology*. Paper presented at the User Modeling.
- Drake, A., J. (1976). *Teaching Critical Thinking: Analysing, learning and teaching critical skills*. Danville: Interstate.
- Driscoll, M., P. (1994). *Psychology of Learning for Instruction*. Needham, MA.: Allyn & Bacon.
- Dufresne, A. (2001). *ExploraGraph : Improving interfaces to improve adaptive support*. Paper presented at the Proceedings of AIED2001, San Antonio, Texas.
- Evens, M., Brandle, S., Freedman, R., Glass, M., Hee-Lee, Y., Scop-shim, L., et al. (2001). *CIRCSIM-Tutor: An intelligent tutoring system using natural language dialogue*. Paper presented at the Proceedings of MAICS 2001, Oxford, OH.
- Faivre, J., Nkambou, R., etFrasson, C. (2002). *Integrative adaptive emotional agents in ITS*. Paper presented at the Proceedings of the International Conference in Intelligent Tutoring Systems, Workshop on Agents, Biarritz, France.
- Ferrero, B., Fernandez-Castro, I., etUrretavizcaya, M. (1999). *Using DETECTive, a generic diagnostic system for procedural domains*. Paper presented at the Proceedings of the International Artificial Intelligence in Education, Le Mans.
- Finin, W., T. (1989). GUMS: A General User Modeling Shell. In A. Kobsa & Wahlster.W. (Eds.), *User Models in Dialog Systems* (pp. 411-430). Berlin, Heidelberg.
- Flavell, J. (1979). Metacognition and cognitive monitoring: New area of cognitive developmental inquiry. *American Psychologist*, 34, 906-911.
- Flor, N., V., etHutchin, E., L. (1991). *Analyzing Distributed Cognition in Software Teams: A Case Study of Team Programming During Perfective Software Maintenance*. Paper presented at the Empirical Studies of Programmers, 4th Workshop.

- française", O. d. I. I. (1997). *Grand dictionnaire terminologique*. Québec: Semantix.
- Frasson, C., Mengelle, T., Aimeur, A., etGouardères, G. (1996). *An actor-based architecture for intelligent tutoring systems*. Paper presented at the Conference on Intelligent Tutoring Systems, Montreal.
- Frasson, C., Mengelle, T., Aimeur, S., etGouarderes, G. T. (1996). *An Actor Based Architecture for Intelligent Tutoring Systems*. Paper presented at the Intelligent Tutoring System, Montreal.
- Frege, G. (1879). Begriffsschrift, A formula language, modeled upon that of arithmetic, for pure thought'. In J. van Heijenoort (Ed.), *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931* (pp. 1-82). Cambridge,Mass.: Harvard University Press.
- Gagne, R. (1992). *The Conditions of Learning and Theory of Instruction*. New York: Holt, Rinehart & Winston.
- Gagne, R. (1996). *Principles of Instructional Design*. Fort Worth, TX: : HBJ College Publishers.
- Gouaderes, G., etMinko, A. (Eds.). (2000). *Simulation et Systèmes Multi-Agents pour la formation professionnelle dans le domaine aéronautique* (Vol. 6). Paris: Hermes Science.
- GrandBastien, J. (1999). Teaching expertise is at the core of ITS research. *International Journal of Artificial Intelligence and education*, 10, 335-349.
- Greeno, J. G., Collins, A. M., etResnick, L. B. (1996). Cognition and learning. In D. Berliner & R. Calfee (Eds.), *Handbook of educational psychology*. New York: Macmillan.
- Greer, J., McCalla, G., Vassileva, J., Deters, R., Bull, S., etKettel, L. (2001). Lessons Learned in Deploying a Multi-Agent Learning Support System: The I-Help Experience. In J. Moore, W. Johnson, L. & C. Redfield, L. (Eds.), *Proceedings of the International Conference on Artificial Intelligence in Education* (pp. 410-421). Amsterdam: IOS Press.
- Gruninger, M., etFox, M., S. (1995). *Methodology for the Design and Evaluation of Ontologies*. Paper presented at the International Joint Conference on Artificial Intelligence. Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal.
- Heffernan, N., etKoedinger, K. (2002). *An Intelligent Tutoring System Incorporating a Model of an Experienced Human Tutor* International Conference on Intelligent Tutoring System Biarritz, France. .

- Hoc, J., M. (1990). Les activités de diagnostic. In J.-F. Richard, C. Bonnet & R. Ghiglione (Eds.), *Traité de Psychologie cognitive Tome 2*. Paris: Dunod.
- Hospers, M., Kroezen, A., Nijholt, R., etHeylen, D. (2003). Developing a generic agent-based intelligent tutoring system and applying it to nurse education. In *Proceedings of the IEEE International Conference on Advanced Language Technologies* (pp. 443-444). Athens, Greece.
- Hume, D. (1748). *Enquiry concerning Human Understanding*: Hackett Publ Co. 1993.
- Ikeda, M., etMizogouchi, R. (1994). FITS: A Framework for ITS - A Computational Model of Tutoring. *Journal of Artificial Intelligence in Education*, 5(3), 319-348.
- Iqbal, A., Oppermann, R., Patel, A., etKinshuk. (1999). A Classification of Evaluation Methods for Intelligent Tutoring Systems. In U. Arend, E. Eberleh & K. Pitschke (Eds.), *Software Ergonomie '99 - Design von Informationswelten* (pp. 169-181). Leipzig: B. G. Teubner Stuttgart.
- Kaptelinin, V. (1996). Computer-mediated activity: functional organs in social and developmental contexts. In B. Nardi, A. (Ed.), *Context and consciousness: Activity theory and Human-Computer Interaction*. Cambridge, MA.: MIT Press.
- Kay, J. (1997). *Learner Know Thyself: Student Models to give Learner Control and Responsibility*. Paper presented at the Proceedings of ICCE97, Malaysia.
- Kobsa, A., etPohl, P. (1995). The user modeling shell system BGP-MS. *User Modeling and User Adapted Interaction*, 4, 59-106.
- Koedinger, K., Anderson, J., Hadley, W., etMark, M. (1997). Intelligent Tutoring Goes To School in the Big City. *International Journal of Artificial Intelligence in Education*, 8, 30-43.
- Koedinger, K., Suthers, D., etForbus, K. (1999). Component-based construction of a science learning space. *International Journal of Artificial Intelligence in Education*, 10, 292-313.
- Kolb, D. (1984). *Experiential Learning: Experience as the Source of Learning and Development*. Englewood Cliffs, NJ: Prentice-Hall Inc.
- Larousse. (2004). *Larousse de poche 2005*. Paris: Larousse.
- Lave, J. (1988). *Cognition in practice, Mind, Mathematics and culture in every day life*. New York: Cambridge University Press.
- Leibniz, W., F. (1714). *La monadologie*. Paris: Le Livre de Poche, 1990.

- Leontiev, A. (1984). *Activité, conscience et personnalité*. Moscou: Editions du progrès.
- Lewin, K. (Ed.). (1948). *Resolving social conflicts; selected papers on group dynamics*. New York: Harper & Row.
- Livergood, L., N. (1991). From Computer-Assisted Instruction to Intelligent Tutoring Systems. *Journal of Artificial Intelligence in Education*, 2(3), 39-50.
- Locke, J. (1690). *An Essay Concerning Human Understanding*. Cleveland: Meridian Books, 1964.
- Lovett, M., C. (1998). *Cognitive task analysis in service of intelligent tutoring system design: A case study in statistics*. Paper presented at the Intelligent Tutoring Systems, Proceedings of the Fourth International Conference., Montréal.
- Makatchev, M., Jordan, P., etVanLehn, K. (2004). Abductive Theorem Proving for Analyzing Student Explanations and Guiding Feedback in Intelligent Tutoring Systems. *Journal of Automated Reasoning and Theorem Proving in Education*, 32(3), 187-226.
- Mark, M., A., etGreer, J. E. (1993). Evaluation methodologies for intelligent tutoring systems. *Journal of Artificial Intelligence in Education*, 4(2/3), 129-153.
- Martin, J., etVanLehn, K. (1995). A bayesian approach to cognitive assessment. In P. Nichols, S. Chipman & R. L. Brennan (Eds.), *Cognitively diagnostic assessment*. Hillsdale, NJ: LEA.
- Maturana, H., etVarela, J., F. (1980). *Autopoiesis and Cognition: The Realization of the Living*. Boston: D. Reidel.
- Mengelle, T., de Léan, C., etFrasson, C. (1998). *Teaching and Learning with Intelligent Agents: Actors*. Paper presented at the Intelligent Tutoring Systems, San Antonio.
- Merrill, D. (2003). Using Knowledge Objects to Design Instructional Learning Environments. In T. Murray, S. Blessing & S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software*. Dordrecht: Kluwer Academic Publishers.
- Merrill, M., D. (1991). Constructivism and instructional design. *Educational Technology*, 31(5), 45-53.
- Merrill, M., D., etID2-Research-Group. (1998). ID Expert: A Second Generation Instructional Development System. *Instructional Science*, 26, 243-262.

- Miles, M., M., etHuberman, A., M. (2003). *Analyse des données qualitatives, traduction de la 2e édition américaine*. Bruxelles: De Boeck.
- Miller, G., A. (1962). Some psychological studies of grammar. *American Psychologist*, 17, 748-762.
- Minsky, M. (1975). A framework for representing knowledge. In P. Winston (Ed.), *The Psychology of Computer Vision*. New York: McGraw-Hill.
- Mislevy, R. (1994). Evidence and Inference in Educational Assessment. *Psychometrika*, 59, 439-483.
- Mitrovic, A. (2003). An Intelligent SQL Tutor on the Web. *International Journal of Artificial Intelligence in Education*, 13(2-4), 171-195.
- Mitrovic, A., etMartin, B. (2002). *Evaluating the Effects of Open Student Models on Learning*. Paper presented at the Adaptive Hypermedia and Adaptive Web-Based Systems.
- Morrison, G., Ross, S., etKemp, J., E. (2004). *Designing effective instruction* (4th ed.). New York: John Wiley & Sons.
- Murray, T. (1993). Formative Qualitative Evaluation for "Exploratory" ITS research. *Journal of Artificial Intelligence in Education*, 4(2/3), 179-207.
- Murray, T. (2003a). Eon: Authoring Tools for Content, Instructional Strategy, Student Model, and Interface Design. In T. Murray, S. Blessing & S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software*. Dordrecht.: Kluwer Academic Publishers.
- Murray, T. (2003b). An overview of intelligent tutoring systems authoring tools: Updated Analysis of the State of the Art. In T. Murray, S. Blessing & S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software*. Dordrecht.: Kluwer Academic Publishers.
- Nardi, B., A. (1996). Studying Context: a comparison of activity theory, situated cognition, and distributed cognition. In B. Nardi, A. (Ed.), *Context and Consciousness: Activity Theory and Human-Computer Interaction*. Cambridge,MA.: MIT Press.
- Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA.: Harvard University Press.

- Newell, A., Rosenbloom, P., S., etLaird, J., E. (1989). Symbolic architectures for cognition. In M. Posner (Ed.), *Foundations of Cognitive Sciences* (pp. 93-131). Cambridge,MA.: MIT Press.
- Newell, A., etSimon, H., A. (1972). *Human Problem Solving*. Englewood Cliffs: Prentice-Hall.
- Nkambou, R., Frasson, C., etGauthier, G. (1996). *CREAM-Tools: An Authoring Environment for Curriculum and Course Building in an Intelligent Tutoring System*. Paper presented at the Proceedings of the International Conference on Computer Aided Learning and Instruction in Science Engineering, San Sebastian, Spain.
- Nkambou, R., Frasson, C., etGauthier, G. (2003a). CREAM-Tools: An Authoring Environment for Knowledge Engineering in Intelligent Tutoring Systems. In T. Murray, S. Blessing & S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software*. Dordrecht: Kluwer Academic Publisher.
- Nkambou, R., Frasson, C., etGauthier, G. (2003b). CREAM-Tools: An Authoring Environment for Knowledge Engineering in Intelligent Tutoring Systems. In T. Murray, S. Blessing & S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software* (pp. 93-138). Dordrecht: Kluwer Academic Publisher.
- Nkambou, R., etKabanza, F. (2001). Designing Intelligent Tutoring Systems: A multiagent Planning Approach. *ACM SIGCUE Outlook*, 27(2), 46-60.
- Nkambou, R., etTchetagni, J. (2002). *Using Bayesian Inference and Neural Network Classification for Student Modeling in a Tutoring System*. Paper presented at the Proceedings of the International Conference on Artificial Intelligence and Applications, Malaga.
- Nkambou, R., etTchetagni, J. (2004). *Diagnosing Learner Errors in a Web Based environments using the MPE*. Paper presented at the Proceedings of WBE2004, Anaheim.
- Noel, B. (1997). *La métacognition*. Bruxelles: De Boeck Université.
- Nwana, H. (Ed.). (1993). *Mathematical Intelligent Learning Environmen*. Oxford: Intellect.
- Ohlsson, S. (1987). Some principles of intelligent tutoring. In R. W. Lawler & Yazdani.M. (Eds.), *Learning Environments and Tutoring Systems: Learning Environments and Tutoring Systems* (Vol. 1, pp. 203-237). Norwood, New Jersey: Ablex Publishing.

- Ohlsson, S. (1994). Constraint-Based Student Modeling. *Journal of Artificial Intelligence in Education*, 3(4), 429-447.
- Paiva, A., etSelf, J. (1995). TAGUS - A User and Learner Modeling Workbench. *User Modeling and User-Adapted Interaction*, 4(3), 197-226.
- Papert, S. (Ed.). (1999). *Logo Philosophy and Implementation*. Montreal: LCSI.
- Paquette, G. (2002). *La modélisation des connaissances et des compétences, pour concevoir et apprendre*: Presses de l'Université du Québec.
- Pavlov, I. P. (1927). *Conditioned reflexes*. London: Routledge and Kegan Paul. London: Routledge and Kegan Paul.
- Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA.: Morgan Kaufman Publishers.
- Pearl, J., etRussell, S. (2003). Bayesian Networks. In M. Arbib, A. (Ed.), *Handbook of Brain Theory and Neural Networks* (pp. 157-160). Cambridge, MA: MIT Press.
- Person, N., Graesser, A., Kreuz, R., Pomeroy, V., etGroup, T. R. (2001). Simulating human tutor dialog moves in AutoTutor. *International Journal of Artificial Intelligence in Education*, 12, 23-39.
- Piaget, J. (1932). *The Moral Judgement of the Child*. New York: Harcourt, Brace Jovanovich.
- Pohl, W. (1999). Logic-based representation and reasoning for user modeling shell systems. *User Modeling and User-Adapted Interaction*, 9(3), 217-282.
- Pylyshyn, Z. (1984). *Computation and Cognition*. Cambridge: Braqundford/MIT Press.
- Pylyshyn, Z. (1989). Computing in Cognitive Sciences. In M. Posner (Ed.), *Foundations of Cognitive Sciences* (pp. 52-91). Cambridge (MA): MIT Press.
- Reigeluth, C., M. (1999). *Instructional Design Theories and Models: A New Paradigm of Instructional Theory, Volume II*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.
- Reye, J. (1998). *Two-Phase Updating of Student Models Based on Dynamic Belief Networks*.
- Reye, J. (2004). Student Modelling based on Belief Networks. *International Journal of Artificial Intelligence in Education*, 14, 63-96.
- Ritter, R. (1997). *Communication, Cooperation and Competition among multiple tutor agents*. Paper presented at the International Conference on Artificial Intelligence in Education, Amsterdam.

- Ritter, S., etKoedinger, K., R. (1996). An architecture for plug-in tutor agents. *Journal of artificial intelligence and education*, 7(3/4), 315-347.
- Rosé, C., etTorrey, C. (2004). *DReSDeN: Towards a Trainable Tutorial Dialogue Manager to Support Negotiation Dialogues for Learning and Reflection*. Paper presented at the Proceedings of ITS2004.
- Rosé, P., Jordan, P., Ringenber, M., Siler, S., VanLehn, K., etWeinstein, A. (2001). *Interactive Conceptual Tutoring in Atlas-Andes*. Paper presented at the Proceedings of AIED2001.
- Rueda, U., Larrañaga, M., Ferrero, B., Arruarte, A., etElorriaga, A. (2003). *Study of Graphical Issues in a Tool for Dynamically Visualising Student Models*. Paper presented at the Artificial Intelligence in Education.
- Rumelhart, D., E. (1989). The architecture of mind: a connectionist approach. In M. Posner (Ed.), *Foundations of Cognitive Sciences* (pp. 136-159). Cambridge (MA): MIT Press.
- Russel, B., etWhitehead, N., A. (1910-1913). *Principia Mathematica*. Cambridge: University Press.
- Russel, S., etNorvig, P. (2003). *Artificial Intelligence, A Modern Approach* (2nde ed.): Prentice Hall.
- Scapin, D., L., etBastien, J., M.,C. (1997). Ergonomic criteria for evaluating the ergonomic quality of interactive systems. *Behaviour & Information Technology*, 16, 220-231.
- Schön, D., A. (1983). *The reflective practitioner. How professionals think in action*. London: TempleSmith.
- Self, J. (1994). Formal Approaches to Student Modeling. In *Student Models: The Key to Individualized Educational Systems*. New York: Springer Verlag.
- Self, J. (1999). The defining characteristics of intelligent tutoring systems research: ITSs care, precisely. *International Journal of Artificial Intelligence and education*, 10, 350-364.
- Shafer, G. A. (1976). *Mathematical Theory of Evidence*. New Jersey: Princeton University Press.
- Shute, V., Lajoie, S., etGluck, K. (2000). Individualized and group approaches to training. In S. Tobias & J. Fletcher, D. (Eds.), *Training and retraining: A handbook for business, industry, government and the military* (pp. 171-207). New York: McMillan.
- SICSTUS-Prolog. (2005). <http://www.sics.se/isl/sicstuswww/site/index.html>.

- Skinner, B., F. (1938). *The behavior of organisms: An experimental analysis*. Copley: Acton, MA.
- Skinner, B., F. (1974). *About behaviorism*. New York: Vintage.
- Sleeman, D., etBrown, J., S. (Eds.). (1982). *Intelligent Tutoring Systems*. New York: Academic Press.
- Solloway, E., etBielaczyc, K. (1995). Interactive Learning Environments: Where They've Come From and Where They're Going. In *Proceedings of CHI95* (pp. 347-348).
- Sowa, J. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Toronto: Brooks/Cole.
- Tchetagni, J., etNkambou, R. (2002). *A framework for the hierarchical representation of the learner model using Bayesian Networks*, Biarritz, France.
- Tchetagni, J., etNkambou, R. (2004a). *Diagnosing Learner Errors in a Web Based environments using the MPE*. Paper presented at the International Conference on Web-Based Educatio, Anheim.
- Tchetagni, J., etNkambou, R. (2004b). *A framework for epistemological diagnosis and remediation in learning Logical Programming*. Paper presented at the Proceedings of CALIE04, Grenoble.
- Tchetagni, J., etNkambou, R. (2004c). *A framework for epistemological diagnosis and remediation in learning Logical Programming*, Grenoble.
- Tchetagni, J., etNkambou, R. (2005). *"Elaboration of Guidance" agent in the interactive remediation of Logic Programming exercises: an approach to student diagnosis revision*. Paper presented at the 18th. Conference of the Canadian Society for Computational Studies of Intelligence. Workshop on Formal AI Techniques in Computer-Based Training.
- Tchetagni, J., Nkambou, R., etBourdeau, J. (2005a). *Supporting Student Reflection in an Intelligent Tutoring System for Logic Programming*. Paper presented at the International Conference on Artificial Intelligence in Education (To appear).
- Tchetagni, J., Nkambou, R., etBourdeau, J. (2005b). A Tool to Specify a Cognitive Diagnosis Component in ILE. *Journal of Interactive Learning Research*, À apparaître.
- Tchetagni, J., Nkambou, R., etKabanza, F. (2004). *Epistemological Remediation in Intelligent Tutoring Systems*. Paper presented at the Proceedings of IEA/AIE2004.
- Thorndike, E., L. (1912). *Education: A first book*. New York: Macmillan Company.

- Thorndike, E., L. (1932). *The Fundamentals of Learning*. New York: AMS Press.
- Towne, D. (2003). Automated Knowledge Acquisition for Intelligent Support of Diagnostic Reasoning. In K. A. Publishers (Ed.), *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software* Dordrecht.: Kluwer Academic Publishers.
- Tsovaltzi, D., etMatheson, C. (2002). *Formalising hinting in tutorial dialogues*. Paper presented at the EDILOG: 6th workshop on the semantics and pragmatics of dialogue, Edinburgh.
- Turing, A. (1950). Computing Machinery and Intelligence. *Mind*, 49, 433-460.
- Van Lehn, K. (1988). Student Modelling. In M. Polson, C. & J. Richardson, J. (Eds.), *Foundations of Intelligent Tutoring Systems*. London: Laurence Erlbaum.
- Van Marcke, K. (1998). GTE: An epistemological approach to instructional modeling. *Instructional Science*, 26, 147-191.
- Vassileva, J., McCalla, G., etGreer, J. (2003). Multi-Agent Multi-User Modelling in I-Help. *User Modelling and User Adapted Interaction*, 13(1), 179-210.
- Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13, 260-267.
- Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA.: Harvard University Press. Originellement publié en Russe en 1930.
- Wasson, B. (1998). Facilitating dynamic pedagogical decision making: PEPE and GTE. *Learning Sciences*, 26(3/4), 299-316.
- Webber, C. (2004). *From Errors to Conceptions - An approach to student diagnosis*. Paper presented at the 7th Conference on Intelligent Tutoring Systems, Maceios, Alagoas, Brazil.
- Webber, C., Pesty, S., etBalacheff, N. (2002). A multi agent and emergent approach to learner modelling. In F. van Harmelen (Ed.), *Proceedings of the European Conference on Artificial Intelligence* (pp. 98-102). Lyon: IOS Press.
- Weber, G., etBrusilovsky, P. (2001). ELM-ART: An adaptive versatile system for Web-based instruction. *International Journal of Artificial Intelligence in Education*, 12(4), 351-384.

- Wenger, E. (1987a). *Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge*. Morgan Kaufman Press.
- Wenger, E. (1987b). *Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge*. San Diego, California: Morgan Kaufman Press.
- Wertheimer, M. (1922). Laws of organization in perceptual forms. In W. Ellis (Ed.), *A source book of Gestalt psychology* (pp. 71-88). London: Routledge & Kegan Paul, 1938.
- Zadeh, L. (1983). The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems. *Fuzzy Sets and Systems*, 11(199-227).
- Zapata-Rivera, J. D., etGreer, J. (2002). *Exploring Various Guidance Mechanisms to Support Interaction with Inspectable Learner Models*. Paper presented at the Intelligent Tutoring Systems ITS 2002.
- Zapata-Rivera, J.-D., etGreer, J. (2003). *Analyzing Student Reflection in The Learning Game*. Paper presented at the Proceedings of AIED2003, Australia.

