



**HAL**  
open science

**Acquisition de connaissances et apprentissage  
automatique: contribution pour le developpement  
incremental d'un systeme à base de connaissances pour  
les situations de crise - application au domaine de l'eau**

Redouane Senoune

► **To cite this version:**

Redouane Senoune. Acquisition de connaissances et apprentissage automatique: contribution pour le developpement incremental d'un systeme à base de connaissances pour les situations de crise - application au domaine de l'eau. Web. INSA de Lyon, 1995. Français. NNT: 1995ISAL0044 . tel-00822335

**HAL Id: tel-00822335**

**<https://theses.hal.science/tel-00822335>**

Submitted on 14 May 2013

**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.

# THESE

présentée

DEVANT L'INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE LYON

pour obtenir

LE GRADE DE DOCTEUR

FORMATION DOCTORALE : INGENIERIE INFORMATIQUE

PAR

**Redouane SENOUNE**

(Ingénieur en Informatique)

---

**Acquisition de Connaissances et Apprentissage Automatique : contribution  
pour le développement incrémental d'un Système à Base de Connaissances  
pour les situations de crise  
- application au domaine de l'eau -**

---

**Soutenue le 19 juin 1995 devant la commission d'Examen**

<b>Jury MM.</b>	<b>J. KOULOUMDJIAN</b>	<b>Examineur</b>
	<b>A. FATHOLAHZADEH</b>	<b>Examineur</b>
	<b>J. QUINQUETON</b>	<b>Rapporteur</b>
	<b>E. CHOURAQUI</b>	<b>Rapporteur</b>
	<b>A. HOCINE</b>	<b>Examineur</b>
	<b>C. DE SAINTE MARIE</b>	<b>Examineur</b>
	<b>A. MATHON</b>	<b>Directeur de Thèse</b>
	<b>D. GRAILLOT</b>	<b>Examineur</b>



# THESE

présentée

DEVANT L'INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE LYON

pour obtenir

LE GRADE DE DOCTEUR

FORMATION DOCTORALE : INGENIERIE INFORMATIQUE

PAR

Redouane SENOUNE

(Ingénieur en Informatique)

---

**Acquisition de Connaissances et Apprentissage Automatique : contribution  
pour le développement incrémental d'un Système à Base de Connaissances  
pour les situations de crise  
- application au domaine de l'eau -**

---

Soutenue le 19 juin 1995 devant la commission d'Examen

Jury MM.	J. KOULOUMDJIAN	Examineur
	A. FATHOLAHZADEH	Examineur
	J. QUINQUETON	Rapporteur
	E. CHOURAQUI	Rapporteur
	A. HOCINE	Examineur
	C. DE SAINTE MARIE	Examineur
	A. MATHON	Directeur de Thèse
	D. GRAILLOT	Examineur



JANVIER 1995

## INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE LYON

Directeur : J.ROCHAT

Professeurs :

S. AUDISIO	PHYSICOCHIMIE INDUSTRIELLE
J.C. BABOUX	GEMPPM
J. BAHUAUD	MECANIQUE DES SOLIDES
B. BALLAND	PHYSIQUE DE LA MATIERE
D. BARBIER	PHYSIQUE DE LA MATIERE
G. BAYADA	EQUIPE MOD. MATH. CALC. SC.
C. BERGER (Melle)	CETHIL/EQ. PHYSIQUE INDUST
M. BETEMPS	AUTOMATIQUE INDUSTRIELLE
J.M. BLANCHARD	CHIMIE PHYSIQUE APPLIQUEE ET ENVIRONNEMENT
C. BOISSON	VIBRATIONS ACOUSTIQUES
M. BOIVIN	MECANIQUE DES SOLIDES
H. BOTTA	EQUIPE DEVELOPPEMENT URBAIN
G. BOULAYE	L.I.S.I.
J. BRAU	CETHIL/EQ. EQUIPEMENT DE L'HABITAT
M. BRUNET	MECANIQUE DES SOLIDES
J.C. BUREAU	THERMOCHIMIE MINERALE
J.P. CHANTE	COMPOSANTS DE PUISSANCE ET APPLICATIONS
B. CHOCAT	METHODES
B. CLAUDEL	CHIMIE PHYSIQUE APPLIQUEE ET ENVIRONNEMENT
M. COUSIN	BETONS ET STRUCTURES
L. CRONENBERGER	CHIMIE BIOLOGIQUE
M. DIOT	THERMOCHIMIE MINERALE
A. DOUTHEAU	CHIMIE ORGANIQUE
J.C. DUPUY	PHYSIQUE DE LA MATIERE
H. EMPTOZ	MOD. SYST. ET REC. DES FORMES
C. ESNOUF	GEMPPM*
G. FANTOZZI	GEMPPM*
J. FAUCHON	C.A.S.M.
J. FAVREL	L.I.S.P.I.
G. FERRARIS-BESSO	MECANIQUE DES STRUCTURES
Y. FETIVEAU	GENIE ELECTRIQUE ET FERROELECTRICITE
L. FLAMAND	MECANIQUE DES CONTACTS
P. FLEISCHMANN	GEMPPM*
A. FLORY	L.I.S.I.
R. FOUGERES	GEMPPM*
L. FRECON	D.E.L.I.A.
R. GAUTHIER	PHYSIQUE DE LA MATIERE
M. GERY	CETHIL/EQ. EQUIPEMENT DE L'HAB.
G. GIMENEZ	TRAITEMENT DU SIGNAL ET ULTRASONS
P. GOBIN	GEMPPM*
P. GONNARD	GENIE ELECTRIQUE ET FERROELECTRICITE
R. GOUTTE	TRAITEMENT DU SIGNAL ET ULTRASONS
G. GRANGE	GENIE ELECTRIQUE ET FERROELECTRICITE
G. GUENIN	GEMPPM*
G. GUILLOT	PHYSIQUE DE LA MATIERE
A. GUINET	L.I.S.P.I.
C. GUITTARD	D.E.L.I.A.
J.L. GUYADER	VIBRATIONS-ACOUSTIQUE
J.P. GUYOMAR	GENIE ELECTRIQUE ET FERROELECTRICITE
J.M. JOLION	L.I.S.P.I.
J. JOUBERT	GENIE MECANIQUE
J.F. JULLIEN	BETONS ET STRUCTURES
A. JUTARD	AUTOMATIQUE INDUSTRIELLE
R. KASTNER	GEOTECHNIQUE
H. KLEIMANN	GENIE ELECTRIQUE ET FERROELECTRICITE
J. KOULOUMDJIAN	L.I.S.I.
M. LAGARDE	CHIMIE BIOLOGIQUE

JANVIER 1995

M. LALANNE	MECANIQUE DES STRUCTURES
A. LALLEMAND	CETHIL/EQ. ENERGETIQUE ET THERMIQUE
M. LALLEMAND (Mme)	CETHIL/EQ. ENERGETIQUE ET THERMIQUE
P. LAREAL	GEOTECHNIQUE
A. LAUGIER	PHYSIQUE DE LA MATIERE
Ch. LAUGIER	PHYSIOLOGIE ET PHARMACODYNAMIE
P. LEJEUNE	G.M.M.O.
C. LESUEUR	VIBRATIONS-ACOUSTIQUE
A. LUBRECHT	MECANIQUE DES CONTACTS
Y. MARTINEZ	L.I.S.P.I.
H. MAZILLE	PHYSICOCHIMIE INDUSTRIELLE
P. MERLE	GEMPPM
J. MERLIN	GEMPPM*
J.P. MILLET	PHYSICOCHIMIE INDUSTRIELLE
M. MIRAMOND	METHODES
N. MONGEREAU	GEOTECHNIQUE
R. MOREL	MECANIQUE DES FLUIDES
P. NARDON	BIOLOGIE APPLIQUEE
A. NAVARRO	CHIMIE PHYSIQUE APPLIQUEE ET ENVIRON.
M. OTTERBEIN	CHIMIE PHYSIQUE APPLIQUEE ET ENVIRON.
J.P. PASCAULT	MATERIAUX MACROMOLECULAIRES
J. PERA	MATERIAUX MINERAUX
G. PERACHON	THERMOCHIMIE MINERALE
M. PERDRIX	TRAITEMENT DU SIGNAL ET ULTRASONS
J. PEREZ	GEMPPM*
P. PINARD	PHYSIQUE DE LA MATIERE
J.M. PINON	L.I.S.I.
D. PLAY	C.A.S.M.
J. POUSIN	EQUIPE MOD. MATH. ET CALCUL SCIEN.
P. PREVOT	L.I.S.P.I.
R. PROST	TRAITEMENT DU SIGNAL ET ULTRASONS
M. RAYNAUD	CETHIL/EQUIPE GMC
J.M. REYNOUARD	BETONS ET STRUCTURES
J. ROBERT-BAUDOY (Mme)	G.M.M.O.
D. ROUBY	GEMPPM*
P. RUBEL	L.I.S.I.
J.F. SACADURA	CETHIL/EQUIPE GMC
H. SAUTEREAU	MATERIAUX MACROMOLECULAIRES
S. SCAVARDA	AUTOMATIQUE INDUSTRIELLE
D. THOMASSET	AUTOMATIQUE INDUSTRIELLE
M. TROCCAZ	GENIE ELECTRIQUE ET FERROELECTRICITE
R. UNTERREINER	TRAITEMENT DU SIGNAL ET ULTRASONS
J. VERON	CHIMIE PHYSIQUE APPLIQUEE ET ENVIRON.
A. VINCENT	GEMPPM*
G. VIGIER	GEMPPM
P. VUILLERMOZ	PHYSIQUE DE LA MATIERE

**Directeurs de recherche C.N.R.S. :**

P. CLAUDY	THERMOCHIMIE MINERALE
M. MURAT	GEMPPM*
A. NOUAILHAT	PHYSIQUE DE LA MATIERE
M.A. MANDRAND (Mme)	GENETIQUE MOLECULAIRE DES MICROORGANISMES

**Directeurs de recherche I.N.R.A. :**

G. BONNOT	BIOLOGIE
S. GRENIER	BIOLOGIE
Y. MENEZO	BIOLOGIE

**Directeurs de recherche I.N.S.E.R.M. :**

A-F. PRIGENT (Mme)	CHIMIE BIOLOGIQUE
N. SARDA (Mme)	CHIMIE BIOLOGIQUE

\* GROUPE D'ETUDE METALLURGIE PHYSIQUE ET PHYSIQUE DES MATERIAUX

## *Remerciements*

Les travaux présentés dans ce mémoire ont été effectués au sein de l'équipe Ingénierie de l'Environnement du département Stratégie de Développement (STRAD) à l'Ecole Nationale Supérieure des Mines de Saint-Etienne.

J'adresse mes remerciements au Professeur Albert Mathon, actuellement Directeur adjoint chargé de l'enseignement et de la formation à l'Ecole des Mines de Saint Etienne, ancien Directeur du département STRAD, pour m'avoir accueilli dans son laboratoire. Qu'il soit également remercié pour la confiance qu'il m'a accordée et pour ses encouragements.

Je tiens à témoigner ma profonde reconnaissance à Monsieur Didier Graillot, Maître de recherche et responsable de l'équipe Ingénierie de l'Environnement au sein du département STRAD, dans un premier temps, et dans le centre SIMADE par la suite, pour avoir dirigé mes travaux et pour m'avoir initié au domaine de l'environnement. Ses nombreux conseils, tant sur le fond que sur la forme, ont toujours été sources d'idées nouvelles.

Je remercie Monsieur J. Quinqueton, Directeur de Recherche au Laboratoire d'Informatique, de Robotique et de Micro-électronique de Montpellier, et Monsieur E. Chouraqui, Directeur du Département de recherche en Informatique, Automatique et Mécatronique de Marseille, pour avoir accepté d'être rapporteur de cette thèse et de participer au jury.

Mes remerciements vont également à Messieurs J. Kouloumdjian, Professeur à l'Institut National des Sciences Appliquées de Lyon, A. Hocine, Maître de Conférences à l'université de Pau, A. Fatholahzadeh, Professeur à l'Ecole Supérieure d'Electricité de Metz, et C. De Sainte Marie, Ingénieur à la société ITMI (Grenoble), pour avoir accepté d'examiner cette thèse et pour tout l'intérêt qu'ils y ont porté.

Je tiens à remercier également Monsieur D. Peter du Ministère de l'Environnement (Direction de l'Eau) pour l'intérêt qu'il a manifesté à l'égard de ce travail et pour ses judicieux conseils et encouragements.



Je remercie Ph. Beaune pour la patience et la générosité avec laquelle il a su apporter des remarques constructives à ce travail.

J'associe mes remerciements à tous mes collègues de SIMADE pour leur sympathie et leur aide. Merci à tous ceux qui ont pris le temps de corriger ce manuscrit.

Qu'il me soit permis d'exprimer toute ma reconnaissance et mes remerciements à Marie Line Barnéoud, Zahia Mazer et Bernadette Zold pour l'aide qu'elles m'ont apportée durant la réalisation de ce travail.

Je tiens aussi à remercier le personnel du service de reprographie de l'EMSE qui s'est toujours montré aimable et disponible.

Je remercie ma famille, mes amis et tous mes proches pour leur compréhension et leurs encouragements.

Enfin je remercie ma femme pour sa patience, son soutien et son amour.

\*\*\*\*\*

*" Il n'est désir plus naturel que le désir de connaissance. Nous essayons tous les moyens qui nous y peuvent mener. Quand la raison nous faut, nous y employons l'expérience, qui est un moyen plus faible et moins digne; mais la vérité est chose si grande, que nous ne devons dédaigner aucune entremise qui nous y conduise.*

*La raison a tant de formes, que nous ne savons à laquelle nous prendre, l'expérience n'en a pas moins. La conséquence que nous voulons tirer de la ressemblance des événements est mal sûre, d'autant qu'ils sont toujours dissemblables."*

(MONTAIGNE, Essais)

*"Pour commander la nature, il faut obéir à ses lois."*

(F. Bacon)

## **Résumé**

Cette thèse a pour objectif la réalisation d'un Système à Base de Connaissances pour les situations de crise. A travers le développement de ce système, deux principaux axes de recherche ont été entrepris : l'acquisition et la validation de connaissances.

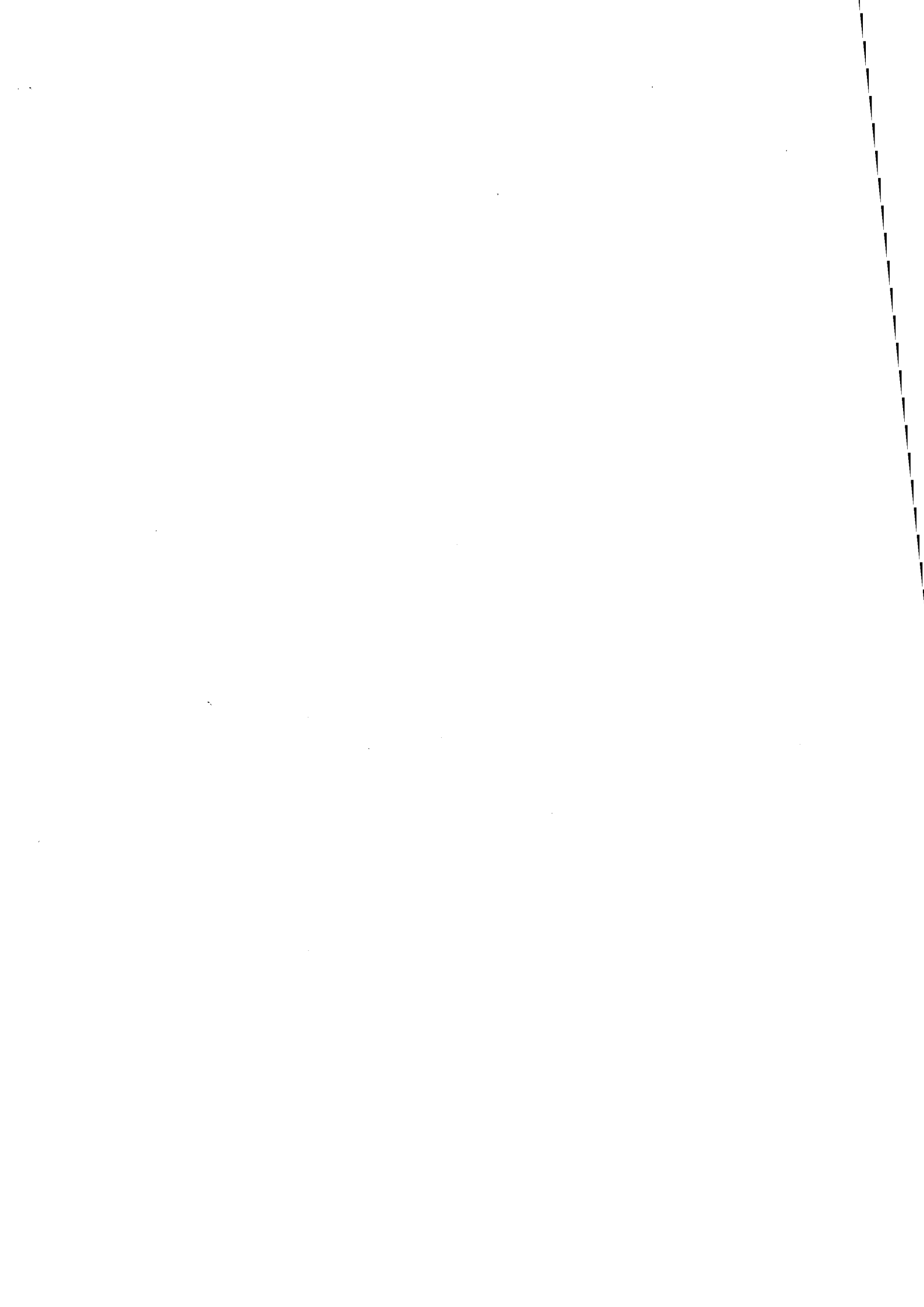
Pour l'étape d'acquisition de connaissances, nous avons intégré une méthodologie d'acquisition de connaissances et une technique d'apprentissage automatique. Dans un premier temps, la méthodologie d'acquisition de connaissances nous a permis de recueillir l'ensemble des connaissances descriptives et stratégiques du domaine, et de construire un langage de description des exemples d'apprentissage. Une technique d'apprentissage est ensuite utilisée pour construire incrémentalement un graphe de connaissances en utilisant des cas d'interventions sur des situations de crise fournis par les experts du domaine.

Pour la phase d'exploitation du système, nous avons proposé deux procédures différentes. La première procédure consiste en l'utilisation interactive du graphe de connaissances construit et la deuxième procédure consiste en l'utilisation en déduction des connaissances contenues dans ce graphe.

L'approche proposée pour la validation des connaissances s'appuie sur l'utilisation interactive du graphe de connaissances construit et sur un suivi des interventions des experts sur des cas de crise.

## **Mots clés**

Base Connaissance, Acquisition, Apprentissage, Aide Décision, validation, Crise.



## **Abstract**

The subject of this thesis is the development of a Knowledge-Based System for situations of crisis. Two main research issues have been studied during the development of the system : knowledge acquisition and knowledge validation.

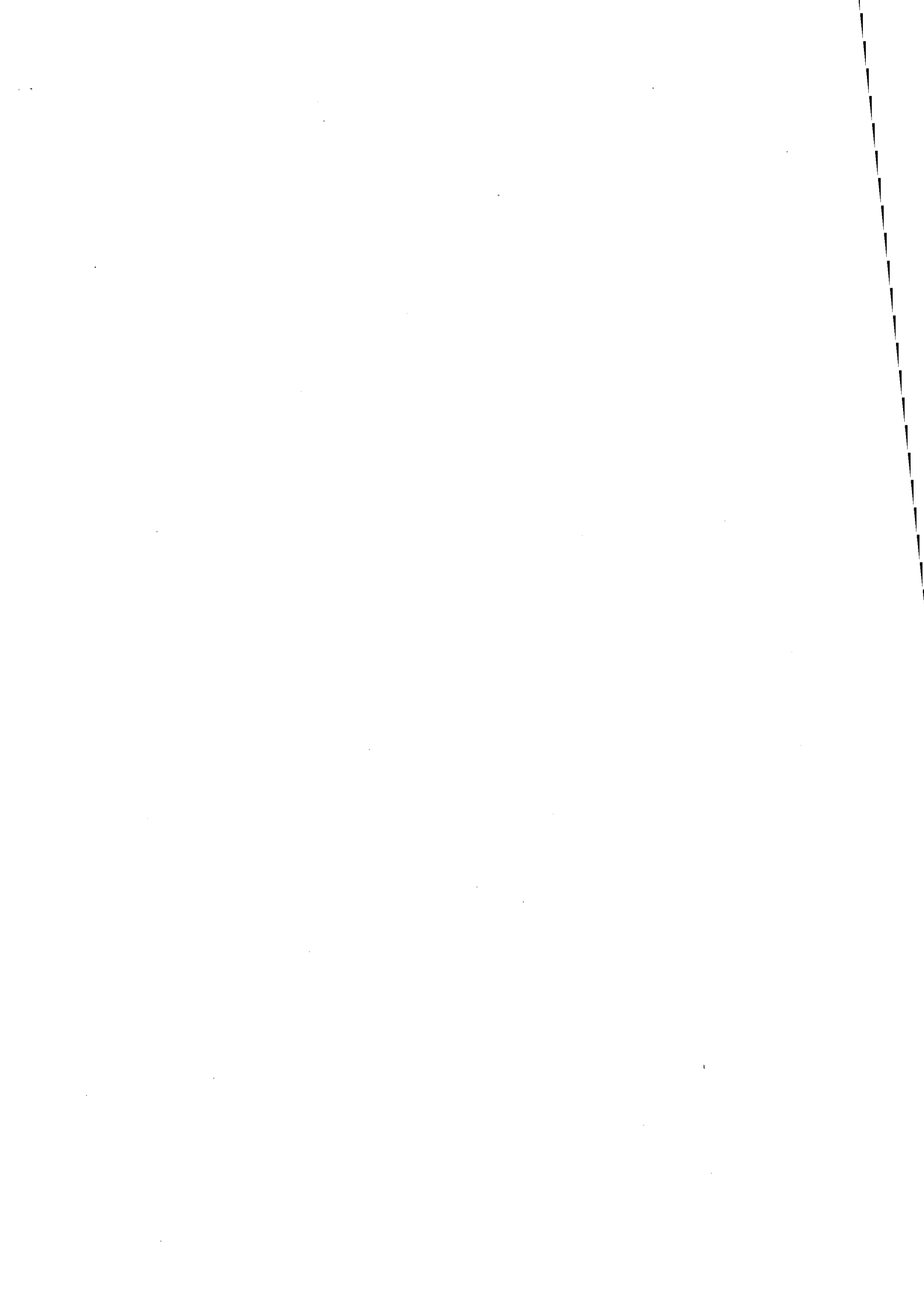
The knowledge acquisition part integrates both knowledge acquisition and machine learning techniques. As a first step, the knowledge acquisition methods have been used to identify the descriptive and strategic domain knowledge and to construct the description language to use for defining the examples needed for the machine learning. The second step is to use a machine learning technique to incrementally construct a knowledge graph using cases on interventions in situations of crisis obtained from the domain experts.

Two different procedures are proposed for the exploitation phase of the system. The first procedure is the interactive use of the knowledge graph, while the second procedure is the deductive use of the knowledge graph.

The knowledge validation approach proposed is based on the interactive use of the knowledge graph and on a follow-up on expert interventions in situations of crisis.

## **Keywords**

Knowledge Base, Acquisition, Machine learning, validation, Situations of crisis.



## *Sommaire*

<b>INTRODUCTION.....</b>	<b>13</b>
<b>1. Chapitre 1 : Outils d'Aide à la Décision en situation de crise.....</b>	<b>18</b>
1.1 LES SITUATIONS DE CRISE.....	18
1.1.1 <i>Le domaine de l'eau ... et les situations de crise dans ce domaine</i> .....	20
1.1.2 <i>Nécessité de méthodes spécifiques pour les situations de crise</i> .....	21
1.2 FONCTIONS ATTENDUES D'UN SYSTEME INFORMATIQUE POUR LES SITUATIONS DE CRISE.....	26
1.2.1 <i>Représentation des connaissances du domaine</i> .....	28
1.2.2 <i>Description de situations de crise</i> .....	28
1.2.3 <i>Sélection des experts appropriés à la résolution de ce type de problèmes</i> .....	29
1.2.4 <i>Argumentation des résultats</i> .....	29
1.2.5 <i>Validation des connaissances</i> .....	29
1.2.6 <i>Maintenance des connaissances</i> .....	30
1.3 APPORT DES SYSTEMES A BASE DE CONNAISSANCES.....	31
1.3.1 <i>Formalismes de représentation des connaissances</i> .....	32
1.4 MEDEXPEAU : UN SYSTEME POUR LES SITUATIONS DE CRISE.....	36
1.4.1 <i>Développement du système MEDEXPEAU</i> .....	37
1.4.2 <i>Architecture du système MEDEXPEAU</i> .....	39
1.4.3 <i>La structure des connaissances</i> .....	40
1.4.4 <i>Reformulation de problèmes dans le système MEDEXPEAU</i> .....	42
1.4.4.1 <i>Le mécanisme de substitution</i> .....	42
1.4.4.1.1 <i>Substitution simple</i> .....	43
1.4.4.1.2 <i>Substitution conditionnelle</i> .....	43
1.4.4.2 <i>Le mécanisme de généralisation</i> .....	44

1.5 BILAN.....	46
1.6 CONCLUSION.....	49
<b>2. Chapitre 2 : Méthodologie d'acquisition de connaissances.....</b>	<b>51</b>
2.1 PROBLEMES LIES A L'ACQUISITION DE CONNAISSANCES.....	51
2.2 LA PSYCHOLOGIE COGNITIVE ET LA MEMOIRE.....	53
2.3 ORGANISATION DU PROJET "ACQUISITION DE CONNAISSANCES".....	55
2.3.1 <i>Le modèle traditionnel</i> .....	56
2.3.2 <i>Le modèle direct</i> .....	57
2.3.3 <i>Le modèle tutoriel (guidé)</i> .....	58
2.4 ETAPES DE DEVELOPPEMENT D'UN SBC.....	59
2.4.1 <i>Approche par "prototypage rapide"</i> .....	59
2.4.2 <i>Approche "structurée"</i> .....	62
2.5 QUELQUES METHODOLOGIES D'ACQUISITION DE CONNAISSANCES.....	65
2.5.1 MACAO.....	66
2.5.1.1 <i>Les étapes de MACAO</i> .....	66
2.5.1.2 <i>Conclusion</i> .....	68
2.5.2 KADS.....	68
2.5.2.1 <i>Les primitives de modélisation KADS</i> .....	69
2.5.2.2 <i>Conclusion</i> .....	71
2.5.3 KOD.....	71
2.5.3.1 <i>L'analyse cognitive</i> .....	72
2.5.3.2 <i>Conclusion</i> .....	74
2.6 NOTRE APPROCHE D'ACQUISITION DE CONNAISSANCES.....	74
2.7 CONCLUSION.....	78
<b>3. Chapitre 3 : Choix d'une technique d'apprentissage adaptée au caractère incrémental et bruité des connaissances.....</b>	<b>79</b>
3.1 POURQUOI UNE TECHNIQUE D'APPRENTISSAGE DANS LE DOMAINE DE L'EAU ?.....	79
3.2 QU'EST-CE QUE L'APPRENTISSAGE AUTOMATIQUE ?.....	80
3.2.1 <i>Les différentes approches d'apprentissage automatique</i> .....	81
3.2.2 <i>Les règles d'inférence pour l'apprentissage symbolique</i> .....	82

3.3 CLASSIFICATION DES TECHNIQUES D'APPRENTISSAGE .....	83
3.3.1 <i>L'apprentissage inductif</i> .....	83
3.3.1.1 <i>Apprentissage à partir d'exemples</i> .....	84
3.3.1.2 <i>Apprentissage par découverte</i> .....	84
3.3.1.3 <i>Apprentissage à partir d'observations</i> .....	85
3.3.1.4 <i>Apprentissage par détection de similarités</i> .....	85
3.3.2 <i>L'apprentissage déductif</i> .....	85
3.3.2.1 <i>Apprentissage par recherche d'explications</i> .....	86
3.3.2.2 <i>Apprentissage par instruction</i> .....	86
3.3.2.3 <i>Apprentissage par coeur</i> .....	86
3.3.3 <i>L'apprentissage par analogie</i> .....	87
3.3.4 <i>Autres stratégies d'apprentissage</i> .....	87
3.3.4.1 <i>Apprentissage connexionniste</i> .....	87
3.3.4.2 <i>Le Raisonnement à Base de Cas</i> .....	88
3.4 QUELQUES SYSTEMES D'APPRENTISSAGE.....	88
3.4.1 <i>ID3</i> .....	88
3.4.2 <i>KATE</i> .....	90
3.4.3 <i>ALEX</i> .....	91
3.4.4 <i>COBWEB</i> .....	92
3.4.5 <i>UNIMEM</i> .....	93
3.4.6 <i>PROTOS</i> .....	93
3.4.7 <i>Les RDP</i> .....	94
3.4.7.1 <i>Les Dépendances Perçues</i> .....	94
3.4.7.2 <i>Formalisation de la relation de Dépendance Perçue</i> .....	95
3.4.7.3 <i>Les connaissances représentées dans le RDP</i> .....	97
3.5 CHOIX D'UN SYSTEME D'APPRENTISSAGE : LES RDP .....	97
3.5.1 <i>Le langage de description des exemples</i> .....	99
3.5.2 <i>Comment se construit un RDP</i> .....	100
3.6 CONCLUSION.....	103
<b>4. Chapitre 4 :Intégration d'une méthodologie d'acquisition de connaissance</b>	
<b>et d'une technique d'apprentissage : le système EXPEAU .....</b>	<b>105</b>



4.1 INTRODUCTION.....	105
4.2 ENVIRONNEMENT D'APPRENTISSAGE .....	106
4.2.1 <i>Environnement des observations</i> .....	107
4.2.1.1 <i>Langage de représentation des observations</i> .....	108
4.2.1.2 <i>Construction de taxinomies</i> .....	111
4.2.1.3 <i>Fonctions de généralisation</i> .....	113
4.2.1.4 <i>Conclusion</i> .....	117
4.2.2 <i>Les exemples d'apprentissage</i> .....	117
4.3 ELEMENTS DU RDP .....	120
4.4 ENTRETIEN INCREMENTAL DU GRAPHE DE CONNAISSANCES .....	123
4.4.1 <i>Schéma de construction de la Base de Connaissances</i> .....	123
4.4.2 <i>Intégration incrémentale des observations</i> .....	125
4.5 FORMAT D'UN EXEMPLE D'APPRENTISSAGE DANS EXPEAU.....	132
4.5.1 <i>Identification de l'intervenant</i> .....	132
4.5.2 <i>Organisme</i> .....	133
4.5.3 <i>Lieu de l'intervention</i> .....	133
4.5.4 <i>Date de l'intervention</i> .....	134
4.5.5 <i>Durée de l'intervention</i> .....	135
4.5.6 <i>Rôle joué par l'intervenant</i> .....	136
4.5.7 <i>Mission de l'intervenant</i> .....	136
4.5.8 <i>Domaine de compétence de l'intervention</i> .....	137
4.5.9 <i>Agents de l'environnement concernés par l'intervention</i> .....	139
4.6 CONCLUSION .....	140
<b>5. Chapitre 5 : Phase d'utilisation et de validation du système EXPEAU..</b>	<b>142</b>
5.1 INTRODUCTION.....	142
5.2 LES CONNAISSANCES DANS LE SYSTEME EXPEAU .....	144
5.2.1 <i>Séparation des connaissances apprises et des connaissances du domaine</i> .....	144
5.2.2 <i>Utilisation des connaissances</i> .....	145
5.2.2.1 <i>Utilisation en déduction</i> .....	146
5.2.2.1.1 <i>Langage d'interrogation</i> .....	147
5.2.2.1.2 <i>Langage des réponses</i> .....	148
5.2.2.1.3 <i>Recherche de solutions</i> .....	149

5.2.2.1.4 <i>Les heuristiques de recherche et de reformulation</i> .....	157
5.2.2.2 <i>Utilisation directe du graphe RDP</i> .....	160
5.2.2.2.1 <i>Utilisation du RDP dans un but pédagogique</i> .....	160
5.2.2.2.2 <i>Utilisation du RDP pour la recherche de solutions</i> .....	161
5.3 LE BRUIT DANS LES CONNAISSANCES .....	166
5.3.1 <i>D'où peuvent venir les erreurs ?</i> .....	166
5.3.2 <i>Besoin de traitement du bruit</i> .....	167
5.4 VALIDATION DES CONNAISSANCES.....	168
5.4.1 <i>Détecter et corriger les incohérences en utilisant le RDP</i> .....	169
5.4.2 <i>Suivi des interventions pour valider les connaissances</i> .....	171
5.5 CONCLUSION.....	174
<b>Conclusion et perspectives.....</b>	<b>175</b>
<b>Références bibliographiques.....</b>	<b>178</b>



## INTRODUCTION

Le développement de nouvelles technologies est à la fois l'un des moteurs essentiels du progrès économique et social et la cause de risques nouveaux. Des catastrophes industrielles comme celles survenues à Bhopal (en 1984, fuite dans un réservoir souterrain contenant de l'isocyanate de méthyle dans l'usine de Bhopal, situé dans l'état de Madhya Pradesh, d'Union Carbide), à Bâle (l'incendie de l'entrepôt de l'usine Sandoz en 1986) et à Tchernobyl (le 26 avril 1986, explosion d'un réacteur dans la centrale nucléaire) représentent des situations de crise et montrent que le combat pour la sécurité n'est jamais définitivement gagné. Le décalage entre les outils existants et les problèmes qu'il faut résoudre en temps de crise nous impose de trouver de nouvelles solutions.

Une situation de crise pose de façon brutale des problèmes complexes qui nécessitent une solution dans des délais très brefs. De plus, une crise peut engendrer une cascade de problèmes si elle n'est pas traitée à temps et avec des moyens efficaces. Des systèmes d'aide à la décision en cas de crise sont utilisés pour donner un ensemble d'indications et de décisions à prendre. Les indications fournies par ces systèmes sont : les causes possibles de la crise, les points qui peuvent être affectés par la crise, l'ensemble des impacts possibles. Les décisions peuvent être : des plans d'évacuation d'urgence, des mesures d'urgence à prendre, etc.

Cependant ce type de décisions est insuffisant pour des situations complexes et imprévues. La rigueur de l'expertise est une condition indispensable, elle nécessite le plus souvent le jugement, l'expérience et le savoir faire d'experts humains. Elle demande aussi des processus de décision plus ouverts et une mise en commun du savoir de plusieurs intervenants. Cela signifie qu'il n'est pas toujours possible de résoudre des problèmes en s'appuyant sur des règles de décisions recueillies auprès d'experts. Dans de telles situations, il est indispensable de faire appel aux experts eux-mêmes qui, par leurs expériences, leurs pouvoir de jugement stratégique et leur intuition peuvent tenir compte de tous les facteurs de la crise ainsi que ceux de son évolution.

Le présent travail a pour objectif principal la réalisation d'un Système à Base de Connaissances pour les situations de crise dans le domaine de l'eau. Ce système n'a pas pour vocation de donner un ensemble de décisions à prendre en cas de crise, mais de proposer une liste d'experts qui, eux, peuvent résoudre les problèmes engendrés par la situation de crise. A travers le développement de ce système, deux principaux axes de recherche ont été entrepris : l'acquisition et la validation de connaissances.

Le domaine de l'eau est caractérisé par des connaissances empiriques, évolutives et dispersées chez plusieurs experts appartenant à différentes disciplines. Cette caractéristique nous a amené à intégrer une méthodologie d'acquisition de connaissances et une technique d'apprentissage automatique pour résoudre le problème du transfert de connaissances depuis les experts vers le système informatique. Dans un premier temps, la méthodologie d'acquisition de connaissances nous a permis de recueillir l'ensemble des connaissances descriptives et stratégiques du domaine, et de construire un langage de description des exemples d'apprentissage. Une technique d'apprentissage automatique est ensuite utilisée pour construire incrémentalement un graphe de connaissances en utilisant des cas d'intervention sur des situations de crise fournis par les experts du domaine.

L'utilisation d'une technique d'apprentissage pour acquérir incrémentalement les connaissances induit des erreurs dans la Base de Connaissances. De plus, comme les rapports d'intervention sur des cas de crise sont fournis par plusieurs experts, le problème d'incohérence dans les connaissances acquises reste un point essentiel à traiter. Ainsi ce système a aussi pour vocation de détecter des incohérences dans les connaissances et de corriger la Base de Connaissances.

## ***PLAN DE LA THESE***

### ***Chapitre 1 : Outil d'aide à la décision en situation de crise***

Dans le premier chapitre de ce mémoire, nous tentons de présenter la problématique des situations de crise dans le domaine de l'eau et nous montrons la nécessité de construire un outil informatique spécifique pour ces situations. Les objectifs et les caractéristiques de cet outil

sont ensuite étudiés et son développement est décrit. Le but de cet outil est de modéliser les connaissances et le savoir-faire du domaine pour permettre une description, à différents niveaux de précision, de situations de crise et de sélectionner des experts capables d'intervenir pour trouver une solution aux problèmes engendrés par de telles situations.

## **Chapitre 2 : Méthodologie d'acquisition de connaissances**

Le développement et l'évaluation d'un Système à Base de Connaissances, décrit dans le chapitre 1, a permis de révéler la difficulté d'actualisation de la Base de Connaissances due principalement à une mauvaise organisation du processus d'acquisition de connaissances. L'étape d'élicitation et de formalisation des connaissances pour notre cas d'application représente une phase délicate du projet car pour résoudre des problèmes dans des situations de crise, les experts du domaine n'interprètent pas toujours leurs connaissances, ils utilisent parfois des automatismes qui sont des connaissances intuitives et qui leur permettent de donner rapidement une bonne réponse à un problème posé. Le chapitre 2 est donc consacré à la recherche d'une méthodologie adaptée à notre cas d'application en examinant la contribution des méthodologies d'acquisition de connaissances existantes. Il en ressort que les méthodologies existantes offrent un guide et un cadre pour aider le cognicien et les experts dans l'étape d'extraction, d'analyse et de formalisation des connaissances, mais ne permettent pas une amélioration continue des connaissances par une acquisition incrémentale. Comme notre application concerne le domaine de l'eau, qui est un domaine pluridisciplinaire et évolutif dont les connaissances sont dispersées à travers plusieurs experts de différentes disciplines, nous montrons que l'automatisation de l'acquisition de connaissances s'impose. Par conséquent, nous avons décidé de construire notre Base de Connaissances en utilisant un apprentissage automatique à partir d'exemples d'intervention sur des cas de crise fournis par les experts du domaine. Néanmoins, le processus d'apprentissage repose sur les observations présentées au système et sur des connaissances générales sur le domaine qui permettent de guider la production de concepts. Par conséquent, nous devons dans un premier temps spécifier un langage de description adéquat pour la représentation des exemples d'intervention et recueillir toutes les connaissances nécessaires pour orienter le processus d'apprentissage vers l'objectif que nous avons fixé.

### **Chapitre 3 : *Choix d'une technique d'apprentissage adaptée au caractère incrémental et bruité des connaissances***

Après avoir décrit, dans le chapitre 2, l'utilisation d'une méthodologie d'acquisition de connaissances pour mettre au point un cadre d'expression des cas d'intervention d'experts dans des situations de crise, nous examinons dans ce chapitre quelques méthodes d'apprentissage existantes pour choisir celle qui répond le mieux aux besoins de notre application. Le développement du Système à Base de Connaissances pour notre cas nécessite la collaboration de plusieurs experts et un enrichissement continu dû à la nature des connaissances dans ce domaine. Pour cela, nous utilisons un apprentissage incrémental fondé sur la théorie de la Dépendance Perçue développée par De Sainte Marie. Nous montrons comment adapter cette méthode d'apprentissage pour notre cas d'application en choisissant une description adéquate des exemples d'interventions fournis par les experts du domaine.

### **Chapitre 4 : *Intégration d'une méthodologie d'acquisition de connaissances et d'une technique d'apprentissage : le système EXPEAU***

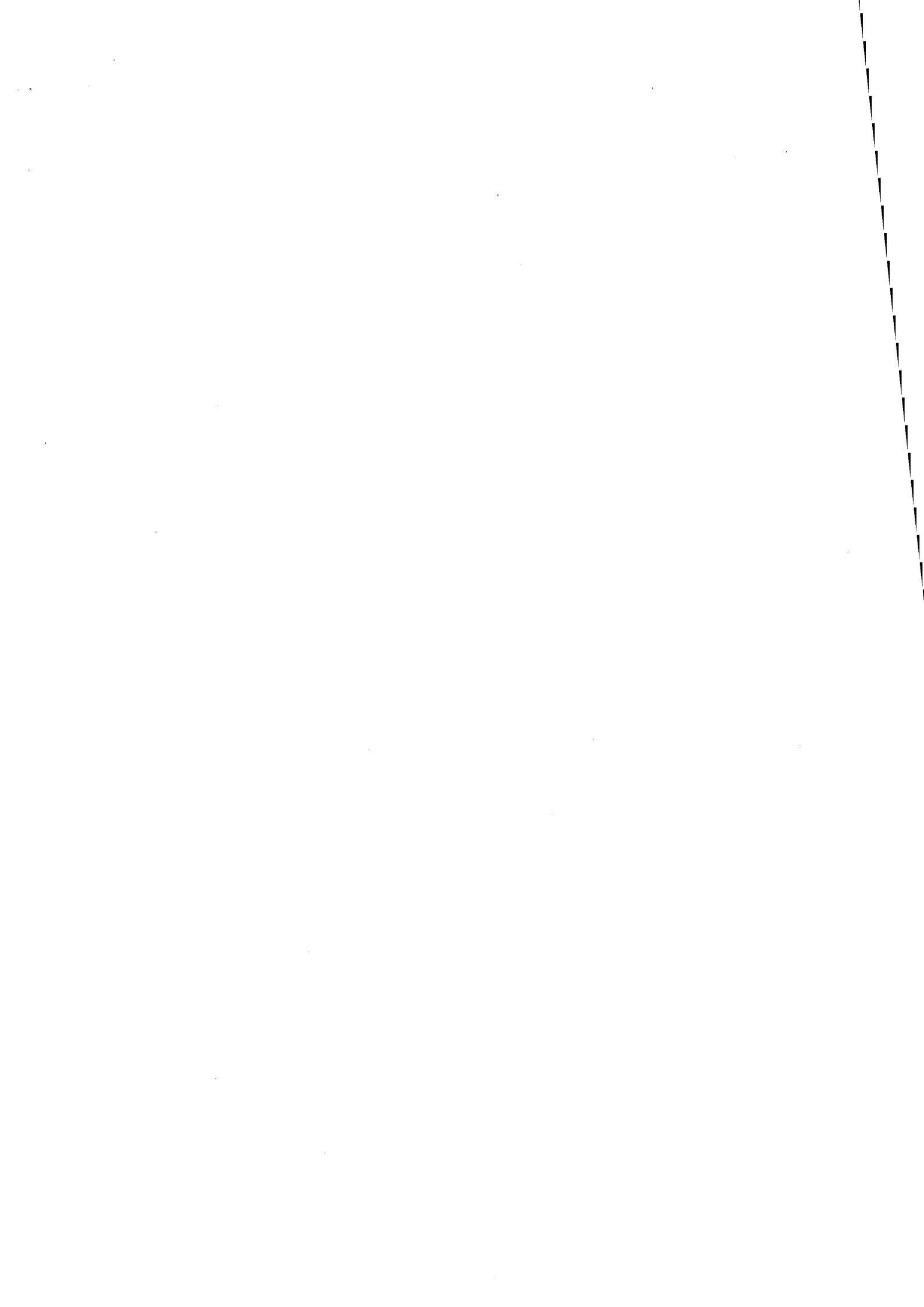
Le chapitre 4, est consacré au développement du système d'apprentissage "EXPEAU". Nous décrivons le langage utilisé pour la description des exemples d'apprentissage et nous explicitons la construction des taxonomies du domaine (taxonomie des professions, taxonomie des régions, taxonomie des compétences) en adoptant la méthodologie d'acquisition décrite au chapitre 2. Nous expliquons comment se fait la construction incrémentale de la Base de Connaissances en décrivant l'algorithme qui permet d'intégrer de nouveaux exemples d'apprentissage.

### **Chapitre 5 : *Phase d'utilisation et de validation du système EXPEAU***

Dans le chapitre 5, nous proposons un algorithme permettant d'exploiter les connaissances produites par le système d'apprentissage présenté au chapitre 4. Nous montrons

comment décrire une situation de crise en manipulant directement le graphe qui représente la connaissance apprise ou en utilisant un langage de requête. Nous décrivons dans ce chapitre les procédures de détection d'incohérences dans la Base de Connaissances et de leur validation. Nous proposons ensuite un schéma de communication entre le système et son environnement pour permettre un suivi efficace des interventions des experts sur le terrain.





# ***1. Chapitre 1 : Outils d'Aide à la Décision en situation de crise***

## **1.1 LES SITUATIONS DE CRISE**

Il est bien reconnu que l'essor des nouvelles technologies fait naître de grands espoirs, mais aussi de grandes peurs. Des catastrophes industrielles comme celles survenues à Bhopal (en 1984, fuite dans un réservoir souterrain contenant de l'isocyanate de méthyle dans l'usine de Bhopal, situé dans l'état de Madhya Pradesh, d'Union Carbide), à Bâle (incendie de l'entrepôt de l'usine Sandoz en 1986) et à Tchernobyl (le 26 avril 1986, explosion d'un réacteur dans la centrale nucléaire) [Kervern 91] ont montré que les organisations peuvent être surprises par des problèmes auxquels elles ne se sont pas préparées. Ces problèmes délicats, qui sont liés aux situations de crise, révèlent l'inefficacité des connaissances disponibles (l'impuissance face à la complexité, le décalage entre les systèmes mis à disposition et l'urgence des problèmes à résoudre, etc.). Ils mettent ainsi en évidence les conséquences dramatiques pour les hommes et l'environnement lorsque des outils et des moyens spécifiques ne sont pas disponibles.

Une situation de crise est une situation qui pose de façon brutale et imprévue un nombre impressionnant de problèmes urgents et difficilement surmontables. Elle est caractérisée par un débordement ou une inadéquation des capacités de réaction et de traitement habituelles et un besoin de jugements stratégiques plus que des réponses tactiques définies a priori [Forgues 91], [Lagadec 91]. L'insuffisance d'informations et de connaissances représente une difficulté non négligeable pour la maîtrise des problèmes [Lagadec 86] ; mais en situation de crise, encore une fois, nous sortons des limites habituelles car nous ne disposons ni d'estimations précises, ni de base d'interprétation des causes et des effets.

En résumé, une situation de crise peut être définie par les éléments suivants :

- l'enchaînement des problèmes : le problème initial peut engendrer une cascade d'autres problèmes,
- l'importance des dégâts : l'impact des situations de crise est de grande ampleur ( pertes humaines, dommages matériels, etc.),
- la nécessité d'une intervention d'urgence : la situation de crise exige une attention immédiate,
- l'absence de procédures programmées pour la résolution du problème : une situation de crise est une situation inattendue, pour laquelle il n'existe donc pas de solutions préparées à l'avance.

Chacun de ces éléments influe sur le processus de prise de décision. Parmi les facteurs les plus importants qui rendent plus difficile une prise de décision, nous pouvons citer :

- la pression temporelle : la nécessité de trouver une solution d'urgence, va empêcher toute réflexion approfondie et toute analyse détaillée de la situation et des conséquences des actions entreprises pour la maîtriser,

- l'incertitude : face à un événement inattendu, le décideur devra prendre une décision en s'appuyant sur des informations parcellaires. De ce fait, il va déclencher une action sans savoir si celle-ci est la plus appropriée, et sans pouvoir évaluer toutes les conséquences,

- la complexité : l'enchaînement des problèmes induit une complexité dans le traitement de la situation de crise et des interactions entre les différents problèmes.

La figure 1.1 illustre les effets induits par les éléments caractérisant une situation de crise qui influent sur le processus de la prise de décision.

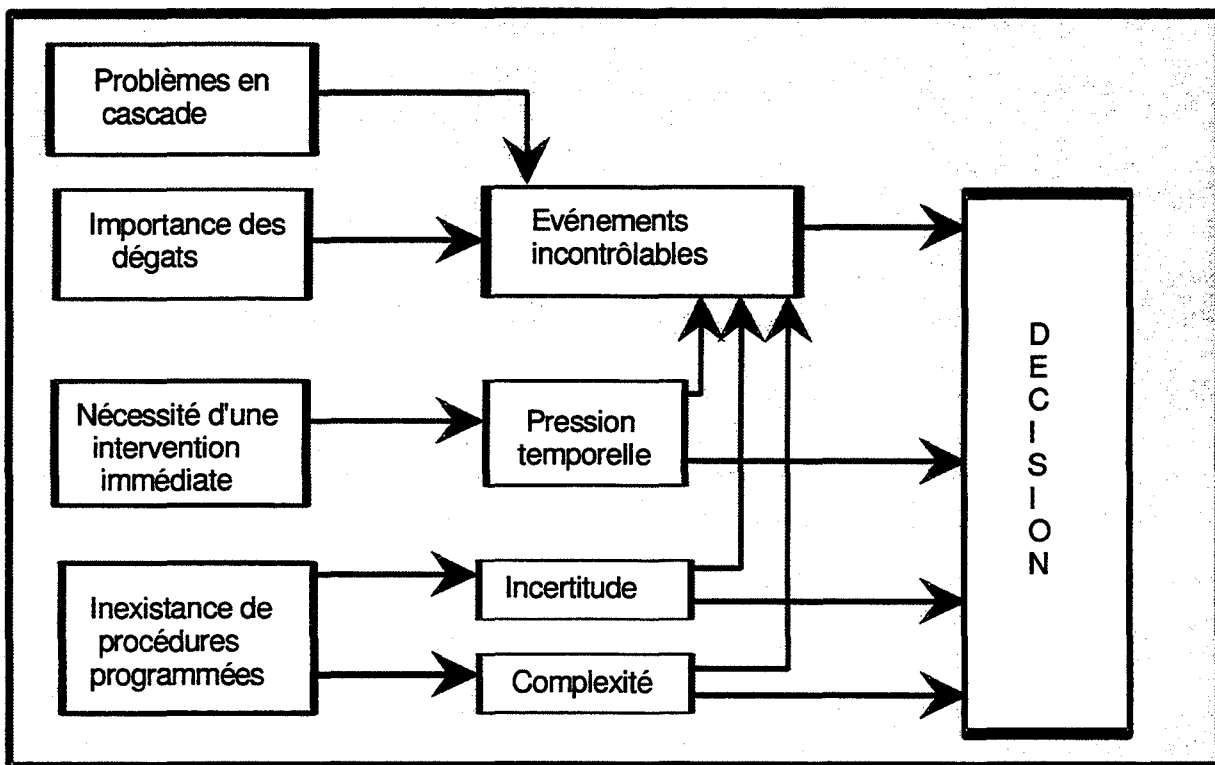


Figure 1.1 : caractéristiques et effets induits d'une situation de crise (inspiré de [Forgues 91])

### 1.1.1 Le domaine de l'eau ... et les situations de crise dans ce domaine

Le comportement des milieux naturels et technologiques est parfois imprévisible, ils peuvent donner lieu à des phénomènes dont l'évolution est indéterminée et incontrôlable dans

sa totalité. La prise de décision dans le domaine de l'eau est complexe car elle implique la prise en compte de l'imbrication et de l'interaction de différentes disciplines (hydrologie, hydrogéologie, pédologie, etc.).

L'eau est une ressource aux usages multiples. Elle est en effet utilisée pour la consommation humaine, pour l'industrie (eau de procédés, eau de refroidissement, de lavage ou d'évacuation des déchets, production d'énergie, etc.), pour l'agriculture, pour les transports, les loisirs, etc. Malheureusement, elle constitue parfois un obstacle aux activités humaines, voire la cause de catastrophes naturelles dont il faut se protéger. La succession d'événements tels que les sécheresses, les inondations, les pollutions dans les milieux lacustres et marins, les pollutions liées à des accidents routiers, les crues de rivières, la rupture de barrages, etc., renforce la prise de conscience des problèmes liés au domaine de l'eau.

Les inondations figurent parmi les plus importantes des catastrophes naturelles dans le monde. Elles apparaissent généralement suite à des orages violents, conduisant à des situations dramatiques dans les zones urbanisées (400 morts au Népal en 1987, inondations de Rio de Janeiro en 1988). Certaines crues, généralisées sur de très grands bassins, ont été des catastrophes majeures (100.000 morts en 1991 en Chine - crue du Yang-Zé).

Dans le paragraphe 1.1.2, nous allons examiner quelques outils qui sont actuellement disponibles et utilisés comme des systèmes d'aide à la décision en cas de crise. Nous montrerons ensuite leurs limites et nous spécifierons les fonctions attendues d'un outil pour de telles situations.

### **1.1.2 Nécessité de méthodes spécifiques pour les situations de crise**

Ces dernières années ont été marquées par une prise de conscience de la nécessité de disposer d'outils pour la maîtrise de situations de crise. Pour ce qui concerne le domaine de l'eau, nous avons indiqué que la maîtrise des phénomènes naturels est très complexe car nous ne pouvons pas prévoir avec certitude les conséquences et les moyens à mettre en oeuvre en situation d'urgence. Les connaissances nécessaires à la résolution des problèmes posés dans ce

domaine sont encore fragmentaires, éclatées dans une multitude de champs scientifiques, de spécialités et dans différents organismes.

Des systèmes d'intelligence artificielle fondés sur des règles de décision et ayant des capacités de raisonnement peuvent être utiles, pour aider ou parfois remplacer l'opérateur humain, dans la résolution de certains problèmes. Comme exemple, nous pouvons citer le système "S-eau-S" [Risler 91] qui est un logiciel d'aide à la décision en cas d'alerte à la pollution des eaux, et le système "SEABEL" [Kervern 91] qui est un outil d'aide à la décision pour les dangers de pollution marine.

- "S-eau-S" fournit à l'utilisateur un diagnostic prenant en compte l'ensemble des scénarios d'évolutions possibles. Ce système peut par exemple (si la cause et le point de pollution sont connus) :

- donner le cheminement vers l'aval de la pollution,
- donner les captages susceptibles d'être atteints.

Il offre aussi la possibilité (si la cause de la pollution n'est pas connue) de passer en revue les établissements ou installations susceptibles d'être la cause potentielle de l'anomalie constatée. D'autres outils de cette catégorie existent et permettent entre autres :

- d'évaluer la situation et d'énumérer les impacts possibles,
- d'établir un plan d'urgence et/ou de donner des mesures d'urgence à prendre.

- "SEABEL" est capable de tirer des conclusions et de faire des prévisions en ce qui concerne les dangers de pollution chimique de la mer. Il est aussi un instrument de simulation des accidents, il peut donc être utilisé comme un moyen d'enquête et comme outil pédagogique. Ce système est composé de quatre modules remplissant les tâches suivantes :

- diagnostic d'accident,
- diagnostic d'impact,
- identification des dangers,
- aide à la décision d'urgence.

Ces deux systèmes utilisent une représentation sous forme de règles de production pour coder les connaissances issues des experts.

D'autres outils informatiques d'aide à l'élaboration de plans d'urgence et d'évacuation destinés aux problèmes d'inondation (bris de barrage, crues exceptionnelles, etc.) sont développés et permettent de :

- déterminer les zones inondées,
- déterminer et caractériser les dommages dans les zones inondées,
- établir les plans d'évacuation des zones qui sont inondées et/ou touchées par une destruction limitrophe des infrastructures (pont, production d'électricité, etc.).

Des fiches synthétiques préparées à l'avance et décrivant les étapes à suivre en cas de crise (crues, inondations, ...) sont parfois utilisées par des organisations diverses. Les services chargés de la mise en alerte de la population en cas de crues (services d'annonce des crues) s'appuient sur de telles fiches comme aide à la décision (figure 1.2).

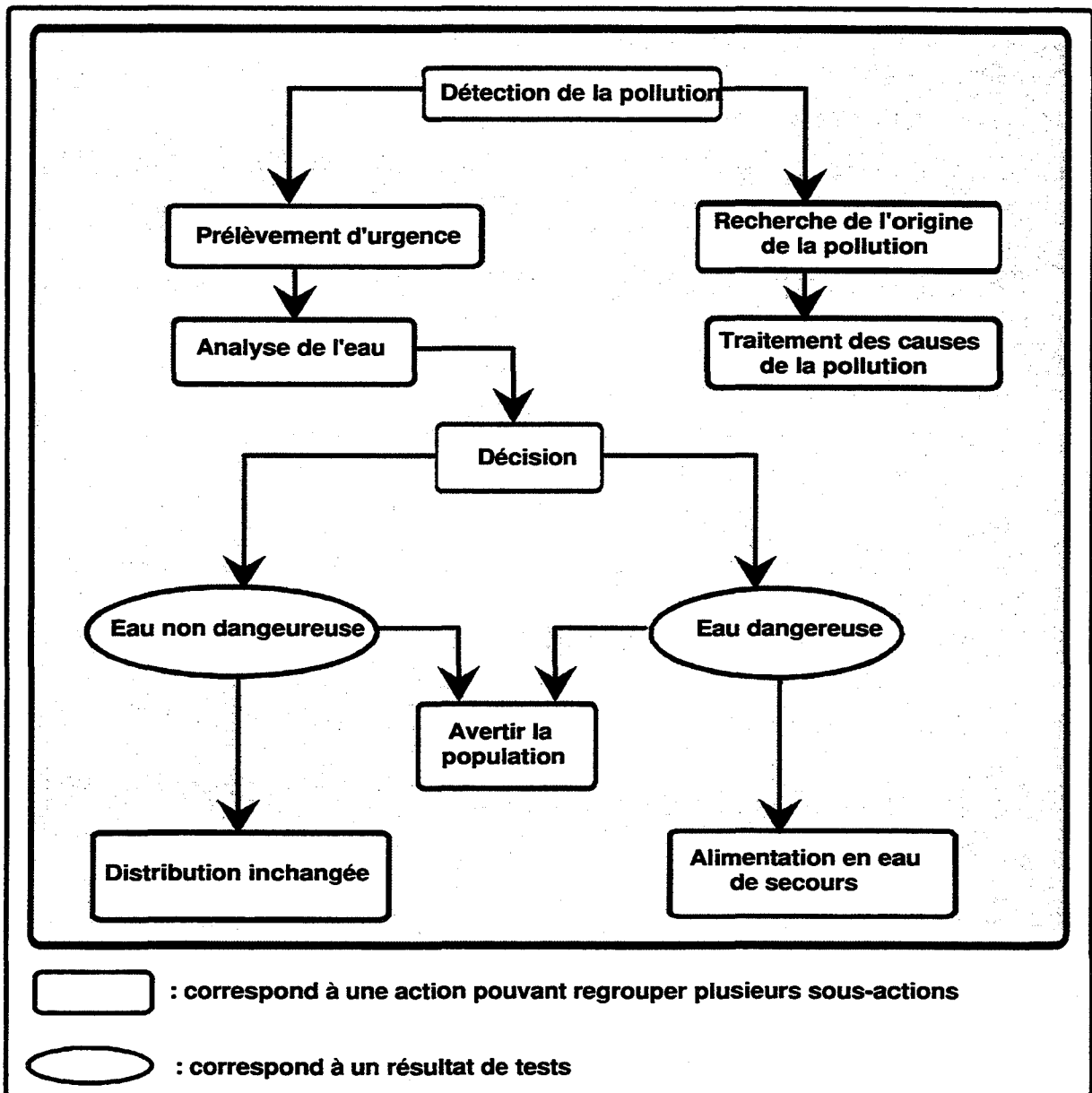


Figure 1.2 : Exemple de fiche synthétique en cas de pollution d'eau

Il se trouve que les outils que nous venons de citer sont davantage orientés vers la résolution de problèmes bien identifiés, ils sont donc insuffisants et inadaptés pour des situations complexes et imprévues. Nous pouvons, en effet, être confrontés à des situations de crise pour lesquelles les logiques préparées sont peu similaires à celles qu'il serait nécessaire d'utiliser et les mécanismes d'exploitation trop réducteurs et contre-indiqués. Tout d'abord, lors de la construction d'un tel système, nous ne pouvons pas penser à toutes les éventualités qui



peuvent se produire afin de prendre en compte tous les phénomènes engendrés par la situation initiale. D'autre part, suivre aveuglement un scénario prédéfini peut conduire à une aggravation de la crise lorsqu'il n'y a pas une bonne adéquation de la situation à traiter par rapport à une typologie de situations connues (il semble que les premières réactions des conducteurs de la centrale nucléaire de Tchernobyl ont amené la crise à un point de non-retour [Forgues 91]). Ainsi si un outil proposant des décisions devient l'unique élément de résolution dans des situations de crise, les solutions proposées peuvent être inadaptées car elles ne prennent pas en compte l'évolution de tous les phénomènes générés par la situation.

Un système d'aide à la décision peut être défini comme un système de résolution de problèmes utilisant des heuristiques [Levine 89]. Un tel système ne prend pas vraiment de décision ; mais propose uniquement des solutions à l'utilisateur à qui, d'ailleurs, revient la charge de contrôler une partie importante de la recherche heuristique. Généralement un système d'aide à la décision accomplit des tâches :

- d'analyse statistique en utilisant des méthodes d'analyse de données,
- de prévision, en se basant sur des modèles (physiques, économétriques, etc.) et les statistiques,
- de simulation, en s'appuyant sur les modèles (physiques, économétriques, etc.) et des méthodes de Monte-Carlo,
- de choix d'actions, en utilisant les techniques de décision multicritère,
- d'optimisation, en faisant appel à la recherche opérationnelle.

Toutefois, même en faisant appel à des modèles mathématiques sophistiqués, le processus de décision global n'est généralement pas complètement déterminé. Nous constatons donc bien que de tels systèmes d'aide à la décision sont insuffisants dans le cas d'une situation de crise qui, elle, nécessite le plus souvent le jugement, l'expérience et le savoir faire d'experts.

Ce qui a été énoncé précédemment montre qu'il n'est pas toujours possible de résoudre les problèmes en se basant sur un guide de conduite s'inspirant des fiches-réflexes élaborées par les spécialistes de l'urgence. Cela peut être même dangereux de recueillir des règles dans le but d'une utilisation pour l'aide à la décision dans des situations délicates nécessitant des jugements et des actions précises. Dans de telles situations, pour arrêter l'escalade des problèmes

engendrés, il est indispensable de mobiliser rapidement des experts du domaine qui, eux, par leurs expériences, leur pouvoir de jugement stratégique et leur intuition peuvent trouver une solution au problème posé. L'intervention pour la résolution de problèmes en situation de crise nécessite des connaissances acquises par la formation sur des bases théoriques et surtout expérimentales. Les bases expérimentales viennent de la recherche fondamentale et appliquée, mais aussi et d'abord de l'expérience vécue : analyses d'accidents, recherches des causes d'accidents, prises de décisions, etc. La difficulté qui se pose, a priori, réside donc au niveau de l'expertise et de la communication, à savoir comment procéder pour trouver les experts les plus appropriés vis-à-vis d'une situation de crise.

Après avoir posé le problème, il est nécessaire maintenant d'explicitier les fonctions que doit remplir un outil informatique pour une utilisation en situation de crise.

## **1.2 FONCTIONS ATTENDUES D'UN SYSTEME INFORMATIQUE POUR LES SITUATIONS DE CRISE**

Nous avons indiqué dans le paragraphe précédent que les situations de crise nécessitent des outils spécifiques autres que ceux utilisés comme aide à la prise de décision dans des domaines où la connaissance est bien établie et pour des problèmes répertoriés. Une situation de crise est, par définition, une situation imprévue pour laquelle nous n'aurions donc pas pu prévoir la résolution (du moins en totalité) par une combinaison d'un certain nombre de procédures. Dans de pareils cas, nous n'attendons pas d'un système de nous dire ce qu'il faut faire mais de nous proposer une liste d'experts capables d'intervenir pour résoudre les problèmes engendrés par la situation de crise. C'est une solution sur laquelle nous avons décidé de nous focaliser. L'obtention d'une telle solution suppose que l'outil ait la capacité d'une part d'analyser la situation et donc de pouvoir comprendre la description d'une situation de crise et d'interpréter les concepts utilisés dans cette description (figure 1.3), et d'autre part d'analyser les expériences et les compétences des experts du domaine.

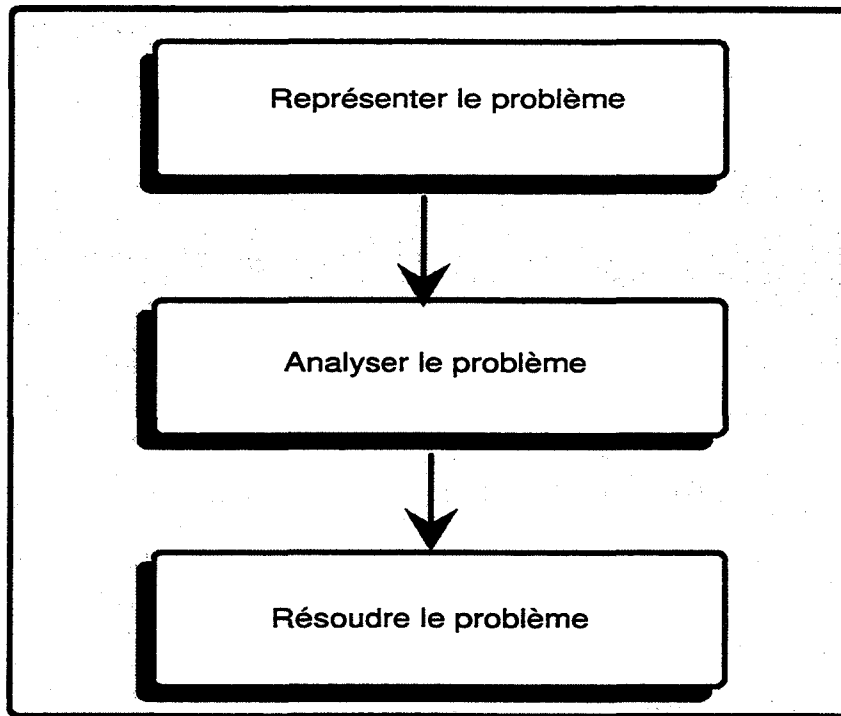


Figure 1.3 : Enchaînement des étapes de représentation, d'analyse et de résolution

Il est donc nécessaire de recueillir toutes les connaissances du domaine qui permettront, par la suite, de décrire des situations de crise et de sélectionner les experts qui pourront apporter une solution aux problèmes posés. Un mécanisme d'argumentation qui permet au système de justifier les résultats qu'il a produit est indispensable pour ce type d'outil. Si les experts proposés par le système n'arrivent pas à trouver une solution au problème décrit, le système doit pouvoir valider les connaissances qui ont permis d'aboutir à de tels résultats. Pour prendre en compte de nouvelles situations de crise et de nouvelles compétences acquises par les experts, le système doit pouvoir actualiser l'ensemble des connaissances qu'il utilise.

Le système doit donc remplir les fonctions qui permettent: la représentation du domaine, la description de situations de crise, la sélection d'experts, l'argumentation des résultats, la validation et la maintenance des connaissances.

### 1.2.1 Représentation des connaissances du domaine

Dans notre cas, nous cherchons un type de représentation qui nous permette d'obtenir un modèle adapté à la recherche d'experts en situation de crise dans le domaine de l'eau. Une des caractéristiques de l'outil que nous devons prendre en compte à ce niveau est l'approche par "problèmes". L'approche par "problèmes" exige que nous traitions l'ensemble des éléments entrant dans la description d'une situation de crise comme un tout et ne pas raisonner en termes de disciplines et de sous-disciplines correspondant à ces éléments. En effet, il s'agit de s'intéresser aux experts du domaine et non pas aux spécialistes de telle ou telle discipline. Nous définissons ci-dessous ce que nous entendons par "expert" et "spécialiste" pour expliciter la nature des connaissances qu'ils détiennent.

*Définition :*

- un expert est une personne qui possède un savoir acquis par la pratique. Il s'appuie sur une longue expérience pour résoudre des problèmes pointus,
- un spécialiste est une personne spécialisée dans un domaine déterminé et restreint. Il possède un savoir régi par un ensemble de théories bien établies.

### 1.2.2 Description de situations de crise

La description de situations de crise est un préalable important à la résolution du problème. Mieux le problème est formulé en utilisant des éléments de connaissances ~~pertinents~~, plus la réponse sera exacte. Pour cela, le système recherché doit comporter une interface qui permette la formulation, à différents niveaux de précision, de toute situation de crise dans le domaine en offrant une aide d'accès aux éléments de connaissances qui sont pertinents pour cette représentation. La description d'une situation de crise consiste à définir en extension un ensemble de descripteurs. La cardinalité de cet ensemble dépend d'une part des éléments de connaissances du domaine et d'autre part du niveau de précision attribuée à la description.

### **1.2.3 Sélection des experts appropriés à la résolution de ce type de problèmes**

Une fois le problème décrit, le système doit proposer, en suivant des stratégies adaptées, une liste d'experts les plus compétents vis-à-vis de la résolution de ce type de problèmes. Une autre caractéristique de l'outil à prendre en compte à ce niveau est "l'obligation de réponse". L'obligation de réponse exige que le système propose au moins un expert à chaque problème qui lui est posé. En d'autres termes, si aucun expert ne répond exactement aux critères d'intervention sur un cas de crise, le système doit rechercher ceux qui sont les plus proches de ce type d'intervention (les experts qui sont les plus suffisamment compétents pour intervenir sur le cas de crise décrit par l'utilisateur). Pour chaque situation décrite et à laquelle aucun expert ne correspond pour trouver une solution, le système doit donc être capable de reformuler la description du problème pour élargir le champ de la recherche. Mais, cette reformulation de problèmes doit se faire de manière à ce que la nouvelle description obtenue soit similaire à la description du problème initial. Cette contrainte qui consiste à rester le plus près possible de la situation initiale permet d'éviter de sélectionner des experts qui ne pourront pas intervenir sur la situation de crise décrite initialement.

### **1.2.4 Argumentation des résultats**

Afin de justifier le raisonnement suivi et de convaincre de la justesse du résultat obtenu, le système doit donner des arguments sur l'adéquation des experts trouvés avec la description de la situation de crise. Dans le cas où le problème initial est reformulé pour trouver une solution, le système doit aussi expliciter les raisons du choix des connaissances qui ont permis d'aboutir à cette reformulation.

### **1.2.5 Validation des connaissances**

Cette activité comporte deux aspects. Le premier est destiné à examiner la complétude et l'utilité des connaissances structurées dans le système. Le deuxième aspect doit permettre de

prendre en compte l'évolution des connaissances. La mise en oeuvre des connaissances dans de nouvelles situations permet de juger grossièrement de la validité des connaissances.

## 1.2.6 Maintenance des connaissances

Suite à la validation, des modifications au niveau du modèle de connaissances doivent être, dans le cas échéant, possibles. L'activité de maintenance des connaissances consiste aussi à faire vivre l'outil par des mises à jour des connaissances recueillies : ajout, modification et suppression des connaissances préalablement organisées dans le système.

La figure 1.4 illustre le schéma SADT (Structured Analysis and Design Technique) [Lissandre 90], [Jaulent 92] de l'enchaînement des fonctions attendues d'un système informatique pour les situations de crise.

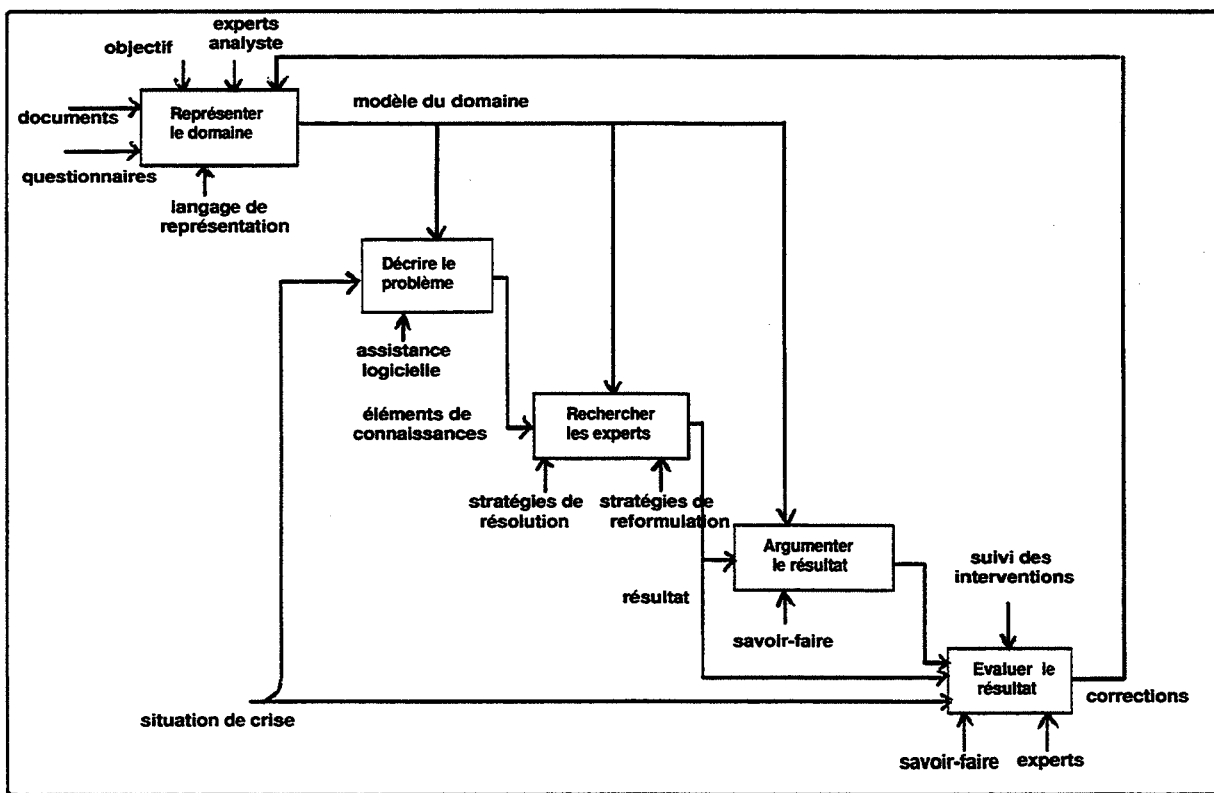


Figure 1.4 : Schéma SADT des fonctions d'un outil informatique pour les situations de crise

Afin de pouvoir manipuler un ensemble de connaissances dispersées dans plusieurs disciplines et de satisfaire les critères énoncés : approche par "problèmes" (raisonnement sur les expériences d'experts) et obligation de réponse (possibilité de reformulation de problèmes en raisonnant sur les connaissances du domaine), nous avons décidé de développer un Système à Base de Connaissances que nous décrirons dans le paragraphe 1.4.

### 1.3 APPORT DES SYSTEMES A BASE DE CONNAISSANCES

Certains auteurs définissent un Système à Base de Connaissances (SBC) comme un outil d'amélioration des compétences et un moyen de diffusion des connaissances [Grundstein 88], [Lewis 94]. Ce type de système a comme vocation de révéler, de structurer et de faire partager une connaissance évolutive. Nous pouvons dire qu'un SBC est tout d'abord un système de traitement de données qui permet de raisonner sur des connaissances recueillies pour aboutir à des conclusions ou découvrir de nouveaux concepts et de résoudre des problèmes en utilisant des heuristiques.

Rappelons qu'un SBC est constitué de différents éléments à savoir :

**une Base de Connaissances (BC)** : contient les connaissances du domaine d'application permettant au système d'agir comme un expert,

**un Système Cognitif** : est l'élément actif du système, il est chargé d'exploiter la BC pour construire des raisonnements permettant de résoudre les problèmes posés,

**un module d'explication** : il permet de rendre compte de la manière dont a été conduite la résolution d'un problème,

**un module d'acquisition de connaissances** : il permet de faciliter les tâches d'acquisition et de révision des connaissances,

**un espace de travail** : est la zone dynamique où s'effectue le raisonnement du système. L'espace de travail est réinitialisé au début de chaque session de travail.

Les SBC ont montré leur utilité dans différents domaines économiques, industriels, etc., grâce au caractère déclaratif des connaissances qu'ils manipulent (séparation des connaissances et des traitements, ce qui permet un développement incrémental) et à la capacité de ces systèmes à donner des explications pour justifier les résultats produits. Les SBC permettent principalement de pallier l'insuffisance du nombre d'experts et de rendre accessible le savoir-faire des experts à des utilisateurs moins expérimentés, et aussi de transmettre une connaissance qui se perdrait avec la disparition de l'expert qui la détient. La manipulation des symboles par ces systèmes ouvre le champ à des applications hors de portée des outils traditionnels qui ne raisonnent que sur la base de données quantifiables et d'algorithmes.

### **1.3.1 Formalismes de représentation des connaissances**

La représentation des connaissances est un problème qui se pose dans le développement de tout système sensé manipuler un ensemble de connaissances. Elle consiste à transcrire des connaissances sous une forme symbolique exploitable par un système de raisonnement. Le choix d'une représentation adéquate est fondamental et détermine toute la suite du développement du système.

Les langages de représentation des connaissances viennent donc en aval du processus de construction de Bases de Connaissances et jouent un rôle important dans la réussite d'un projet "Système à Base de Connaissances". La finalité d'un langage de représentation est de définir une structure d'accueil adaptée à l'implantation, la maintenance et l'exploitation des connaissances (et notamment à la production d'explications satisfaisantes pour justifier les résultats obtenus).

Les principaux formalismes de représentation de connaissances utilisés pour le développement de Systèmes à Base de Connaissances sont [Barr 81], [Kauffmann 85], [Dominé 88] :



### **Représentations logiques**

Les représentations logiques permettent de formaliser des propositions logiques et des relations entre les propositions. La logique mathématique est l'un des premiers formalismes utilisés pour représenter des connaissances dans le domaine de l'intelligence artificielle. Ces représentations incluent la logique des propositions et la logique des prédicats du premier ordre.

### **Règles de production**

Une règle de production est caractérisée par la formulation **SI** condition(s) **ALORS** action(s), qui indique que dans une situation donnée (vérification des prémisses contenues dans la partie gauche de la règle), certaines conclusions peuvent être inférées ou établies.

### **Représentations par objets structurés (les frames)**

Les frames constituent des structures de données dans lesquelles toute la connaissance relative à un événement ou à un objet particulier est regroupée ensemble. Un frame peut être décrit comme un arbre dont la racine est étiquetée par son nom. Le premier niveau de l'arbre est celui de ses attributs (slots), le second celui des facettes (facets), le troisième celui des valeurs (values). Les frames sont des structures qui peuvent être utilisées pour représenter différents types d'objets. Ils peuvent être liés les uns aux autres et organisés hiérarchiquement et ainsi, un frame peut alors hériter des propriétés d'autres frames positionnés plus haut dans la hiérarchie.

### **Réseaux sémantiques**

Les réseaux sémantiques ont été introduits pour la première fois dans les travaux de Quillian [Quillian 68] sur la modélisation de la mémoire sémantique de l'être humain. Les réseaux sémantiques, comme les frames, organisent la connaissance autour des objets à décrire. Les objets sont représentés par des noeuds dans un graphe et les relations entre ces objets par des arcs. Un des intérêts principaux de ce formalisme est la capacité d'atteindre à partir d'un point du réseau toutes les informations conceptuellement associées à ce point. Les réseaux sémantiques sont adaptés aux domaines d'expertise dans lesquels les connaissances sont interdépendantes et structurées.

## **Représentations procédurales**

Dans les représentations procédurales, la connaissance est représentée sous forme de procédures (programme).

Le tableau 1.1 résume les avantages et les inconvénients des formalismes de représentation de connaissances que nous avons cités dans ce paragraphe.

	<b>Avantages</b>	<b>Inconvénients</b>
<b>Représentations logiques</b>	<ul style="list-style-type: none"> <li>- Existence d'une théorie de la logique</li> <li>- Modularité de la représentation (facilité d'ajout, de suppression et de modification de connaissances)</li> <li>- Facilité de représentation de connaissances descriptives</li> </ul>	<ul style="list-style-type: none"> <li>- Difficulté de représentation de connaissances heuristiques</li> </ul>
<b>Règles de production</b>	<ul style="list-style-type: none"> <li>- Lisibilité: les connaissances sont facilement compréhensibles (structure de représentation normalisée)</li> <li>- Capacité d'explication du raisonnement</li> <li>- Facilité d'ajout, de modification et de suppression de connaissances</li> </ul>	<ul style="list-style-type: none"> <li>- Lourdeur d'exploitation lorsqu'il s'agit de traiter un nombre important de connaissances</li> <li>- Difficulté de contrôler et de maintenir la cohérence lorsqu'il y a un grand nombre de règles</li> <li>- Difficulté de représentation de connaissances descriptives</li> </ul>
<b>Réseaux sémantiques</b>	<ul style="list-style-type: none"> <li>- Capacité de représentation de connaissances descriptives et de relations entre les connaissances</li> <li>- Capacité de structuration de connaissances</li> </ul>	<ul style="list-style-type: none"> <li>- Difficulté d'explication du raisonnement</li> <li>- incapacité de représenter des connaissances heuristiques</li> </ul>
<b>Représentations par objets structurés</b>	<ul style="list-style-type: none"> <li>- Possibilité de structuration des connaissances et des raisonnements</li> <li>- clarté de la représentation</li> </ul>	<ul style="list-style-type: none"> <li>- Difficulté d'explication du raisonnement</li> </ul>
<b>Représentations procédurales</b>	<ul style="list-style-type: none"> <li>- Facilité de représentation des connaissances heuristiques</li> </ul>	<ul style="list-style-type: none"> <li>- Difficulté de mise en oeuvre d'un mécanisme d'explication</li> <li>- Manque de modularité (l'ajout de nouvelles connaissances peut nécessiter une modification des procédures de traitement)</li> </ul>

Tableau 1.1 : Avantages et inconvénients des langages de représentation de connaissances

Nous avons précisé précédemment que l'outil recherché doit assurer une assistance aux utilisateurs dans le processus de description de situation de crise. La description de telles situations doit pouvoir être faite à différents niveaux de précision. Cela nécessite un accès facile et guidé aux connaissances représentées. Une solution qui paraît satisfaisante consiste à structurer les connaissances du domaine sous une forme hiérarchique et de permettre un accès aux éléments de connaissances spécifiques à partir d'éléments de connaissances généraux.

Pour cela, nous avons opté pour une représentation sous forme de réseaux sémantiques. Cette structure va nous permettre également de répondre au critère "d'obligation de réponse" en mettant en relation les éléments de connaissances qui peuvent contribuer à la reformulation de problèmes.

Dans le paragraphe suivant nous présenterons le SBC que nous avons développé pour répondre à la problématique des situations de crise.

#### **1.4 MEDEXPEAU : UN SYSTEME POUR LES SITUATIONS DE CRISE**

Le système MEDEXPEAU (Méditerranéen-Experts en Eau) a pour objectif de représenter et d'exploiter des connaissances issues des organisations compétentes dans le domaine de l'eau des pays du bassin méditerranéen. Il s'agit de constituer un réseau de connaissances et d'experts afin de pouvoir mobiliser les compétences existantes et de lutter contre les problèmes pouvant survenir dans ce domaine [Senoune 92].

La première étape de notre travail a consisté en une familiarisation avec le domaine au travers de réunions de travail avec les experts et de consultation de documents techniques. Un corpus de connaissances a été constitué pour former une structure de base servant de questionnaire à soumettre à tous les experts.

### 1.4.1 Développement du système MEDEXPEAU

Pour la réalisation du système MEDEXPEAU, nous nous sommes appuyés sur une méthode de conception évolutive par raffinements successifs [Feigenbaum 89]. Cette approche dite "évolutive" consiste à développer un prototype et à l'adapter aux besoins. Le prototype réalisé pour servir de tests est progressivement amélioré jusqu'à l'obtention d'un outil jugé satisfaisant par les utilisateurs. Cette technique de développement partiel de systèmes informatiques, utilisée dans le but de mieux cerner le problème et la solution nous permet, entre autres :

- d'évaluer la faisabilité des exigences,
- de valider la nécessité de toutes les fonctions,
- de découvrir des fonctions manquantes,
- de déterminer la convivialité de l'interface utilisateur.

La figure 1.5 illustre l'enchaînement des étapes dans le processus de développement du système MEDEXPEAU.

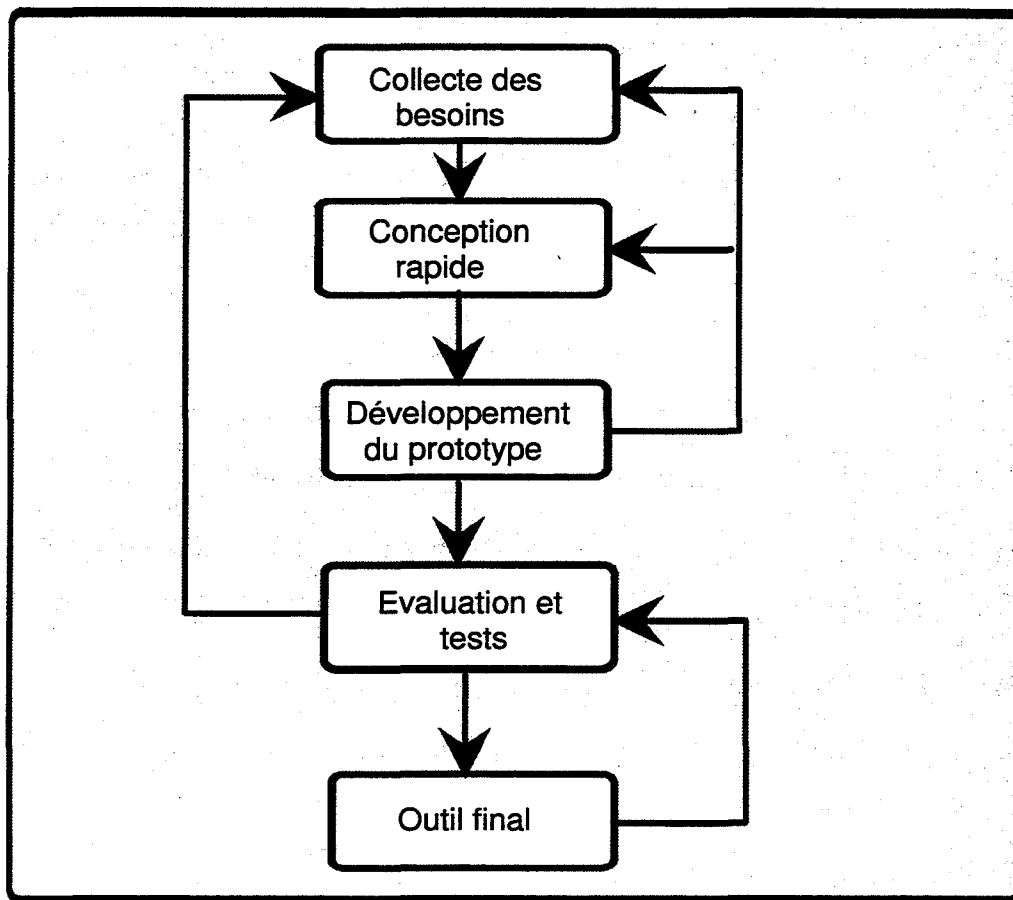


Figure 1.5 : Le cycle de vie du système MEDEXPEAU

Comme illustré sur la figure 1.5, nous avons adopté un cycle de vie fondé sur un modèle incrémental pour le développement de MEDEXPEAU. Ce modèle correspond à un parcours des étapes de développement en profondeur d'abord. Cela signifie que nous traitons un ensemble de fonctions jusqu'à l'implantation avant de revenir sur d'autres fonctions pour combler les erreurs et les oublis du cycle précédent. En effet, au cours du traitement correspondant à une étape donnée, il arrive souvent de découvrir un nouvel aspect du problème non traité dans l'étape précédente. D'ailleurs, au début d'un projet, le domaine d'étude n'est pas totalement identifié et nous ne sommes jamais sûr de la complétude des connaissances nécessaires.

Avant de décrire comment sont structurées les connaissances du domaine, nous allons tout d'abord présenter les différentes composantes du système MEDEXPEAU.

### **1.4.2 Architecture du système MEDEXPEAU**

Comme le montre la figure 1.6, le système MEDEXPEAU est constitué des éléments suivants [Senoune 92] :

- une Base de Connaissances : constituée des connaissances du domaine d'application et des informations (coordonnées, compétences, références des interventions) sur les experts du domaine,
- un module d'aide à l'acquisition de connaissances : donne la possibilité d'insérer et de modifier les éléments de connaissances dans la BC,
- un module d'aide à la formulation de problèmes : permet d'aider les utilisateurs dans le processus de description de situations de crise,
- un module d'analyse et de résolution : analyse le problème décrit et sélectionne les experts les plus compétents vis-à-vis de ce problème,
- une interface : est l'élément qui permet une interaction entre l'environnement (experts, utilisateurs) et le système.

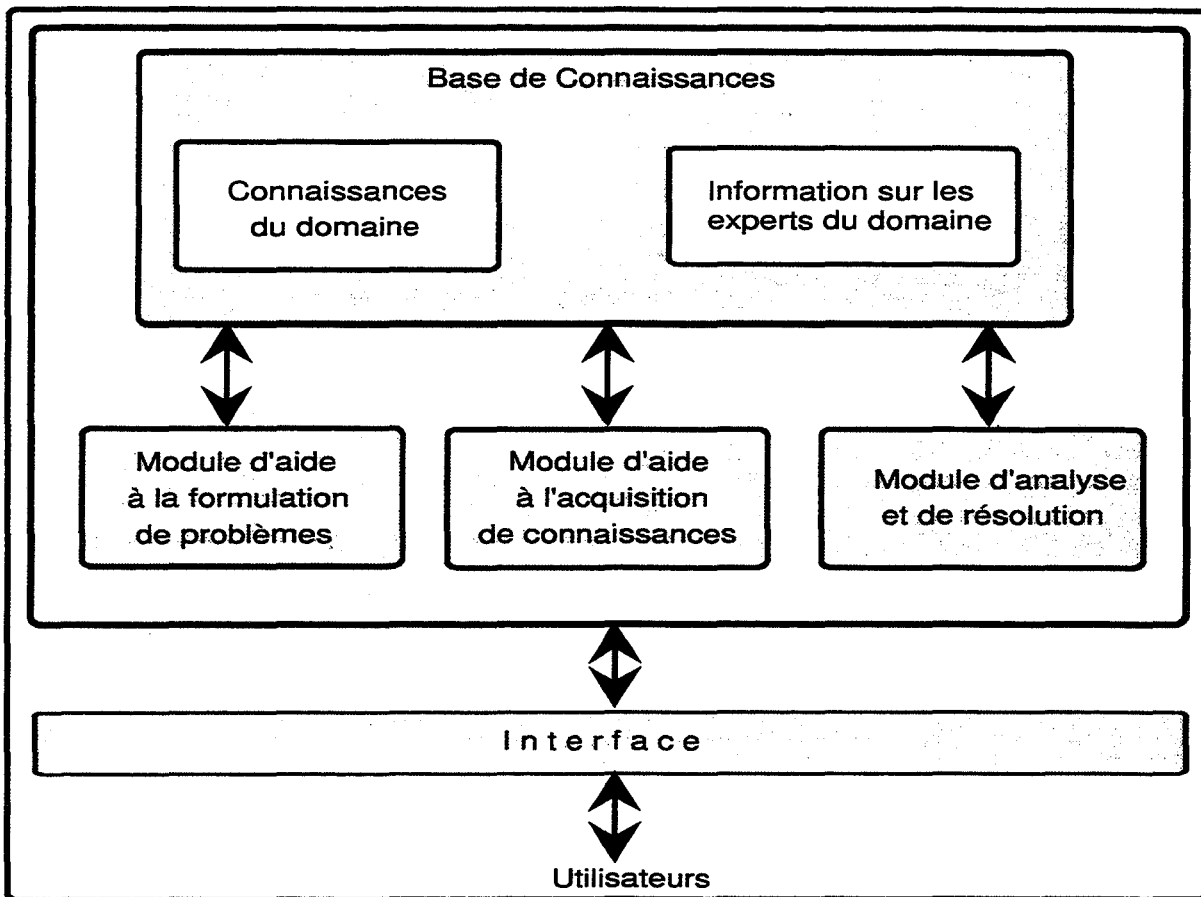


Figure 1.6 : Principaux éléments constituant le système MEDEXPEAU

Après avoir décrit l'architecture globale de MEDEXPEAU, nous présenterons la structure des connaissances du domaine et les types de liens qui existent entre ces connaissances.

### 1.4.3 La structure des connaissances

Dans le paragraphe 1.2.2, nous avons indiqué qu'une situation de crise doit pouvoir être décrite à différents niveaux de précision selon les informations dont nous disposons sur celle-ci. Ainsi les descriptions suivantes :

- "pollution des eaux",



- "pollution des eaux par des éléments chimiques",
- "pollution des eaux marines par des détergents, la pollution est provoquée par des eaux usées industrielles et les espèces touchées sont les mollusques et les algues".

montrent les différents niveaux de détail dans la description de situation de crise dans le domaine de l'eau. Certaines informations peuvent encore lui être apportées comme, par exemple, "l'usage de l'eau polluée (baignade, aquaculture, alimentation en eau potable, etc.)", ce qui permet de préciser davantage le problème en question.

Dans le but de pouvoir décrire des situations de crise sous de telles formes, nous avons dû hiérarchiser les connaissances du domaine. Rappelons que c'est l'une des raisons pour lesquelles nous avons opté pour l'utilisation d'une structure de type "réseau sémantique" [Barr 81], [Sabah 88], [Sabah 89]. Cette structuration permet l'accès aux connaissances du plus général au plus spécifique, elle est d'ailleurs compatible avec l'organisation des connaissances chez les experts humains. Les types de liens qui constituent les relations entre les noeuds du réseau sémantique représentant les connaissances du domaine de l'eau sont :

- le lien "**est-un**" : correspond à une typologie des connaissances. Par exemple, Rhône-Alpes (est-un) région,
- le lien "**partie-de**" : permet de décrire comment sont composés les éléments complexes en fonction d'autres éléments plus simples. Ainsi, ce type de lien va servir, par exemple, pour spécifier que l'Isère se situe en Rhône-Alpes : Isère (partie-de) Rhône-Alpes,
- le lien "**synonyme**" sert à représenter des relations de synonymie entre les connaissances du domaine (exemple : bassin\_versant "est synonyme de" aire\_de\_drainage, dragage "est synonyme de" curage),
- Le lien "**substitution**" sert à mettre en relation une connaissance donnée à d'autres connaissances qui peuvent lui être substituées du point de vue de la description des problèmes.

Ces différents types de lien permettent d'explicitier les relations qui existent entre les éléments de connaissance du domaine et de faciliter les reformulations de problèmes.

#### 1.4.4 Reformulation de problèmes dans le système MEDEXPEAU

Comme nous l'avons déjà indiqué auparavant, l'une des caractéristiques de notre système est "l'obligation de réponse". Cela signifie que les réponses du type "il n'y a aucun expert pouvant intervenir pour résoudre des problèmes engendrés par la situation de crise" ne sont pas acceptables. Dans de tels cas, le système doit proposer une liste d'experts qui ont un savoir et une expérience sur des situations de crise les plus similaires de celle décrite.

Le traitement d'un ensemble de situations de crise  $\{S_1, \dots, S_n\}$  similaires à une situation de crise donnée  $S'$  exige une possibilité de passage de la description de la situation  $S'$  à la description des situations similaires  $\{S_1, \dots, S_n\}$ . Avant d'expliquer comment peut se faire ce passage de la description d'une situation à la description d'une autre situation, nous allons définir d'abord ce que nous entendons par "situation similaire" à une autre situation dans notre cas d'application.

Nous dirons qu'une situation de crise  $S_1$  est similaire à une autre situation de crise  $S_2$  si les méthodes de résolution des problèmes de la situation  $S_2$  peuvent être appliquées à la résolution des problèmes de la situation  $S_1$ .

Les descriptions de situations de crise  $\{S_1, \dots, S_n\}$  similaires à celle spécifiée  $S'$  doivent donc pouvoir être déduites à partir de la description de  $S'$ . Pour cela nous avons prévu des mécanismes (substitution et généralisation) permettant de passer d'une description d'une situation à une autre qui lui est similaire.

##### 1.4.4.1 Le mécanisme de substitution

La substitution consiste à identifier les éléments de connaissances du domaine qui peuvent remplacer ceux appartenant à la description d'une situation de crise donnée dans le but d'obtenir une description d'une situation de crise similaire à la précédente.

*Définition :*

soit  $\mathbf{C}$  l'ensemble des connaissances du domaine,

Nous dirons que le descripteur  $\mathbf{A} \in \mathbf{C}$  est substituable par un autre descripteur  $\mathbf{B} \in \mathbf{C}$  si les connaissances requises pour intervenir sur les situations de crise dont la description est  $d(s) = (c_1, \dots, \mathbf{B}, \dots, c_n)$  sont suffisantes pour intervenir sur des situations de crise dont la description est  $d(s) = (c_1, \dots, \mathbf{A}, \dots, c_n)$ .

Dans certains cas, les substitutions ne sont valables que dans un contexte bien précis. Le contexte est défini par les éléments décrivant la situation de crise. Pour prendre en compte cette contrainte nous avons défini deux types de substitution :

#### 1.4.4.1.1 Substitution simple

Dans ce cas, un élément de connaissance peut être substitué par un autre auquel il est relié par un lien de type "substitution".

Ce mécanisme de "substitution simple" peut être formalisé comme suit :

Soient  $\mathbf{C}$  l'ensemble des connaissances du domaine et  $\mathbf{DS}$ , l'ensemble des connaissances appartenant à la description du problème.

Nous dirons qu'une connaissance  $d$  est substituable par une autre connaissance  $c$  si :

$d \in \mathbf{DS}$ ,  $c \in \mathbf{C} - \mathbf{DS}$  et il existe un lien de substitution entre  $d$  et  $c$ .

Exemple : si la description d'une situation de crise comporte le descripteur "aquifère superficiel", celui-ci peut être remplacé par le descripteur "galerie".

#### 1.4.4.1.2 Substitution conditionnelle

Dans ce cas, un élément de connaissance ne peut être substitué par un autre auquel il est relié par un lien "substitution" qu'à la seule condition que d'autres éléments définis

appartiennent à la description du problème. Cela nécessite des règles pour spécifier les conditions de cette substitution.

Ce mécanisme de "substitution conditionnelle" peut être formalisé comme suit :

Soient **C** l'ensemble des connaissances du domaine, **DS**, l'ensemble des connaissances appartenant à la description du problème, et **SE**, un sous ensemble de **C**.

Nous dirons qu'une connaissance **d** est substituable par une autre connaissance **c** sachant **SE** si :

$d \in DS, c \in C - DS, SE \subset DS$ , et il existe un lien de substitution entre **d** et **c**.

Exemple : si la description d'une situation de crise comporte le descripteur "lac", il peut être remplacé par le descripteur "retenue de barrage" s'il s'agit d'une pollution d'origine chimique par les nitrates, les phosphates et l'ammonium.

#### 1.4.4.2 Le mécanisme de généralisation

Dans le cas où même en appliquant le mécanisme de substitution, le système n'aboutit pas à un résultat positif, le mécanisme de généralisation est appliqué pour étendre le champ des solutions.

En pratique, une fois défini un langage de représentation des connaissances, deux étapes sont nécessaires pour définir une méthode de généralisation [Pottier 86] :

- définir l'expression "une connaissance est plus générale qu'une autre",
- définir un algorithme déterminant les connaissances qui sont plus générales que les connaissances initiales.

Dans les domaines où il est nécessaire de transformer des descriptions en d'autres plus générales, des règles de généralisation peuvent être utilisées [Mitchell 83], [Bollinger 86].

Nous citerons ici les règles les plus importantes :

- **La transformation de constantes en variables** : Suivant cette règle, la description "compétence (X)" est plus générale que la description "compétence (eaux-usées)".

- **La suppression de conjonctions (oubli de conjonctions)** : Suivant cette règle, la description "compétence(eaux-usées) & compétence(déchets-industriels)" peut être généralisée en la description "compétence(eaux-usées)".

- **La remontée dans des hiérarchies de concepts** : En utilisant les taxinomies de concepts représentant la connaissance du domaine.

Pour notre cas, comme la connaissance du domaine est représentée sous une forme structurée à l'aide d'un réseau sémantique, nous utilisons une sorte de généralisation qui consiste à remplacer les descripteurs du problème par d'autres descripteurs plus généraux en utilisant la règle de la remontée dans les hiérarchies de concepts. Le fait de procéder à une action de généralisation ne nécessite donc pas d'informations supplémentaires.

Supposons la hiérarchie illustrée par la figure 1.7 dans le domaine de l'eau (un sous ensemble de la taxinomie du domaine). Prenons un cas de crise dans lequel nous voulons remplacer le descripteur "Haute\_mer" par un autre descripteur plus général. Sans aucune connaissance de la structuration du domaine, nous ne pouvons pas trouver quel est le descripteur qui est plus général qu'un autre, mais si nous avons la hiérarchie (figure 1.7) qui permet de classer les connaissances du domaine, nous remarquons que nous pouvons généraliser le descripteur "Haute\_mer" par le descripteur "Eaux\_marines".

Nous aurions aussi pu généraliser le descripteur "Haute\_mer" par le descripteur "Eaux" mais le descripteur "Eaux\_marines" est plus spécifique.

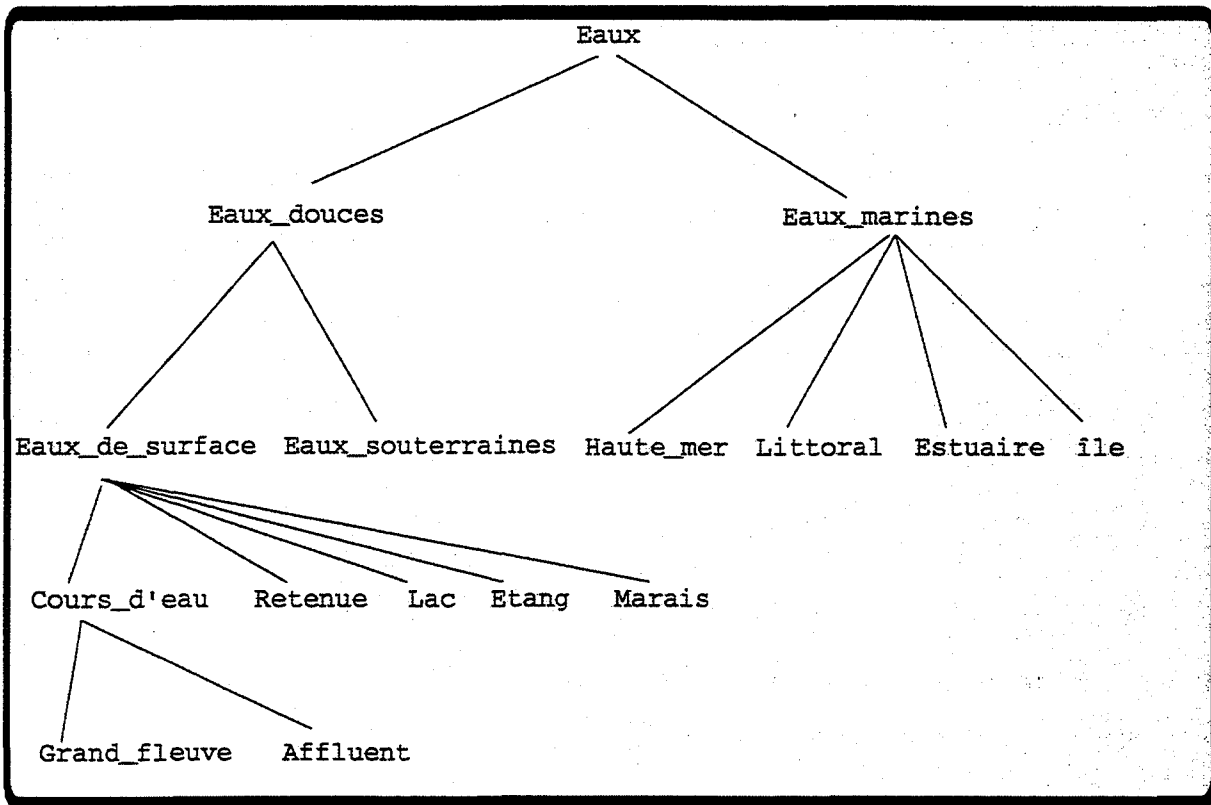


Figure 1.7 : Exemple de classification des connaissances du domaine de l'eau

## 1.5 BILAN

Le système MEDEXPEAU est réalisé en TURBO-C, il fonctionne sur les micro-ordinateurs IBM PC et compatible, sous le système d'exploitation MS-DOS. Il a été testé plusieurs fois pendant l'année 1992 par les différents experts et utilisateurs (Ecole des Mines de Saint Etienne, Institut Méditerranéen de l'Eau, Ministère de l'Environnement). Ces tests ont permis de montrer la faisabilité du SBC pour la recherche d'experts pouvant intervenir sur des situations de crise et de valider les fonctions d'un tel système. Ils ont permis aussi de révéler l'incomplétude et la difficulté d'actualisation de la BC dues principalement à une mauvaise organisation du processus d'acquisition de connaissances. En effet, tout au long de la réalisation du système MEDEXPEAU, notre démarche pour la construction de la BC a été empirique et ne s'est pas appuyée sur des méthodologies d'acquisition de connaissances ; mais plutôt sur l'expérience des différents participants à ce projet.

Ainsi, étant donné que le domaine d'application (le domaine de l'eau) est caractérisé par :

- des connaissances empiriques : les connaissances des experts de ce domaine sont fondées essentiellement sur une longue expérience et par conséquent, ces connaissances sont propres à chaque expert,
- une imbrication de plusieurs disciplines : le domaine de l'eau recouvre plusieurs disciplines (hydrologie, hydrogéologie, sédimentologie, pédologie, etc.), ce qui implique que les connaissances de ce domaine sont dispersées à travers plusieurs experts de formation et de culture différentes,
- une variation (nuance) géographique du sens des concepts : dans le domaine de l'eau, il n'existe pas une uniformité géographique de l'expertise, une technique utilisée, par exemple, pour lutter contre la désertification en France n'est pas appropriée à un pays d'Afrique du nord. En effet, la désertification en France consiste en un abandon d'un territoire par les hommes alors qu'en Afrique du nord, elle consiste en l'aridification et la disparition de végétation. Aussi, le concept de rivière en France est différent du même concept au Maroc du fait que les rivières sont pérennes en France et temporaires au Maroc. Par conséquent, la notion de régime hydraulique est différente dans les zones géographiques "France" et "Maroc",
- un savoir et un savoir-faire évolutif : l'apparition de nouveaux cas de crise dans le domaine de l'eau et la mise au point de nouvelles techniques pour y faire face font que les connaissances dans le domaine et le savoir faire des experts évoluent continuellement.

La construction de la BC du système nécessite l'intégration continue de nouvelles connaissances. Pour résoudre ce problème, l'utilisation d'une technique d'apprentissage automatique s'impose, afin de permettre une acquisition incrémentale et sans intermédiaire humain des connaissances dans ce domaine. Ainsi les experts n'auront pas à délivrer explicitement toutes leurs connaissances, mais fourniront des rapports sur les interventions qu'ils ont effectués sur des cas réels de crise. Ces rapports serviront alors d'exemples d'apprentissage pour permettre au système d'acquies automatiquement les connaissances des experts. En effet, comme les experts dans le domaine de l'eau appartiennent à différentes disciplines et possèdent une connaissance empirique (leur savoir est construit dans la pratique au fil des interventions qu'ils effectuent sur des situations de crise), certains liens interdisciplinaires ne peuvent être découverts qu'en examinant les interventions réelles d'experts.

Pour pouvoir prendre en compte l'évolution des connaissances dans le domaine de l'eau, la technique d'apprentissage choisie doit permettre un traitement incrémental des rapports d'intervention fournis par les experts. Comme les experts du domaine de l'eau appartiennent à différentes disciplines, les descripteurs utilisés pour décrire ces rapports d'intervention peuvent être différents d'une discipline à l'autre. Ainsi, le système d'apprentissage doit pouvoir utiliser des exemples d'apprentissage ayant des formats différents. C'est à dire qu'un expert appartenant à une discipline donnée peut utiliser des descripteurs qu'un expert d'une autre discipline n'aura pas à utiliser nécessairement. Le nombre de descripteurs utilisés dans un exemple d'apprentissage peut aussi varier d'une intervention à l'autre. En effet, si nous prenons le seul exemple des compétences qui sont mises en oeuvre au cours d'une intervention, il est clair que leur nombre dépend uniquement de la situation de crise. Cela signifie que l'intervention sur un cas de crise  $C_1$  peut nécessiter un nombre  $X_1$  de compétences ; alors que l'intervention sur un cas de crise  $C_2$  peut nécessiter un nombre  $X_2$  de compétences (avec  $X_1 \neq X_2$ ). Cette souplesse dans la description des exemples d'apprentissage permet donc de faciliter la tâche de description des cas d'intervention par les différents experts. Mais, même si le système d'apprentissage accepte un format variable des exemples d'apprentissage, il est tout de même indispensable de construire, d'abord, un langage de description adéquat. Ainsi, chacun des experts va utiliser une partie de ce langage qui est propre à la nature des interventions qu'il a effectué.

En plus du savoir faire associé aux interventions effectuées, les experts possèdent une importante connaissance descriptive du domaine et des connaissances stratégiques. Les connaissances descriptives du domaine vont permettre de représenter la sémantique des connaissances et les relations qui existent entre elles. Les connaissances stratégiques représentent les métaconnaissances qui vont être utilisées lors de la résolution de problèmes. Nous allons montrer dans le deuxième chapitre pourquoi la construction d'un langage de description et le recueil des connaissances descriptives et stratégiques du domaine exigent l'utilisation d'une méthodologie d'acquisition de connaissances.



## 1.6 CONCLUSION

Dans ce chapitre nous avons d'abord montré l'intérêt d'un SBC spécifique pour les situations de crise et nous avons décrit ensuite les fonctions que doit remplir un tel système.

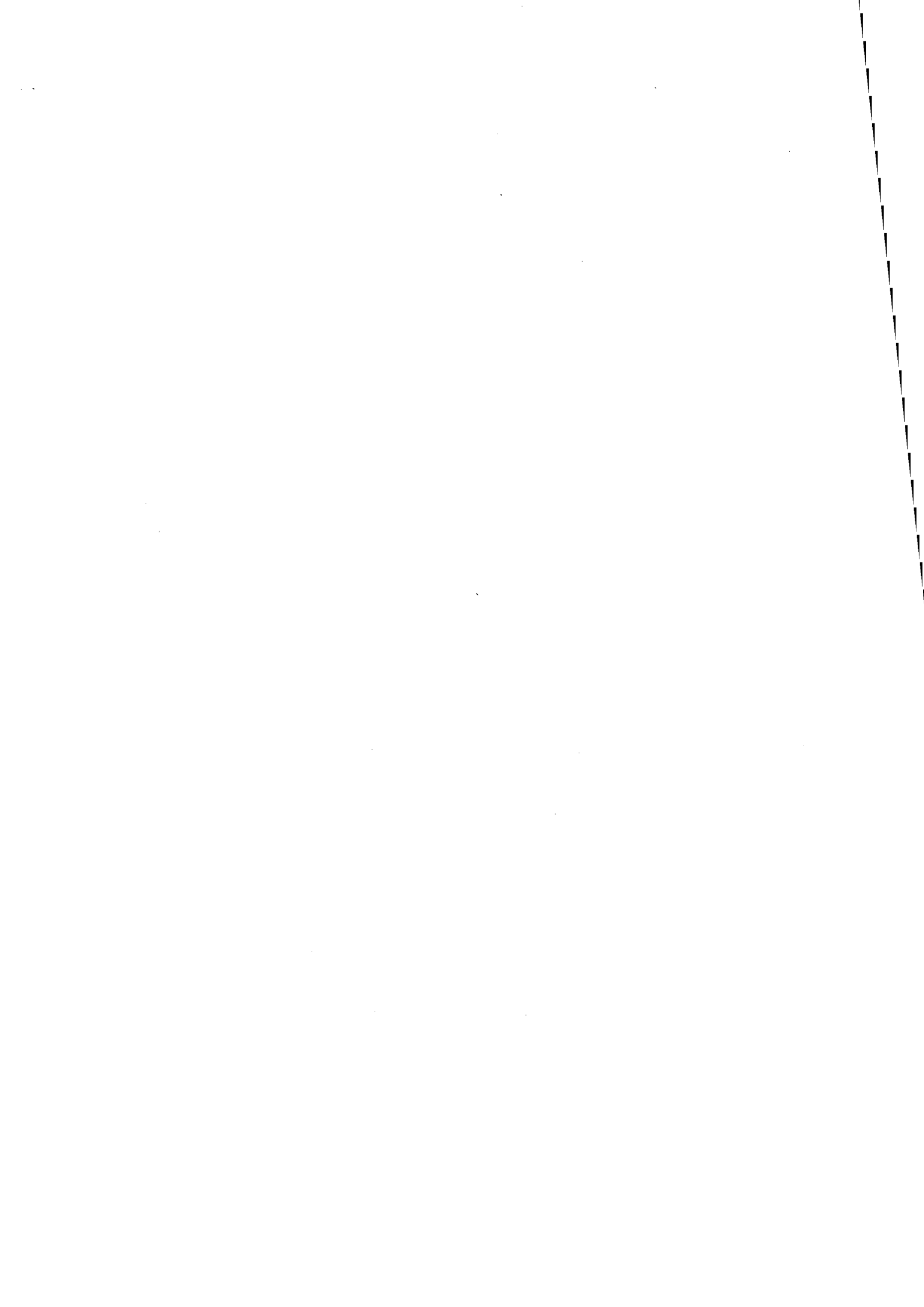
Le système développé n'a pas pour vocation de donner un ensemble de décision à appliquer en cas de crise, mais de proposer une liste d'experts les plus compétents vis-à-vis de la résolution de problèmes engendrés par une situation de crise. En effet, comme une situation de crise est par définition une situation imprévue pour laquelle il n'y a pas de procédures de résolution programmées à l'avance, il est indispensable de mobiliser rapidement des experts du domaine qui, eux, par leurs expériences et leur pouvoir de jugement stratégique peuvent trouver une solution aux problèmes posés.

Lorsqu'un utilisateur du système désire formuler une situation de crise, il utilise des éléments d'information contenus dans la BC. Le système permet un accès rapide et guidé à l'ensemble des connaissances du domaine pour expliciter le problème à résoudre. Ces connaissances sont présentées de manière "compréhensible" (en présentant à l'utilisateur les connaissances du domaine par niveau de spécification. C'est-à-dire que l'utilisateur commence à traiter des connaissances générales qui le conduisent aux connaissances spécifiques).

Comme nous l'avons indiqué, la description de situations de crise peut se faire à différents niveaux de précision. Pour aboutir à des résultats pertinents, il est nécessaire de faire ces descriptions de manière rigoureuse. Il est un fait que lorsque nous voulons être rigoureux, nous ne savons pas où il faut s'arrêter dans la description des circonstances : faut-il dire si la rivière polluée par des déchets industriels est lente ou rapide, poissonneuse ou non, importante ou indifférente pour tel secteur de vie, etc. Il est important de trouver un niveau de détail qui soit suffisant pour qu'une description correspondant à une situation de crise soit représentative sans être trop générale afin de ne pas couvrir un trop grand ensemble de situations. Car si la description est trop générale, nous aurons une solution étendue qui répond au problème posé sans avoir des résultats spécifiques.

Dans le chapitre 2, nous aborderons plus en profondeur le problème de l'acquisition de connaissances qui se pose dans tout projet SBC. Nous examinerons quelques méthodologies d'acquisition de connaissances et nous présenterons la démarche que nous allons adopter pour

recueillir les connaissances descriptives et stratégiques du domaine et pour construire un langage de description des cas d'interventions d'experts qui vont servir d'exemples d'apprentissage.



## ***2. Chapitre 2 : Méthodologie d'acquisition de connaissances***

### **2.1 PROBLEMES LIES A L'ACQUISITION DE CONNAISSANCES**

Pour résoudre des problèmes dans un domaine précis, l'être humain doit tout d'abord avoir une connaissance sur le domaine en question. Cette connaissance permet d'orienter le raisonnement vers le but à atteindre. Un domaine peut être essentiellement défini par des objets et des relations entre ces objets qui prennent un sens différent selon l'univers du problème (contexte de l'environnement de l'application) et de la finalité de leur utilisation. Ainsi, nous pouvons trouver des relations mettant en association des objets et des actions ou des relations entre des objets. En d'autres termes, nous pouvons dire que le raisonnement utilisé pour la résolution d'un problème s'effectue en manipulant des concepts abstraits qui modélisent les entités du problème et de son contexte.

Depuis longtemps, l'homme a eu l'idée de faire résoudre des problèmes par des machines automatiques. De la tentative de mise en pratique de cette idée, il s'est avéré que cela n'est

vraiment possible qu'en dotant ces dites "machines" de connaissances utilisées par les humains lorsqu'ils accomplissent des tâches données. Désormais tout développeur ou utilisateur d'un système intelligent sait que la capacité du système à résoudre des problèmes et à fournir des explications dépend des connaissances qu'il exploite : les concepts, les relations, les représentations, les méthodes et les heuristiques portant sur son domaine de travail. Ces connaissances sont obtenues par le biais d'un processus appelé "acquisition de connaissances" qui consiste à transférer des connaissances depuis les sources d'expertise (experts, documents, Bases de Données, etc.) vers le système informatique.

L'expertise détenue par les experts représente la connaissance la plus difficile à extraire et à modéliser dans un SBC. En effet, cette expertise est acquise par une longue pratique (on peut dire, en quelques sortes, que ces connaissances se sont formées à l'insu de l'expert). Ces connaissances sont utilisées comme des automatismes pour trouver rapidement une bonne solution à un problème donné. Puisque l'expert n'a pas un accès conscient à ces connaissances, leur explicitation par celui-ci n'est pas évident et leur acquisition représente donc une tâche délicate.

Le processus d'acquisition de connaissances renferme des étapes d'élicitation, d'analyse, de formalisation et de retranscription du savoir-faire dans un système de représentation. Ce processus s'avère complexe et peut produire des connaissances invalides dues principalement à la déformation des connaissances des experts. Cette déformation des connaissances peut être provoquée par :

- le décalage qui existe entre le langage utilisé par les experts pour délivrer leurs connaissances et les langages informatique disponibles,
- une interprétation invalide des connaissances des experts de la part du cognicien,
- une synthèse invalide des expertises émises par plusieurs experts (dans les domaines où il y a plusieurs experts).

Un autre problème auquel sont confrontés les développeurs de SBC est celui de l'oubli de la part des experts de délivrer une partie de leurs connaissances. Cela est dû au fait que l'expert ne se met pas toujours en situation réelle lorsqu'il fournit ses connaissances, par

conséquent il ne décompose pas explicitement toutes les étapes du raisonnement pour fournir la totalité de son savoir à propos de la tâche qu'il traite.

La confrontation à ces problèmes a donné naissance à des méthodologies d'acquisition de connaissances [Boose 86], [Eshelman 86], [Vogel 88], [Aussenac 89], [Dieng 90], [Brunet 91], [Compagnon 91], [Tong 92], [Zacklad 92], [Angele 93], [Neubert 93], [Poisvert 93], [Vicat 93], élaborées dans le but d'aider à mener correctement le processus de l'acquisition de connaissances et ainsi faciliter le développement de Bases de Connaissances.

Dans le paragraphe suivant, nous allons décrire une classification des différents modes de raisonnement et l'organisation des connaissances dans le système cognitif des experts.

## **2.2 LA PSYCHOLOGIE COGNITIVE ET LA MEMOIRE**

La psychologie cognitive se propose d'étudier l'organisation des connaissances, leurs enchaînements et leurs interactions lors de la résolution de problèmes par des experts [Benkirane 90]. Comme nous l'avons déjà mentionné, au cours de la résolution de problèmes, l'être humain raisonne sur des concepts abstraits qui modélisent les entités de l'univers du problème, en tire des conclusions qu'il interprète ensuite dans cet univers [Chouraqui 85]. Durant ce processus de résolution, l'expert met en oeuvre des connaissances de différents types selon la nature du problème à résoudre. Ainsi lorsqu'il résout un problème qu'il a déjà rencontré auparavant, il utilise des règles de décision qui sont des associations empiriques entre des conditions et une conclusion. En revanche, lorsqu'il résout un problème qu'il n'a jamais rencontré et lorsque la référence aux situations connues ne permet pas de prendre une décision, alors cette dernière est élaborée à partir d'un ensemble de connaissances plus structurées qui font partie de la théorie du domaine [Benkirane 90].

Ces modes de raisonnement ont été identifiés en ingénierie cognitive et en ergonomie par Rasmussen [Rasmussen 83] qui a défini un modèle de réaction en trois niveaux dans l'activité mentale humaine :

- le **niveau réflexe** : est le plus bas niveau dans lequel l'activité mentale consiste à sélectionner et à activer des schémas de résolution de problèmes grâce à une impulsion informationnelle provenant de l'environnement,
- le **niveau règle** : est le niveau intermédiaire dans lequel l'expert applique un ensemble de règles pour résoudre le problème lorsque la situation lui est familière ou préplanifiée,
- le **niveau connaissance** : est le plus haut niveau dans lequel l'expert ne sait pas immédiatement que faire face au problème qui lui est posé (lorsque la situation est nouvelle et/ou imprévue). Il adopte alors un comportement basé sur la connaissance en faisant recours à des connaissances profondes ou à des méta-connaissances. C'est ce niveau de connaissances qui est souvent mis en oeuvre par les experts pour résoudre des problèmes dans les situations de crise.

En ce qui concerne l'organisation des connaissances dans le système cognitif des experts, les travaux récents en psychologie cognitive ont montré que le processus de mémorisation n'est pas une accumulation passive d'informations, mais le résultat du traitement actif de connaissances par le système nerveux. La mémoire humaine n'enregistre pas les événements à la manière d'un magnétophone, mais trie, sélectionne, associe et recompose les informations qui lui parviennent. Les résultats expérimentaux de la psychologie et de la neurophysiologie montrent, par exemple, qu'il existe au moins trois niveaux différents de stockage de la connaissance [Velichkovsky 92], [Tiberghien 94] :

- la **mémoire procédurale** contient les processus automatiques et inconscients qui contrôlent l'usage des capacités acquises,
- la **mémoire sémantique** qui stocke les informations sur le sens des mots, les faits d'ordre général, les environnements particuliers et les autres formes usuelles de connaissances,
- la **mémoire épisodique** ou **autobiographique** qui stocke les événements et expériences spécifiques à l'individu.

L'information sémantique est indépendante du contexte dans lequel elle se manifeste ou dans lequel elle doit être retrouvée. Un expert du domaine de l'eau peut, par exemple, se

souvenir qu'un type de poisson est sensible à la pollution de l'eau, qu'il se trouve au bureau, en montagne ou au bord de la mer.

A l'inverse de l'information sémantique, la représentation épisodique dépend des conditions dans lesquelles elle s'est constituée. Un expert peut se souvenir plus facilement des actions nécessaires pour résoudre un problème en cas de crise s'il se trouve actuellement dans la même situation.

### 2.3 ORGANISATION DU PROJET "ACQUISITION DE CONNAISSANCES"

Tout projet informatique s'appuie sur une organisation définissant les acteurs qui y interviennent et délimitant leur champ d'action. Le projet "acquisition de connaissances" n'échappe pas à la règle et de plus, il nécessite l'implication d'autres acteurs qui ne sont pas toujours nécessaires dans les projets classiques. Ces nouveaux acteurs sont les experts et le cogniticien.

**Les experts** : sans experts dans un domaine, nous ne pouvons pas envisager le développement d'un SBC. Les experts sont des personnes ayant des connaissances et un savoir-faire acquis par la pratique sur des tâches et des problèmes du domaine.

**Le cogniticien** : le cogniticien est une personne qui intervient comme intermédiaire entre les experts et la machine pour améliorer le processus d'acquisition de connaissances dans un projet SBC. Il joue un rôle important pour amorcer le processus d'acquisition, et pour analyser les connaissances que livre l'expert au sujet d'un problème donné. Généralement, c'est au cogniticien de prouver l'utilité du projet et convaincre l'expert de s'y intéresser afin qu'il soit prêt à y consacrer du temps et à se rendre disponible. Le cogniticien doit donc avoir des qualités parmi lesquelles nous pouvons citer :

- **un sens de la communication** : le cogniticien a un rôle important à jouer dans l'établissement d'une bonne relation entre les différents intervenants dans le projet. Il doit



utiliser des techniques de communication pour amener les experts à fournir le plus de connaissances possible sur le domaine,

- **une capacité d'adaptation à différents domaines** : lorsque le cogniticien prend en charge un nouveau projet, il doit se familiariser avec la terminologie utilisée dans ce domaine pour pouvoir communiquer avec les experts en utilisant leur langage d'expression. Il doit aussi avoir quelques connaissances de base du domaine considéré comme, par exemple, les types de problèmes rencontrés et les méthodes utilisées, qui lui permettront de poser les bonnes questions et d'orienter le dialogue,
- **une grande patience** : le cogniticien doit apprécier les problèmes auxquels l'expert est confronté, il doit discuter avec lui et demander des éclaircissements sans avoir l'air de critiquer. Si l'expert semble être hésitant pour donner une connaissance, le cogniticien doit montrer suffisamment de patience,

Selon le type de connaissances à acquérir et le degré de motivation des experts du domaine pour le développement d'un SBC, le cogniticien peut prendre une place différente dans la phase d'acquisition de connaissances. Dans la suite, nous donnerons les principaux modèles d'organisation utilisés.

### 2.3.1 Le modèle traditionnel

Dans ce modèle, c'est le cogniticien qui est chargé de recueillir le savoir des experts du domaine pour les structurer ensuite dans la Base de Connaissances. Afin de ne pas perdre de vue les besoins des utilisateurs potentiels du système à développer, le cogniticien travaille en collaboration avec eux. Comme le montre la figure 2.1, c'est le cogniticien qui joue le rôle principal. Il doit non seulement faire réagir les experts de façon à répondre aux exigences des utilisateurs, mais aussi aider les utilisateurs à énoncer leurs besoins de manière à éviter les retours en arrière pour corriger les insuffisances dues à l'oubli de spécification de quelques fonctions.

Dans un tel modèle, le cogniticien qui a une expérience dans le domaine de l'acquisition de connaissances va construire un modèle de connaissances accessible aux experts qui n'ont pas participé à cette étape ; ce qui facilite la maintenance du système. Mais il existe un risque d'aboutir à un ensemble de connaissances biaisées, dû à une mauvaise interprétation par le cogniticien des connaissances de l'expert.

Ce modèle est conseillé lorsque les experts ne sont pas très coopératifs et semblent hésitants face à la décision de développement d'un SBC .

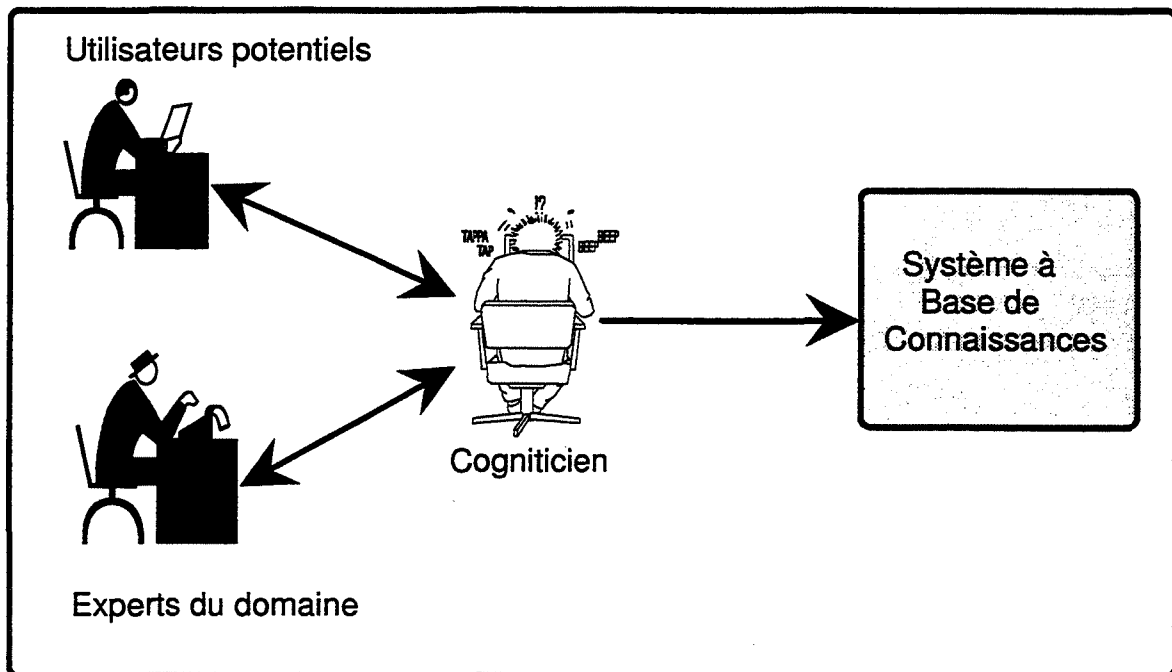


Figure 2.1 : Le modèle traditionnel

### 2.3.2 Le modèle direct

Le rôle du cogniticien est supprimé dans ce modèle et l'expert délivre directement ses connaissances au SBC. L'engagement des experts du domaine est la clé de réussite d'un projet basé sur ce modèle. L'expérience a montré que lorsque les experts expriment leurs connaissances, il est préférable de leur accorder une grande autonomie et de leur permettre

ainsi d'intervenir directement dans la construction de la BC. Dans une telle organisation, des problèmes peuvent se poser dès lors que les experts chargés de la construction de la BC abandonnent le projet en cours de réalisation. Ceux qui seront chargés de reprendre et de compléter la BC trouveront des difficultés pour décoder les connaissances qui sont déjà intégrées dans le système. La maintenance du système obtenu sera difficile à assurer par des acteurs autres que ceux chargés de son développement.

Ce modèle qui permet de mettre au point rapidement une BC est conseillé lorsque les experts sont très coopératifs et possèdent une expérience dans le développement de SBC. Néanmoins, le système doit assurer grossièrement les fonctions du cogniticien, c'est à dire qu'il doit inciter les experts à fournir des connaissances complètes et bien structurées à travers une interface évoluée.

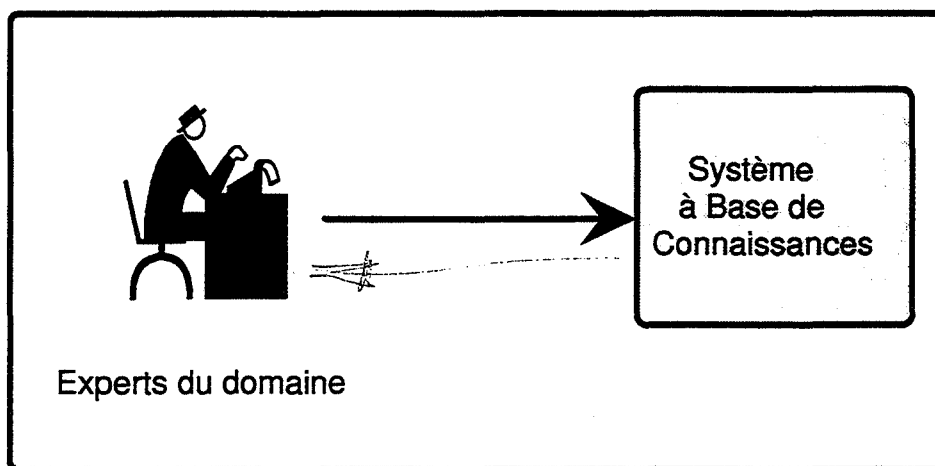


Figure 2.2 : Le modèle direct

### 2.3.3 Le modèle tutoriel (guidé)

Dans ce modèle, le cogniticien agit comme tuteur aidant les experts à fournir leur savoir et les conseille sur la représentation des connaissances qui correspond le mieux à leur retranscription. Ce modèle cherche à réduire l'intervention du cogniticien dans les étapes où il est préférable de laisser les experts s'exprimer sans les contraindre, mais le cogniticien peut

intervenir pour gérer les désaccords sur la formalisation des connaissances, conseiller, orienter les experts et donner son avis sur les résultats obtenus.

Ce modèle est conseillé lorsque les experts sont coopératifs mais manquent d'expérience sur le développement de projets SBC.

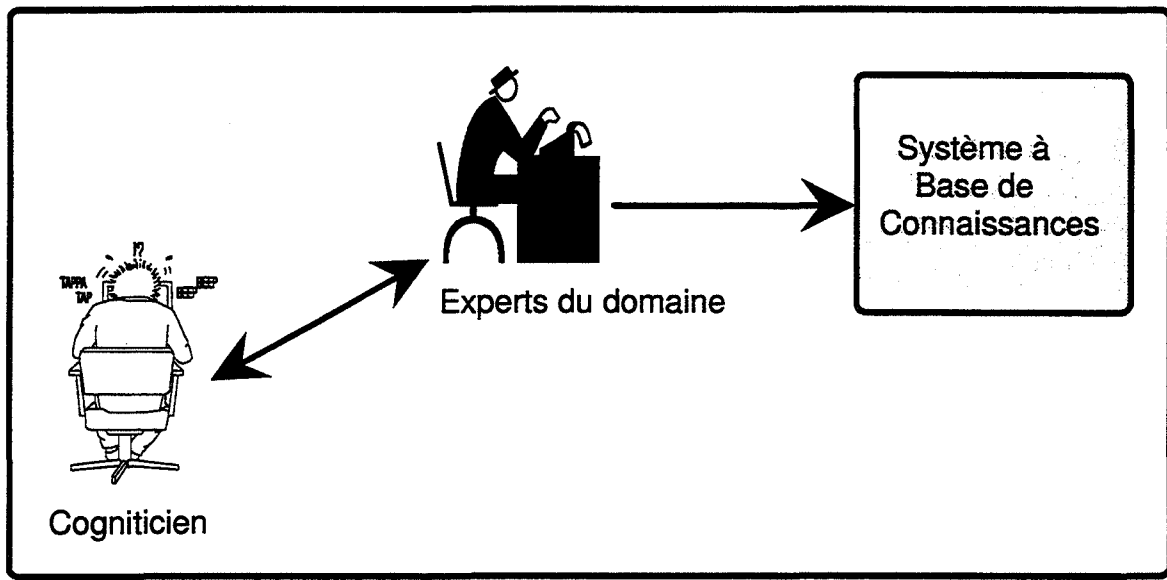


Figure 2.3 : Le modèle tutoriel

Toutefois, même en adoptant un modèle d'organisation approprié au domaine d'application, le problème de l'acquisition de connaissances n'est pas totalement résolu. Dans le paragraphe qui suit, nous montrerons qu'il est nécessaire de construire un modèle des connaissances appelé "modèle cognitif", destiné à servir d'intermédiaire entre l'expression brute des connaissances et leurs spécifications informatiques.

## 2.4 ETAPES DE DEVELOPPEMENT D'UN SBC

### 2.4.1 Approche par "prototypage rapide"

Les premiers projets SBC ont été développés selon les méthodes appliquées aux projets informatiques classiques, en adoptant une approche par "prototypage rapide". Dans ce cas,

l'expert formule ses connaissances et une transposition en représentation informatique est effectuée en tenant compte de l'environnement logiciel choisi a priori [Dieng 90]. Un projet basé sur une telle approche suit les étapes suivantes :

- **Identification** : cette étape permet de mettre en place les personnes concernées et les ressources nécessaires. Elle permet aussi de préciser les caractéristiques du problème et les objectifs attendus.

- **Développement en quelques mois d'une maquette** : cette étape permet de conceptualiser le problème (à partir d'entretiens cognicien-expert, explication des principaux concepts, relations et stratégies de résolution du problème), de formaliser les connaissances (représentation dans un formalisme), de les implanter dans un outil choisi et enfin de tester la maquette pour la valider.

- **Développement d'un système complet** : cette étape permet de compléter ou de modifier la BC de la maquette.

- **Validation du système complet** : à ce niveau, le système est testé et évalué par les experts et les utilisateurs.

- **Intégration dans l'environnement d'application et maintenance** : consiste à exploiter le SBC et à assurer sa maintenance.

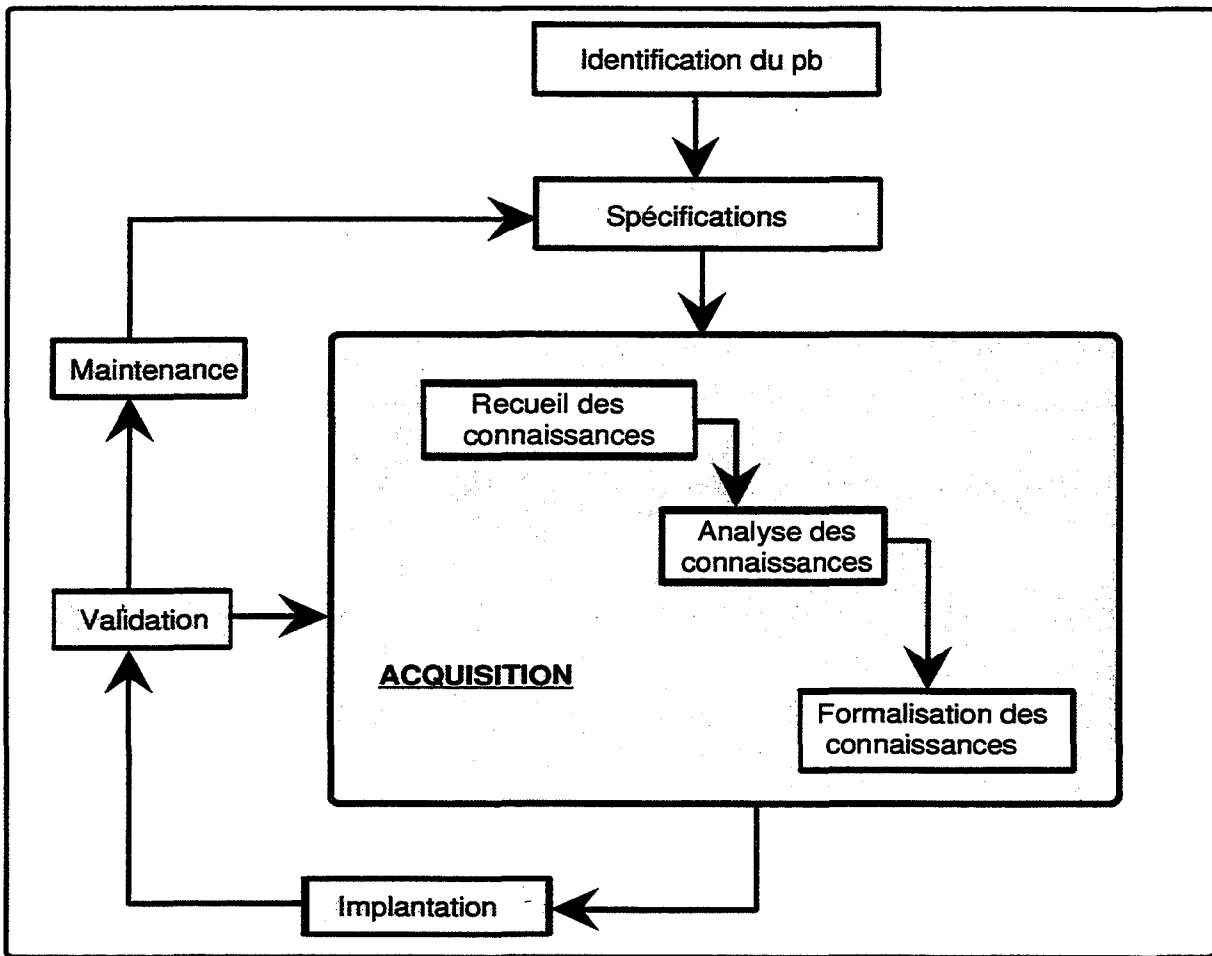


Figure 2.4 : Cycle de vie d'un SBC - l'approche par prototypage rapide -

Les outils développés en suivant cette approche souffrent de deux principales lacunes à savoir :

- La difficulté pour les non-spécialistes des techniques des SBC d'accéder aux connaissances constituées dans la BC,
- La difficulté de maintenance de l'outil, due principalement à une mauvaise structuration des connaissances.

## 2.4.2 Approche "structurée"

Pour surmonter les lacunes de l'approche par "prototypage rapide", les chercheurs dans ce domaine ont vite eu conscience de la nécessité de conceptualiser les connaissances et de passer par une représentation intermédiaire (se situant entre le niveau linguistique et le niveau symbolique d'implémentation). A. Newell [Newell 82] est le premier à avoir introduit la notion de *knowledge level*. Cette notion consiste à décrire les connaissances du domaine étudié à un niveau conceptuel abstrait, sans tenir compte des considérations informatiques.

La figure 2.5 montre la place du niveau "connaissances" dans l'environnement d'acquisition de connaissances.

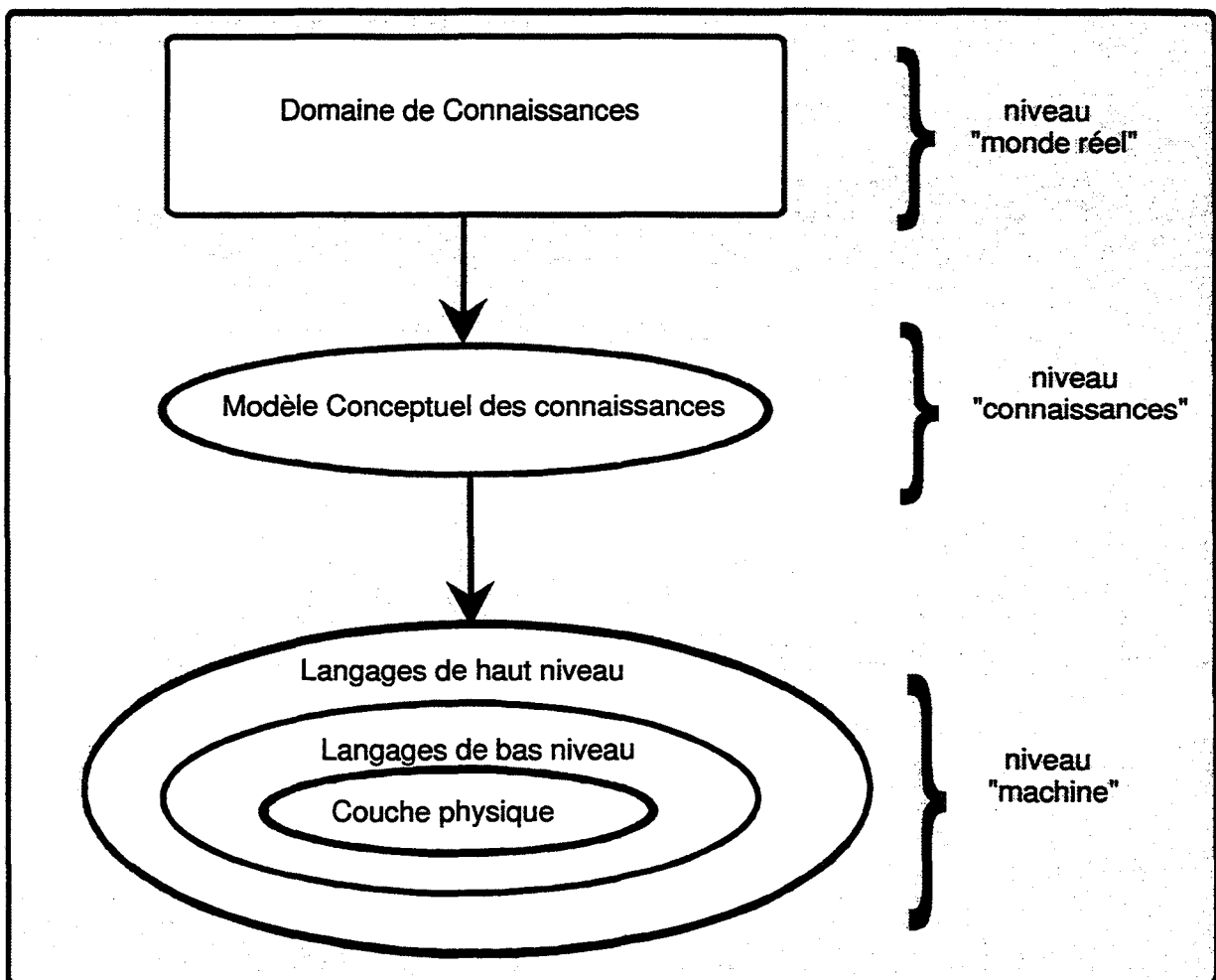


Figure 2.5 : Place du niveau "connaissances"

Cette approche considère donc la tâche d'acquisition de connaissances comme une activité de modélisation. Un projet basé sur cette approche contient les phases suivantes :

- **Identification du problème** : Il s'agit d'identifier les besoins et de préciser les caractéristiques du problème et les objectifs attendus.

- **Modélisation de la connaissance** : cette phase permet de recueillir la connaissance, de l'analyser et de l'organiser pour aboutir à un modèle de connaissance. La conception du système se base alors sur le modèle de connaissance obtenu. Dans cette étape, nous ne tenons pas compte des considérations informatiques, ainsi nous avons plus de liberté pour modéliser.

- **Conception du système** : cette phase permet de produire une architecture logicielle qui réalise les fonctions spécifiées par le modèle des connaissances.

- **Implantation et tests** : il s'agit de retranscrire la connaissance dans un langage de représentation et d'effectuer des tests qui consistent à valider les résultats de l'implantation.

- **Intégration dans l'environnement d'application et maintenance** : il s'agit d'exploiter le SBC et d'assurer sa maintenance.

Cette approche qui s'appuie sur une analyse approfondie des connaissances se fixe comme objectif la construction d'un modèle conceptuel d'expertise exprimant la sémantique des connaissances ainsi que les principes et les mécanismes de base du raisonnement. Le modèle de connaissances construit offre plusieurs avantages parmi lesquels : la possibilité de valider les connaissances indépendamment de l'implantation informatique et son utilité de vecteur de communication entre les différents participants au projet [Curtis 92].



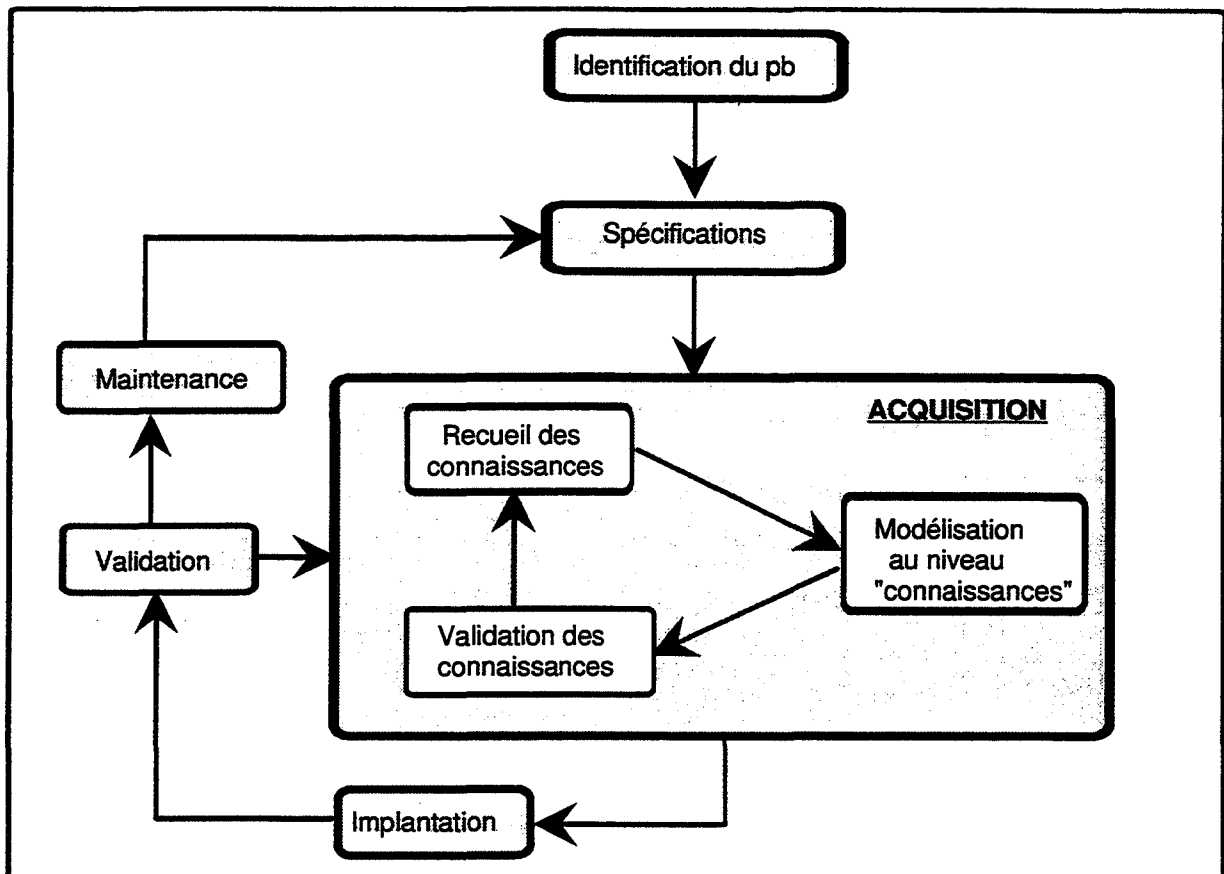


Figure 2.6 : Cycle de vie d'un SBC - l'approche structurée -

Mais avant de pouvoir construire un modèle de connaissances nous devons d'abord recueillir l'expertise pour pouvoir l'analyser et la structurer. Le résultat du recueil de connaissances dépend en partie des techniques utilisées pour favoriser l'expression des experts, car il y a des techniques qui sont plus adaptées que d'autres pour le recueil de certaines formes de connaissances. Parmi les différentes techniques qui sont utilisées pour faciliter l'identification de connaissances nous pouvons citer [Hart 88], [Aussenac 89], [Dieng 90] :

- **Consultation de documents** : le cogniticien lit des documents conseillés par l'expert (documents généraux sur le domaine, documents particuliers rédigés par l'expert, etc.), et se familiarise avec le vocabulaire du domaine.

- **Entretiens informels** : A partir d'entretiens d'information et de présentation générale de l'IA et du domaine d'expertise ressortira une démarche générale ainsi que les principales

tâches à traiter. Les entretiens informels sont indispensables pour amorcer le dialogue entre expert et cogniticien.

- **Observation directe de l'expert en situation** : le cogniticien observe l'expert en train d'exercer son activité dans son environnement habituel et note les actions de l'expert, sa démarche générale, les cas et types de problèmes traités. Cette technique est un bon moyen d'accéder aux connaissances de contrôle qui échappent aux verbalisations.

- **Entretiens centrés** : l'expert et le cogniticien dialoguent pour préparer l'extraction des connaissances (à partir de cas observés, de notes prises au cours des observations, des opinions de l'expert) en faisant ressortir un échantillon de cas représentatif de l'expertise et des conditions dans lesquelles vont être recueillis les protocoles (conditions habituelles, simulation, etc.).

- **Grilles-répertoires** : la grille répertoire est une représentation de la façon dont l'expert appréhende un problème particulier [Hart 88]. Elle permet de représenter la vision qu'a une personne d'un problème. Cette méthode est utile dans les premiers stades de l'extraction du savoir, en particulier si l'expert n'arrive pas à expliquer clairement ce qu'il pense. Elle soulève néanmoins des critiques du fait que les résultats seront très subjectifs. Deux experts, face au même problème, peuvent aboutir à deux ensembles de résultats complètement différents.

Après avoir noté que le processus d'acquisition de connaissances pour le développement de SBC est une tâche complexe et doit être abordé de manière rigoureuse, nous présenterons dans le paragraphe qui suit quelques méthodologies développées pour faciliter ce processus de transfert de connaissances.

## 2.5 QUELQUES METHODOLOGIES D'ACQUISITION DE CONNAISSANCES

Les méthodologies d'acquisition de connaissances ont pour objectif de faciliter le transfert de connaissances depuis des experts vers des systèmes informatiques. Elles s'appuient sur des techniques spécifiques pour mener à bien les phases propres aux SBC et notamment,

l'identification, le recueil et la modélisation des connaissances. Le point fort de ces méthodes consiste en leur contribution à construire un modèle de connaissances (le knowledge level) intermédiaire entre le monde réel et le monde symbolique. Ce modèle permet une meilleure appréhension et une meilleure validation des connaissances acquises.

Parmi les méthodologies d'acquisition de connaissances les plus avancées nous pouvons citer MACAO, KADS, KOD que nous présenterons dans la suite de ce paragraphe. Ces méthodes ont comme principe la séparation des phases d'analyse et de conception pour aboutir à la construction d'un modèle de connaissances fiable.

## 2.5.1 MACAO

La méthodologie MACAO (Méthodologie d'Acquisition de Connaissances Assistée par Ordinateur) [Aussenac 89] a été développée dans le souci d'aider le cogniticien et l'expert à travailler en collaboration pour aboutir à de meilleurs résultats. Elle représente à la fois une méthode complète de développement d'un SBC et un outil supportant cette méthode.

L'aide apportée à l'expert par MACAO, consiste à faciliter l'expression directe de ses connaissances lorsque cela est nécessaire pour éviter les contraintes dues à la subjectivité du cogniticien et limiter ainsi une partie des biais habituellement introduits par l'interprétation des connaissances par ce dernier. MACAO apporte aussi une aide importante au cogniticien dans les processus d'analyse et de structuration des connaissances fournies par l'expert.

### 2.5.1.1 Les étapes de MACAO

La méthode MACAO se déroule en quatre étapes principales : • identification de l'expertise, • recueil de protocoles, • analyse des protocoles, • validation des connaissances structurées par l'expert.

**Identification de l'expertise :** cette étape consiste en un dialogue expert-cogniticien et en l'observation de l'expert en situation réelle de résolution de problèmes. L'expert classe les

différents problèmes par catégories, en utilisant les techniques de "grilles-répertoires". L'identification de l'expertise permet d'éviter l'oubli d'une partie des connaissances, un malentendu de départ entre expert et cogniticien [Agapeyeff 88], [Aussenac 89], une réaction de méfiance ou même de refus de coopération de la part de l'expert ou encore la mise en route d'un projet irréaliste. Le cogniticien doit chercher non seulement à identifier l'expertise, mais il doit aussi vérifier que tous les éléments sont favorables à la réussite du projet avant de s'y engager.

**Recueil de protocoles :** cette étape consiste à recueillir des connaissances liées à la mise en oeuvre de l'expertise utilisée en résolution de problèmes et des explications sur ces résolutions. L'expert saisit au clavier les informations qu'il traite et commente oralement ce qu'il fait pour l'enregistrer sur un magnétophone. A ce niveau, le cogniticien peut poser des questions pour orienter les verbalisations et recueillir des informations plus précises. A la fin de cette étape, un **glossaire** est constitué à partir de l'ensemble des mots inscrits dans le logiciel par l'expert au cours de la résolution du problème.

**Analyse des protocoles :** à ce niveau le cogniticien est chargé d'analyser, de structurer et de formaliser les connaissances de l'expert dans un mode de représentation intermédiaire. Les données de l'expert sont transcrites dans un formalisme interne proche des réseaux sémantiques et l'expert intervient alors pour valider, confirmer ou modifier les résultats de l'analyse.

**Validation interactive des connaissances par l'expert :** une fois les connaissances structurées par le cogniticien, l'expert va tenter de les valider. Les connaissances recueillies sont présentées à l'expert sous forme de graphes au cours d'entretiens centrés afin de pouvoir détecter facilement toutes les informations manquantes. Ainsi, un dialogue s'établit entre l'expert et le cogniticien pour justifier le choix des connaissances à représenter et la façon de les formaliser. Dans un premier temps le cogniticien intervient pour justifier sa formalisation puis, il est souhaitable que l'expert réagisse librement, fournisse des explications complémentaires sans intervention du cogniticien.

### **2.5.1.2 Conclusion**

L'ensemble des connaissances structurées sur l'expertise acquise au niveau du recueil de protocoles permet au système de fournir des explications. L'utilisation d'un logiciel pour garder les traces des résolutions issues de la phase du recueil de protocoles évite les biais habituels de l'analyse de protocole comme l'interprétation subjective des comportements par le cogniticien ou l'utilisation de termes étrangers au vocabulaire du domaine. La validation des connaissances peut se faire directement par l'expert en manipulant un graphe, ce qui facilite la détection d'oublis et d'incohérences dans les connaissances.

L'inconvénient de cette méthode est que le recueil et l'analyse de protocoles peuvent être longs et conduire à des données incomplètes. La méthode KADS que nous allons décrire dans le paragraphe qui suit, propose un cadre structuré en quatre niveaux pour orienter et assister l'expert et le cogniticien dans les tâches de recueil et d'analyse de connaissances ; ce qui comble les lacunes de la méthode MACAO.

### **2.5.2 KADS**

Développée dans le cadre d'un projet ESPRIT (n° 1098), la méthode KADS (Knowledge Acquisition and Design Support) a comme objectif d'offrir un support pour le développement commercial de SBC [Brenker 85], [Brunet 90], [Brunet 91], [Krisch 93]. Pour cela, elle propose un cycle de vie complet, un ensemble d'outils conceptuels pour la modélisation des connaissances et un support logiciel, l'atelier SHELLEY [Bouchet 89].

Le processus de développement du SBC est décomposé dans KADS en la construction successive de plusieurs modèles : modèle d'expertise, modèle de coopération entre le futur système et ses utilisateurs, modèle externe qui intègre les contraintes qui vont influencer sur la conception et la réalisation du système final, modèle de conception fonctionnelle, modèle logique puis modèle physique du système. Une bibliothèque de modèles de tâches génériques, appelés modèles d'interprétation permet une acquisition structurée des connaissances.

La définition du modèle d'expertise dans KADS s'appuie sur une structure à quatre niveaux.

### 2.5.2.1 Les primitives de modélisation KADS

Afin d'aboutir à un modèle de l'expertise représenté à un bon niveau d'abstraction, KADS propose un cadre sémantique organisé à travers une structure à quatre niveaux (figure 2.7).

**Le niveau Domaine** décrit la connaissance statique du domaine (indépendamment de son utilisation) en termes d'objets, de concepts, d'attributs et de relations entre ces divers éléments.

**Le niveau Inférence** décrit les inférences qui peuvent être faites sur les connaissances du niveau précédent. Il contient de la méta-connaissance sur le niveau domaine, décrivant l'ensemble des fonctions permettant de faire des inférences lors de la résolution de problèmes. Ces fonctions sont appelées *sources de connaissances* et leurs arguments *méta-classes*. Une méta-classe indique comment un ensemble spécifique de concepts du domaine peut être utilisé.

**Le niveau Tâche** décrit comment appliquer les inférences possibles (les étapes du raisonnement) en fonction des buts. Mettre en application une structure d'inférence signifie, d'une part, préciser comment elle sera parcourue et, d'autre part, associer les éléments de contrôle nécessaires.

**Le niveau Stratégique** contient les connaissances qui conduisent à l'élaboration du plan d'action. Il décrit le plan de résolution du problème posé et les connaissances nécessaires au contrôle de celui-ci. Un plan de résolution précise l'ordonnancement des buts à atteindre pour résoudre le problème et ainsi formalise le comportement de résolution du problème. Par exemple, nous y précisons comment résoudre le problème, réaliser le contrôle de l'exécution des buts, réagir en cas d'impasse.

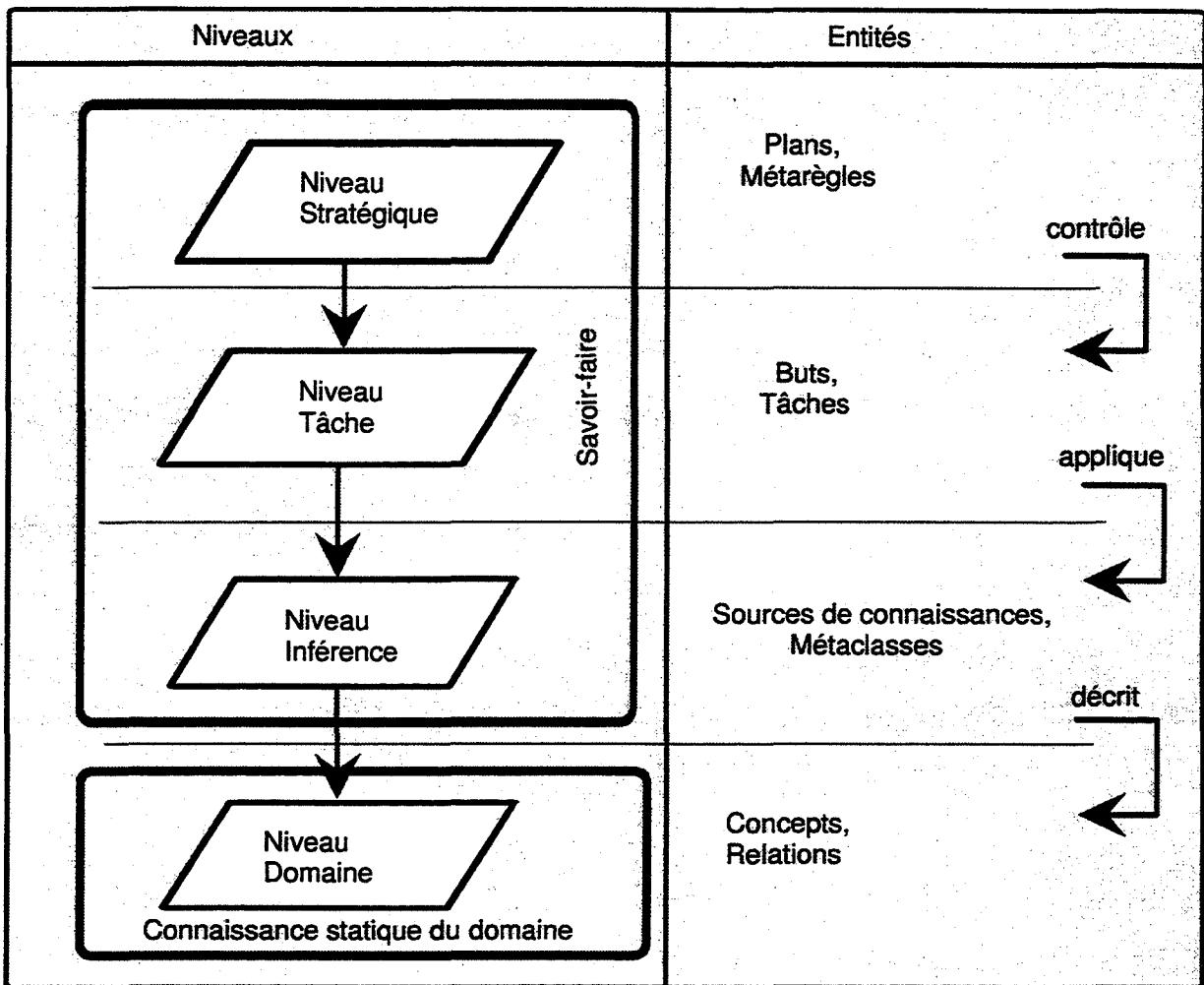


Figure 2.7 : Le modèle d'expertise à quatre niveaux de KADS

Afin de pouvoir identifier ces éléments, KADS offre :

- des techniques d'analyse de protocoles pour identifier les éléments du niveau domaine,
- un ensemble de "Knowledge Sources", primitives d'inférence, pour l'identification des connaissances du niveau inférences,
- une bibliothèque de modèles d'interprétation, qui sont des modèles génériques pour la résolution d'un type de problème particulier (typologie de problème) et qui servent de cadre pour l'élaboration du modèle des tâches et de guide pour les séances de recueil des connaissances [Brenker 85].

Le niveau stratégique ne bénéficie, quant à lui, d'aucun outil. Son identification étant laissée à l'initiative du cognicien.

### 2.5.2.2 Conclusion

La méthode KADS repose sur la compréhension des connaissances et des mécanismes de raisonnement de l'expert lors de la réalisation d'une tâche d'expertise. Pour faciliter l'analyse des connaissances, KADS offre une structure à quatre niveaux pour formaliser les connaissances et une bibliothèque de modèles d'interprétation. L'utilisation des modèles génériques, qui sont définis comme des typologies d'éléments de base et de relations de structuration pour certaines classes de problèmes, permet de faciliter la construction d'un modèle d'expertise (le cognicien travaille par affinements successifs des éléments des structures génériques). Le recueil des données est alors guidé par les modèles. Cependant, l'approche "Model Driven" transforme l'activité d'analyse en une activité de choix de modèles qui nécessite une bonne expérience de manipulation des modèles d'interprétation.

Nous pouvons constater qu'à la fin de la construction des différents modèles (modèle d'expertise, modèle de coopération, modèle externe), KADS ne propose aucun lien pour passer des différents niveaux de connaissances vers les langages de représentation disponibles actuellement. Dans le paragraphe qui suit, nous présenterons la méthode KOD qui, elle, spécifie les langages correspondants à chaque type de connaissance obtenu par cette méthode.

### 2.5.3 KOD

KOD (Knowledge Oriented Design) est une méthode d'acquisition et de formalisation de connaissances mise au point par Claude Vogel [Vogel 88], [Vogel 89], [Albert 90]. Contrairement à KADS, le cycle de vie de KOD ne couvre que l'aspect Base de Connaissances. L'atelier qui lui est associé comme support logiciel est la K-STATION. La méthode KOD sépare le transfert d'expertise en deux étapes séquentielles à savoir :

**L'analyse cognitive** : consiste à comprendre et à modéliser le savoir de l'expert. Cette étape s'appuie sur l'utilisation de techniques issues des travaux en linguistique et en anthropologie pour le découpage et l'analyse du discours de l'expert.



**La réalisation informatique** : consiste à développer le SBC proprement dit en utilisant les résultats obtenus lors de l'analyse cognitive.

### 2.5.3.1 L'analyse cognitive

La méthode KOD s'intéresse au recueil des connaissances d'un expert, à leur modélisation à partir de la retranscription systématique des entretiens avec l'expert, puis à la spécification de la Base de Connaissances du futur système selon une analyse en plusieurs étapes des données verbales. Pour mener l'analyse cognitive, KOD propose une décomposition en trois étapes donnant lieu à trois niveaux de modèles : (le modèle pratique, le modèle cognitif et le modèle informatique).

- **le modèle pratique** : permet de définir des dictionnaires à partir d'une analyse textuelle des entretiens. Cette étape consiste à analyser le discours de l'expert en identifiant les termes pertinents et en les étiquetant selon une typologie clairement définie : les **taxèmes**, les **actèmes** et les **inférences**.

Les **taxèmes** désignent les objets du monde physique. Les **actèmes** décrivent le changement d'état causé par un sujet sur un destinataire. Les **inférences** permettent d'établir de nouvelles propositions à partir de celles déjà acquises. Une inférence produite par l'expert dans un contexte particulier fait appel à des schémas d'interprétation, relativement indépendants du contexte. Destinées au cognicien, les chaînes d'inférence sont énoncées pour justifier ou expliquer une interprétation ou une prévision.

- **le modèle cognitif** : permet la structuration des entités précédemment identifiées. Il se compose de trois structures principales : une **taxinomie** qui organise les taxèmes selon la relation "est-un", des **actinomies** qui articulent les décompositions et les séquences d'actèmes et des **schémas d'interprétation** qui regroupent les inférences pouvant être activées dans un même contexte.

Contrairement à la construction du modèle pratique, la démarche suivie pour l'élaboration du modèle cognitif peut demander un certain travail d'interprétation de la part du

cogniticien. Ainsi, la création de la taxinomie peut nécessiter l'introduction d'entités qui n'apparaissent pas explicitement dans le discours de l'expert. A chacun de ces niveaux, l'analyse prend en compte trois paradigmes de représentation : la description statique (**taxèmes, taxinomies**), les actions dynamiques (**actèmes, actinomies**) et l'interprétation ou modélisation déclarative (**schémèmes, schémas**).

• **le modèle informatique** : décrit le système exécutable. La taxinomie devient une hiérarchie d'objets, les actinomies donnent les méthodes, et les schémas donnent des règles. Vogel considère que les différents modèles de programmation actuels se réfèrent aux paradigmes suivants :

- la programmation par objets qui permet d'exprimer les données et les taxinomies,
- la programmation par acteurs qui permet d'exprimer les traitements et les actinomies,
- la programmation par la logique qui permet d'exprimer les contraintes et les règles.

MODELE PRATIQUE	MODELE COGNITIF	MODELE INFORMATIQUE
FORMES STATIQUES (données)	TAXINOMIES	Programmation par objets
FORMES CAUSATIVES (traitements)	ACTINOMIES	Programmation par acteurs
FORMES DECLARATIVES (contraintes et règles)	SCHEMAS	Programmation par la logique

Figure 2.8 : Cycle de vie de KOD

### 2.5.3.2 Conclusion

KOD constitue un cadre général qui peut s'adapter à la représentation de tout type de connaissance, puisqu'il s'agit en fait d'une reformulation dans un cadre plus strict des concepts d'objets et d'actions. La K-Station qui se présente comme une plate-forme logicielle offre un ensemble d'outils pour aider à la mise en oeuvre de la méthode KOD. Certains de ces outils, de type hypertexte, permettent de gérer les fichiers de retranscription, de conceptualiser les informations qu'ils contiennent à l'aide des types de structures constituées.

L'avantage essentiel de KOD réside dans la décomposition d'un texte en vue de sa modélisation cognitive avec relativement peu de concepts (objet, action, contrainte, règle). Cet avantage devient inconvénient dans la mesure où le peu de concepts proposés ne suffit généralement pas à modéliser finement la complexité d'une expertise.

On peut lui reprocher sa lourdeur, le manque de méthode dans la conduite des entretiens, le coût et la lenteur de leur retranscription systématique ainsi que la formalisation presque automatique de connaissances dont beaucoup ne sont pas utiles à la conception du système.

## 2.6 NOTRE APPROCHE D'ACQUISITION DE CONNAISSANCES

Notre approche ne consiste pas à inventer une n<sup>ième</sup> méthodologie d'Acquisition de Connaissances, mais de s'inspirer des méthodologies existantes pour recueillir un ensemble de connaissances dans le domaine de l'eau. Les connaissances à acquérir sont, d'une part, celles qui sont nécessaires à la construction d'un langage de description qui sera utilisé par les experts du domaine pour décrire les cas d'intervention qu'ils ont effectués pour résoudre des problèmes engendrés par des situations de crise et d'autre part, celles qui sont nécessaires pour l'exploitation des connaissances apprises à partir des exemples d'apprentissage (description des cas d'intervention) fournis au système. Ces connaissances peuvent être réparties en deux classes, à savoir [Senoune 95] :

- les connaissances descriptives du domaine : il s'agit des concepts, des relations qui existent entre les concepts, des taxinomies des concepts, etc. Le langage de description recherché est constitué d'un ensemble de ces concepts et de leurs domaines de valeurs qu'ils peuvent prendre,
- les connaissances stratégiques : il s'agit des heuristiques qui permettent de reformuler automatiquement des descriptions de situations de crise en des descriptions similaires. La reformulation d'une description est nécessaire à chaque fois que le système ne trouve pas de solution au problème décrit.

Comme KADS adopte une philosophie qui intègre les différents types de connaissances à travers des niveaux séparés, nous avons choisi de travailler avec la même idée de division des connaissances par niveaux. Le niveau domaine nous permet de constituer un vocabulaire à utiliser pour décrire les cas d'intervention et à construire les différentes taxinomies du domaine définissant certains éléments du vocabulaire constitué. Le niveau stratégie regroupe les stratégies de reformulation de problèmes. Les informations recueillies au cours des réunions, des interviews et de la consultation des documents sur le domaine sont structurées et reformulées pour obtenir un premier modèle de connaissances. Ce modèle est destiné à éliminer les ambiguïtés et les imprécisions ou inexactitudes inhérentes à toute transmission de connaissances opérée sans support méthodologique. Il permet aux experts de revoir les informations qu'ils ont indiqué pour les vérifier et les valider.

Pour cette étape d'acquisition de connaissances, nous avons adopté une organisation du projet qui s'appuie sur le modèle traditionnel (cf. paragraphe 2.3.1) afin de recueillir le plus possible de connaissances descriptives et stratégiques. Pour l'étape d'apprentissage, le modèle direct (cf. paragraphe 2.3.2) sera utilisé pour permettre aux experts de délivrer leurs connaissances en fournissant des cas d'intervention qu'ils ont effectué dans le cadre de la résolution de problèmes engendrés par des situations de crise.

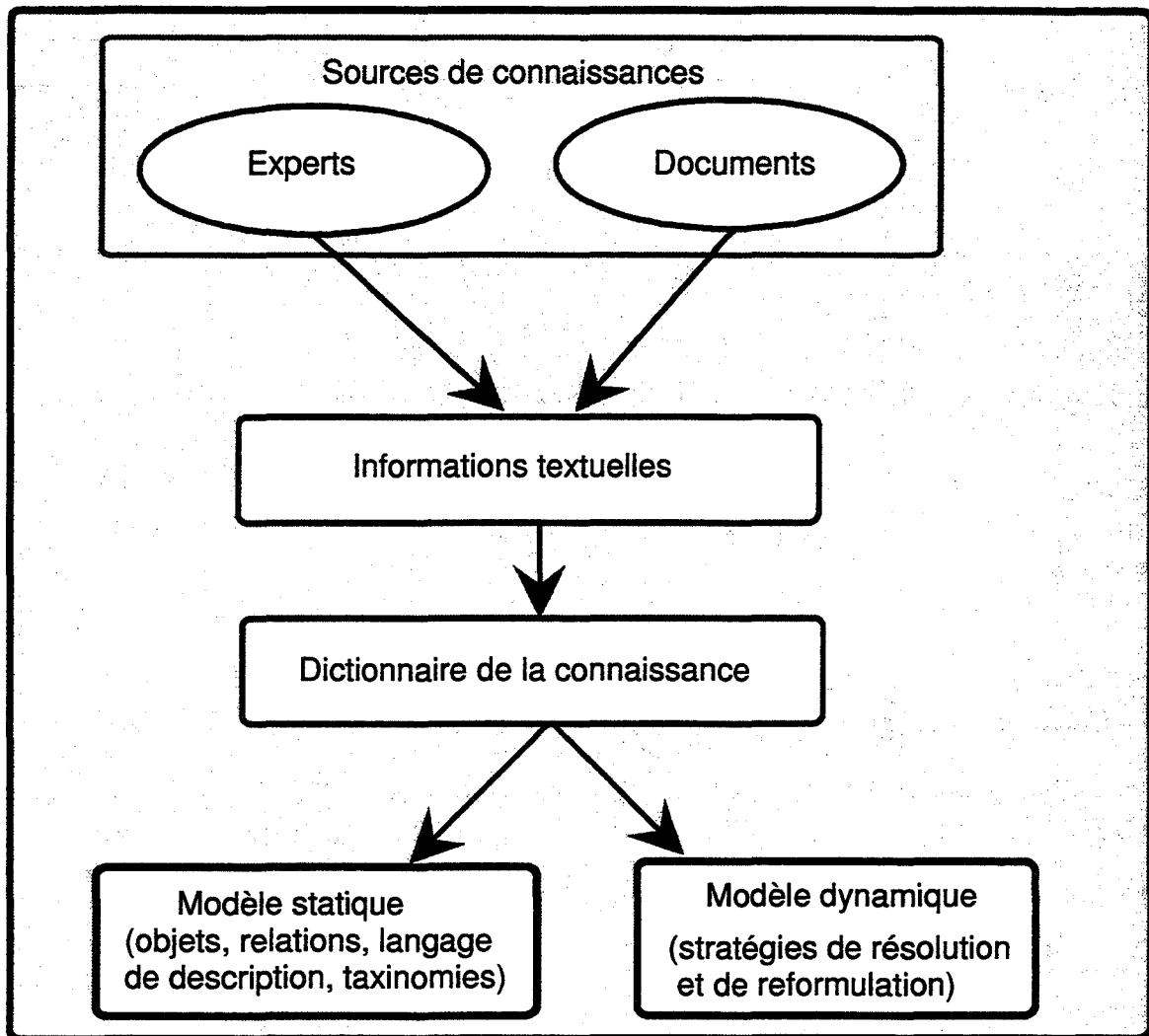


Figure 2.9 : Etapes d'acquisition de connaissances

Le processus d'acquisition des connaissances correspondant aux deux niveaux (domaine, stratégie) ainsi que la constitution du langage de description des exemples d'intervention suit les quatre phases suivantes :

- Identification des connaissances : à partir de la consultation de documents sur le domaine, d'analyse des rapports d'intervention des experts sur des cas de crise, et d'entretiens avec les experts, nous constituons un ensemble de connaissances contenant tous les éléments (objets, relations entre objets, règles) donnés par les experts.

- **Organisation des connaissances** : après avoir identifié et spécifié les connaissances, nous demandons aux experts d'organiser et d'abstraire cet ensemble de connaissances pour donner une forme aux régularités qui apparaissent dans les connaissances qu'ils ont fournis.

- **Validation des connaissances** : nous demandons aux experts d'explicitier les choix qui les ont amené à organiser les objets connus sous la forme proposée (dans le cas de taxinomie, par exemple), et/ou de compléter ou de clarifier les différents niveaux de la taxinomie.

- **Recherche d'un format de représentation** : A partir des connaissances recueillies, les experts déterminent les descripteurs qui vont leur servir pour décrire les exemples d'apprentissage.

La figure 2.10 illustre le schéma SADT du processus d'acquisition de connaissances pour notre cas d'application.

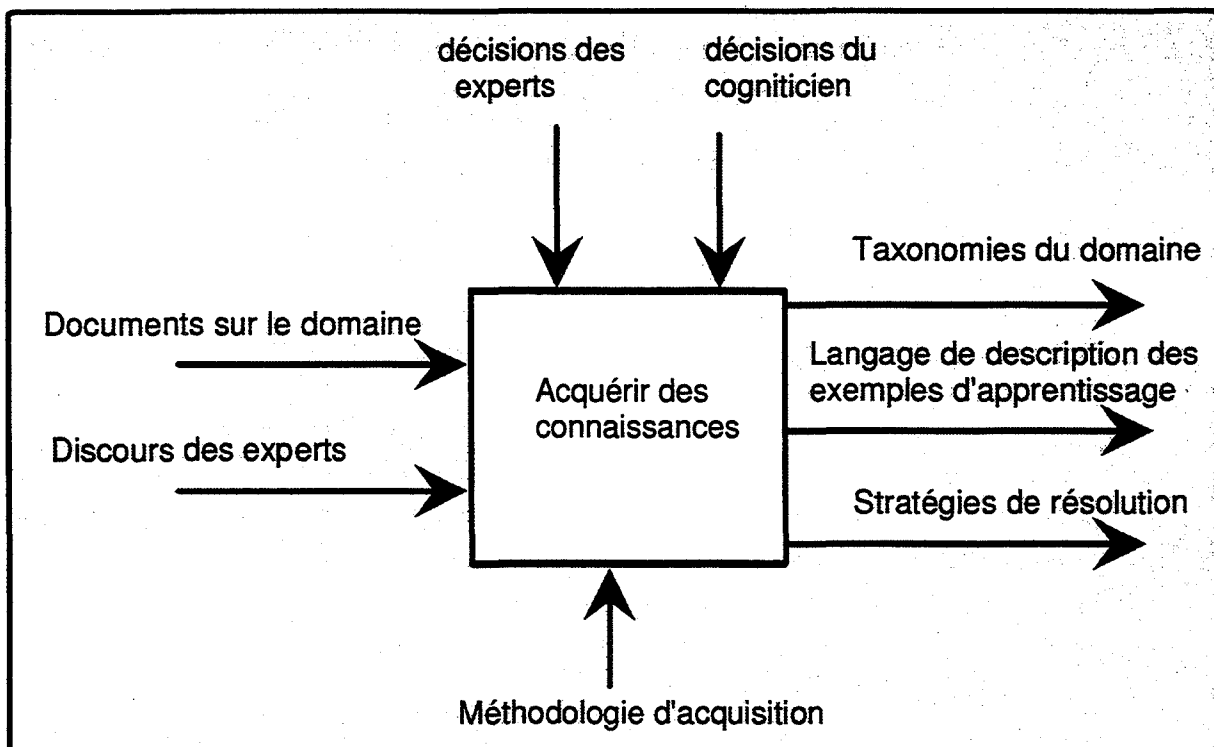


Figure 2.10 : Diagramme SADT du processus d'acquisition de connaissances

## 2.7 CONCLUSION

Nous avons vu à travers les méthodes étudiées que l'extraction des connaissances recouvre les tâches de recueil de connaissances à l'aide de différentes techniques, de conceptualisation (identification des concepts, propriétés, relations et procédures) et d'analyse à l'aide d'outils structurants permettant aussi une validation des connaissances. Nous avons vu aussi que ces tâches sont effectuées par deux catégories de personnes, l'une possédant des compétences propres au domaine à modéliser - les experts -, l'autre possédant des connaissances générales sur la modélisation des raisonnements. Le résultat de leur travail est, dans un premier temps, un modèle de connaissances.

Nous avons montré aussi que pour notre cas d'application, il s'agit de recueillir deux types de connaissances dans le domaine de l'eau. Le premier type représente les connaissances nécessaires pour le processus d'apprentissage utilisant des exemples d'intervention sur des cas de crise fournis par les experts. Elles doivent donc permettre de spécifier un langage de représentation de ces exemples et représenter la sémantique du domaine de l'eau. Le deuxième type représente les connaissances stratégiques que les experts utilisent pour passer de la description d'une "situations de crise" à une autre description similaire.

Dans le chapitre 3, nous décrivons les différents types d'apprentissage et de systèmes d'apprentissage utilisés pour l'acquisition automatique de connaissances et nous expliciterons les raisons qui nous ont amené à choisir une technique fondée sur les Dépendances Perçues.

### ***3. Chapitre 3 : Choix d'une technique d'apprentissage adaptée au caractère incrémental et bruité des connaissances***

#### **3.1 POURQUOI UNE TECHNIQUE D'APPRENTISSAGE DANS LE DOMAINE DE L'EAU ?**

Dans le chapitre précédent (chapitre 2), nous avons expliqué pourquoi le développement d'un SBC est un processus complexe et long. Nous avons décrit, par la suite, quelques méthodologies d'acquisition de connaissances qui sont utilisées pour construire une BC. Mais, même en utilisant ces techniques d'acquisition, l'extraction de connaissances empiriques des experts représente une tâche difficile. En effet, les experts n'arrivent pas à expliciter les connaissances qu'ils ont acquis par une longue expérience.

Les méthodologies d'acquisition de connaissances se sont avérées d'un grand intérêt pour le développement de projets SBC, mais elles ont vite montré leurs limites dans les domaines où



les connaissances évoluent et où l'expertise est empirique et détenue par plusieurs experts. En effet, lorsque le domaine d'application concerné est bien déterminé et non évolutif, les connaissances peuvent être transférées dans le système informatique en utilisant une méthode d'acquisition de connaissances. Mais lorsque le domaine est très large et évolutif, ce qui nécessite la collaboration de plusieurs experts et un enrichissement continu des connaissances, l'utilisation des méthodes classiques d'acquisition de connaissances est insuffisante. Dans de tels domaines, il est nécessaire d'utiliser une méthode d'Apprentissage Automatique pour pallier aux lacunes soulevées par les méthodes classiques d'acquisition de connaissances [Senoune 94b]. L'acquisition de connaissances dans le domaine de l'eau qui est caractérisé essentiellement par des connaissances empiriques, évolutives et dispersées à travers plusieurs experts, nécessite donc l'utilisation d'une méthode d'apprentissage automatique. Pour notre cas d'application, les connaissances à acquérir par apprentissage sont fondées sur des interventions effectuées par les experts du domaine sur des cas de crise. Ainsi, nous ne demanderons pas aux experts de délivrer explicitement leur savoir, mais de fournir des exemples d'intervention qu'ils ont effectué pour résoudre des problèmes en situation de crise.

### **3.2 QU'EST-CE QUE L'APPRENTISSAGE AUTOMATIQUE ?**

Les premiers travaux en apprentissage ont été utilisés pour le développement de systèmes informatiques pour la reconnaissance et la classification de formes et de systèmes de jeux [Samuel 59], [Samuel 67]. Ces travaux sur l'apprentissage portaient sur l'ajustement automatique des paramètres de classement et d'évaluation élémentaires.

La notion d'apprentissage automatique est assez large, elle se réfère à des situations où une machine accroît son niveau de connaissances à accomplir certaines tâches. Sur le plan pratique, le but de l'apprentissage est d'automatiser l'acquisition de connaissances dans le développement de SBC. Cela permet de remplacer l'interaction expert-cogniticien ainsi que les techniques d'extraction de connaissances. Sur le plan théorique et cognitif, le but de l'apprentissage est de comprendre et d'identifier les mécanismes généraux liés au processus d'apprentissage, donc orienté vers la modélisation des processus d'apprentissage humain et le développement d'algorithmes généraux d'apprentissage.

L'apprentissage automatique comme l'apprentissage humain est un processus qui consiste à acquérir de nouvelles connaissances ou à améliorer celles déjà acquises. Les connaissances acquises sont utilisées pour acquérir de nouvelles connaissances et les nouvelles connaissances peuvent corriger, structurer et améliorer les connaissances déjà acquises. Le système améliore son comportement au fur et à mesure qu'il acquiert des connaissances. A partir d'une certaine "maturité", d'un certain seuil de connaissance et d'apprentissage, le système intelligent novice devient lui même expert et il a de moins en moins besoin d'aide extérieure [Cavarero 86]. Le schéma ci-dessous illustre l'utilisation des connaissances apprises pour améliorer le processus d'apprentissage.

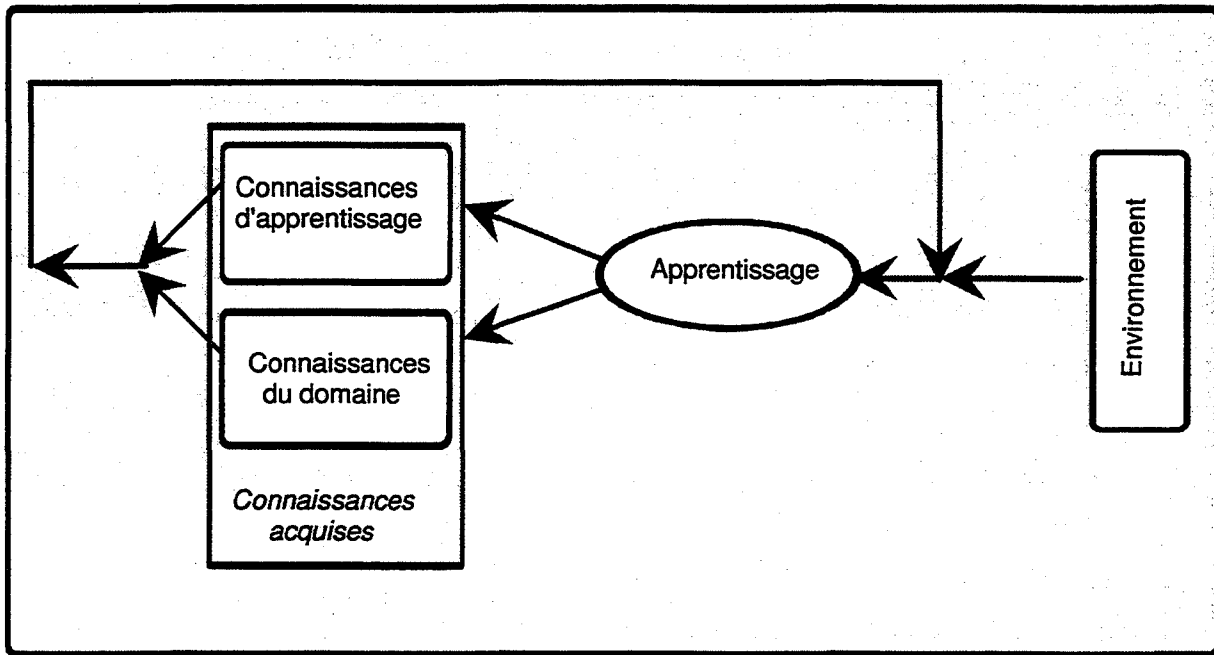


Figure 3.1 : Utilisation des connaissances acquises pour améliorer le processus d'apprentissage

### 3.2.1 Les différentes approches d'apprentissage automatique

Les méthodes d'apprentissage automatique peuvent être divisées en deux classes, à savoir "l'approche numérique" et "l'approche symbolique" :

**l'approche numérique** [Kodratoff 86], [Kodratoff 91] : cette approche est issue des méthodes d'analyse de données et des méthodes connexionnistes, fondées sur les modèles des réseaux de neurones. Elle est efficace et résistante au bruit, mais elle donne lieu à des règles ou des concepts incompréhensibles. L'analyse de données est une méthode très ancienne de traitement des observations qui s'appuie sur les théories de statistiques et de probabilités. Cependant, ces théories produisent des résultats difficiles à interpréter par des non spécialistes. Les énoncés sur lesquels s'effectue l'apprentissage, dans cette approche, sont représentés par des vecteurs numériques. Ce mode de représentation est mal adapté lorsqu'il s'agit de manipuler des objets structurés.

**l'approche symbolique** [Kodratoff 86], [Kodratoff 91] : traite un ensemble d'énoncés décrits dans une représentation symbolique. Cette approche est bien adaptée à une interaction avec des experts humains, mais elle est très sensible au bruit. Les résultats sont plus facilement compréhensibles grâce à la sémantique intégrée dans la représentation des objets. Dans l'approche symbolique, nous trouvons plusieurs types d'apprentissage selon les stratégies utilisées dans le processus d'apprentissage.

Avant de décrire les différentes techniques d'apprentissage, nous allons d'abord préciser les règles d'inférence utilisées dans les processus d'apprentissage.

### 3.2.2 Les règles d'inférence pour l'apprentissage symbolique

Le processus d'apprentissage s'appuie sur des règles d'inférence telles que la déduction, l'induction, l'analogie et l'abduction [Charniak 85], [Thayse 91] :

**La déduction** est un processus de dérivation qui permet de conclure de nouvelles propositions spécifiques à partir d'un ensemble de propositions générales, par application d'un ensemble de règles logiques. La déduction est une opération qui préserve la vérité (toute inférence par déduction produit une proposition qui est vraie dans tous les modèles de la théorie de départ).

**L'induction** est un processus de dérivation de propositions plus générales à partir de propositions établies pour certains cas spécifiques, par application d'un ensemble de règles logiques.

**L'analogie** est un processus de dérivation de propositions caractérisant une situation inconnue à partir de propositions caractérisant une situation connue et similaire. Les termes "situation" et "similarité" sont à prendre dans un sens très large. Une situation peut se référer à un objet, une classe d'objet, un problème posé, une règle ou procédure de résolution, une action, un plan, un comportement, etc. ; une similarité peut porter sur des attributs physiques ou structurels, des liens conceptuels, des historiques de résolution, etc.

**L'abduction** est l'opération d'inférence de propositions qui expliquent d'autres propositions données.

### **3.3 CLASSIFICATION DES TECHNIQUES D'APPRENTISSAGE**

Dans tout processus d'apprentissage, le système apprenant reçoit des données de la part de l'environnement. Ces données peuvent subir une transformation dans le but d'aboutir à un ensemble de connaissances organisées sous une nouvelle forme qui est adaptée à l'utilisation auxquelles elles sont destinées. Les systèmes d'apprentissage automatique peuvent être classifiés selon différents critères à savoir : le domaine d'application, la représentation des connaissances acquises, le type des données en entrée, l'inférence, etc. [Michalski 90].

Dans ce qui suit, nous présentons les différentes techniques d'apprentissage en les classifiant selon le type d'inférence utilisé par le processus d'apprentissage.

#### **3.3.1 L'apprentissage inductif**

En apprentissage inductif [Dietterich 84], [Kodratoff 86], [Kodratoff 87], [Fisher 87], une masse de données est présentée au système d'apprentissage qui infère certaines connaissances considérées comme étant pertinentes par application d'un processus de

recherche de descriptions générales plausibles. Le système d'apprentissage fait des généralisations sur les connaissances reçues et des sélections des résultats les plus plausibles. Le processus d'induction permet donc de déterminer des connaissances considérées a priori vraies à partir d'autres connaissances qui sont vraies. L'apprentissage inductif comprend l'apprentissage à partir d'exemples, l'apprentissage par découverte, l'apprentissage à partir d'observations et l'apprentissage par détection de similarités.

### **3.3.1.1 Apprentissage à partir d'exemples**

Il est assez difficile de donner une définition claire et qui synthétise l'apprentissage à partir d'exemples, car tous les systèmes d'apprentissage qui utilisent des descriptions d'exemples pour arriver à une classification de connaissances sont dits "systèmes d'apprentissage à partir d'exemples" et la plupart des systèmes d'apprentissage peuvent être définis ainsi. Une caractéristique de l'apprentissage à partir d'exemples [Michalski 83a], [Dietterich 84], [Clark 90] est la classification des exemples dans un ensemble de classes. L'apprentissage est basé sur un algorithme qui permet de trouver les classes auxquelles appartiendront de nouveaux exemples présentés au système. Il s'agit donc d'utiliser des descriptions de faits préclassifiés par l'expert pour engendrer des connaissances de classification d'autres nouveaux faits.

### **3.3.1.2 Apprentissage par découverte**

En apprentissage par découverte [Langley 84], [Langley 86], le système considère aussi des ensembles d'objets, actions, etc., mais il n'est pas précisé si ces instances appartiennent ou non à la même classe. L'Analyse Symbolique des Données (Conceptual Clustering), un aspect important de l'apprentissage par découverte, a pour tâche d'effectuer les meilleurs regroupements possibles des instances et d'en trouver une description en intention.

### **3.3.1.3 Apprentissage à partir d'observations**

L'apprentissage à partir d'observations est un apprentissage non supervisé car il n'a pas recours à un tuteur qui étiquette au préalable les exemples et les contre-exemples. Il consiste à établir une classification à partir d'un ensemble d'objets ou de situations donnés de manière à mettre en évidence des structures et des concepts n'apparaissant pas a priori. Les classifications obtenues permettent donc de mieux comprendre le monde sur lequel sont faites les observations. Cette technique d'apprentissage comble les limites des techniques traditionnelles de classification telles que l'analyse de données qui construisent des regroupements d'objets uniquement en fonction de la similarité numérique de ces objets.

### **3.3.1.4 Apprentissage par détection de similarités**

En apprentissage par détection de similarités (traduction de Similarity Based Learning, ou SBL) [Mitchell 82], [Kodratoff 86], l'apprentissage se fait en détectant des similarités dans un ensemble d'exemples et des dissimilarités entre les exemples et les contre-exemples. Cela consiste à prendre un grand nombre d'exemples et à les analyser en termes de similarités et de différences et créer une description généralisée par abstraction des similarités.

## **3.3.2 L'apprentissage déductif**

L'apprentissage est dit déductif lorsque toutes les connaissances nécessaires sont données au départ [Kodratoff 86]. Il consiste à déduire de nouvelles connaissances à partir de connaissances existantes. La déduction détermine les conséquences que nous pouvons dégager des données et produit de nouveaux faits. L'apprentissage déductif comprend l'apprentissage par recherche d'explications, l'apprentissage par instruction et l'apprentissage par coeur.

### **3.3.2.1 Apprentissage par recherche d'explications**

En apprentissage par recherche d'explications (traduction de Explanation Based Learning, ou EBL), l'apprentissage se fait à partir d'explications dérivées de l'analyse d'un exemple ou d'un contre-exemple du concept ou de la règle qu'il s'agit d'apprendre. L'apprentissage par recherche d'explications consiste à obtenir de nouvelles connaissances (appelées explications ou hypothèses) qui vont permettre d'expliquer des faits observés vis-à-vis de la connaissance déjà existante dans le système [Marquis 89]. Toutefois, il ne faut retenir que les explications les plus intéressantes, ce qui nécessite un certain nombre d'indications qui permettent de les déterminer.

Nous pouvons constater que l'apprentissage par recherche d'explication ne permet pas d'acquérir de nouvelles connaissances, mais transforme des connaissances existantes qui s'avéraient inutilisables par rapport à une tâche donnée, en des connaissances plus opérationnelles. L'apprentissage par recherche d'explications est adapté lorsqu'il existe une connaissance suffisante du domaine qui permettra d'expliquer pourquoi un exemple appartient ou non au concept en cours d'étude.

### **3.3.2.2 Apprentissage par instruction**

En apprentissage par instruction, la charge du système revient à reformuler la connaissance qui lui est fournie en une connaissance plus propice à son utilisation ultérieure. Le processus d'acquisition est guidé par une utilisation déductive par le système de métaconnaissances sur le type de connaissance à acquérir.

### **3.3.2.3 Apprentissage par coeur**

En apprentissage par coeur, il n'y a pas de transformation de la connaissance reçue par le système. Celle-ci est acceptée et mémorisée telle qu'elle est fournie par l'environnement. Le seul problème dans ce cas réside dans la manière d'indexer l'information stockée pour des recherches ultérieures. Cet apprentissage consiste à mémoriser les problèmes déjà résolus et

lorsqu'un nouveau problème est posé, soit il est reconnu et la solution est immédiatement trouvée, soit il ne l'est pas et la solution est indéfinie.

### **3.3.3 L'apprentissage par analogie**

En apprentissage par analogie [Carbonell 83], [Carbonell 86], [Burstein 86], [Chouraqui 86], [Dorrough 86] il y a combinaison d'apprentissage inductif et déductif. Cela consiste à reconnaître des similarités entre le concept à apprendre et un concept connu, et à déterminer quelles caractéristiques pertinentes peuvent être transférées du concept connu vers le concept à apprendre. En d'autres termes, l'apprentissage par analogie consiste à apprendre à reconnaître un problème déjà résolu et à exploiter la solution pour résoudre un nouveau problème. La résolution de problèmes par analogie consiste donc à transférer de la connaissance à partir des épisodes précédents de résolution de problèmes aux nouveaux problèmes qui partagent des aspects significatifs de l'expérience précédente correspondante, et à utiliser les connaissances transférées pour construire des solutions pour les nouveaux problèmes. Les conclusions obtenues dans ce cas correspondent à des hypothèses plausibles, devant être validées et éventuellement modifiées. L'analogie comporte donc un certain risque lié à l'évaluation du degré de similarité de deux concepts.

### **3.3.4 Autres stratégies d'apprentissage**

#### **3.3.4.1 Apprentissage connexionniste**

L'apprentissage connexionniste est un apprentissage inductif qui repose sur l'utilisation des réseaux de neurones [Hopfield 82], [Ackley 85]. Chaque neurone du réseau possède des entrées et une sortie caractérisées par des valeurs numériques. La sortie est obtenue par application d'une fonction sur la somme pondérée des entrées du neurone. L'apprentissage connexionniste consiste à modifier les poids ou valeurs seuils des interconnexions entre les neurones. Un neurone cumule les signaux de ces prédécesseurs et lorsque la somme dépasse



son seuil, il active son état de sortie vers les neurones successeurs. Ce processus est répété successivement jusqu'à ce que le réseau ait une forte probabilité de réponse correcte.

### **3.3.4.2 Le Raisonnement à Base de Cas**

D'un point de vue psychologique, le Raisonnement à Base de Cas (CBR : Case-Based Reasoning) [Kolodner 87], [Harmon 93], [Rissland 93] se réfère au raisonnement à l'aide duquel l'être humain résout des problèmes en se basant sur des problèmes déjà résolus précédemment. L'application de stratégies qui se sont avérées satisfaisantes dans la résolution de problèmes similaires constitue une étape importante de cette méthode. Si le cas traité ne ressemble à aucun des cas déjà résolus dans le passé, le nouveau problème ainsi que sa solution seront mémorisés afin de pouvoir être exploités dans la résolution de nouveaux problèmes qui vont se présenter. Le raisonnement sur des exemples appris consiste à interpréter un nouveau cas à l'aide d'un cas similaire extrait du système et choisi comme guide [Bareiss 90].

Après avoir décrit les différentes méthodes d'apprentissage, nous examinerons quelques systèmes d'apprentissage qui s'appuient sur ces méthodes pour la construction de Bases de Connaissances.

## **3.4 QUELQUES SYSTEMES D'APPRENTISSAGE**

### **3.4.1 ID3**

ID3 (Interactive Dichotomiser Three) [Quinlan 86], [Quinlan 87], [Smith 92] est un système d'apprentissage qui construit, par raffinements successifs, un arbre de décision à partir d'exemples. L'ensemble d'apprentissage est partitionné en sous-ensembles en fonction des attributs jusqu'à l'obtention d'observations d'une seule classe dans chaque sous-ensemble. Les tests sur les attributs constituent les branches de l'arbre et les sous-ensembles constituent les

feuilles. Les arbres de décision sont construits à partir de la connaissance au sujet d'une situation ou d'un système.

Pour classer une nouvelle observation, en phase d'exploitation, le système parcourt l'arbre déjà construit, en partant de la racine et en parcourant le chemin formé de branches qui correspondent aux attributs de l'observation. Lorsqu'il aboutit à une feuille, il déduit alors que l'observation appartient à la classe relative aux observations de cette feuille.

La construction des arbres de décision est une méthode très intéressante pour produire un ensemble de règles de décision, mais l'inconvénient est que cette construction fondée sur les attributs des observations nécessite l'utilisation d'attributs discriminants. La présence d'attributs non discriminants inadéquats ou de bruit dans l'ensemble d'apprentissage peut conduire à une partition en classes n'ayant qu'un seul élément, ce qui peut amener à la construction d'arbres de décision complexes et surchargés.

Un autre inconvénient apparaît lorsqu'il s'agit de classer une observation pour laquelle des valeurs manquent pour un ou plusieurs attributs (si un test portant sur un tel attribut est rencontré dans le parcours de l'arbre, il n'est pas possible de déterminer quelle est la branche adéquate à choisir).

Afin d'obtenir un arbre de taille raisonnable, ayant un bon taux de prédiction sur les observations nouvelles et afin de pouvoir traiter des observations comportant des attributs à valeurs inconnues, Quinlan [Quinlan 90] a proposé un certain nombre d'extensions pour améliorer la méthode ID3 : "technique d'élagage de l'arbre", "prise en compte des valeurs inconnues et imprécises d'attributs", "assouplissement des règles de décision".

**Elagage de l'arbre** : consiste à calculer le taux d'erreur de chaque sous-arbre à partir de l'ensemble d'apprentissage, d'estimer le taux d'erreur si le sous-arbre est remplacé par une feuille et ne garder que les sous-arbres qui ne peuvent pas être remplacés sans un accroissement significatif du taux de mauvais classement de l'arbre. Des conditions d'arrêt dans le développement de l'arbre doivent être définies. Elles prennent en compte le nombre d'éléments d'un noeud, le nombre de niveaux de l'arbre, etc.

**Traitement des valeurs inconnues** : consiste à calculer les probabilités (moyenne, supérieure et inférieure) de l'affectation à une classe de l'observation compte tenu des valeurs d'attributs présentes pour traiter les observations comportant des attributs à valeurs inconnues.

**Seuils plus souples** : consiste à associer à chaque question binaire du type  $A < T$ , un intervalle  $[T^-, T^+]$  encadrant  $T$ . Cet intervalle est calculé de manière à avoir un taux d'erreur acceptable sur l'ensemble d'apprentissage en remplaçant  $T$  par  $T^-$  ou  $T^+$ . Plus la taille de l'intervalle est petit plus la conclusion est précise.

Les arbres de décision sont une bonne représentation pour reconnaître des concepts. Leurs inconvénients est que leurs conclusions sont catégoriques, ils ne permettent pas d'intégrer des informations incertaines dans leurs classifications. Des données manquantes ou des informations imprécises contraignent la classification d'une observation. Nous pouvons donc constater que ce système ne peut pas être utilisé dans les domaines dans lesquels les exemples d'apprentissage ne contiennent pas tous les mêmes attributs. De plus, ID3 ne permet pas de construire la BC de manière incrémentale, c'est à dire que l'ensemble des données doit être fourni au système en une seule fois.

### 3.4.2 KATE

Comme ID3, KATE [Manago 89] est un système d'apprentissage à partir d'exemples. Il construit automatiquement des arbres de décision à partir d'exemples préclassés. Dans KATE les connaissances sont représentées sous formes de schémas (frames) à l'aide du langage Le\_Frame qui permet en particulier d'organiser l'information selon une hiérarchie (taxonomie de prédicats), de représenter des contraintes sur la valeur des descripteurs et de représenter des règles permettant de déduire la valeur d'un descripteur à partir de celle d'autres descripteurs.

KATE utilise une stratégie de recherche heuristique avec optimisation de la mesure du gain d'information. Il exploite la structure hiérarchique des schémas pour engendrer dynamiquement les tests pour le calcul de ce gain d'information. La stratégie standard de génération de tests consiste à traverser la hiérarchie des classes du plus général au plus spécifique. Les conditions suivantes permettent de contraindre l'espace de recherche :

- un noeud dans l'arbre de décision, porte soit sur l'introduction d'un nouvel objet soit sur un champ d'un objet apparaissant déjà dans l'arbre de décision,

- avant d'introduire une classe il faut d'abord introduire toutes les classes auxquelles elle appartient, ainsi il faut commencer par les classes les plus générales. Ceci permet de mener une recherche du plus général au plus spécifique, guidée par un modèle conforme à la stratégie originale utilisée dans ID3.

Nous pouvons constater que KATE est une extension de ID3 qui permet une prise en compte de données complexes. L'utilisation d'une représentation de connaissances à base de schémas (frames) permet de mieux structurer les connaissances sur le domaine d'application. Toutefois, ce système n'est pas incrémental et nécessite que les exemples d'apprentissage aient le même format (comportent les mêmes attributs).

### 3.4.3 ALEX

ALEX [Seroussi 89] est un système d'apprentissage par analogie. Des exemples représentant des problèmes résolus sont proposés séquentiellement au système. A partir de ces exemples, il construit une base de connaissances qui joue le rôle de base d'expériences pour résoudre de nouveaux problèmes par analogie.

Dans un tel système, une fois les connaissances de base acquises, la capacité à résoudre des problèmes dépend de l'expérience. Les nouveaux problèmes considérés comme des variantes de problèmes déjà résolus, sont en effet résolus par analogie. La résolution de problèmes apparaît en fait, comme le processus de compréhension qui utilise l'analogie pour établir un lien entre tout nouveau problème à résoudre et une séquence de résolution connue. L'établissement de ce lien est facilité par la construction de classes de problèmes, dont chacune d'elles est caractérisée par une description générale des problèmes qui la constituent, et par un type général de résolution.

Pour faire une correspondance entre un problème à résoudre et un autre déjà résolu, ALEX compare les caractéristiques de surface de la description du premier problème à celles du deuxième problème contenu dans sa base d'expériences. Il calcule un score d'appariement

partiel entre eux et estime le résultat par rapport à un seuil. Plus le score est élevé, plus l'analogie est grande. Mais le résultat de l'appariement dépend de la qualité des descriptions. Il faut que les descriptions à comparer soient précises pour que l'appariement soit satisfaisant.

Le système ALEX peut donc être utilisé dans les domaines dans lesquels les problèmes sont bien définis et les caractéristiques de surface déduites des énoncés sont pertinentes pour les décrire. Un problème à résoudre doit être apparié avec un grand nombre de problèmes déjà résolus afin de trouver ceux qui sont les plus proches. Une mesure de similarité est nécessaire comme une base de sélection des problèmes résolus les plus appropriés. Le calcul du score d'appariement structurel nécessite une importante connaissance implicite sur le domaine.

#### **3.4.4 COBWEB**

COBWEB [Fisher 87] est un système d'apprentissage à partir d'observations qui est fondé sur le regroupement conceptuel incrémental pour construire des modèles prédictifs de l'environnement. Il transforme une collection de descriptions d'objets en un arbre de classification dans lequel les objets sont décrits en terme de paires attribut-valeur. Chaque noeud de l'arbre de classification est un concept probabiliste qui représente une classe d'objets. La classification d'objets dans COBWEB s'appuie sur les similarités qui existent entre les objets d'une même classe et les dissimilarités entre les objets de classes différentes.

COBWEB requiert une connaissance sur le domaine pour pouvoir évaluer le degré de similarité et de dissimilarité entre les objets. Il permet d'acquérir des connaissances incertaines, mais les descriptions obtenues sont aussi des descriptions incertaines du fait qu'il s'appuie sur des traitements probabilistes. COBWEB permet de traiter incrémentalement des exemples d'apprentissage. Cependant, l'ordre d'arrivée des exemples peut influencer sur le processus de construction de nouvelles classes d'objets.

### 3.4.5 UNIMEM

UNIMEM (UNiversal MEMory) [Lebowitz 86], [Lebowitz 87] est un système d'apprentissage à partir de similarités qui a pour but de découvrir des liens entre concepts. Il construit automatiquement une hiérarchie de généralisation en utilisant des méthodes fondées sur les similarités appliquées incrémentalement à des exemples représentés sous forme de triplets (propriété, facette, valeur). Chaque noeud de la hiérarchie représente un concept décrit par un vecteur (attribut, valeur). UNIMEM est capable de traiter un grand nombre d'exemples qui peuvent être des exemples incomplets. Il peut, par conséquent, être utilisé dans les domaines où la connaissance est imprécise.

L'inconvénient de ce système est qu'il nécessite, comme COBWEB, une connaissance implicite sur le domaine pour pouvoir évaluer la distance entre les concepts et intégrer des traitements probabilistes. De plus, les concepts obtenus sont des descriptions incertaines.

### 3.4.6 PROTOS

PROTOS [Bareiss 90] est un système d'apprentissage à partir de cas, dédié aux tâches de classification heuristique. Pour classer de nouveaux cas, PROTOS utilise des modèles de concepts lui permettant de guider efficacement l'interprétation de cas individuels. Un modèle est un cadre qui peut correspondre à plusieurs cas qui sont proches les uns des autres par rapport à des caractéristiques définies. Les cas typiques sont représentés sous forme d'un réseau appelé "structure de catégories". Chaque noeud de cette structure renferme un ensemble de cas typiques. Chaque cas typique de la structure des catégories sert de modèle pour interpréter de nouveaux cas. Un appariement fondé sur les connaissances du domaine est utilisé pour sélectionner un cas typique. Pour que PROTOS puisse raisonner et apprendre à partir de cas, il est doté de fonctions de regroupement de cas typiques d'une catégorie, d'indexage des cas, d'utilisation des cas pour la résolution de problème, d'apprentissage de la structure des catégories et d'apprentissage des indexes des cas typiques.

Pour aboutir à une classification pertinente, les cas typiques nécessitent la représentation de toutes les propriétés des objets qui les constituent. Mais comme il est souvent nécessaire de

faire une abstraction sur les objets en ne prenant en compte que certaines de leurs propriétés et en ignorant les autres, il existe un risque de mettre ensemble des cas qui ne sont pas tout à fait similaires. D'une part, la représentation de toutes les propriétés des objets mène à une lourdeur dans leur traitement, et d'autre part les abstractions faites sur les objets peuvent contribuer à une perte de cohérence dans les catégories de cas typiques. Un autre inconvénient de ce système est que le processus d'appariement devient coûteux dès qu'il s'agit de traiter un grand nombre de cas typiques.

### 3.4.7 Les RDP

les Réseaux de Dépendances Perçues (RDP) [Sainte Marie 89], [Sainte Marie 90a], [Sainte Marie 90b] permettent de représenter, sous forme de graphe, un ensemble de Dépendances Perçues (DP). Ces Dépendances Perçues sont issues du traitement incrémental d'un ensemble d'observations constituant des exemples d'apprentissage.

#### 3.4.7.1 Les Dépendances Perçues

La théorie de la Dépendance Perçue, développée par Sainte Marie [Sainte Marie 89], [Sainte Marie 90a], [Sainte Marie 90b] est fondée sur les relations naturelles qui accompagnent les observations pratiques du monde réel. La Dépendance Perçue s'appuie sur les relations qui existent entre des objets dans un ensemble d'observations dans un domaine donné. Ainsi, si pour chaque observation ayant une propriété  $P_1$  nous constatons aussi l'existence de la propriété  $P_2$ , nous dirons que  $P_1$  et  $P_2$  sont en relation de Dépendance Perçue. Cette relation entre les deux propriétés  $P_1$  et  $P_2$  est notée par  $P_1 \rightarrow P_2$ . Toutefois, cette dépendance entre les éléments observés n'est valable que sur l'ensemble d'observations qui a permis sa formation. En d'autres termes, la relation de Dépendance Perçue  $P_1 \rightarrow P_2$  qui est vérifiée sur l'ensemble des observations actuelles peut ne plus être valable dès lors que nous étendons le champ d'observation par une nouvelle observation dans laquelle nous pouvons constater l'existence de la propriété  $P_1$  et l'absence de la propriété  $P_2$ .

### 3.4.7.2 Formalisation de la relation de Dépendance Perçue

D'un point de vue mathématique, la relation de Dépendance Perçue peut être définie comme suit :

Soient  $\mathbf{D}$  un langage de description ;  $\mathbf{R}$  une relation dans  $\mathbf{D} \times \mathbf{D}$ , appelée "relation fondatrice" ;  $\mathbf{O}$  un sous-ensemble de  $\mathbf{D}$ , appelé "ensemble d'observation" ; et  $f_a, f_b$  deux éléments de  $\mathbf{D}$ , appelés descripteurs,

**Définition 1** :  $f_a$  et  $f_b$  sont en relation de Dépendance Perçue ( $f_a \rightarrow f_b$ ) si et seulement si :

$$\exists o \in \mathbf{O}, \forall e \in \mathbf{O}, (o \mathbf{R} f_a) \wedge (e \mathbf{R} f_a \Rightarrow e \mathbf{R} f_b)$$

cela peut se traduire par : nous avons observé au moins un exemple  $o$  de  $\mathbf{O}$  contenant le descripteur  $f_a$  dans sa description, et tous les exemples  $e$  de  $\mathbf{O}$  observés et qui contiennent  $f_a$  dans leur description, contiennent aussi  $f_b$  dans leur description. L'ensemble des exemples qui contiennent  $f_a$  dans leur description est donc inclus dans l'ensemble des exemples qui contiennent  $f_b$  dans leur description.

Si nous désignons par  $I_{O,R}$  l'image de  $\mathbf{O}$  dans  $\mathbf{D}$  par  $\mathbf{R}$ , c'est à dire que :

$$I_{O,R} = \{d \in \mathbf{D} / \exists e \in \mathbf{O}, e \mathbf{R} d\}$$

alors la Dépendance Perçue est une relation de  $I_{O,R} \times I_{O,R}$ .

**Définition 2** :  $f_a$  et  $f_b$  sont en relation d'Equivalence Perçue si et seulement si :

$$f_a \rightarrow f_b \text{ et } f_b \rightarrow f_a, \text{ cette relation est notée par : } f_a \leftrightarrow f_b.$$

$$\exists o \in \mathbf{O}, \forall e \in \mathbf{O}, (o \mathbf{R} f_a) \wedge [(e \mathbf{R} f_a \Rightarrow e \mathbf{R} f_b) \wedge (e \mathbf{R} f_b \Rightarrow e \mathbf{R} f_a)]$$



cela peut se traduire par : nous avons observé au moins un exemple  $o$  de  $O$  contenant l'élément  $f_a$  dans sa description, et tous les exemples  $e$  de  $O$  observés et qui contiennent  $f_a$  dans leur description, contiennent aussi  $f_b$  dans leur description, et tous les exemples  $e$  de  $O$  observés et qui contiennent  $f_b$  dans leur description, contiennent aussi  $f_a$  dans leur description. L'ensemble des exemples qui contiennent  $f_a$  dans leur description est donc égal à l'ensemble des exemples qui contiennent  $f_b$  dans leur description.

Le langage de description  $D$  est constitué d'un ensemble de formules dans lequel le seul connecteur utilisé est la conjonction. Une observation est donc décrite sous forme d'une conjonction d'éléments atomiques.

Nous pouvons vérifier facilement que la relation de Dépendance Perçue constitue un préordre (ordre partiel) et que la relation d'Equivalence Perçue constitue une relation d'équivalence.

**Définition 3** : Une classe d'Equivalences Perçues est une partie de  $O$  formée de tous les éléments qui sont en relation d'Equivalence Perçue à l'un d'entre eux.

**Théorème** : Deux classes d'équivalences Perçues sont disjointes ou confondues.

**Preuve** : soit  $CE_a$  et  $CE_b$  deux classes d'Equivalences Perçues non confondues, définies respectivement par  $a$  et  $b \in O$ . Supposons que ces deux classes ne sont pas disjointes et soit  $c \in CE_a \cap CE_b$ . Nous avons  $c \in CE_a \Rightarrow c R a$  ; et  $c \in CE_b \Rightarrow c R b$ . Nous déduisons, de la symétrie et de la transitivité, que  $a R b$ , autrement dit que  $b \in CE_a$  ; tous les éléments de  $CE_b$  appartiennent alors à  $CE_a$  puisqu'ils sont en relation d'Equivalence Perçue avec  $b$  ; donc  $CE_a = CE_b$ . Ainsi, deux classes sont confondues si elles ont un élément en commun.

Il résulte du théorème ci-dessus que les classes d'Equivalences Perçues forment une partition de  $O$ . Autrement dit,  $O$  est partagé en un nombre fini de classes d'Equivalence  $CE_i$ , disjointes deux à deux, et dont la réunion est  $O$ . Nous avons donc :

$$O = \bigcup_i CE_i, \text{ et } CE_i \cap CE_j = \emptyset \text{ si } i \neq j$$

### 3.4.7.3 Les connaissances représentées dans le RDP

Un Réseau de Dépendances Perçues (RDP) est la structure de graphe orienté simple, sans circuit, permettant une représentation des Dépendances Perçues (DP). Ce graphe peut s'entretenir incrémentalement pour intégrer de nouvelles observations. Chaque noeud du graphe est constitué par les éléments de  $\mathbf{O}$  qui sont en relation d'Equivalence Perçue. Le RDP est constitué donc d'un ensemble de classe d'Equivalences Perçues.

D'une façon plus formalisée, le graphe  $RDP = (V_{RDP}, \varepsilon_{RDP})$  est le couple constitué par :

- 1)° un ensemble de noeuds  $V_{RDP} = \{n_1, n_2, \dots, n_m\}$
- 2)° une famille  $\varepsilon_{RDP} = \{u_1, u_2, \dots, u_k\}$  d'éléments du produit cartésien :  $V_{RDP} \times V_{RDP} = \{(x, y) / x \in V_{RDP}, y \in V_{RDP}\}$  représentant l'ensemble des arcs orientés (paires de noeuds).

*Remarque :*

Deux éléments  $f_a$  et  $f_b$  appartenant respectivement aux classes d'Equivalences Perçues associées aux noeuds  $n_a$  et  $n_b$  seront en relation de Dépendance Perçue si et seulement si il existe un chemin dans le graphe allant de  $n_a$  à  $n_b$  ; ils seront en relation d'Equivalence Perçue si et seulement si  $n_a = n_b$ , c'est-à-dire s'ils appartiennent au même noeud.

## 3.5 CHOIX D'UN SYSTEME D'APPRENTISSAGE : LES RDP

Parmi les systèmes d'apprentissage que nous avons décrit, nous pouvons constater que les arbres de décision (ID3, KATE) ne permettent pas un apprentissage incrémental et nécessitent que la description des exemples comportent tous les attributs appartenant au langage de description. Le système ALEX nécessite une définition précise des problèmes et de leurs solutions pour construire une base d'expériences. De plus, le calcul d'un score d'appariement structurel qui permet de déduire des problèmes similaires à d'autres nécessite une connaissance implicite sur le domaine. Les systèmes COBWEB et UNIMEM permettent

de traiter les exemples d'apprentissage de façon incrémentale, mais l'ordre d'arrivée de ces exemples peut influencer sur la structure des classes de concepts construites. Ces deux systèmes nécessitent aussi des connaissances implicites sur le domaine pour pouvoir évaluer la distance entre les concepts et prendre en compte des informations incertaines. PROTOS qui est un système d'apprentissage à partir de cas, nécessite une description précise de toutes les propriétés des objets constituant les cas typiques pour aboutir à des catégories de classes significatives. Les RDP permettent un apprentissage incrémental à partir d'exemples. A l'inverse de COBWEB et de UNIMEM, les RDP ne nécessitent pas des connaissances implicites pour construire des classes d'objets.

Un autre avantage des RDP est que l'ordre d'arrivée des exemples d'apprentissage n'influe pas sur la construction du graphe de connaissances. Puisque, dans notre cas d'application, les observations peuvent être fournis par les experts dans un ordre quelconque, cette propriété est indispensable pour la construction de la Base de Connaissances.

Etant donné que la connaissance du domaine de l'eau évolue dans le temps, et qu'elle est dispersée chez plusieurs experts, un traitement incrémental des cas d'intervention s'impose [Senoune 94a], [Senoune 94b]. Les RDP permettent non seulement d'intégrer incrémentalement de nouveaux exemples dans la structure de connaissances construite, mais aussi d'améliorer le langage de description pour la prise en compte de nouveaux concepts qui apparaissent dans le domaine.

Pour aboutir à un modèle représentatif du domaine, il faut traiter un grand nombre d'observations. Les RDP, en ne représentant que les objets nécessaires à un niveau donné, optimisent la représentation du domaine de connaissances et permet ainsi de prendre en charge plusieurs observations. La représentation des connaissances sous forme d'un graphe facilite leur validation ainsi que leur exploitation.

Ces critères de choix nous ont amené à utiliser la théorie de la Dépendance Perçue pour construire incrémentalement la Base de Connaissances de notre système.

### 3.5.1 Le langage de description des exemples

Les travaux sur les RDP sont proches de ceux de Mitchell sur les espaces de version [Mitchell 82], qui eux aussi permettent de traiter progressivement un ensemble d'observations. Dans un RDP, les éléments les plus généraux sont représentés dans le noeud correspondant au niveau le plus haut du graphe. Ce noeud constitue une tautologie du fait qu'il couvre et décrit tous les exemples d'apprentissage. Dans les espaces de version le niveau correspondant à la tautologie est appelé "la description nulle", et les descriptions les plus spécifiques sont les exemples d'apprentissage eux-mêmes.

D'autres travaux proches des RDP sont présentés dans [Mineau 90]. Il s'agit d'une méthode de classification symbolique d'un ensemble d'objets de connaissances qui sont décrits comme étant des objets structurés. La structure de représentation des connaissances est basée sur les graphes conceptuels. Dans ce formalisme, les connaissances sont représentées sous forme de graphes où les boîtes (rectangles) représentent des concepts et les cercles représentent des relations entre les concepts. Les exemples sont considérés incrémentalement et le but est d'élaborer une classification hiérarchique.

Nous avons indiqué auparavant que la théorie de la Dépendance Perçue permet de révéler des relations qui existent dans un ensemble d'observations dans un domaine donné. Il est donc nécessaire, pour utiliser cette théorie, de disposer d'observations qui sont faciles à collecter et qui couvrent bien le domaine de connaissances. Pour ce qui concerne le domaine de l'eau, nous avons montré dans le premier chapitre, que les experts font naturellement référence à leurs expériences passées pour accomplir des tâches données. Ils travaillent aussi par adaptation des processus de résolutions qu'ils ont déjà effectués dans des situations similaires. La première étape, qui a consisté à utiliser une méthodologie pour acquérir les connaissances établies du domaine et trouver une forme adéquate de description des observations, permet donc de faciliter aux experts du domaine de délivrer leurs savoir sous formes d'exemples d'apprentissage.

Dans notre cas, la notion d'apprentissage à partir d'exemples est fondée sur les régularités qui existent entre les objets dans ces exemples. Pour pouvoir détecter ces régularités, il faut d'abord pouvoir décrire les exemples avec des concepts et des relations entre ces concepts.

C'est pourquoi le choix des éléments qui vont constituer le langage de description des exemples est important.

Nous allons maintenant décrire sur un exemple simple comment à partir d'un ensemble d'observations, le système construit un RDP représentant les connaissances et les relations apprises.

### 3.5.2 Comment se construit un RDP

Les exemples d'apprentissage, sensés décrire des états représentatifs de l'application, sont écrits sous forme d'une conjonction de descripteurs et constituent l'ensemble des observations servant de base au processus de construction du RDP. L'ensemble des descripteurs utilisés pour représenter les exemples d'apprentissage doit être défini. Leur définition comprend la spécification de leur nom, leur type, leur domaine (les valeurs autorisées) et les restrictions qui portent sur leurs conditions d'utilisation.

Soit  $e$  un exemple, nous appellerons  $d(e)$  sa description.

$d(e) = d_1 \& d_2 \& d_3 \& \dots \& d_n$  où  $d_1, d_2, \dots, d_n$  sont des descripteurs.

Chaque descripteur utilisé est représenté par un triplet :  $d_i = (\text{attribut}, \text{sélecteur}, \text{valeur})$ .

Supposons que les observations dans notre domaine, nécessitent l'utilisation de trois attributs:

- nom : (jean, pierre, alain),
- lieu : (lyon, nice, paris),
- compétence : (cours\_eau, eau\_de\_surface, infrastructure),

et supposons deux exemples  $e_1$  et  $e_2$  dont la description est :

$d(e_1) : \{(nom=jean)\&(lieu=paris)\&(compétence=eau\_de\_surface)\}$

$d(e_2) : \{(nom=pierre)\&(lieu=paris)\&(compétence=infrastructure)\}$

pour les deux exemples  $e_1$  et  $e_2$ , nous obtenons la représentation suivante :

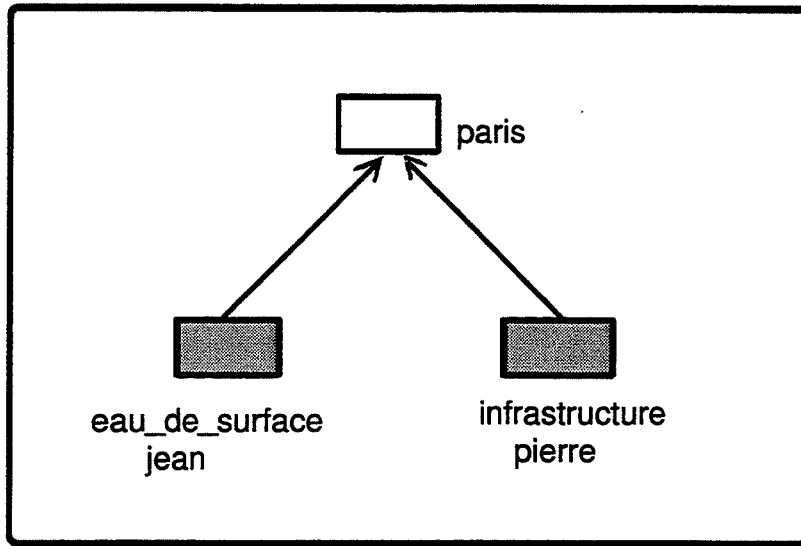


Figure 3.2 : RDP correspondant aux exemples d'apprentissage  $\{e_1, e_2\}$

Si nous considérons les trois exemples dont la description est :

$d(e_1) : \{(nom=jean)\&(lieu=paris)\&(competence=eau\_de\_surface)\}$

$d(e_2) : \{(nom=pierre)\&(lieu=paris)\&(competence=cours\_d'eau)\&(competence=infrastructure)\}$

$d(e_3) : \{(nom=alain)\&(lieu=paris)\&(competence=cours\_d'eau)\&(competence=infrastructure)\}$

le graphe résultant de ces observations est le suivant :

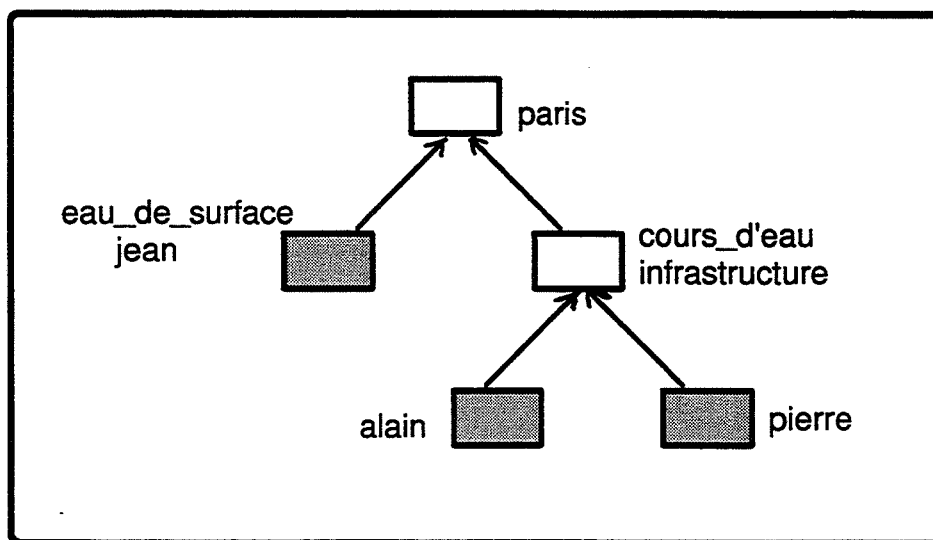


Figure 3.3 : RDP correspondant aux exemples d'apprentissage  $\{e_1, e_2, e_3\}$

Si nous analysons la description de ces exemples d'apprentissage, nous constaterons que "cours\_d'eau" apparaît à chaque fois que "infrastructure" apparaît et réciproquement. Cela signifie que ces deux éléments sont en relation d'Equivalence Perçue, et par conséquent ils appartiennent au même noeud du graphe représentant les observations. La même constatation peut être faite pour "eau\_de\_surface" et "jean".

A chaque fois que nous avons observé "alain", nous avons observé aussi "cours\_d'eau" et "infrastructure" mais la réciproque n'étant pas vraie. Il en ressort que "alain" et "cours\_d'eau, infrastructure" sont en relation de Dépendance Perçue ; ce qui se traduit par un arc allant du noeud associé à "alain" au noeud associé à "cours\_d'eau, infrastructure".

Le terme "paris" apparaît dans la description de tous les exemples, il est contenu dans le noeud correspondant au sommet du graphe pour pouvoir être partagé par toutes les observations.

Les noeuds caractérisés par un fond plein représentent la base des d'observations qui constituent l'ensemble d'apprentissage. Partant de l'un de ces noeuds, nous pouvons reconstruire un exemple unique en remontant jusqu'à la racine du graphe et en formant des conjonctions entre les termes contenues dans les noeuds. Ainsi, nous avons trois noeuds qui correspondent chacun à un exemple d'apprentissage.

Si l'observation obs1 devient obsolète et que nous la retirons de l'ensemble d'apprentissage, la reconstruction du RDP donnera comme résultat le graphe de la figure 3.4. Nous constatons alors l'existence des relations suivantes:

paris  $\longleftrightarrow$  cours\_d'eau,  
 paris  $\longleftrightarrow$  infrastructure,  
 cours\_d'eau  $\longleftrightarrow$  infrastructure.

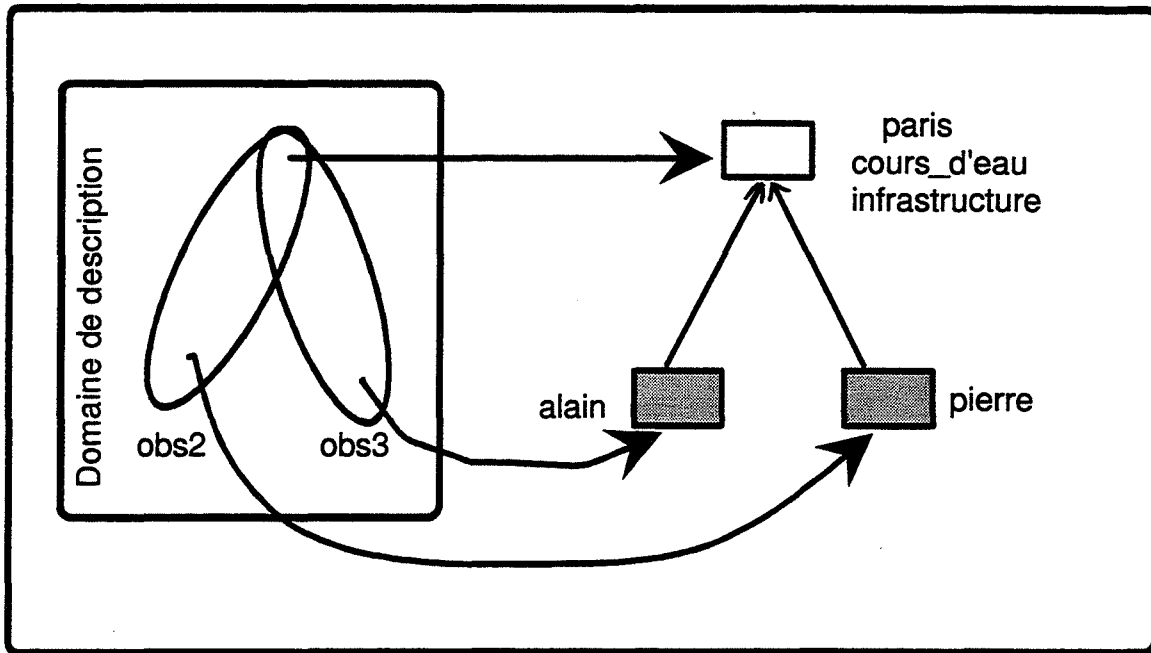


Figure 3.4 : RDP correspondant aux exemples  $\{e_2, e_3\}$

Les exemples donnés ci-dessus ont été choisis volontairement simples pour mieux illustrer comment des éléments de connaissances sont structurés et hiérarchisés dans un Réseau de Dépendances Perçues. Le regroupement de concepts équivalents dans les mêmes noeuds et les liens existants (explicitement schématisés par des arcs) entre les noeuds contenant des concepts dépendants les uns des autres, permettent de découvrir des connaissances intermédiaires [Senoune 94a]. Ces connaissances sont constituées de sous ensembles de descripteurs qui représentant les observations fournies au système. Elles servent, par conséquent, à faciliter la description de problèmes en termes de blocs d'éléments génériques.

### 3.6 CONCLUSION

Dans certains domaines, la construction de Bases de Connaissances est facilitée par le fait que l'expertise nécessaire au bon fonctionnement du système est bien déterminée et disponible en totalité chez un ou plusieurs experts. Dans d'autres cas, les connaissances évoluent et les experts du domaine acquièrent de plus en plus de savoir. Il est alors nécessaire



d'appliquer un apprentissage incrémental pour recueillir les informations au fur et à mesure qu'elles arrivent dans le système. De plus, nous savons que l'être humain, pour exprimer ou faire valoir son savoir, a tendance à donner des exemples. Il est donc plus facile de demander à un expert de fournir des exemples traitant d'un sujet particulier que d'exiger de lui une description directe de son savoir. C'est pour ces raisons que nous avons choisi de développer la Base de Connaissances de notre système en exploitant incrémentalement des exemples d'intervention fournis par les experts de ce domaine.

Dans le chapitre 4, nous décrivons les étapes de construction d'un environnement d'observation qui est nécessaire à la description des exemples d'apprentissage, au processus d'apprentissage et à l'exploitation du graphe de connaissances construit. Nous décrivons aussi l'algorithme permettant d'intégrer incrémentalement des exemples d'apprentissage dans le RDP, et nous expliciterons le format utilisé pour la description des cas d'intervention fournis par les experts du domaine. Le système ainsi construit est appelé "EXPEAU" (EXPerTs en EAU).

## ***4. Chapitre 4 :Intégration d'une méthodologie d'acquisition de connaissance et d'une technique d'apprentissage : le système EXPEAU***

### **4.1 INTRODUCTION**

Dans le chapitre 1, nous avons montré que le développement d'un SBC pour les situations de crise dans le domaine de l'eau nécessite l'utilisation d'une méthodologie d'acquisition de connaissances pour acquérir les connaissances descriptives du domaine ainsi que des stratégies de résolution ; et d'une technique d'apprentissage pour acquérir incrémentalement les connaissances en exploitant des cas d'interventions effectuées par les experts pour résoudre des problèmes en situation de crise. Dans le deuxième chapitre nous avons montré comment nous pouvons se rapprocher des niveaux de connaissances utilisés dans la méthodologie KADS pour acquérir la connaissance profonde du domaine et la méta-connaissance sur son utilisation. Dans le troisième chapitre, nous avons expliqué pourquoi la

théorie de la Dépendance Perçue peut résoudre le problème de l'intégration automatique des exemples d'intervention fournis par les experts du domaine dans la BC. Dans ce chapitre nous montrerons comment les connaissances du domaine peuvent être utilisées pour la spécification d'un environnement sur lequel s'appuiera le processus d'apprentissage. Nous décrirons, par la suite, l'algorithme utilisé pour intégrer incrémentalement dans la BC, les exemples d'apprentissage représentant des cas d'intervention effectués par les experts ; et nous donnerons, enfin, le langage de description utilisé par les experts du domaine pour décrire les cas d'intervention.

Dans un premier temps, nous devons donc recueillir la connaissance profonde du domaine qui servira non seulement pour le processus de résolution mais aussi pour le processus d'apprentissage. Par la suite, une technique d'apprentissage à partir d'observations est utilisée pour construire incrémentalement la BC du système EXPEAU. Cet apprentissage s'effectue en s'appuyant sur les connaissances recueillies auparavant et qui constituent **l'environnement d'apprentissage**. Dans le paragraphe qui suit nous décrivons un tel environnement.

## 4.2 ENVIRONNEMENT D'APPRENTISSAGE

Avant de rentrer dans le vif du sujet concernant la construction incrémentale du système EXPEAU, nous allons présenter le cadre dans lequel l'acquisition de connaissances va être effectuée. Pour pouvoir représenter les exemples d'apprentissage qui expriment, dans notre cas, des observations du monde réel, il est nécessaire de déterminer un langage de description approprié. Or ce langage de formulation dépend en partie de la nature des observations qu'il exprime, ce qui signifie que le choix du langage est contraint par la nature des connaissances à représenter. Mais nous savons aussi que le mode de représentation utilisé peut influencer sur l'opération de détection de régularités dans un ensemble d'observations [Amarel 68], [Ganascia 87]. Il est donc évident que le choix d'un langage pertinent pour décrire les observations est un préalable important pour le processus d'apprentissage.

Pour ce qui concerne notre application, il s'agit de spécifier un langage qui permettra de traduire les cas d'intervention fournis par les experts du domaine. Comme nous le verrons plus

loin, ces observations sont constituées d'un ensemble de connaissances portant sur le domaine de l'intervention, la date de l'intervention, le rôle joué pendant cette intervention, les compétences mises en oeuvre, etc. Le langage de représentation utilisé repose sur la logique des prédicats du premier ordre. Ce formalisme se trouve approprié et suffisant pour représenter les cas d'intervention effectués par les experts du domaine.

L'environnement d'apprentissage de notre système est composé de deux parties dépendantes l'une de l'autre, à savoir :

- un environnement des observations : cet environnement spécifie le langage dans lequel seront représentés les exemples d'apprentissage,
- des exemples d'apprentissage : les observations fournies par les experts du domaine et représentées dans le langage spécifié dans l'environnement des observations constituent les exemples d'apprentissage qui seront utilisés par le système d'apprentissage pour construire la BC.

#### **4.2.1 Environnement des observations**

La définition de l'environnement des observations, nécessite tout d'abord de déterminer l'ensemble des éléments permettant de décrire les exemples d'apprentissage. Pour chacun de ces éléments, il faut préciser le domaine de définition et la fonction de généralisation correspondante. Nous verrons que les éléments du langage peuvent avoir des domaines de définition structurés différemment selon leur nature. Ainsi, la définition des domaines des éléments de nature "hiérarchique" exigent la construction de taxinomies. Dans ce qui suit, nous présenterons ces trois étapes qui permettent de spécifier l'environnement des observations.

#### 4.2.1.1 Langage de représentation des observations

Pour traduire les observations fournies par les experts en un ensemble d'exemples d'apprentissage, un langage de description pertinent est construit. Il renferme tous les éléments nécessaires pour décrire les exemples d'intervention des experts sur des cas de crise.

Une observation est décrite par une liste de descripteurs reliés entre eux par l'opérateur de "conjonction". Cette forme de description permet en effet de représenter totalement une intervention effectuée par un expert. L'opérateur de "disjonction" n'a pas d'utilité pour la description des exemples d'apprentissage (nous verrons par la suite que pour ce qui concerne l'étape d'exploitation des connaissances, la reformulation d'une description d'une situation de crise prend implicitement en charge la notion de "disjonction"). Chaque descripteur utilisé est représenté par un triplet  $d_i = (\text{attribut}, \text{sélecteur}, \text{valeur})$ . La sémantique qui est liée à ces descripteurs dépend du domaine d'application. Si, par exemple, nous définissons un attribut pour identifier l'expert X qui a fourni une observation, la sémantique de cet attribut sera "l'expert X a résolu le problème dont les caractéristiques sont données par le reste des descripteurs de l'observation". Si pour cette même observation, nous définissons un attribut pour identifier les compétences  $C_1, C_2, \dots, C_n$  requises pour résoudre le problème, la sémantique de cet attribut sera "l'expert X a les compétences  $C_1, C_2, \dots, C_n$ ". Nous constatons qu'en liant une sémantique à chaque attribut défini, nous aboutissons à une sémantique globale d'une observation représentant un exemple d'intervention sur des cas réels [Senoune 95].

Par rapport à la sémantique associée à chaque attribut, il est nécessaire de définir le domaine de valeurs que nous pouvons lui affecter. Pour mieux prendre en compte les différents "types" de données que nous pouvons trouver dans une observation, les attributs sont caractérisés par un "type" qui peut être : **nominal**, **hiérarchique** ou **linéaire**. Cette structuration des attributs permet de prendre en compte un certain nombre de propriétés du langage de description au cours du processus d'apprentissage et d'explicitier des relations entre descripteurs qui ne sont pas mises en évidence initialement dans les cas d'observation.

- Le domaine de définition d'un attribut de type **nominal** est l'ensemble amorphe de ses éléments. Il est constitué d'un ensemble d'éléments non ordonnés et non structurés ; ce qui signifie que ces éléments sont indépendants les uns des autres.

- Le domaine de définition d'un attribut de type **linéaire** est un ensemble d'éléments ordonnés. Ce domaine de définition est muni d'un ordre total.

- Le domaine de définition d'un attribut de type **hiérarchique** est un ensemble d'éléments structurés. Ce domaine de définition est muni d'un ordre partiel.

Pour définir les éléments constituant le domaine de définition d'un attribut, nous pouvons procéder de trois manières différentes :

1)° la **définition en extension** : la définition en extension consiste à fournir explicitement la liste des éléments du domaine. Pour les attributs de type hiérarchique, l'extension est donnée sous forme d'une liste de listes. Chaque élément de la structure est défini de la manière suivante : (élément . père\_de\_cet\_élément), à l'exception de l'élément maximal qui est sous forme du seul élément correspondant : (élément\_maximal).

**Exemples :**

**nom de l'attribut : booléen, type de l'attribut : nominal, domaine de définition : {oui, non}**

Dans ce type de définition, il s'agit d'énumérer la liste des valeurs que peut prendre un attribut.

**nom de l'attribut : nombre, type de l'attribut : linéaire, domaine de définition : {1,2 ... 20}**

Nous pouvons vérifier facilement l'existence d'un ordre total sur un tel domaine de définition. En effet, quels que soient les éléments  $x$  et  $y$  de cet ensemble, nous avons toujours  $(x \neq y) \Rightarrow ((x < y) \text{ ou } (y < x))$  ; ce qui signifie que cet ensemble est totalement ordonné.

**nom de l'attribut : compétence, type de l'attribut : hiérarchique, domaine de définition : {(Eaux\_de\_surface, Eaux\_douces) (Eaux\_souterraines, Eaux\_douces) (Haute\_mer, Eaux\_marines) (Littoral, Eaux\_marines) (Estuaire, Eaux\_marines) (Île, Eaux\_marines) (Eaux\_douces, Eaux) (Eaux\_marines, Eaux) (Eaux)}**

Pour les attributs de type hiérarchique, l'ensemble des éléments du domaine de définition est partiellement ordonné car la relation d'ordre qui existe sur un tel domaine ne permet pas de comparer tous les éléments entre eux.

**2)° la définition en compréhension** : au lieu d'énumérer explicitement tous les éléments du domaine de définition d'un attribut, il est parfois préférable (lorsque cela est possible) de donner une définition en compréhension de cet ensemble. La définition en compréhension du domaine d'un attribut consiste à fournir une fonction qui détermine le domaine d'un attribut. Pour ce qui concerne les attributs de type linéaire, comme l'ensemble des éléments de leur domaine de définition est muni d'une relation d'ordre total, nous pouvons donc indiquer les bornes inférieure et supérieure du domaine de définition. Toutefois, comme nous l'avons indiqué auparavant, les attributs de type hiérarchique ne peuvent en aucun cas être défini en compréhension. Ils nécessitent donc une définition explicite (définition en extension).

### Exemples

**nom de l'attribut : booléen, type de l'attribut : nominal, domaine de définition : chaîne\_de\_caractères**

Tel qu'il est défini ici, le domaine des valeurs de l'attribut peut prendre n'importe quelle chaîne de caractères. Mais, nous pouvons tout de même restreindre ce domaine en construisant des fonctions d'appartenance spécifiques.

**nom de l'attribut : nombre, type de l'attribut : linéaire, domaine de définition : nombre\_entier :borne\_inférieure 1 :borne\_supérieure 20**

Ici, l'attribut prend ces valeurs dans l'intervalle [1, 20].

**3)° la définition par "union de sous-domaines"** : dans ce cas, nous devons définir les sous-domaines constituant le domaine de définition de l'attribut. L'utilisation de ce mode de définition peut être implicite ou explicite. Par exemple, un domaine nominal qui n'est défini ni en compréhension ni en extension est défini implicitement comme l'union de ses sous-domaines. Si un domaine est défini en compréhension ou en extension et de plus des sous-domaines sont indiqués, alors son extension complète sera l'union des éléments de son extension et de l'union implicite des extensions de ses sous-domaines. Pour un domaine de type

hiérarchique, il est nécessaire d'indiquer explicitement l'élément supérieur auquel sera ajouté l'union des sous-domaines.

Mais pour résoudre le problème des incohérences qui peuvent résulter de la définition par "union de sous-domaines", il faut tenir compte du type des attributs à spécifier. Ainsi, lors de la définition implicite ou explicite par "union de sous-domaines", quelques contraintes doivent être vérifiées, à savoir :

- un domaine hiérarchique ne peut contenir que des sous-domaines de type hiérarchique et linéaire de façon à préserver la relation d'ordre partiel entre les éléments du domaine.

- un domaine linéaire ne peut contenir que des sous-domaines de type linéaire.

A l'inverse d'un domaine linéaire ou hiérarchique, un domaine nominal peut contenir des sous-domaines de n'importe quel type (hiérarchique, linéaire et nominal).

Cette connaissance sur le domaine est très intéressante lors du processus d'apprentissage pour restreindre et orienter la production de description de concepts ainsi que lors de l'exploitation proprement dite des connaissances apprises pour situer les éléments de connaissances qui ne sont pas représentés sur le RDP.

Pour définir le domaine des valeurs que peuvent prendre des attributs de type hiérarchique, il est nécessaire de construire des taxinomies. Dans ce qui suit, nous allons voir l'intérêt d'avoir des taxinomies pertinentes et les difficultés rencontrées pour les construire.

#### **4.2.1.2 Construction de taxinomies**

Plusieurs Systèmes d'Apprentissage utilisent des connaissances représentées sous forme de taxinomies exprimant, par nature, des relations de généralité et de spécificité entre les objets d'un domaine. De par leur forme de représentation, les taxinomies permettent de faire facilement des généralisations sur les éléments de connaissances du domaine. Pour ce qui concerne notre cas d'application, nous allons voir par la suite qu'une partie des attributs du



langage de description des exemples d'apprentissage peut être définie sous une forme structurée ; ce qui contribue à améliorer l'exploitation des connaissances du domaine.

Nous avons indiqué précédemment que nous pouvons définir le domaine de valeurs des attributs de trois manières différentes : en **compréhension**, en **extension** et par **union de sous-domaines**. Nous avons indiqué aussi que la définition des éléments d'un attribut de type "hiérarchique" ne peut en aucun cas se faire en "compréhension" et qu'ils doivent être déclarés en extension. Par conséquent, il est nécessaire de classifier toutes les valeurs des éléments structurés sous forme d'une taxinomie. C'est aux experts du domaine en collaboration avec le cogniticien qu'incombe la tâche d'imaginer une classification d'un ensemble d'éléments qui convient le mieux à l'utilisation auquel elle est destinée. Mais, l'expérience a montré que même au bout d'un grand effort, les taxinomies produites ne représentent pas toujours des classifications uniques et universelles que nous pouvons appliquer dans n'importe quel contexte. Nous savons aussi que la construction d'une taxinomie dépend de plusieurs facteurs, parmi lesquels nous pouvons citer :

- le problème à résoudre,
- le point de vue et l'expérience du constructeur.

Ainsi le problème de subjectivité des experts peut influencer sur la forme des classifications obtenues. Dans le domaine de l'eau qui renferme plusieurs disciplines, les informations considérées comme pertinentes pour un chimiste, par exemple, peuvent ne pas forcément l'être pour un biologiste. Le contexte d'utilisation des connaissances peut aussi invalider la forme d'une taxinomie construite pour un contexte différent. Dans un contexte particulier une personne considérera, par exemple, les nitrates et les phosphates comme des polluants chimiques et dans un autre contexte la même personne trouvera préférable de dire que les nitrates et les phosphates sont des sels. La validation de ces structures par les différents experts est donc indispensable pour aboutir à des classifications qui conviennent aux experts des différentes disciplines.

Pour notre cas, les taxinomies sont construites au cours d'une série d'entretiens dirigés où les experts doivent classifier les objets qu'ils manipulent. Nous aurions pu adopter la méthode des entretiens libres (qui consiste à permettre aux experts d'exprimer leurs connaissances sans les interrompre et sans les orienter vers un but précis), mais cela aurait abouti à une masse

importante d'information dont une partie peut être inutile [Feng 93]. En plus de cet inconvénient qui rend lourd l'analyse de ces informations, les experts peuvent ne pas aborder les problèmes essentiels vis-à-vis de l'application. Nous avons donc préféré utiliser les entretiens dirigés pour mieux structurer le recueil de connaissances et faciliter leur analyse.

Les différentes classifications sont effectuées en tenant compte de l'objectif du système, à savoir la description de situation de crise et la sélection des experts pouvant résoudre des problèmes engendrés par une situation de crise donnée. Pour arriver à collecter le maximum d'information, l'entretien à tenir avec les experts est préparé à l'avance. Cela consiste en une familiarisation avec les connaissances du domaine, et en une préparation d'un ensemble de questions pertinentes et organisées en tenant compte du but à atteindre. Ainsi pour ce point bien précis qui consiste à construire des taxinomies, les questions peuvent être de la forme : est-ce que l'élément X renferme l'élément Y ?, quelles sont les éléments que vous connaissez et qui sont une sorte de l'élément Z ?, etc.

Lorsque les experts estiment avoir fourni tous les éléments du domaine de définition relatif à un attribut de type hiérarchique, nous construisons la taxinomie de ces éléments en utilisant une représentation graphique [Desfray 92]. Cette représentation graphique est alors présentée aux différents experts pour qu'ils la valident et la complètent en cas d'oubli d'informations.

Après avoir décrit la manière de définir les domaines des différents attributs du langage de description des exemples d'apprentissage, nous allons présenter maintenant les fonctions de généralisation appliquées à chaque type d'attribut.

### **4.2.1.3 Fonctions de généralisation**

Pour notre système d'apprentissage à partir d'un ensemble d'observations, déterminer le généralisé de deux ou de plusieurs descripteurs de ces observations est une opération essentielle. Les systèmes d'apprentissage portent une attention particulière sur les fonctions de généralisation qui permettent d'aboutir aux généralisés les plus pertinents. Une définition formelle de la notion de "plus général" est nécessaire pour pouvoir vérifier la validité des

généralisés obtenus. Dans ce qui suit, nous allons donc tenter de donner une telle définition formalisée de cette notion et de la fonction de généralisation elle-même.

Soient  $\mathfrak{G}$  une fonction de généralisation et  $\mathbf{R}$  la relation fondatrice ; si  $\mathbf{d}_1$  et  $\mathbf{d}_2$  sont deux descripteurs de  $\mathbf{D}$ , alors  $\mathfrak{G}(\mathbf{d}_1, \mathbf{d}_2)$  correspondra au généralisé le plus spécifique de ces deux éléments et peut être formalisée comme suit [Sainte Marie 89], [Royer 90] :

$$\mathfrak{G} : \mathbf{D} \times \mathbf{D} \rightarrow \mathbf{D}$$

$$(\mathbf{d}_1, \mathbf{d}_2) \rightarrow \{\mathbf{d}_g \in \mathbf{D} / \mathbf{d}_1 \mathbf{R} \mathbf{d}_g \wedge \mathbf{d}_2 \mathbf{R} \mathbf{d}_g \wedge (\forall \mathbf{d}'_g \in \mathbf{D}, \mathbf{d}_1 \mathbf{R} \mathbf{d}'_g \wedge \mathbf{d}_2 \mathbf{R} \mathbf{d}'_g \Rightarrow \mathbf{d}_g \mathbf{R} \mathbf{d}'_g)\}$$

Pour expliciter plus clairement cette formulation, nous introduisons la notion intuitive de "plus général", pour remplacer la relation " $\mathbf{R}$ " par la relation " $<$ " ; où  $\mathbf{d}_i < \mathbf{d}_j$  signifie : " $\mathbf{d}_j$  est plus général que  $\mathbf{d}_i$ ". Ainsi nous obtenons la formalisation suivante :

$$\mathfrak{G} : \mathbf{D} \times \mathbf{D} \rightarrow \mathbf{D}$$

$$(\mathbf{d}_1, \mathbf{d}_2) \rightarrow \{\mathbf{d}_g \in \mathbf{D} / \mathbf{d}_1 < \mathbf{d}_g \wedge \mathbf{d}_2 < \mathbf{d}_g \wedge (\forall \mathbf{d}'_g \in \mathbf{D}, \mathbf{d}_1 < \mathbf{d}'_g \wedge \mathbf{d}_2 < \mathbf{d}'_g \Rightarrow \mathbf{d}_g < \mathbf{d}'_g)\}$$

par extension, si  $\mathbf{D}_1$  et  $\mathbf{D}_2$  sont deux listes de descripteurs de  $\mathbf{D}$ , alors  $\mathfrak{G}(\mathbf{D}_1, \mathbf{D}_2)$  correspond aux généralisés les plus spécifiques de chacun des attributs communs à  $\mathbf{D}_1$  et  $\mathbf{D}_2$ .

Nous pouvons constater que  $\mathbf{d}_g = \mathfrak{G}(\mathbf{D}_1, \mathbf{D}_2)$  est unique et symétrique. Par conséquent, l'ordre d'arrivée des exemples d'apprentissage n'influe pas sur la construction du graphe de connaissances. De plus, la détermination des "généralisés" se fait incrémentalement, c'est à dire qu'il est possible de déterminer la généralisation d'un certain nombre de formules et ensuite la généralisation d'autres formules avec les généralisés déjà obtenus.

Pour construire la BC de notre système en utilisant des cas d'intervention, ces deux propriétés (unicité, symétrie) sont indispensables. En effet, comme les observations qui servent d'exemples d'apprentissage peuvent arriver dans un ordre quelconque, si la fonction de généralisation ne vérifie pas la propriété de la "symétrie", les connaissances acquises seront donc différentes selon l'ordre d'arrivée de ces exemples.

Comme les attributs du langage de description peuvent avoir des types différents (nominal, linéaire, hiérarchique), nous définissons une fonction de généralisation pour chacun de ces types d'attributs :

- pour les attributs de type **nominal**, nous choisissons la fonction de **remplacement des constantes par des variables** [Vere 75], [Vere 80], [Michalski 83a], [Bollinger 86]. Cette fonction de généralisation consiste à remplacer des occurrences d'une constante par une variable.

$\mathfrak{G}(d=a, d=b) = d=x$ , où  $a$  et  $b$  sont des constantes et  $x$  une variable, avec  $x = \{a, b\}$

- pour les attributs de type **hiérarchique**, nous choisissons la fonction de **remontée dans une hiérarchie conceptuelle** [Vere 75], [Vere 80], [Michalski 83a], [Bollinger 86]. Cette fonction de généralisation donne comme résultat les plus proches successeurs communs des éléments à généraliser.

$\mathfrak{G}(d=a, d=b) = d=e$ , où  $a$  et  $b$  sont des constantes et  $e$  une autre constante qui est le plus petit ascendant commun à  $a$  et  $b$ .

Si, par exemple, l'attribut "compétence" est de type hiérarchique et qu'il est représenté par la taxinomie illustrée sur la figure 4.1. En supposant l'existence des deux descriptions  $d_1$  et  $d_2$  suivantes, tel que :

$d_1=(\text{compétence}=\text{Eaux\_de\_surface})\&(\text{région}=\text{Rhône\_alpes}),$

$d_2=(\text{compétence}=\text{Eaux\_souterraines})\&(\text{région}=\text{Rhône\_alpes}),$

nous obtenons alors le généralisé suivant :

$\mathfrak{G}(d_1, d_2) = d_g = (\text{compétence}=\text{Eaux\_douces})\&(\text{région}=\text{Rhône\_alpes})$

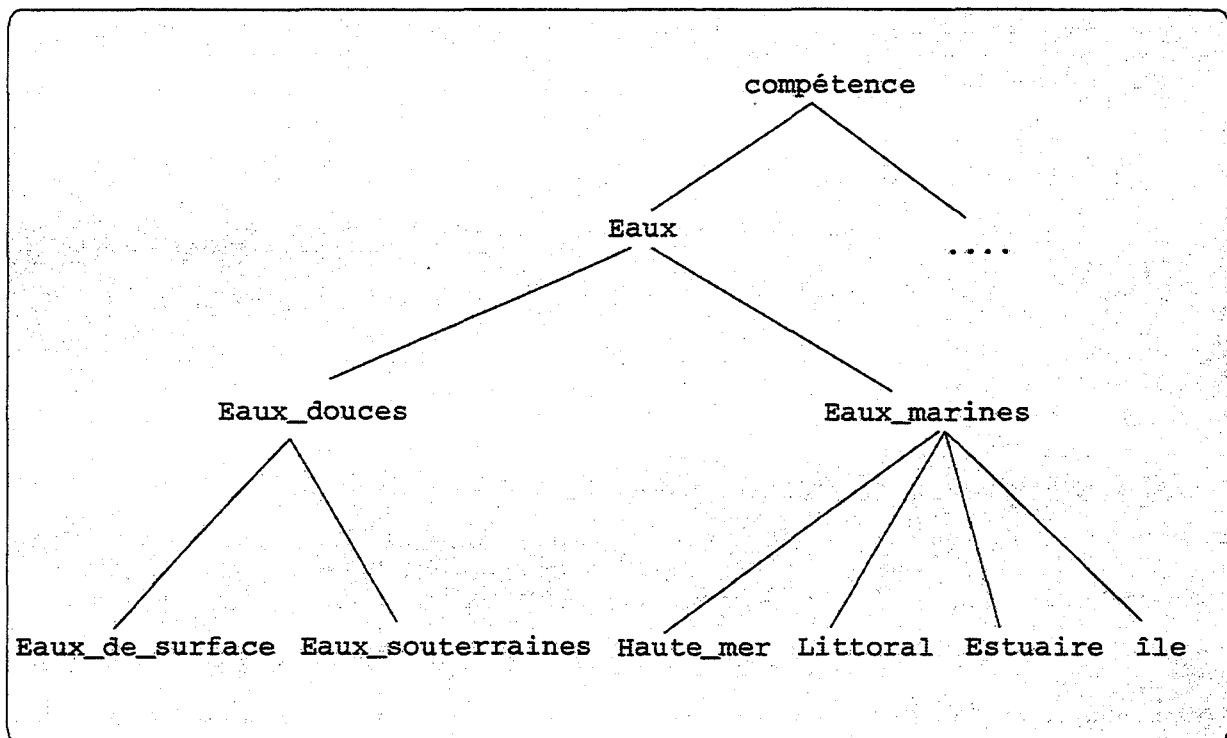


Figure 4.1 : Exemple de taxinomie d'un attribut de type hiérarchique

• pour les attributs de type **linéaire**, nous choisissons la fonction de **fermeture d'un intervalle** [Vere 75], [Vere 80], [Michalski 83a], [Bollinger 86]. Cette fonction de généralisation donne comme résultat le plus petit domaine de valeurs contenant les deux éléments à généraliser.

$\mathfrak{G}(d=a, d=b) = d \in [\inf(a, b), \sup(a, b)]$ , où a et b sont deux constantes.

Si, par exemple, l'attribut "date" est de type linéaire et si nous avons les deux descriptions suivantes :

$d_1=(date=1980)\&(région=Rhône\_alpes)$ ,  $d_2=(date=1990)\&(région=Rhône\_alpes)$ ,

nous obtenons alors le généralisé suivant :

$\mathfrak{G}(d_1, d_2) = d_g = (date \in [1980, 1990])\&(région=Rhône\_alpes)$

**Remarque :**

Dans tous les cas, la fonction de généralisation appliquée à deux éléments qui ont la même valeur, rend comme résultat cette valeur.

**4.2.1.4 Conclusion**

Nous avons vu que la spécification de l'environnement des observations est un processus incrémental. Il est donc possible d'enrichir le langage de description des observations pour pouvoir prendre en compte de nouvelles connaissances du domaine sans modifier les connaissances apprises. En effet, la nature des connaissances de notre domaine (distribution des connaissances à travers plusieurs disciplines, évolutivité des connaissances) exige une possibilité d'améliorer cet environnement incrémentalement. Il est clair que même en s'appuyant sur une méthodologie d'acquisition de connaissances, les experts du domaine ne peuvent pas déterminer au début du projet la totalité des éléments constituant le langage de description des exemples d'apprentissage. En effet, à partir des nouveaux problèmes traités par les experts, vont surgir de nouveaux descripteurs qui nécessitent d'être intégrés dans l'environnement des observations.

**4.2.2 Les exemples d'apprentissage**

Une fois que l'environnement des observations est spécifié (les attributs du langage, les domaines de définition, les fonctions de généralisation), les experts peuvent commencer à exprimer les observations qui vont constituer l'ensemble des exemples d'apprentissage. Les attributs définis sont utilisés pour représenter les descripteurs des observations.

Supposons, par exemple, que pour représenter des observation dans notre domaine, nous ayons construit le langage de description constitué de trois attributs :

identif : pour identifier l'expert ayant fourni l'exemple d'apprentissage décrivant une intervention,

région : pour indiquer la zone géographique dans laquelle est effectuée cette intervention,

compétence : pour indiquer les compétences mises en oeuvre au cours de cette intervention.

Soit un ensemble d'observation  $\{e_1, e_2, e_3, e_4, e_5\}$  dont les descriptions correspondantes sont  $\{d(e_1), d(e_2), d(e_3), d(e_4), d(e_5)\}$  tel que :

$d(e_1) : \{(identif = azili) \& (région = maroc) \& (compétence = cours\_d'eau) \& (compétence = eau\_de\_surface) \& (compétence = ressources\_en\_eau)\}$

$d(e_2) : \{(identif = berrada) \& (région = maroc) \& (compétence = eau\_de\_surface) \& (compétence = ressources\_en\_eau)\}$

$d(e_3) : \{(identif = salem) \& (région = tunisie) \& (compétence = réhabilitation) \& (compétence = infrastructure)\}$

$d(e_4) : \{(identif = kadiri) \& (région = tunisie) \& (compétence = audit) \& (compétence = infrastructure)\}$

$d(e_5) : \{(identif = couloum) \& (région = tunisie) \& (compétence = eau\_de\_surface) \& (compétence = ressources\_en\_eau)\}$

Nous pouvons tout d'abord constater que les deux attributs "identif" et "région" n'apparaissent qu'une seule fois dans la description d'un exemple  $d(e_i)_{i=1,5}$ , alors que l'attribut "compétence" apparaît plusieurs fois. En effet, comme chaque exemple d'apprentissage porte sur une intervention effectuée par un expert du domaine, l'attribut "identif" qui est associé à l'identification de cet expert n'apparaît donc qu'une seule fois seulement dans la description d'un exemple. Ainsi, si plusieurs experts sont intervenus au même titre sur un cas de crise, chacun d'eux doit fournir une observation concernant cette intervention. Il en est de même pour l'attribut "région" qui est associé à la zone géographique localisant le lieu de l'intervention. Par contre l'attribut "compétence" qui est associé aux compétences mises en oeuvre dans cette intervention apparaît plusieurs fois lorsque plusieurs compétences sont nécessaires pour résoudre des problèmes engendrés par une situation de crise. A partir de cet

ensemble d'observations  $\{e_1, e_2, e_3, e_4, e_5\}$ , le système d'apprentissage construit le RDP illustré sur la figure 4.2.

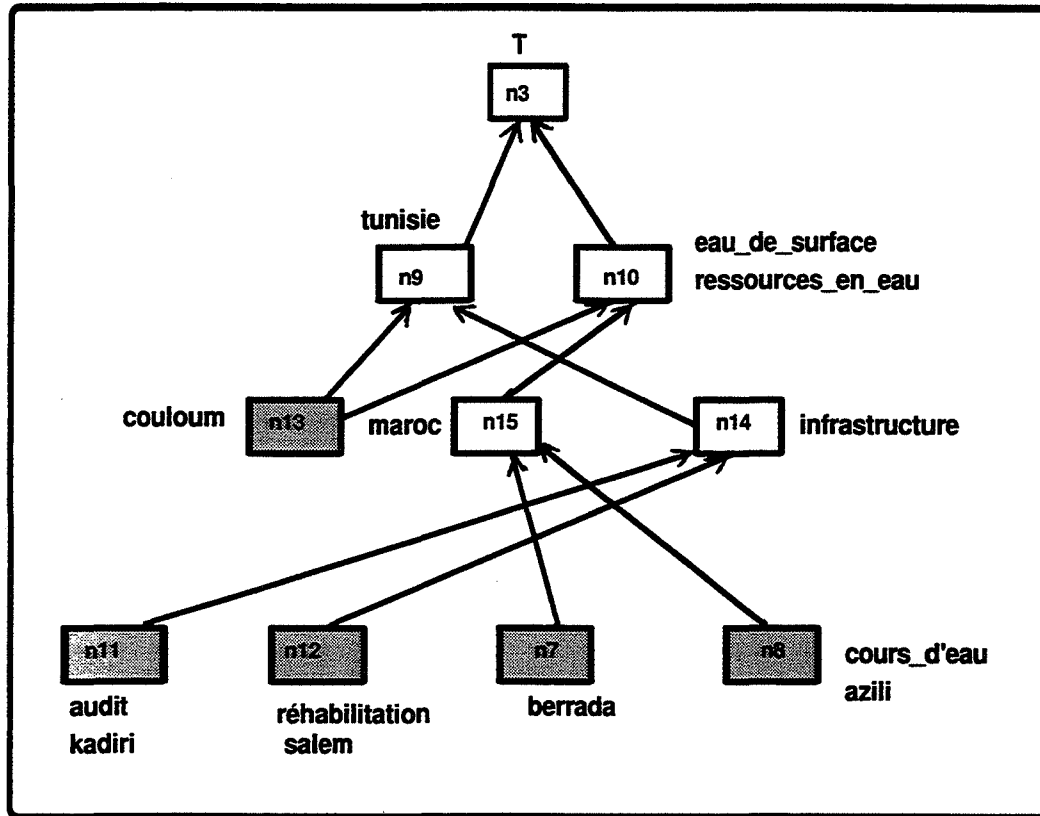


Figure 4.2 : Graphe de connaissances correspondant aux observations  $\{e_1, e_2, e_3, e_4, e_5\}$

Les cinq observations  $\{e_1, e_2, e_3, e_4, e_5\}$  peuvent être reconstruites à partir des noeuds  $n_{13}$ ,  $n_{11}$ ,  $n_{12}$ ,  $n_7$  et  $n_8$  qui sont caractérisés sur le graphe par un "fond plein". Chacun de ces noeuds contient les éléments de connaissances qui ne rentrent que dans la description de l'observation qu'il représente. Les éléments qui appartiennent à la description de plusieurs observations sont mis à un niveau plus général pour pouvoir être partagés par les différentes observations et ainsi minimiser le nombre de noeuds du graphe.

L'observation ( $e_5$ ), par exemple, représentée par le noeud  $n_{13}$  est constituée des éléments correspondant aux trois noeuds  $n_{13}$ ,  $n_9$  et  $n_{10}$ . Nous voyons donc que pour trouver la



description de cet exemple, il faut partir du noeud  $n_{13}$  et parcourir tous les autres noeuds  $n_i$  pour lesquels les éléments de la classe d'Equivalences Perçues correspondant au noeud  $n_{13}$  sont en relation de Dépendance Perçue avec la classe d'Equivalences Perçues correspondant au noeud  $n_i$ .

Le descripteur (région = maroc) correspondant au  $n_{15}$  rentre dans la description des deux exemples d'apprentissage  $e_1, e_2$ . Ces deux exemples d'apprentissage contiennent aussi dans leur description les descripteurs (compétence = eau\_de\_surface) et (compétence = ressources\_en\_eau) correspondant au noeud  $n_{10}$ . Mais, comme ces descripteurs rentrent dans la description de l'exemple d'apprentissage  $e_5$ , le noeud  $n_{10}$  est situé à un niveau supérieur de celui du noeud  $n_{15}$ .

Avant de décrire le processus qui permet d'intégrer de nouvelles observations dans le RDP, nous allons définir d'abord quelques notions sur le graphe de connaissances.

### 4.3 ELEMENTS DU RDP

Nous allons, dans ce qui suit, définir les différents éléments traités dans un RDP.

#### *Définitions :*

- **Formule atomique** : nous appelons formule atomique toute description constituée d'un seul descripteur.

**Exemple** : (compétence = eau\_de\_surface).

- **Formule** : nous appelons formule ou formule complexe la conjonction de  $n$  descripteurs (avec  $n \geq 2$ ) dans lequel un attribut quelconque peut apparaître plus d'une fois mais avec des valeurs différentes.

**Exemple** : (compétence = audit)&(compétence = infrastructure)&(région = tunisie).

- **Langage de description** : le langage de description  $D$  est l'ensemble de toutes les formules que nous pouvons produire à partir d'un ensemble d'attributs qui sont définis.

- **Ensemble d'apprentissage** : l'ensemble d'apprentissage  $E_a$  est constitué de l'ensemble des exemples  $e_i$  décrits par  $d(e_i)$ .

- **Noeuds et arcs du RDP** : Rappelons que le Réseau de Dépendances Perçues est défini par un couple  $(V_{RDP}, \varepsilon_{RDP})$  où  $V_{RDP}$  est l'ensemble des noeuds du RDP, et  $\varepsilon_{RDP}$  est l'ensemble des arcs orientés (paires de noeuds) tel qu'une paire de noeuds  $(n_i, n_j)$  de  $V_{RDP}$  appartient à  $\varepsilon_{RDP}$  si et seulement si les éléments de la classe d'Equivalences Perçues correspondant au noeud  $n_i$  sont en relation de Dépendances Perçues avec les éléments de la classe d'Equivalence Perçue correspondant au noeud  $n_j$ . En d'autres termes, nous pouvons dire que la description de tous les exemples associés au noeud  $n_j$  contient la description des exemples associés au noeud  $n_i$ . L'existence d'un arc  $u = (n_i, n_j)$  implique donc que l'ensemble des formules associées au noeud  $n_i$  et l'ensemble des formules associées au noeud  $n_j$  appartiennent au moins à la description d'un même exemple de l'ensemble d'apprentissage  $E_a$ .

- **Chemin entre deux noeuds** : un chemin entre deux noeuds est une suite d'arcs  $u = (u_1, u_2, \dots, u_n)$ , dans lequel l'extrémité terminale de l'arc  $u_i$  coïncide avec l'extrémité initiale de l'arc  $u_{i+1}$ .

L'existence d'un chemin entre deux noeuds  $n_i$  et  $n_j$  signifie que les éléments de la classe d'Equivalences Perçues correspondant au noeud  $n_i$  sont en Dépendance Perçue avec la classe d'Equivalences Perçues correspondant au noeud  $n_j$ .

- **Étiquette d'un noeud** : l'étiquette d'un noeud est l'ensemble des formules atomiques correspondant à ce noeud. L'étiquette d'un noeud constitue donc une classe d'Equivalences Perçues.

**Exemple** :

étiquette( $n_{14}$ ) = {compétence=infrastructure}

étiquette( $n_{10}$ ) = {compétence = eau\_de\_surface, compétence = ressources\_en\_eau}

L'étiquette d'un noeud peut être un ensemble vide, une formule atomique ou une formule. Nous verrons par la suite la signification d'un noeud ayant une étiquette vide.

- **Couverture d'un noeud** : la couverture d'un noeud est constituée de l'ensemble des exemples d'apprentissage  $\mathbb{E}_s$  dont la description contient la formule appartenant à l'étiquette de ce noeud.

**Exemple :**

$$\text{couv}(n_{10}) = \{e_1, e_2, e_5\}$$

$$\text{couv}(n_{15}) = \{e_1, e_2\}$$

$$\text{couv}(n_{14}) = \{e_3, e_4\}$$

Nous pouvons constater que l'existence d'un arc entre les noeuds  $n_i$  et  $n_j$ , c'est-à-dire si  $(n_i, n_j) \in \mathcal{E}_{\text{RDP}}$ , implique que :  $\text{couv}(n_i) \subseteq \text{couv}(n_j)$ . Par conséquent, l'existence d'un chemin entre les noeuds  $n_k$  et  $n_r$  implique que  $\text{couverture}(n_k) \subseteq \text{couverture}(n_r)$ .

- **Noeuds antécédents** : l'ensemble des noeuds antécédents d'un noeud  $n$  est l'ensemble des noeuds  $n_i$  pour lesquels il existe un chemin de  $n_i$  à  $n$ .

**Exemple :**  $\text{ant}(n_{10}) = \{n_{13}, n_{15}, n_7, n_8\}$

Si un noeud du RDP ne possède aucun noeud antécédent, il est dit noeud terminal.

- **Noeuds successeurs** : l'ensemble des noeuds successeurs d'un noeud  $n$  est l'ensemble des noeuds  $n_i$  pour lesquels il existe un chemin de  $n$  à  $n_i$ .

**Exemple :**  $\text{suc}(n_7) = \{n_{15}, n_{10}, n_3\}$

Si un noeud ne possède aucun noeud successeur, il est dit racine du graphe.

- **Noeuds pères** : l'ensemble des noeuds pères d'un noeud  $n$  est l'ensemble des noeuds antécédents de  $n$  les moins spécifiques (les antécédents directs de  $n$ ). En d'autres termes, le noeud  $n_i$  est dit "noeud père" du noeud  $n$  s'il existe un arc ayant son extrémité initiale en  $n_i$  et son extrémité terminale en  $n$ .

**Exemple** :  $\text{pères}(n_{10}) = \{n_{15}, n_{13}\}$

- **Noeuds fils** : l'ensemble des noeuds fils d'un noeud  $n$  est l'ensemble des noeuds successeurs de  $n$  les moins généraux (les successeurs directs de  $n$ ). En d'autres termes, le noeud  $n_i$  est dit "noeud fils" du noeud  $n$  s'il existe un arc ayant son extrémité initiale en  $n$  et son extrémité terminale en  $n_i$ .

**Exemple** :  $\text{fils}(n_7) = \{n_{15}\}$

Après avoir défini l'environnement d'apprentissage pour notre système ainsi que les différentes notions rentrant dans un RDP, nous allons maintenant décrire le processus d'intégration incrémentale des exemples dans le graphe de connaissances.

## 4.4 ENTRETIEN INCREMENTAL DU GRAPHE DE CONNAISSANCES

Le caractère évolutif et multidisciplinaire des connaissances du domaine de l'eau nécessite un traitement incrémental de ces connaissances pour les intégrer dans le système EXPEAU. Nous présenterons, dans ce qui suit, le schéma global de construction de la BC du système et nous donnerons l'algorithme qui permet d'intégrer incrémentalement des observations fournies par les experts du domaine.

### 4.4.1 Schéma de construction de la Base de Connaissances

Les étapes de construction de la Base de Connaissances du système sont illustrées par le schéma SADT de la figure 4.3. Nous constatons clairement qu'une phase d'acquisition de connaissances du domaine est nécessaire pour mener à bien le processus d'apprentissage. Une fois que l'environnement des observations est constitué, les exemples d'intervention peuvent être pris en compte séquentiellement. Nous verrons dans le cinquième chapitre comment la phase d'exploitation du RDP construit à ce niveau, utilise aussi les connaissances spécifiées dans l'environnement des observations. Les stratégies de résolution qui constituent la méta-

connaissance sur le domaine sont aussi utilisées dans la phase d'exploitation du système EXPEAU.

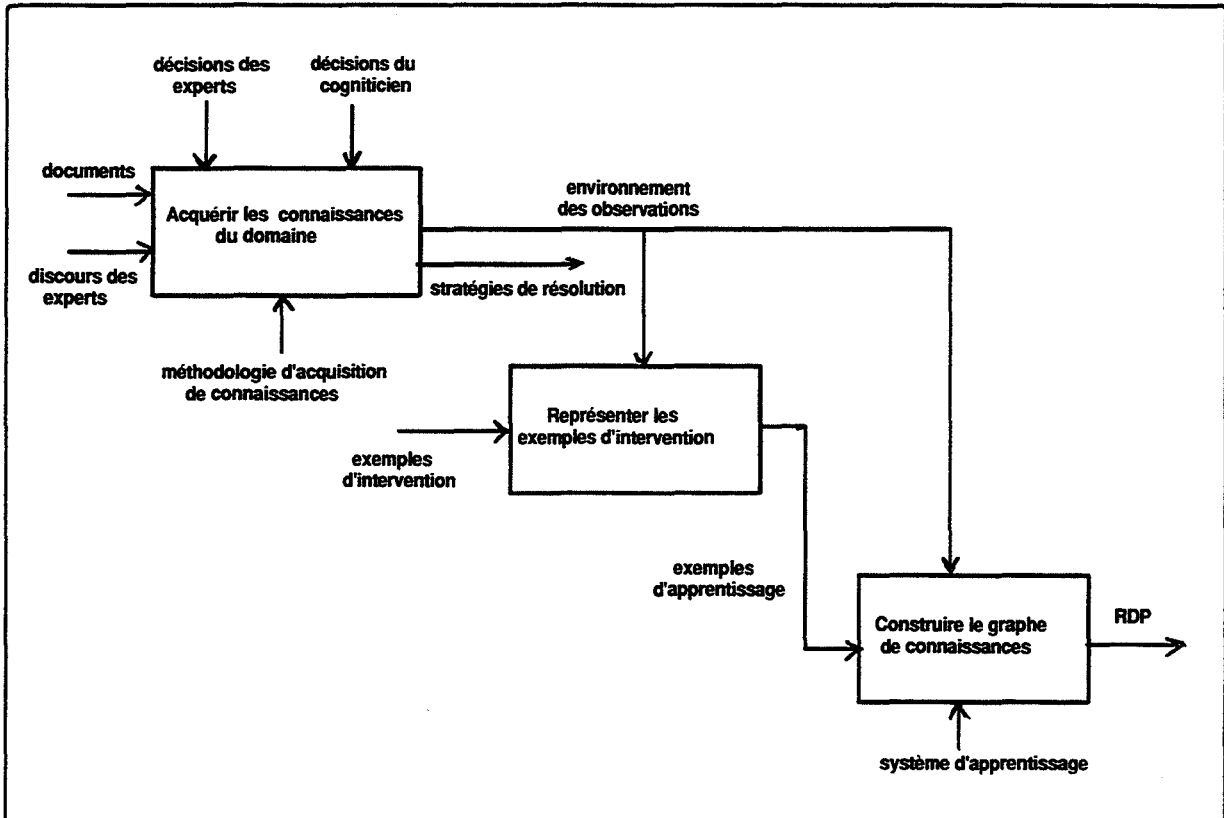


Figure 4.3 : Schéma SADT de la construction de la Base de Connaissances du système EXPEAU

Les connaissances du domaine représentent les connaissances générales décrivant le domaine d'application. Elles permettent aussi de spécifier le langage de description des exemples d'intervention sur des cas de crise fournis par les experts du domaine. Ce langage de description définit la syntaxe à adopter pour représenter des exemples d'apprentissage et définit l'ensemble des domaines de définition des éléments de ce langage. En s'appuyant sur ce langage

de description, nous produisons donc des observations compréhensibles par le système d'apprentissage.

#### 4.4.2 Intégration incrémentale des observations

Dans EXPEAU, la Base de Connaissances est construite incrémentalement au fur et à mesure que les cas d'intervention sont fournis par les experts. Cette construction incrémentale du graphe consiste à intégrer séquentiellement de nouveaux exemples d'apprentissage à la connaissance déjà apprise.

Nous désignerons par  $\mathbf{E}_a$  l'ensemble des cas d'observation appelé ensemble d'apprentissage, par  $\mathbf{G}$  la fonction de généralisation, par  $f_n$  l'étiquette du noeud  $n$  (ensemble des formules atomiques contenues dans le noeud  $n$ ), et par  $\mathbf{Gr}$  le graphe représentant la connaissance apprise. Ce graphe subit des changements (ajout de noeuds, ajout d'arcs, modification de noeuds, modification d'arcs, suppression de noeuds, suppression d'arcs) lorsque l'ensemble d'apprentissage  $\mathbf{E}_a$  change (ajout ou retrait de cas d'observation).

La racine du graphe constitue le noeud "Tautologie", il présente la caractéristique suivante :  $\text{couv}(\mathbf{T}) = \mathbf{E}_a$  et  $f_{\mathbf{T}}$  correspond aux formules communes à tous les exemples.

Supposons qu'un nouvel exemple  $e_N$  soit ajouté à l'ensemble d'apprentissage  $\mathbf{E}_a$ , initialement cet exemple est associé à un nouveau noeud  $N$  caractérisé par sa couverture  $\text{couv}(N) = \{e_N\}$  et l'ensemble des formules atomiques contenues dans ce noeud  $f_N = d(e_N)$ .

La procédure de construction du graphe [Sainte Marie 89], [Beaune 92a] visite les noeuds déjà existants en adoptant la stratégie de parcours en "largeur d'abord". Ce qui revient à commencer par le noeud le plus général  $\mathbf{T}$ .

#### Début

Créer et initialiser le noeud  $N$  correspondant à  $e_N$

**Pour** chaque noeud  $n$  visité lors du parcours du graphe **faire**

$F \leftarrow \emptyset$

**Pour** tout noeud  $x$  tel que il existe un chemin de  $N$  à  $x$  **faire**

$F \leftarrow F \cup f_x$

**fin faire**

**si**  $((G(f_N, f_n) - F) \neq \emptyset$  et  $(G(f_N, f_n) \neq f_n)$  **alors**

créer un nouveau noeud  $n'$

$V_{RDP} \leftarrow V_{RDP} \cup n'$

$f_{n'} \leftarrow f_n - G(f_N, f_n)$

$f_n \leftarrow G(f_N, f_n)$

**Pour** tout arc  $(n_i, n) \in \varepsilon_{RDP}$  **faire**

supprimer  $(n_i, n)$  de  $\varepsilon_{RDP}$  (on supprime les pères du noeud  $n$ )

ajouter  $(n_i, n')$  à (ils deviennent ceux de  $n'$ )

**fin faire**

ajouter  $(n', n)$  à  $\varepsilon_{RDP}$  ( $n'$  devient l'unique père de  $n$ )

**fin si**

**si**  $(G(f_N, f_n) = f_n)$  **alors**

$f_N \leftarrow f_N - f_n$

**modifier\_** $\varepsilon_{RDP}(N, n)$

**fin si**

**fin faire****si**  $((f_N = \emptyset) \text{ et } (\exists m \in V_{RDP} / (m \neq N), (f_n = \emptyset) \text{ et}$  $(\forall n_i \in V_{RDP}, (m, n_i) \in V_{RDP} \Leftrightarrow (N, n_i) \in V_{RDP})))$  **alors**

retirer le noeud N du RDP (il est identique à un exemple déjà fourni)

**fin si****fin****Procédure modifier\_εRDP (N, n)****début****Pour** tout  $s \in V_{RDP} / (n, s) \in \epsilon_{RDP}$  et il n'existe pas de chemin de N à s **faire**supprimer  $(n, s)$  de  $\epsilon_{RDP}$  (car en reliant N à n, nous créons un lien de N à s)**Pour** tout  $a \in V_{RDP} / (a, n) \in \epsilon_{RDP}$  et il n'existe pas de chemin de a à s **faire**ajouter  $(a, s)$  à  $\epsilon_{RDP}$  (car le lien de a à s est perdu en supprimant le lien de n à s)**fin faire****Pour** tout  $x \in V_{RDP} /$  il n'existe pas de chemin de s ou de n à x et il existe unchemin de N à x **faire**ajouter  $(n, x)$  à  $\epsilon_{RDP}$ **fin faire****fin faire****Pour** tout  $s \in V_{RDP} / (N, s) \in \epsilon_{RDP}$  et il existe un chemin de n à s **faire**



supprimer  $(N, s)$  de  $\varepsilon_{RDP}$  (car en reliant  $N$  à  $n$ , nous créons un double lien de  $N$  à  $s$ )

**fin faire**

ajouter  $(N, n)$  à  $\varepsilon_{RDP}$

**fin**

L'intégration d'un nouvel exemple consiste à modifier incrémentalement le RDP existant de manière à prendre en compte les descripteurs de cet exemple. De même, la procédure qui permet de retirer une observation modifie aussi incrémentalement le graphe en retirant les connaissances correspondant à tous les exemples d'apprentissage qui satisfont la condition représentée par l'observation.

Dans le paragraphe 4.2.2, nous avons illustré un exemple de graphe (figure 4.2) correspondant à l'ensemble d'observation  $\{e_1, e_2, e_3, e_4, e_5\}$  dont les descriptions correspondantes sont  $\{d(e_1), d(e_2), d(e_3), d(e_4), d(e_5)\}$ . Essayons maintenant d'intégrer une autre observation ( $e_6$ ) dont la description est  $d(e_6)$ . Nous aurons alors un ensemble d'apprentissage constitué par les exemples  $\{e_1, e_2, e_3, e_4, e_5, e_6\}$  dont les descriptions sont :

$d(e_1) : \{(identif = azili) \& (région = maroc) \& (compétence = cours\_d'eau) \& (compétence = eau\_de\_surface) \& (compétence = ressources\_en\_eau)\}$

$d(e_2) : \{(identif = berrada) \& (région = maroc) \& (compétence = eau\_de\_surface) \& (compétence = ressources\_en\_eau)\}$

$d(e_3) : \{(identif = salem) \& (région = tunisie) \& (compétence = réhabilitation) \& (compétence = infrastructures)\}$

$d(e_4) : \{(identif = kadiri) \& (région = tunisie) \& (compétence = audit) \& (compétence = infrastructure)\}$

$d(e_5) : \{(identif = couloum) \& (région = tunisie) \& (compétence = eau\_de\_surface) \& (compétence = ressources\_en\_eau)\}$

$d(e_6) : \{(identif = azili) \& (région = maroc) \& (compétence = audit) \& (compétence = infrastructure)\}$

Nous obtenons alors le graphe de connaissances ci-dessous.

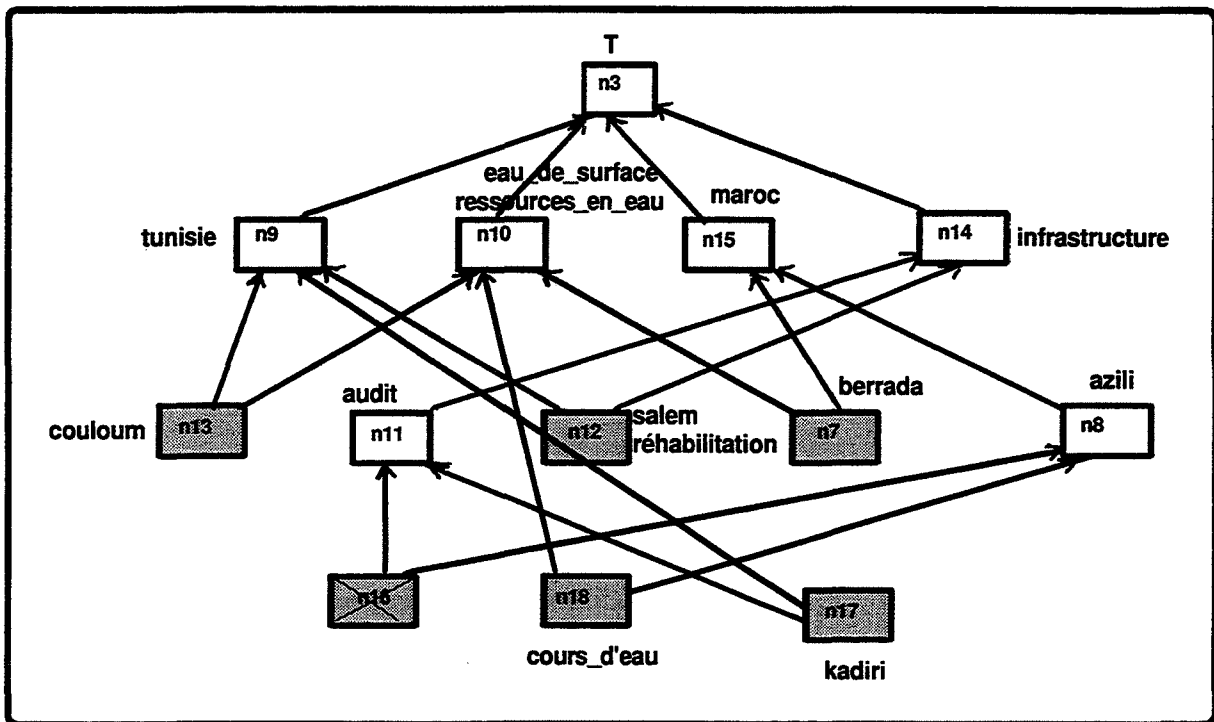


Figure 4.4 : Graphe de connaissances correspondant aux exemples  $\{e_1, e_2, e_3, e_4, e_5, e_6\}$

Nous pouvons constater que le noeud "n<sub>16</sub>" ne contient aucune formule, ce qui revient à dire que son étiquette est vide :  $étiquette(n_{16}) = \emptyset$ . En effet, ce noeud représente l'observation ( $e_6$ ), dont la description est constituée des éléments se trouvant dans les noeuds ( $n_8, n_{15}, n_{11}, n_{14}$ ). Il n'est donc construit que pour pouvoir établir un lien entre les noeuds qui contiennent une formule de la description de l'observation ( $e_6$ ).

Nous constatons aussi que la relation d'Equivalence Perçue entre les deux formules atomiques ( $compétence = cours\_d'eau$ ) et ( $identif = azili$ ) qui existait sur le premier

RDP (figure 4.2) n'est plus valable sur le nouvel RDP obtenu. Cela est dû à l'ajout d'une nouvelle observation dont la description contient la formule atomique (identif=azili) mais qui ne contient pas la formule atomique (compétence=cours\_d'eau). Cette relation d'Equivalence Perçue est donc réduite à une relation de Dépendance Perçue car il demeure que toutes les observations comportant la formule atomique (compétence=cours\_d'eau) dans leur description comportent aussi la formule atomique (identif=azili). Il en est de même pour la relation d'Equivalence Perçue entre les deux formules atomiques (compétence=audit) et (identif=kadiri) qui existe sur le premier RDP (figure 4.2), mais que nous ne retrouvons plus sur le nouveau RDP (figure 4.4).

Si l'ajout de l'observation ( $e_6$ ) a provoqué une transformation dans le RDP précédent et a remis en cause quelques relations existantes entre les éléments de connaissances constituant le RDP, il est possible d'intégrer des exemples qui, eux, ne changent pas la structure du RDP et ne remettent pas forcément en cause les différentes relations de Dépendance et d'Equivalence Perçue existantes. Ainsi si nous voulons ajouter l'observation ( $e_7$ ) dont la description est  $d(e_7)$  à cet ensemble tel que :

$d(e_1)$  : {(identif = azili) & (région = maroc) & (compétence = cours\_d'eau) & (compétence = eau\_de\_surface) & (compétence = ressources\_en\_eau)}

$d(e_2)$  : {(identif = berrada) & (région = maroc) & (compétence = eau\_de\_surface) & (compétence = ressources\_en\_eau)}

$d(e_3)$  : {(identif = salem) & (région = tunisie) & (compétence = réhabilitation) & (compétence = infrastructures)}

$d(e_4)$  : {(identif = kadiri) & (région = tunisie) & (compétence = audit) & (compétence = infrastructure)}

$d(e_5)$  : {(identif = couloum) & (région = tunisie) & (compétence = eau\_de\_surface) & (compétence = ressources\_en\_eau)}

$d(e_6)$  : {(identif = azili) & (région = maroc) & (compétence = audit) & (compétence = infrastructure)}

$d(e_7) : \{(identif = azili) \& (région = maroc) \& (compétence = barrage) \& (compétence = eau\_de\_surface) \& (compétence = ressources\_en\_eau)\}$

Nous constatons que la formule atomique (compétence=barrage) n'apparaît que dans la description de l'observation ( $e_7$ ). Par conséquent, l'ajout de l'observation ( $e_7$ ) va nécessiter la création d'un nouveau noeud.

Le RDP correspondant à cet ensemble d'apprentissage est le suivant :

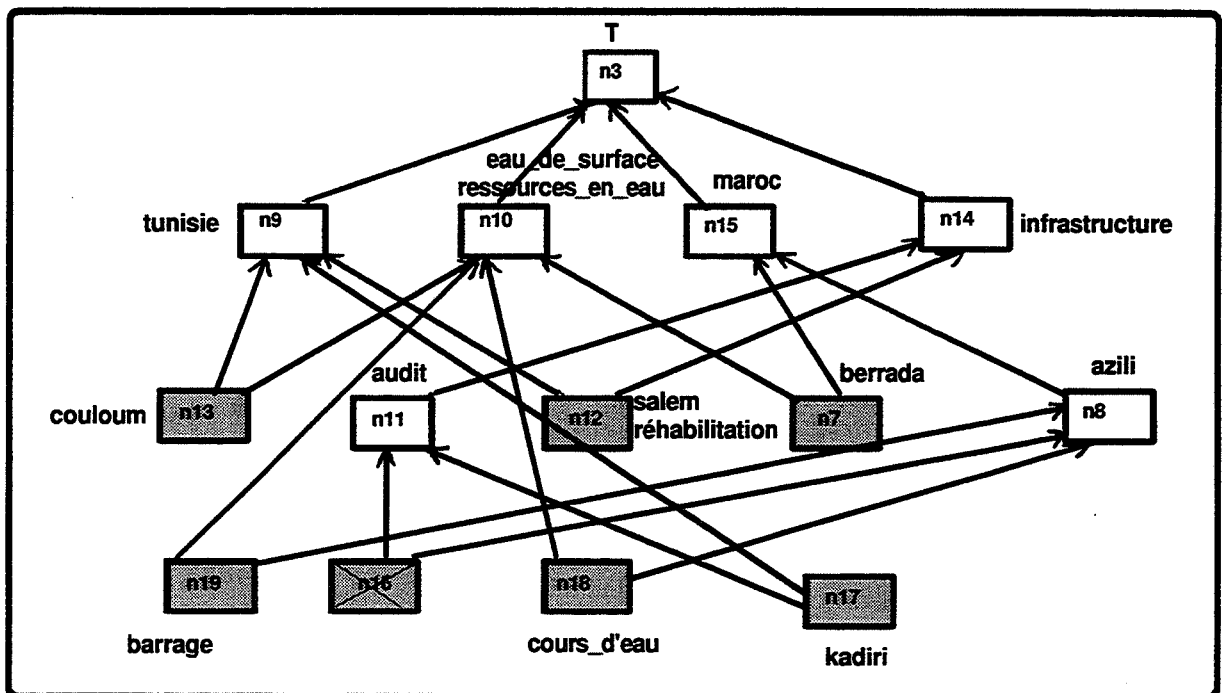


Figure 4.5 : Graphe de connaissances correspondant aux exemples  $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$

L'exemple d'apprentissage  $e_7$  est représenté par les noeuds  $n_{19}$ ,  $n_{10}$ ,  $n_8$ ,  $n_{15}$ . La formule complexe  $\{compétence=eau\_de\_surface, compétence=ressources\_en\_eau\}$  correspondant à l'étiquette du noeud  $n_{10}$  est partagée maintenant par quatre exemples d'apprentissage à savoir :  $\{e_1, e_2, e_5, e_7\}$ . Ainsi nous constatons que les classes d'Equivalences Perçues qui existent dans le RDP constituent des éléments génériques de description de situation de crise. En effet, toute observation qui contient dans sa description le descripteur  $d_i$  de la classe d'Equivalences Perçues  $CE_i$ , contient aussi tous les autres descripteurs de cette classe. Le système permet

donc d'apprendre des connaissances intermédiaires qui vont servir à faciliter la description de situation de crise en utilisant les formules constituants des classes d'Equivalences Perçues.

Dans le paragraphe suivant nous allons détailler les éléments du langage de description qui sont utilisés dans le système EXPEAU.

## 4.5 FORMAT D'UN EXEMPLE D'APPRENTISSAGE DANS EXPEAU

L'étape d'acquisition de connaissances qui s'est déroulée avec les experts du domaine a donné lieu à un langage de description permettant de représenter les cas d'intervention. Pour arriver à avoir un langage aussi complet que possible, les experts ont travaillé sur la description des cas d'intervention qu'ils ont effectués sur des cas réels. Quelques 200 cas d'intervention ont ainsi été traités pour faire ressortir un format de description avec lequel nous pouvons les représenter. Ce langage de description est constitué des éléments suivants : l'identification de l'intervenant, l'organisme auquel appartient l'intervenant, le lieu de l'intervention, la date de l'intervention, la durée de l'intervention, le rôle joué par l'intervenant, la mission de l'intervenant, le domaine de compétence de l'intervention, les agents de l'environnement qui sont concernés par l'intervention.

### 4.5.1 Identification de l'intervenant

nom\_inter : nom de l'intervenant : nominal.

prenom\_inter : prénom de l'intervenant : nominal.

Ce prédicat sert à identifier l'expert correspondant à une observation qui est fournie au système. Dans le but de faire associer chaque intervention à un seul intervenant, toute observation doit nécessairement inclure dans sa description une et une seule fois ce prédicat. Si un expert  $X_i$  donné du domaine est intervenu plusieurs fois sur des cas de crise, la formule (identif : (nom\_inter =  $X_i$ , prenom\_inter =  $Y_i$ )) apparaîtra dans autant d'observations.

## 4.5.2 Organisme

organisme\_inter : organisme dans lequel travail un expert : nominal.

Ce prédicat peut prendre les valeurs suivantes : {société, entreprise, ministère, administration, bureau d'étude, collectivité locale}.

Il permet d'indiquer à quel organisme appartient l'expert qui a fourni une observation. A l'inverse du premier prédicat (Identification de l'intervenant), le prédicat "Organisme" n'est pas obligatoire dans la description d'une observation. Il peut aussi apparaître plusieurs fois et avec des valeurs différentes dans une même observation dans le cas où l'expert exerce dans plusieurs organismes.

## 4.5.3 Lieu de l'intervention

lieu\_inter : lieu de l'intervention : hiérarchique.

Ce prédicat sert à indiquer la zone géographique dans laquelle est effectuée l'intervention en question. Comme le prédicat "Organisme", le prédicat "Lieu de l'intervention" n'est pas obligatoire, mais à l'inverse du premier, il ne peut apparaître au plus qu'une seule fois dans la description d'une observation. Il permettra lors de l'exploitation des connaissances de :

- s'intéresser aux experts qui sont intervenus sur des cas de crise survenus dans une zone géographique donnée,
- déduire les types de problèmes spécifiques aux zones géographiques,
- déduire les régions qui sont les plus touchées par des catastrophes,
- etc.

L'argument de ce prédicat est défini sous forme d'une structure. La taxinomie qui représente les zones géographiques d'intervention est illustrée sur la figure 4.6.

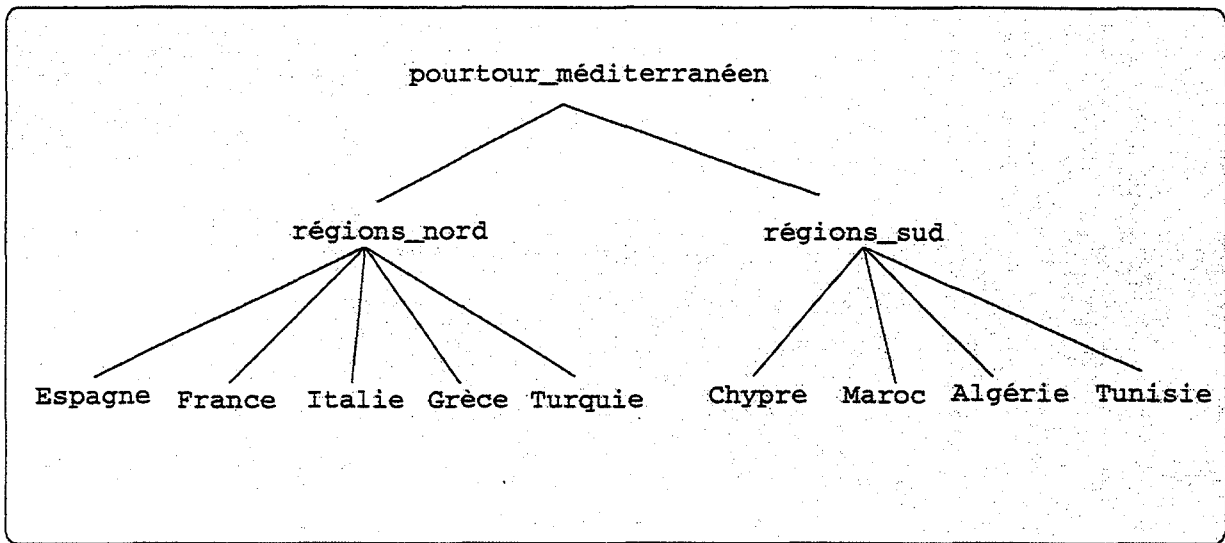


Figure 4.6 : Taxinomie des régions

#### 4.5.4 Date de l'intervention

date\_an\_inter : année de l'intervention : linéaire.

date\_mois\_inter : mois de l'intervention : hiérarchique.

La date de l'intervention permet de situer dans le temps (quelle saison ?, quelle année ?) une intervention effectuée par un expert du domaine. Ce prédicat ne peut apparaître qu'une seule fois au plus dans la description d'une observation donnée. L'année de l'intervention qui est de type "linéaire", est donnée sous forme d'une valeur numérique. Le mois de l'intervention qui est de type "structure" (figure 4.7) permet de classer les mois de l'année en "saison sèche" ou "saison pluvieuse". Pour une intervention qui s'étale sur plusieurs mois, nous indiquons uniquement le mois dans lequel a commencé l'intervention ; le prédicat "durée de l'intervention" que nous allons décrire par la suite, permettra de déduire les autres mois.

Ce prédicat permettra lors de l'exploitation des connaissances de :

- déduire les périodes pendant lesquelles surviennent certains types de problèmes,
- déduire les experts qui interviennent à certaines périodes,
- etc.

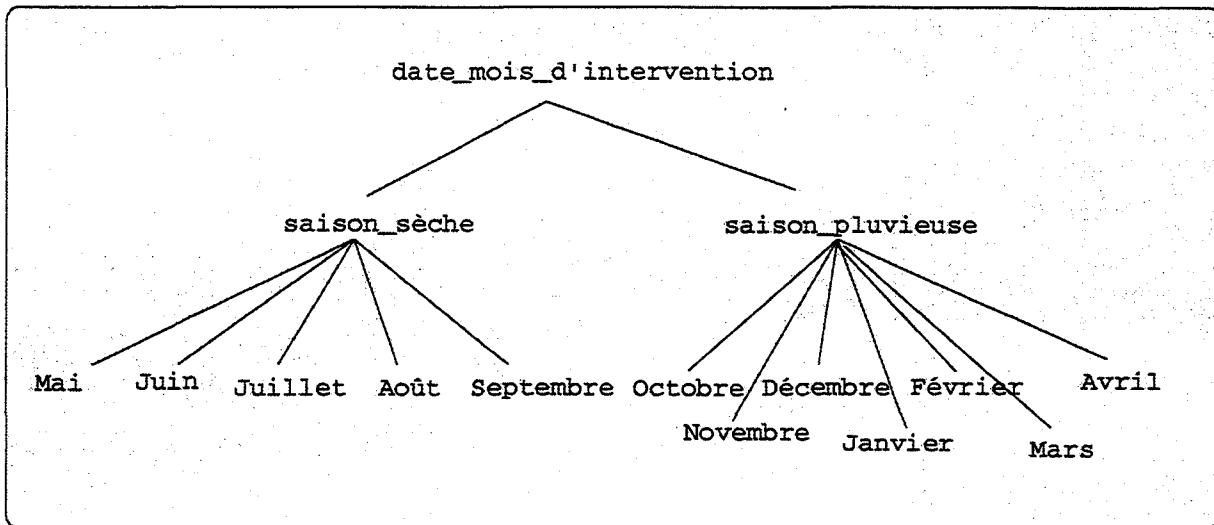


Figure 4.7 : Taxinomie des dates d'intervention

#### 4.5.5 Durée de l'intervention

`duree_inter` : durée de l'intervention : linéaire.

Ce prédicat prend ses valeurs sur l'ensemble suivant : {ponctuelle, courte, moyenne, longue}. Il permet d'indiquer le temps passé pour traiter un cas de crise et ainsi de :

- déduire les durées de travail qu'exigent les différents problèmes survenus dans le domaine de l'eau,
- quantifier l'expérience acquise par les experts,
- etc.

Afin de permettre aux différents experts d'utiliser la même précision sur l'échelle des valeurs de type symbolique {ponctuelle, courte, moyenne, longue}, nous avons déterminé un intervalle de valeurs numériques qui lui est associé. Ainsi, à partir de valeurs numériques affectées par l'expert à ce prédicat, le système effectue les transformations suivantes :

si (`duree_inter` > 6mois) alors (`duree_inter` = longue),

si ( $3 \text{ mois} \leq \text{duree\_inter} \leq 6 \text{ mois}$ ) alors (`duree_inter` = moyenne),



si  $(0.5 \text{ mois} \leq \text{duree\_inter} < 3 \text{ mois})$  alors ( $\text{duree\_inter} = \text{courte}$ ),

si ( $\text{duree\_inter} < 0.5 \text{ mois}$ ) alors ( $\text{duree\_inter} = \text{ponctuelle}$ ),

Puisqu'une intervention quelconque sur un cas de crise s'effectue pendant un temps déterminé, ce prédicat ne doit apparaître qu'une seule fois au plus dans la description d'une observation donnée. Toutefois, il doit apparaître dans toutes les descriptions d'observations qui contiennent le prédicat "Date de l'intervention". Cette contrainte permet, en effet, d'éviter des descriptions dans lesquelles la date de début d'une intervention est précisée, sans pour cela pouvoir déterminer la date de fin de cette intervention.

#### **4.5.6 Rôle joué par l'intervenant**

$\text{role\_inter}$  : rôle de l'intervenant pendant cette intervention : type nominal.

Ce prédicat peut prendre les valeurs suivantes : { $\text{maître\_d'oeuvre}$ ,  $\text{chef\_de\_projet}$ ,  $\text{directeur}$ ,  $\text{conducteur\_des\_etudes}$ ,  $\text{coordinateur\_technique}$ }.

Il sert à indiquer le niveau d'intervention de l'expert car pour un cas de crise donné, plusieurs personnes peuvent intervenir mais chacune avec une responsabilité définie. Ce prédicat n'est pas obligatoire dans la description d'une observation, mais il peut aussi apparaître plusieurs fois si c'est nécessaire pour indiquer les différents rôles tenus par l'expert au cours de cette intervention.

#### **4.5.7 Mission de l'intervenant**

$\text{mission\_inter}$  : mission de l'intervenant : type nominal.

Ce prédicat prend ses valeurs dans l'ensemble suivant : { $\text{élaboration}$ ,  $\text{suivi}$ ,  $\text{étude}$ ,  $\text{réalisation}$ ,  $\text{expertise}$ ,  $\text{participation}$ ,  $\text{création}$ ,  $\text{enquête}$ ,  $\text{maintenance}$ ,  $\text{réhabilitation}$ ,  $\text{diagnostic}$ ,  $\text{investigation}$ ,  $\text{renovation}$ ,  $\text{mise en valeur}$ ,  $\text{gestion}$ ,  $\text{assistance technique}$ }.

Il permet d'indiquer le type d'opération effectué par l'expert au cours de cette intervention. Tout comme le prédicat précédent, ce prédicat n'est pas obligatoire et peut apparaître plusieurs fois dans la description d'une observation et ceci du fait qu'un expert peut effectuer plusieurs opérations au cours d'une même intervention.

#### **4.5.8 Domaine de compétence de l'intervention**

`competence_inter` : les compétences définissant l'intervention : hiérarchique

Ce prédicat est le plus important pour la description d'une observation. Il permet de spécifier les compétences mises en oeuvre par l'expert au cours d'une intervention sur un cas de crise. Puisque toute intervention sur un cas de crise fait nécessairement référence à au moins une compétence, ce prédicat apparaît donc au moins une fois dans la description d'une observation.

La figure 4.8 illustre la classification des compétences du domaine de l'eau sous forme d'une taxinomie.

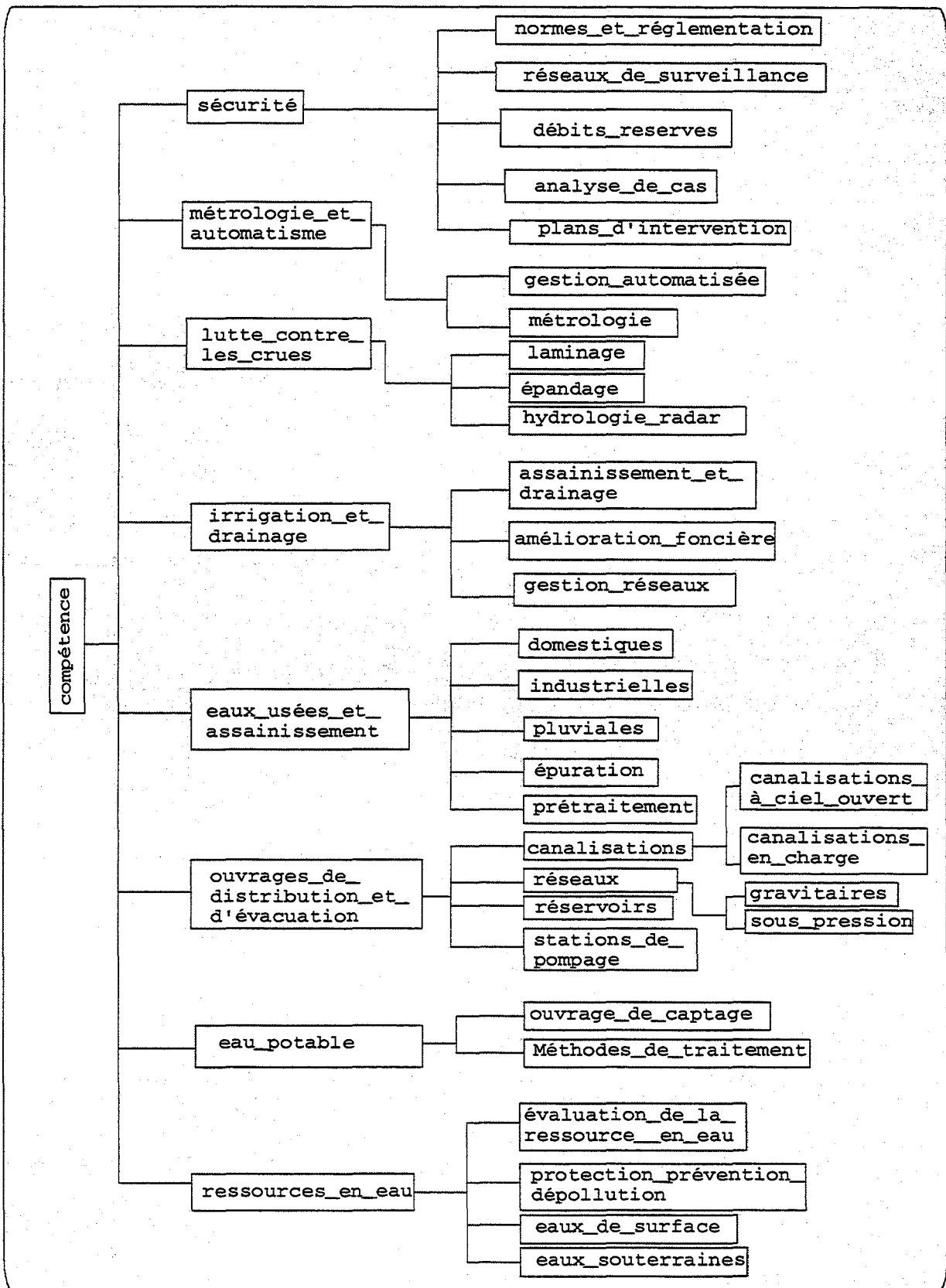


Figure 4.8 : Une partie de la taxinomie des compétences

Ce prédicat permet, en phase d'exploitation des connaissances, de déduire :

- les compétences que possèdent les experts du domaine,
- les compétences nécessaires dans une zone géographique donnée,
- les domaines de compétences qui ne sont pas couverts par les experts,
- etc.

#### **4.5.9 Agents de l'environnement concernés par l'intervention**

`agents_inter` : les agents de l'environnement qui sont touchés par la situation de crise : structure.

Ce prédicat sert à indiquer les agents de l'environnement qui sont touchés par les effets d'un cas de crise. Ce prédicat n'est pas obligatoire, mais il peut apparaître plusieurs fois dans la description d'une même observation (car une crise peut porter atteinte à plusieurs agents de l'environnement ; une pollution des eaux peut, par exemple, toucher des poissons, des végétaux, etc. (figure 4.9)).

En phase d'exploitation, ce prédicat permet de déduire :

- les agents de l'environnement qui sont touchés par certains types de problèmes,
- les experts qui ont une expérience pour le traitement de problèmes ayant de telles conséquences,
- les compétences requises pour le traitement de telles conséquences du problème,
- etc.

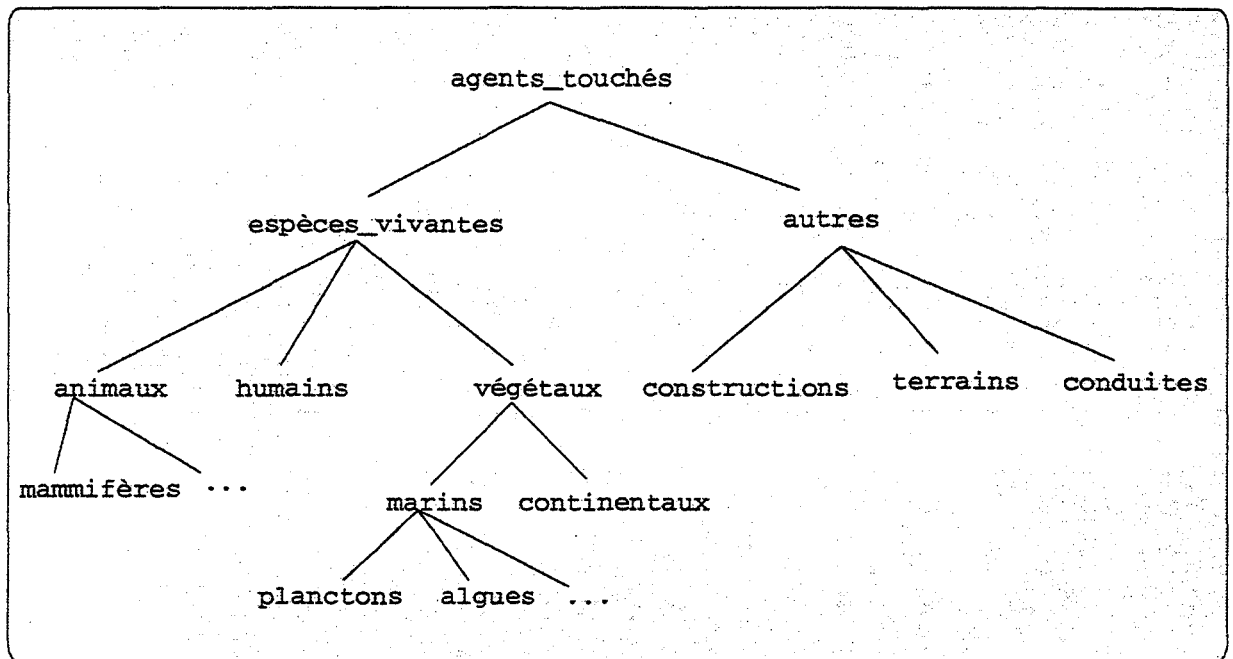


Figure 4.9 : Une partie de la taxinomie des agents de l'environnement

Afin de pouvoir prendre en compte le caractère évolutif des connaissances du domaine de l'eau, le langage de description que nous avons défini peut être amélioré soit en spécifiant de nouveaux prédicats soit en élargissant le champ du domaine de définition d'un attribut existant.

Pour ce qui concerne l'identification des experts, d'autres informations générales (adresses, numéros de téléphone, fonctions, etc. ) sont renseignés dans un fichier indépendant. Cela évite d'introduire ces informations dans la description de chaque observation et d'encombrer le graphe de connaissances RDP.

## 4.6 CONCLUSION

Dans ce chapitre, nous avons décrit le processus incrémental de construction de la BC du système EXPEAU. L'incrémentalité qui consiste à développer la BC en plusieurs étapes est très utile pour les domaines d'application dans lesquels : - le volume des données à traiter est important, - les données évoluent dans le temps. Pour le domaine de l'eau qui est un domaine pluridisciplinaire dans lequel les experts enrichissent leurs connaissances au fur et à mesure de

leurs interventions pour la résolution de problèmes en situation de crise, l'apprentissage incrémental s'est avéré d'un grand intérêt et a permis de résoudre le problème de l'acquisition de connaissances.

L'apprentissage incrémental nécessite, toutefois, la possibilité de constituer un ensemble d'observations pour servir d'exemples d'apprentissage. La construction d'une Base de Connaissances complète et efficace dans le domaine de l'eau, est fondée sur une spécification correcte de l'environnement des observations et le choix d'observations représentatives du domaine. Le langage de description des exemples d'apprentissage est choisi de manière à permettre aux experts du domaine de fournir facilement leurs expériences sous forme d'intervention sur des cas réels. Ce langage peut aussi être enrichi incrémentalement en intégrant la spécification de nouveaux descripteurs dans l'environnement des observations. Cela permet de décrire des données auxquelles les experts n'ont pas pensé au début du projet ou pour exprimer des concepts nouveaux qui n'existaient pas auparavant (nouvelles techniques, nouvelles compétences, etc.).

Nous avons vu également comment le graphe de connaissances construit à partir d'un ensemble d'observations, permet de révéler des relations de régularités qui existent entre les connaissances du domaine.

Par ailleurs, la construction incrémentale de la BC du système EXPEAU nécessite un suivi continu des connaissances acquises pour détecter et vérifier les incohérences et supprimer les redondances. Dans le chapitre 5, nous allons montrer comment exploiter les connaissances dans EXPEAU. Nous allons montrer aussi comment les résultats fournis par le système sont utilisés pour valider les connaissances par un suivi des interventions sur le terrain.



## ***5. Chapitre 5 : Phase d'utilisation et de validation du système EXPEAU***

### **5.1 INTRODUCTION**

Les systèmes d'apprentissage permettent d'acquérir et de structurer automatiquement des connaissances dans un domaine donné. Afin d'éviter le recueil de connaissances inutiles, l'apprentissage doit se faire par rapport à un objectif qui est fixé à l'avance. Cela permet de s'intéresser aux seules connaissances qui sont susceptibles de contribuer à la résolution des problèmes à traiter. Cette "orientation" de l'apprentissage pour acquérir des connaissances nécessaires pour la résolution d'une catégorie de problèmes, peut se faire en choisissant le langage de description des données d'entrée, le langage de représentation des connaissances acquises et des contraintes sur le processus d'apprentissage lui-même [Michalski 86a], [Ganascia 87], [Ganascia 88]. Pour ce qui concerne notre domaine d'application,



l'environnement d'apprentissage spécifié dans le chapitre 4 permet au système EXPEAU d'acquérir des connaissances nécessaires aux tâches de description de situations de crise et de sélection d'experts pouvant résoudre les problèmes engendrés par de telles situations.

Les connaissances apprises par les systèmes d'apprentissage peuvent être structurées sous différentes formes, telles que des arbres de décision, des frames, des graphes, des règles de production, etc. Dans notre cas, nous avons vu dans les chapitres 3 et 4 que la connaissance apprise est représentée dans un Réseau de Dépendances Perçues et forme un modèle évolutif du domaine.

Le processus de construction de la Base de Connaissances du système EXPEAU est décrit dans le chapitre 4. Ce processus a nécessité deux tâches qui ne sont pas indépendantes l'une de l'autre : la première tâche consiste à définir l'environnement sur lequel seront fondés les exemples d'apprentissage ; la deuxième tâche consiste à construire un RDP à partir d'un ensemble d'observations fourni par les experts du domaine. L'environnement d'apprentissage dépend du type d'observations à décrire pour servir d'exemples d'apprentissage ; et la construction du RDP dépend aussi de l'environnement des observations dans lequel sont spécifiées les modalités et les contraintes sur l'utilisation des exemples d'apprentissage. Puisque l'apprentissage dans notre cas se fait de manière incrémentale, il est nécessaire d'enrichir incrémentalement l'environnement des observations pour pouvoir prendre en compte l'évolution des connaissances du domaine.

Dans ce présent chapitre, nous allons décrire les procédures d'exploitation du modèle de connaissances construit ainsi que les procédures utilisées pour détecter des erreurs et incomplétudes que ce soit dans l'environnement des observations ou dans le RDP construit. Mais, ne perdons pas de vue que la détection des erreurs n'a d'intérêt que si des procédures de traitement et de correction peuvent être mises en oeuvre pour permettre une validation des connaissances. Pour cela, nous décrivons aussi un schéma de communication entre le système EXPEAU et son environnement permettant d'exploiter les rapports d'intervention des experts sur des cas de crise pour lesquelles ils ont été sélectionnés.

## **5.2 LES CONNAISSANCES DANS LE SYSTEME EXPEAU**

L'ensemble des connaissances du système EXPEAU est constitué d'une part des éléments de spécification de l'environnement des observations et d'autre part des connaissances contenues dans le RDP construit à partir d'exemples d'apprentissage fournis par les experts du domaine. En phase d'exploitation du système, les connaissances contenues dans le RDP sont utilisées pour donner une réponse à des requêtes formulées par les utilisateurs. Le système ne fait appel aux connaissances spécifiées dans l'environnement des observations que lorsque les connaissances contenues dans le RDP ne permettent pas d'aboutir à une solution.

### **5.2.1 Séparation des connaissances apprises et des connaissances du domaine**

Nous avons vu dans le chapitre 4 que le Réseau de Dépendances Perçues ne contient que les connaissances apprises à partir des observations fournies au système EXPEAU. Ces connaissances sont donc guidées par la description des interventions effectuées par les experts du domaine pour résoudre des problèmes engendrés par des situations de crise. Elles représentent les connaissances de surface, obtenues incrémentalement au fur et à mesure que des exemples d'apprentissage sont présentés au système. Les connaissances profondes qui, elles, sont définies dans la spécification de l'environnement des observations ne sont pas intégrées dans le RDP. Le système EXPEAU n'utilise ces connaissances qui permettent, entre autre, de décrire la sémantique du domaine que pendant le processus d'exploitation du graphe. C'est à dire que le système fait appel aux connaissances définies dans l'environnement des observations lorsque l'exploitation de la connaissance apprise requiert une connaissance générale qui détermine les domaines de définition, les règles d'application, etc.

La séparation en deux ensembles distincts de la connaissance dans le système EXPEAU offre plusieurs avantages permettant entre autres :

- d'assurer une cohérence sémantique des relations qui existent entre les éléments de connaissances du RDP. En effet, le RDP étant constitué uniquement des connaissances tirées à

partir des exemples d'apprentissage fournis au système, les relations qui existent entre ces connaissances sont des relations de régularités observées sur ces exemples,

- de faciliter la réutilisation de la connaissance profonde pour différents RDP construits dans ce domaine,
- d'avoir une souplesse d'utilisation des connaissances par le système. La connaissance profonde du domaine n'est utilisée que lorsque les connaissances contenues dans le RDP ne permettent pas d'aboutir à une solution,
- de faciliter la validation des connaissances. Ainsi, si les observations ayant permis de construire un RDP deviennent obsolètes, elles peuvent être retirées de l'ensemble d'apprentissage sans se soucier de l'environnement des observations dans lequel elles sont construites.

## 5.2.2 Utilisation des connaissances

Le graphe de connaissances construit à partir des cas d'intervention représente un modèle du domaine adapté pour la recherche d'experts. Ce modèle est constitué d'un ensemble de classes d'Equivalences Perçues et d'un ensemble de liens représentant les relations de Dépendances Perçues qui existent entre ces classes. Au cours de l'exploitation de ce graphe de connaissances, le système utilise les connaissances du domaine spécifiées dans l'environnement des observations.

Les connaissances dans le système EXPEAU peuvent être utilisées de deux manières différentes. D'une part, le RDP peut être exploité directement, tel qu'il est construit, soit dans un but pédagogique, soit pour la recherche d'une solution à une requête décrite par l'utilisateur. D'autre part, les connaissances contenues dans l'environnement des observations et celles représentées sur le RDP peuvent être utilisées en déduction pour la sélection des experts pouvant résoudre des problèmes engendrés par une situation de crise donnée.

### 5.2.2.1 Utilisation en déduction

Les observations qui servent d'exemples d'apprentissage sont décrits en utilisant des descripteurs qui sont spécifiés dans l'environnement des observations. Cet environnement permet de donner aux exemples d'apprentissage une forme compréhensible par le système (le format des exemples d'apprentissage est décrit dans le chapitre 4). La description d'une observation est constituée d'une liste d'au moins deux descripteurs, à savoir le descripteur qui sert à identifier l'intervenant et celui indiquant les compétences mises en oeuvre au cours de l'intervention sur un cas de crise. Les autres descripteurs servent à donner plus de précision sur la situation de crise sur laquelle est effectuée l'intervention (la zone géographique de l'intervention, la durée de l'intervention, le rôle joué par l'intervenant, etc.).

Comme règle sur le format des exemples d'apprentissage, nous pouvons constater que la description de chaque intervention sur un cas de crise est liée à un intervenant. Nous ne pouvons donc pas fournir au système une observation qui ne contient pas l'acteur de cette observation. Si plusieurs intervenants ont collaboré au même titre et dans les mêmes conditions à une intervention sur une situation de crise, l'observation est représentée autant de fois qu'il y a d'intervenants, et à chaque fois nous précisons l'intervenant correspondant.

Par contre, lors de l'exploitation de la connaissance apprise, nous cherchons plutôt à identifier les experts qui pourraient résoudre des problèmes engendrés par une situation de crise. Une requête formulée au système prend donc la forme d'une observation incomplète. Nous pouvons dire, en effet, qu'une telle observation représente un exemple d'apprentissage auquel il manque le descripteur identifiant l'intervenant. De la même manière avec laquelle nous avons défini un langage de description des observations servant d'exemples d'apprentissage, nous définissons aussi un langage d'interrogation qui permet de formuler des requêtes au système et un langage de réponse dans lequel seront décrits les résultats de la requête. Toutefois, pour ne pas se retrouver avec des requêtes que le système ne peut pas traiter, il est nécessaire que le langage d'interrogation soit un sous ensemble du langage d'observation.

### 5.2.2.1.1 Langage d'interrogation

En utilisant le langage d'interrogation, nous décrivons des requêtes constituées d'une liste de descripteurs représentant des situations de crise. Il est clair que le langage utilisé à ce niveau est un sous ensemble du langage de description qui est spécifié dans l'environnement des observations. Car si des descripteurs de la requête ne sont pas connus de l'environnement des observations et donc ne sont pas non plus intégrés dans le RDP, le système ne pourra pas satisfaire cette requête. Le langage d'interrogation est choisi de manière à ce qu'il offre la possibilité de décrire des situations de crise qui puissent survenir dans le domaine de l'eau. Il est constitué des éléments suivants :

#### **Domaine de compétence de l'intervention**

competence\_inter : compétences que doit posséder l'intervenant.

Dans la description d'une requête, ce descripteur est le plus important. Il permet, en effet, de représenter le domaine de compétence de la situation de crise décrite par cette requête.

exemple : (competence\_inter=hydrologie\_radar), (competence\_inter=réseaux), etc.

#### **Agents de l'environnement concernés par l'intervention**

agents\_inter : les agents touchés par la situation de crise et que doit traiter l'intervenant.

exemple : (agent\_inter = algues), (agent\_inter=mammifères), etc.

Ce prédicat permet de préciser les agents de l'environnement qui sont touchés par une situation de crise. Cela permet donc au système de se référer aux experts qui ont traité des situations ayant les mêmes conséquences.

#### **Lieu de l'intervention**

lieu\_inter : lieu où doit se dérouler l'intervention.

Ce descripteur permet au système de déduire les experts qui ont une expérience sur des problèmes qui surviennent dans une zone géographique donnée.

**Rôle joué par l'intervenant**

role\_inter : rôle à jouer par l'intervenant.

Ce descripteur permet de préciser le niveau d'intervention requis.

exemple : (role\_inter = maître\_d'oeuvre), (role\_inter = coordinateur\_technique), etc.

**Mission**

mission\_inter : mission que doit mener l'intervenant

Ce descripteur permet de préciser la tâche que doit réaliser l'expert dans l'intervention pour laquelle il est sélectionné.

exemple : (mission\_inter = étude), (mission\_inter = réalisation), etc.

**5.2.2.1.2 Langage des réponses**

Sachant que les résultats fournis par le processus d'exploitation ne contiennent que les descripteurs qui permettent d'identifier les intervenants potentiels, il est nécessaire de définir le langage dans lequel vont être formulées les réponses pour restreindre leurs descriptions aux seuls éléments qui intéressent l'utilisateur du système. Le langage des réponses est constitué des éléments suivants :

**Identification de l'intervenant**

nom\_inter : nom de l'expert sélectionné.

prenom\_inter : prénom de l'expert sélectionné.

**Organisme**

organisme\_inter : organisme auquel appartient l'expert sélectionné.

A partir du descripteur "identification de l'intervenant", nous pouvons accéder aux autres informations au sujet des experts sélectionnés (coordonnées, fonctions, langues, etc.), en utilisant le fichier dans lequel sont renseignés ces informations.

### 5.2.2.1.3 Recherche de solutions

La recherche de la solution est une suite d'inférences et de reformulations de la requête. A partir d'une description constituée d'une liste d'éléments du langage d'interrogation, le système sélectionne l'ensemble des noeuds du RDP qui contiennent au moins un descripteur de la requête dans leur étiquette. Pour chaque noeud  $n_i$  sélectionné, nous déduisons tous les autres noeuds  $n_j$  tel que :  $\text{chemin}(n_i, n_j) = \text{vrai}$  ou  $\text{chemin}(n_j, n_i) = \text{vrai}$ . L'analyse des éléments constituant l'étiquette de ces noeuds permet d'aboutir à un résultat positif dans le cas où une réponse est trouvée ou à un résultat négatif dans le cas où aucune réponse n'est trouvée. Si, dans un premier temps, le système ne trouve pas de réponse, il reformule la requête en utilisant la connaissance profonde.

L'organigramme illustrée sur la figure 5.1 montre bien comment à partir d'une première formulation décrivant le problème à résoudre, la description est modifiée à chaque itération jusqu'à ce qu'une description équivalente pour laquelle il existe une réponse soit obtenue. La reformulation de problèmes s'appuie sur les connaissances du domaine spécifiées dans l'environnement des observations.

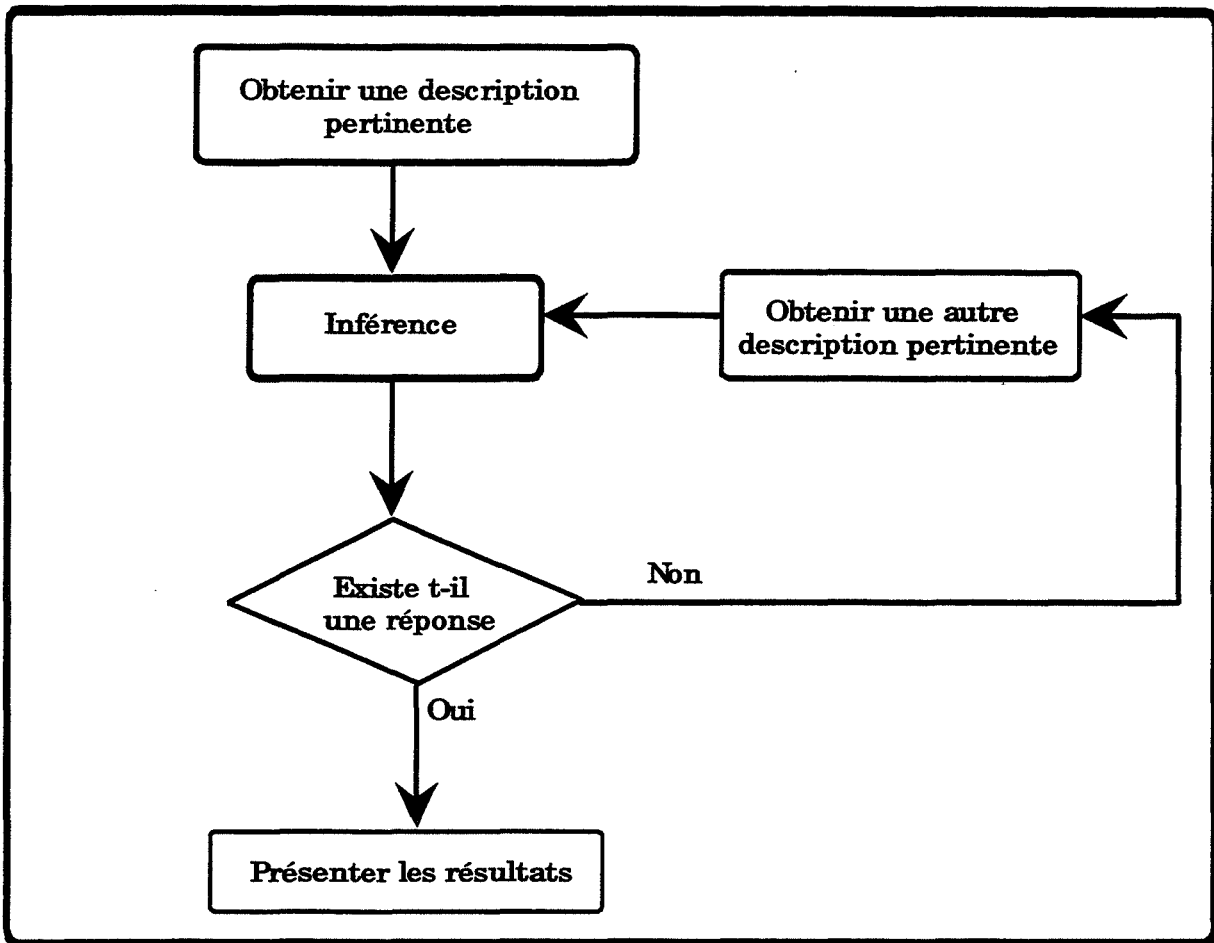


Figure 5.1 : Schéma de recherche d'une solution

Nous allons présenter, maintenant, l'algorithme utilisé par le système pour aboutir à des éléments de réponse (une liste d'experts) à partir de la description d'un cas de crise.

Etant donnés les éléments suivants (décrits dans le chapitre 4) :

- $T$  : le noeud le plus général du RDP (la tautologie),
- $f_n$  : l'étiquette du noeud  $n$ ,
- $\text{suc}(n)$  : l'ensemble des noeuds successeurs du noeud  $n$ ,
- $\text{ant}(n)$  : l'ensemble des noeuds antécédents du noeud  $n$ .

L'algorithme consiste en la réalisation des étapes suivantes :



- pour chaque descripteur  $d_i$  de la requête, localiser le noeud  $n_i$  du RDP dont l'étiquette contient ce descripteur  $d_i$ , c'est à dire le noeud qui vérifie la condition :  $d_i \in f_{n_i}$ ,

- visiter tous les noeuds  $n_j$  ( $i \neq j$ ) qui ont un lien avec le noeud  $n_i$ , c'est à dire tous les noeuds pour lesquels :  $f_{n_i} \rightarrow f_{n_j}$  ou  $f_{n_j} \rightarrow f_{n_i}$ ,

- pour chaque noeud visité, ajouter son étiquette à la liste des éléments liés au descripteur  $d_i$ ,

- lorsque tous les descripteurs  $d_i$  sont traités, faire une intersection entre les éléments liés à chacun de ces descripteurs. Si cette opération donne comme résultat un ensemble vide, il faut effectuer une reformulation de la requête, sinon le résultat représente la réponse à la requête.

début

Desc  $\leftarrow$  description du problème

Tant que Desc  $\neq \emptyset$  faire

N  $\leftarrow$  nombre(descripteurs(Desc)) /\* nombre de descripteurs de la requête \*/

Pour tout  $i = 1, N$  faire

Localiser le noeud  $n$  qui contient le  $i^{\text{ème}}$  descripteur ( $d_i$ ) de Desc

$Et_n \leftarrow f_n - d_i$  /\*  $Et_n$  est l'étiquette d'un noeud  $n$  diminuée du descripteur  $d_i$  appartenant à la description du problème \*/

$S_n \leftarrow \text{suc}(n) - T$

$A_n \leftarrow \text{ant}(n)$

$P(i) \leftarrow Et_n \cup S_n \cup A_n$

fin pour

$Pr \leftarrow \bigcap_{i=1, N} P(i)$

Si Pr =  $\emptyset$  alors

Desc  $\leftarrow$  reformulation (Desc) /\* procédure de reformulation de la requête\*/

Sinon

Choisir (Pr) /\* les éléments de Pr qui appartiennent au langage de réponse et  
qui appartiennent à un plus grand nombre d'exemples \*/

Desc ←  $\emptyset$

**fin si**

**fin tant que**

afficher résultat

**fin**

### **Procédure reformuler(Desc)**

#### **Début**

- utiliser les heuristiques de reformulation pour élire un descripteur
- utiliser les connaissances spécifiées dans l'environnement des observations  
pour généraliser le descripteur
- remplacer le descripteur élu par le descripteur résultant de la généralisation
- retourner la nouvelle description

**fin**

### **Exemple :**

Supposons, par exemple, l'environnement des observations dans lequel sont spécifiés les éléments suivants :

identif : type nominal : domaine (azili, badot, berrada, matia, mujeriego, pares)

région : type structure (figure 5.2)

compétence : type structure (figure 5.3)

Ces éléments permettent d'identifier l'intervenant sur un cas de crise et de spécifier les zones géographiques de l'intervention ainsi que les compétences requises pour cette intervention.

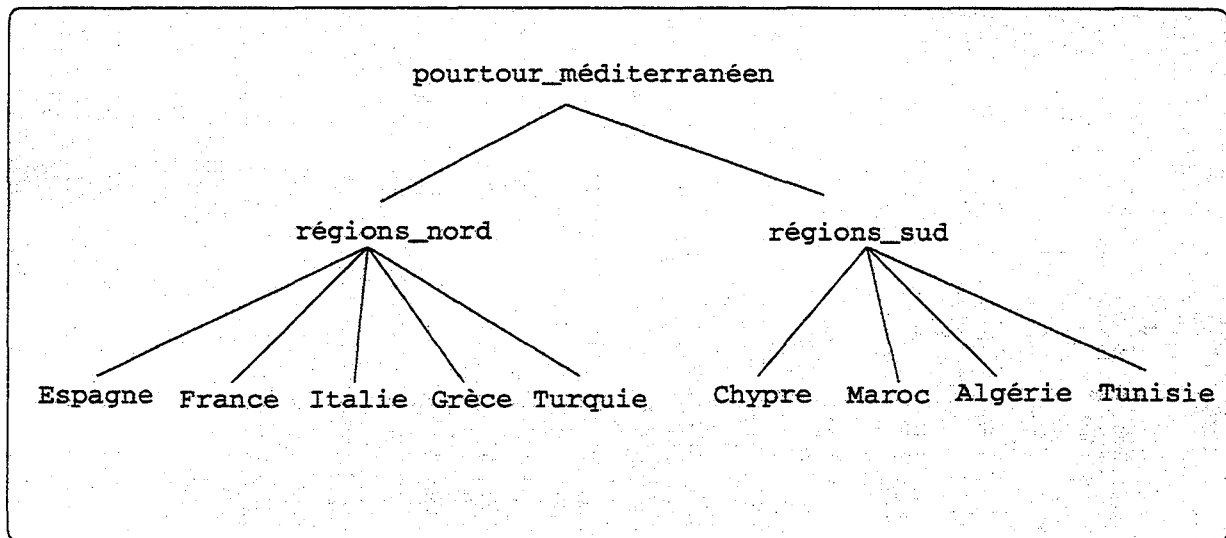


Figure 5.2 : Taxinomie des régions

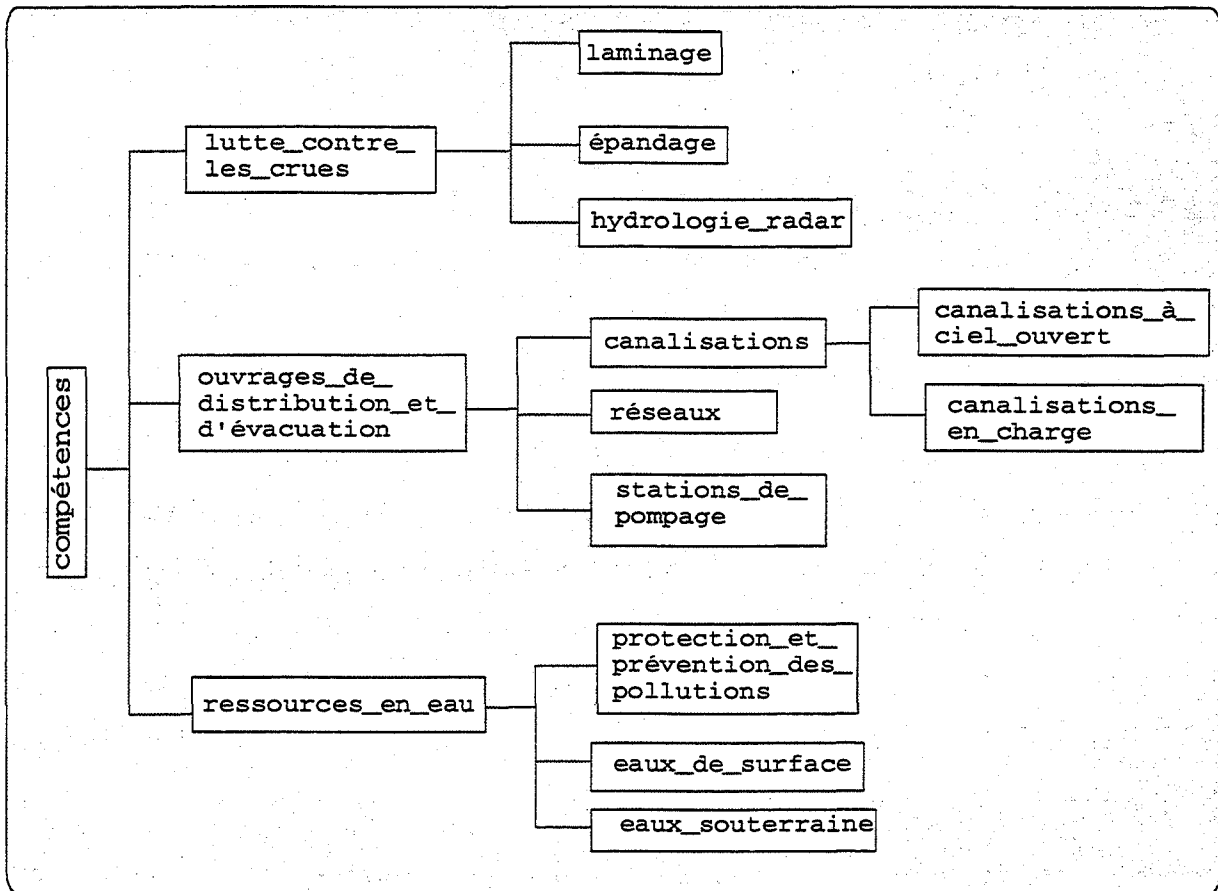


Figure 5.3 : Une partie de la taxinomie des compétences

Supposons aussi que nous avons fourni au système l'ensemble des observations  $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$  décrites par  $\{d(e_1), d(e_2), d(e_3), d(e_4), d(e_5), d(e_6), d(e_7)\}$  tel que :

$d(e_1) : \{(identif = azili) \& (région = maroc) \& (compétence = épandage) \& (compétence = canalisations) \& (compétence = ressources\_en\_eau)\}$

$d(e_2) : \{(identif = berrada) \& (région = maroc) \& (compétence = canalisations) \& (compétence = ressources\_en\_eau)\}$

$d(e_3) : \{(identif = pares) \& (région = espagne) \& (compétence = réseaux) \& (compétence = stations\_de\_pompage)\}$

$d(e_4) : \{(identif = mujeriego) \& (région = espagne) \& (compétence = canalisations\_en\_charge) \& (compétence = stations\_de\_pompage)\}$

$d(e_5) : \{(identif = matia) \& (région = espagne) \& (compétence = canalisations) \& (compétence = ressources\_en\_eau)\}$

$d(e_6) : \{(identif = azili) \& (région = maroc) \& (compétence = canalisations\_en\_charge) \& (compétence = stations\_de\_pompage)\}$

$d(e_7) : \{(identif = badot) \& (région = france) \& (compétence = laminage) \& (compétence = canalisations) \& (compétence = ressources\_en\_eau)\}$

Le RDP construit par le système à partir de ces observations est illustré sur la figure 5.4. Les noeuds  $n_{12}$ ,  $n_{17}$ ,  $n_{18}$ ,  $n_{21}$ ,  $n_{22}$ ,  $n_{23}$ ,  $n_{24}$  caractérisés par un fond plein, représentent la base des observations fournies au système. L'étiquette du noeud  $n_{21}$  est vide, son existence a uniquement pour but d'offrir la possibilité de reconstituer l'observation qui lui correspond. Les noeuds vides dans un RDP ne peuvent donc être que des noeuds n'ayant aucun noeud antécédent et ayant nécessairement au moins deux noeuds successeurs. Ainsi  $\text{succ}(n_{21}) = \{n_{13}, n_{16}, n_{19}, n_{20}\}$  permet de reconstituer la description de l'observation ( $e_6$ ).

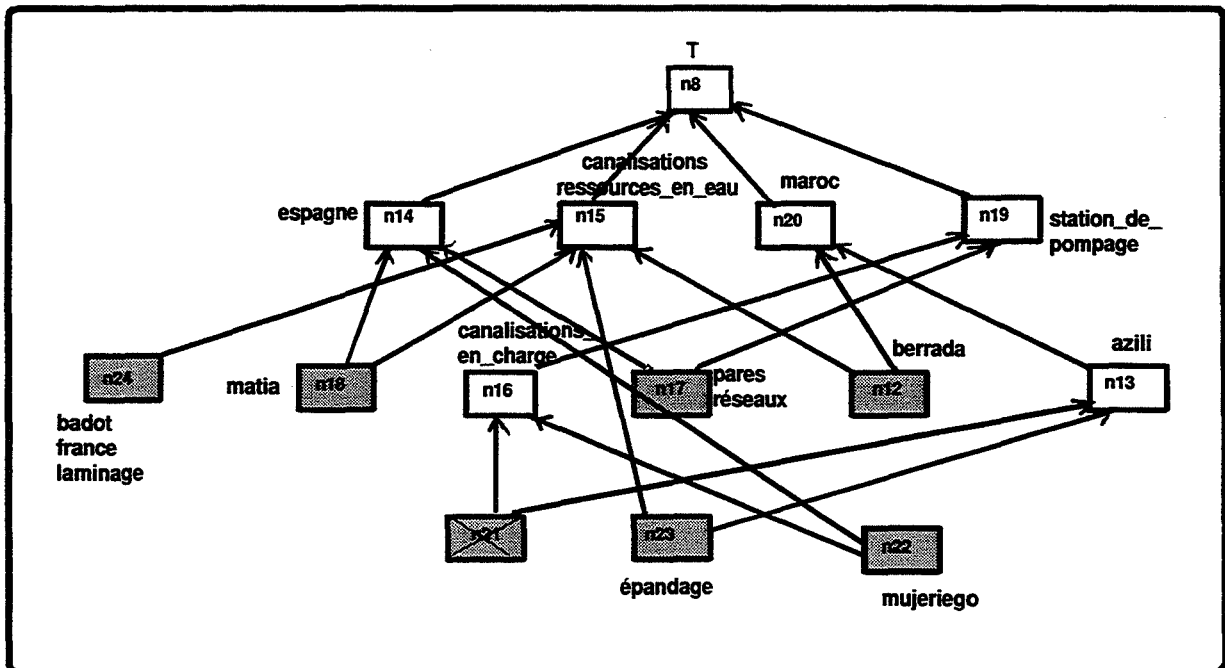


Figure 5.4 : RDP résultant des observations  $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$

Si maintenant nous présentons au système une requête de la forme : (région = maroc) & (compétence = canalisations) & (compétence = eaux\_de\_surface), l'exploitation des connaissances contenues dans le RDP ne permet pas de trouver directement une réponse adéquate. D'ailleurs comme nous pouvons le constater, le descripteur "eaux\_de\_surface" n'appartient même pas à l'ensemble des connaissances contenues dans le RDP. Ainsi, la connaissance du domaine peut être mise en jeu pour sortir de cette situation qui est inconnue du système. En utilisant la taxinomie des compétences, le système va alors reformuler la requête sous la nouvelle forme suivante : (région = maroc) & (compétence = canalisations) & (compétence = ressources\_en\_eau), qui devient une situation connue du système. Le résultat retourné par le système en utilisant l'algorithme décrit ci-dessus est : (identif = azili), (identif = berrada).

Nous avons vu sur un exemple simple comment passer d'une description d'une requête initiale à une autre requête en utilisant les connaissances spécifiées dans l'environnement des observations. Il demeure que pour des requêtes plus complexes se pose le problème du choix des descripteurs à remplacer par d'autres pour aboutir à de nouvelles requêtes.

Il se trouve que pour l'exemple que nous avons traité précédemment, il paraît évident de commencer par remplacer le descripteur (compétence = eaux\_de\_surface) dans le but d'obtenir une requête que pourra traiter le système. Mais, supposons, que même en ayant effectué cette opération, le système ne trouve pas de solution ; il est nécessaire donc de passer à une autre description, en partant des trois descripteurs décrivant la requête courante. Les problèmes qui se posent sont les suivants : quel descripteur prendre en compte en premier ? Jusqu'à quel niveau peut-on généraliser un descripteur avant de passer à un autre descripteur ?

Une solution peut être donnée par l'indication d'un ordre de priorité sur les généralisés potentiels ainsi que des contraintes permettant de contrôler le nombre de généralisations sur un descripteur. C'est, en quelques sortes, une façon de restreindre les possibilités de généralisation pour un besoin d'exploitation des connaissances. Ce mécanisme de contrôle existe dans les systèmes d'apprentissage, tel que CHARADE [Ganascia 88] mais utilisé comme "learning bias" pour un besoin d'apprentissage et non pas pour l'exploitation de connaissances.

Dans ce qui suit, nous allons présenter quelques heuristiques d'exploitation et de reformulation de requêtes utilisées dans le système EXPEAU.

#### **5.2.2.1.4 Les heuristiques de recherche et de reformulation**

Les requêtes présentées au système EXPEAU sont constituées d'un ensemble de descripteurs appartenant au langage d'interrogation que nous avons décrit dans le paragraphe 5.2.2.1.1. La stratégie de reformulation d'une requête initiale s'appuie tout d'abord sur le fait que nous ne donnons pas la même importance aux différents descripteurs de la requête. Ainsi si nous avons une requête qui consiste à interroger le système dans le but de sélectionner des experts pouvant traiter un cas de crise tel que : "pollution d'un cours d'eau", nous constatons en premier lieu que les descripteurs de cette requête sont : "pollution" et "cours d'eau". Intuitivement, en cas de besoin de reformulation de cette requête, il paraît évident de commencer par remplacer le descripteur "cours d'eau" par un autre plus général (le descripteur "eau de surface" par exemple). La requête résultante sera alors sous la forme : "pollution de l'eau de surface". S'il paraît évident de faire cette opération, c'est parce que parmi les deux descripteurs constituant la requête, le descripteur "pollution" est plus important que le descripteur "cours d'eau" dans ce contexte particulier. Mais, le système EXPEAU n'a pas cette capacité de classification intuitive des descripteurs d'une requête et si des informations complémentaires ne lui sont pas indiquées d'une manière ou d'une autre, il ne pourra pas effectuer cette reformulation. C'est pour répondre à ces besoins que nous devons intégrer dans le système EXPEAU des connaissances stratégiques (heuristiques) sur l'utilisation des connaissances du domaine et des connaissances représentées sur le RDP. Un type de ces connaissances stratégiques représente alors un ensemble de règles de reformulation de requêtes.

##### **Exemples de règles stratégiques :**

- **Si** le problème courant est un problème de pollution des eaux marines

**alors** prendre en compte des interventions dans lesquelles sont traités les poissons et les algues

- **Si** la description de la requête renferme les descripteurs (compétence requise, rôle à jouer par l'expert, mission de l'intervenant)

**alors** effectuer les généralisations dans l'ordre suivant : (mission de l'intervenant, rôle à jouer par l'expert, compétence requise)

- **Si** la description de la requête renferme plusieurs fois le même descripteur de type "hiérarchique"

**alors** effectuer les généralisations en commençant par les éléments de niveau le plus bas dans la taxinomie de ce descripteur

Les heuristiques d'exploitation de connaissances et de reformulation de requêtes représentent la métaconnaissance du domaine [Lenat 82], [Pitrat 90]. Cette métaconnaissance est donnée par les experts du domaine dans la phase d'acquisition de connaissances qui consiste à recueillir les connaissances profondes et les stratégies de résolution de problèmes. Contrairement aux connaissances profondes du domaine, la métaconnaissance n'est pas intégrée dans l'environnement des observations. Elle est représentée sous une forme procédurale dans la partie qui contrôle les connaissances de l'environnement des observations et celles contenues dans le RDP.

Dans le cas où aucune heuristique de reformulation n'est indiquée pour un contexte donné, le système suppose que les descripteurs de la requête sont donnés par ordre d'importance. Ainsi, le processus de reformulation commencera par généraliser le descripteur qui est en fin de la liste, et puis l'avant dernier de la liste, etc. En effet, comme nous l'avons vu auparavant pour l'exemple d'une requête qui est "pollution d'un cours d'eau", nous décrivons une requête en commençant par les descripteurs les plus importants. Pour reformuler la requête sans s'éloigner du contexte initial, nous commençons par généraliser les descripteurs qui ont



moins d'importance, et donc ceux qui viennent en fin de la liste contenant les différents descripteurs.

Cette heuristique représente une heuristique standard que le système utilise à chaque fois qu'il se trouve dans une situation inconnue et qu'aucune heuristique n'est donnée pour le cas correspondant. Cette heuristique est sous la forme suivante :

- **Si** la requête renferme respectivement les descripteurs ( $d_1, d_2, \dots, d_n$ )

**alors** effectuer les généralisations dans l'ordre suivant : ( $d_n, d_{n-1}, \dots, d_1$ )

D'autres métaconnaissances, liées au problème de la reformulation de requêtes, permettent d'indiquer à quel niveau de généralité d'un descripteur, le système doit-il arrêter la généralisation d'un descripteur pour passer à un autre descripteur.

- **Si** le descripteur élu pour la généralisation est de type "hiérarchique" **et si** ( $n - m > 2$ )

/\* avec n : niveau du descripteur dans la taxinomie et m : nombre de fois que ce descripteur est généralisé \*/ ;

**alors** effectuer la généralisation sur ce descripteur

Si aucune heuristique n'est donnée pour cela, lorsqu'un descripteur  $d_i$  est élu pour être généralisé, le système EXPEAU essaie toutes les possibilités de généralisation de ce descripteur, en commençant par les plus spécifiques, avant de passer à un autre descripteur.

- **Si** le descripteur élu pour la généralisation est de type "hiérarchique" **et si** ( $n > 0$ )

/\* avec n : niveau du descripteur dans la taxinomie \*/

**alors** effectuer la généralisation sur ce descripteur

### **5.2.2.2 Utilisation directe du graphe RDP**

Nous avons déjà indiqué que le modèle de connaissances représenté par le Réseau de Dépendances Perçues peut être utilisé par les experts du domaine dans un but pédagogique ou pour trouver une solution à une requête. Dans ce qui suit, nous allons décrire les procédures d'utilisation du RDP pour les deux cas.

#### **5.2.2.2.1 Utilisation du RDP dans un but pédagogique**

Le RDP est constitué d'un ensemble de connaissances hiérarchisées sous forme de classes appelées "classes d'équivalences perçues", reliées entre elles par des liens qui explicitent des relations de dépendances perçues. Ainsi, le RDP met en évidence des relations qui existent entre les connaissances tirées des observations fournies par les experts du domaine. Mais ces relations entre les classes de connaissances ne sont représentatives de la réalité que lorsque les observations fournies au système couvrent suffisamment le domaine. Lorsque l'ensemble des exemples d'apprentissage n'est constitué que d'une seule observation, le RDP résultant est réduit à un seul noeud dont l'étiquette contient tous les éléments de connaissances de l'observation ; ce qui ne donne pas vraiment les relations qui peuvent exister entre les connaissances du domaine. Mais au fur et à mesure que l'ensemble des exemples d'apprentissage s'enrichit, des relations entre les classes d'équivalences perçues apparaissent et permettent d'affiner le modèle construit.

Partant du fait que les connaissances constituant une classe d'équivalences perçues sont celles qui apparaissent ensemble dans les observations fournies au système, l'expert peut ainsi découvrir certaines relations entre les connaissances du domaine qu'il ignorait auparavant.

En examinant l'ensemble des relations de dépendance perçue entre les classes d'équivalences perçues du RDP, l'expert peut aussi tirer des conclusions telles que :

- les catastrophes qui se produisent dans une région donnée,

- les compétences requises pour résoudre les problèmes engendrés par une catastrophe donnée,
- la durée d'intervention qui est nécessaire pour résoudre un type de problème,
- le nombre d'experts pouvant intervenir pour résoudre les problèmes engendrés par une catastrophe donnée,
- les experts qui interviennent dans des régions données,
- les pays qui sont spécialisés dans le traitement de certaines catastrophes,
- etc.

En utilisant directement le RDP, nous pouvons ainsi déduire les sous domaines qui ne sont pas couverts par les observations fournies au système. Nous pouvons savoir, par exemple, quels sont les types de problèmes qui ne sont pas représentés sur le RDP ou pour lesquels il n'existe pas d'experts compétents. Nous allons voir plus loin comment il est aussi possible de détecter des incohérences dans les connaissances en utilisant le RDP.

Après avoir montré la possibilité d'utiliser directement le modèle de connaissances représenté par le RDP pour découvrir des relations entre les connaissances du domaine de l'eau, nous allons décrire la procédure permettant de trouver une solution en réponse à une requête donnée.

#### **5.2.2.2 Utilisation du RDP pour la recherche de solutions**

En utilisant de manière interactive le RDP construit, il est possible de formuler des requêtes et d'aboutir aux éléments de réponse correspondant. Seulement, à l'inverse de ce que nous avons présenté au paragraphe traitant de l'utilisation des connaissances en déduction, ici nous ne définissons pas un langage d'interrogation et un autre langage de réponses, mais les éléments de connaissances sont utilisés indifféremment (comme éléments d'interrogation ou de réponse). Les connaissances spécifiées dans l'environnement des observations ne sont pas

utilisées dans ce cas, ce qui induit une perte d'efficacité dans l'exploitation du système. Mais, cette procédure peut être utilisée pour trouver rapidement des solutions (sans faire appel à la connaissance profonde) à des requêtes simples. Elle consiste à sélectionner sur le RDP l'ensemble des éléments qui vont constituer une requête et de déterminer les éléments de réponses en examinant les relations qui existent entre les classes d'équivalences de ce RDP.

Supposons l'ensemble des exemples d'apprentissage constitué des observations  $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$  décrites par  $\{d(e_1), d(e_2), d(e_3), d(e_4), d(e_5), d(e_6), d(e_7), d(e_8)\}$  tel que :

$d(e_1) : \{(identif = azili) \& (région = maroc) \& (compétence = épandage) \& (compétence = canalisations) \& (compétence = ressources\_en\_eau)\}$

$d(e_2) : \{(identif = berrada) \& (région = maroc) \& (compétence = canalisations) \& (compétence = ressources\_en\_eau)\}$

$d(e_3) : \{(identif = pares) \& (région = espagne) \& (compétence = réseaux) \& (compétence = stations\_de\_pompage)\}$

$d(e_4) : \{(identif = mujeriego) \& (région = espagne) \& (compétence = canalisations\_en\_charge) \& (compétence = stations\_de\_pompage)\}$

$d(e_5) : \{(identif = matia) \& (région = espagne) \& (compétence = canalisations) \& (compétence = ressources\_en\_eau)\}$

$d(e_6) : \{(identif = azili) \& (région = maroc) \& (compétence = canalisations\_en\_charge) \& (compétence = stations\_de\_pompage)\}$

$d(e_7) : \{(identif = badot) \& (région = france) \& (compétence = laminage) \& (compétence = canalisations) \& (compétence = ressources\_en\_eau)\}$

$d(e_8) : \{(identif = deschamp) \& (région = france) \& (compétence = laminage) \& (compétence = canalisations) \& (compétence = ressources\_en\_eau) \& (compétence = stations\_de\_pompage)\}$

Le RDP construit par le système à partir de ces observations est le suivant :

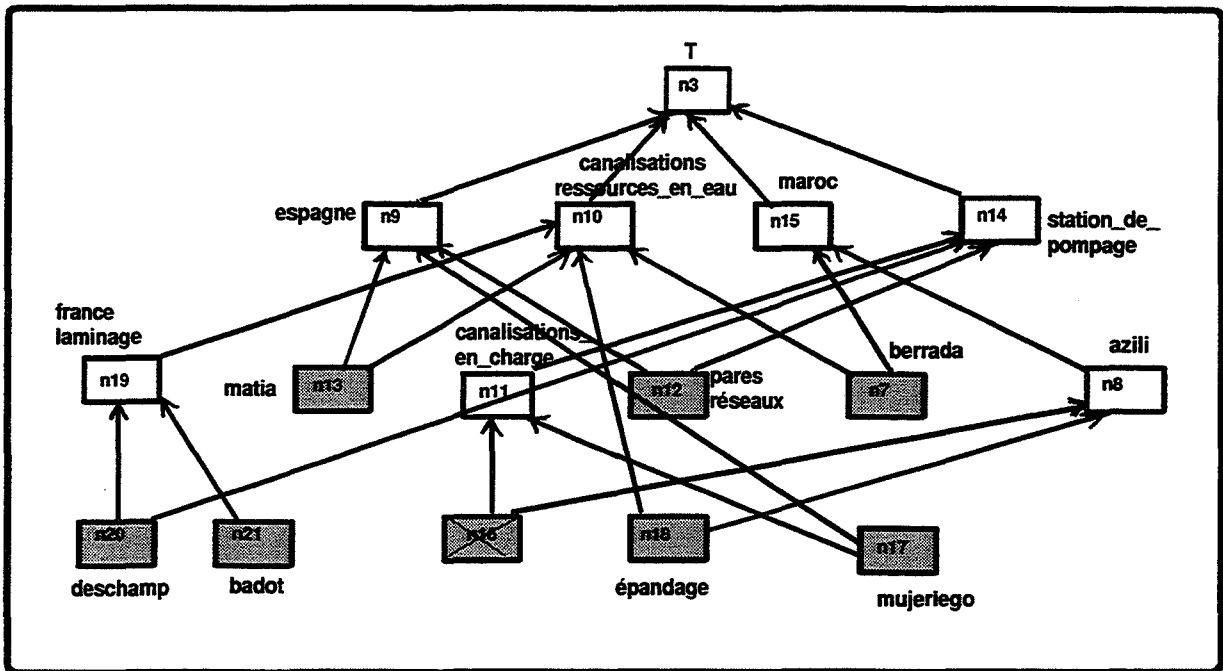


Figure 5.5 : RDP résultant des observations  $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$

On peut constater que l'ajout d'une observation au RDP de la figure 5.4, n'a pas modifié les classes d'équivalences perçues existantes, à l'exception de la classe constituée des éléments :  $\{(identif = badot), (région = france), (compétence = laminage)\}$  qui est transformée en l'ensemble  $\{(région = france), (compétence = laminage)\}$  pour être utilisée aussi pour l'observation contenant l'élément (identif = deschamp). Avant l'ajout de la nouvelle observation ( $e_8$ ), la classe des éléments  $\{(identif = badot), (région = france), (compétence = laminage)\}$  représentait l'étiquette du noeud  $n_{24}$  sur la figure 5.4 pour lequel  $ant(n_{24}) = \emptyset$ , ce qui signifie qu'il représente un noeud terminal. Après l'ajout de l'observation ( $e_8$ ), cette classe renferme l'ensemble des éléments  $\{(région = france), (compétence = laminage)\}$  qui représente l'étiquette du noeud  $n_{19}$  pour lequel  $ant(n_{19}) = \{n_{20}, n_{21}\}$  (sur la figure 5.5). Ces deux noeuds  $n_{20}$  et  $n_{21}$  sont étiquetés respectivement par les éléments (identif = deschamp) et (identif = badot).

Si, par exemple, nous voulons sélectionner les experts qui peuvent intervenir sur une situation décrite par le seul descripteur (compétence = stations\_de\_pompage), nous examinons

les liens qui interviennent en sélectionnant le noeud correspondant ( $n_{14}$ ) et nous obtenons alors le schéma illustré sur la figure 5.6. Les noeuds qui sont concernés dans cette première étape sont ceux qui sont reliés au noeud ( $n_{14}$ ) par un trait continu, à savoir :  $n_3$ ,  $n_{12}$ ,  $n_{11}$ ,  $n_{20}$ ,  $n_{16}$ ,  $n_{17}$ . Si nous voulons s'intéresser uniquement au descripteur "identif" qui identifie les experts, nous trouverons : (identif = deschamp), (identif = pares), (identif = mujeriego), qui correspondent respectivement aux noeuds :  $n_{20}$ ,  $n_{12}$ ,  $n_{17}$ . Mais si nous examinons l'ensemble des observations fournies, nous trouverons que (identif = azili) est intervenu sur un cas comportant (compétence = stations\_de\_pompage), donc la solution que nous avons obtenu n'est pas complète. Pour couvrir tous les éléments qui sont en lien avec le descripteur (compétence = stations\_de\_pompage), il faut activer tous les noeuds qui sont sélectionnés et qui sont des noeuds terminaux (avec un fond plein) représentant des observations. Ainsi, en activant le noeud  $n_{16}$ , les noeuds  $n_8$  et  $n_{15}$  seront sélectionnés, et nous aurons donc abouti à (identif = azili).

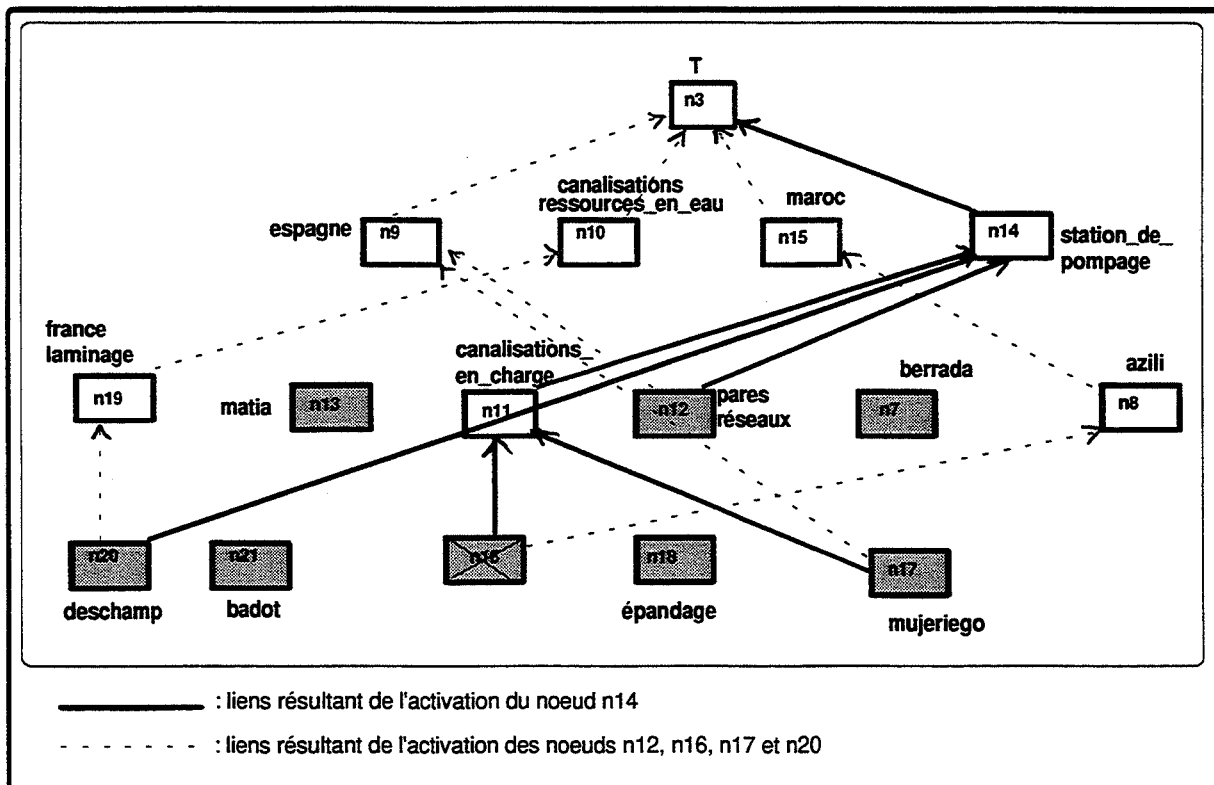


figure 5.6 : Activation d'un noeud du RDP

Cette opération nous permet de sélectionner tous les noeuds dont l'étiquette a un lien avec l'étiquette d'un noeud donné. Ainsi, à partir d'un noeud  $n$ , nous obtenons tous les noeuds  $n_i$  pour lesquels :  $\text{étiquette}(n) \rightarrow \text{étiquette}(n_i)$  ou  $\text{étiquette}(n_i) \rightarrow \text{étiquette}(n)$ .

En appliquant cette procédure pour un ensemble de noeuds correspondant à une requête que nous voulons décrire, nous obtenons l'ensemble des autres noeuds qui renferment le résultat recherché. L'algorithme qui décrit cette procédure est le suivant.

**Début**

Noeuds[s]<sub>s=1,N</sub>  $\leftarrow \emptyset$  /\* Noeuds est une liste de liste \*/

k  $\leftarrow 1$

**Pour** tout noeud  $n_i$  activé par l'utilisateur **faire**

Noeuds[k]  $\leftarrow \text{ant}(n_i) \cup \text{suc}(n_i)$

**Pour** tout noeud  $n_j \in \text{Noeuds}[k]$  et qui n'est pas encore traité **faire**

/\* ne traiter que les noeuds qui n'ont aucun noeud antécédent  
et qui ont au moins deux noeuds successeurs \*/

**Si**  $\text{ant}(n_j) = \emptyset$  et  $\text{cardinalité}(\text{suc}(n_j)) > 1$  **alors**

Noeuds[k]  $\leftarrow \text{Noeud}[k] \cup (\text{ant}(n_j) \cup \text{suc}(n_j))$

**fin si**

**fin faire**

k  $\leftarrow k + 1$

**fin faire**

**Si** k > 1 **alors**

/\* le résultat est représenté par les éléments qui sont sélectionnés  
à chaque activation d'un noeud  $n_i$  \*/

Rep  $\leftarrow \bigcap \text{Noeuds}[r] ; r = 1, \dots, k$

**sinon**

Rep  $\leftarrow \text{Noeuds}[1]$

**fin si**

**fin**

## 5.3 LE BRUIT DANS LES CONNAISSANCES

Si l'apprentissage automatique offre un intérêt pour acquérir de manière automatique et incrémentale des connaissances à partir d'observations représentant les expériences d'experts, il est un fait que ces connaissances peuvent toutefois intégrer des erreurs. Pour corriger ces erreurs, et si possible prévenir leurs sources, il est nécessaire de détecter d'abord leurs causes principales.

### 5.3.1 D'où peuvent venir les erreurs ?

Dans le système EXPEAU, le processus d'acquisition de connaissances se fait en plusieurs étapes pour pouvoir prendre en compte séquentiellement les observations fournies par des experts appartenant à différentes disciplines. La description de ces observations s'appuie sur un langage de description qui, lui aussi, est construit incrémentalement. Ainsi, les connaissances acquises peuvent être entachées de bruit pour différentes raisons [Manago 87] :

- Il est toujours possible que les experts se trompent lorsqu'ils fournissent leurs observations. Si, par exemple, en décrivant une observation, un expert omet de préciser un descripteur qui fait partie du cas de crise sur lequel il est intervenu, les relations qui existent entre les descripteurs peuvent être erronées,

- une autre cause de bruit dans les connaissances est due au langage de description qui est spécifié dans l'environnement des observations. Soit que ce langage ne permet pas aux experts de représenter la totalité de leurs cas d'intervention, soit que cette retranscription dans le langage spécifié induit des déformations de la connaissance (le langage de description utilisé peut être source de bruit),

- l'évolution des connaissances du domaine peut d'une part rendre les connaissances déjà acquises invalides, et d'autre part rendre le langage de description insuffisant pour représenter correctement de nouvelles observations,



- comme les exemples d'apprentissage sont fournis par plusieurs experts, des connaissances comportant des contradictions peuvent être intégrées dans le système,
- mises à part des erreurs qui existent explicitement dans les connaissances acquises, l'incomplétude des connaissances, due au fait que les experts n'ont pas encore fourni la totalité des informations les concernant, produit des erreurs dans les résultats fournis par le système,
- en plus des erreurs intégrées au sein de la connaissance apprise, les taxinomies décrites dans l'environnement des observations peuvent se révéler incorrectes. Il va donc falloir les corriger au fur et à mesure de l'exploitation du système, pour construire aussi incrémentalement ses taxinomies [Kietz 88].

Nous constatons que le langage de description des exemples d'apprentissage est un facteur important dans l'apparition d'erreurs dans les connaissances apprises. Les connaissances spécifiées dans l'environnement des observations et qui définissent le modèle de la connaissance profonde du domaine [Morik 87] doivent donc être acquises et structurées de manière efficace.

### **5.3.2 Besoin de traitement du bruit**

Dans toutes les applications utilisant l'apprentissage automatique pour acquérir des connaissances, apparaît le besoin de traitement des données bruitées. En effet, si les êtres humains arrivent à raisonner sur des connaissances bruitées, les systèmes cognitifs des SBC sont, eux, très sensibles aux erreurs et aux incohérences qui peuvent se trouver dans leur Base de Connaissances [Hayes-Roth 83]. Dans le cas où la spécification de l'environnement des observations est erronée, les erreurs peuvent se présenter à deux niveaux différents. D'une part, le Réseau de Dépendances Perçues qui est construit en s'appuyant sur les connaissances profondes du domaine peut contenir des erreurs, et d'autre part l'exploitation des connaissances apprises qui utilisent également les connaissances profondes du domaine peut donner des résultats incorrects. Il est donc indispensable de corriger les connaissances contenues dans l'environnement des observations (le langage de description, les domaines de définition, les règles de généralisation) afin d'aboutir à des résultats corrects.

Même si la spécification de l'environnement des observations est correcte, les connaissances contenues dans le RDP peuvent tout de même être erronées pour les raisons cités dans le paragraphe 5.3.1 et l'exploitation du RDP donnera des résultats incorrects. Il faut dans ce cas passer en revue les observations fournies par les experts pour les vérifier et les corriger.

Après avoir fait un diagnostic sur les erreurs qui peuvent être intégrées dans la Base de Connaissances du système EXPEAU, nous allons présenter dans ce qui suit les procédures qui permettent de les détecter et de les corriger.

## 5.4 VALIDATION DES CONNAISSANCES

Pour prévenir la production de résultats incorrects par le système EXPEAU, les connaissances acquises doivent être validées. La validation d'un SBC consiste à vérifier et à corriger les connaissances pour aboutir à une adéquation entre la connaissance modélisée dans le SBC et la connaissance réelle du domaine traité [Ayel 86], [Ayel 88]. Dans la pratique l'évaluation et la validation des connaissances acquises se fait de deux manières :

- le système présente, directement les connaissances recueillies, à l'expert sous une forme expressive qui facilite leur compréhension (en offrant, par exemple, la possibilité d'afficher les connaissances acquises sous forme de graphes),
- le système offre la possibilité d'évaluer les résultats produits lors des sessions de résolution de problèmes.

En général, la validation des connaissances dans un SBC nécessite de confronter les résultats produits par le système en utilisant ces connaissances aux résultats réels. Les systèmes qui permettent une évaluation et une justification des résultats en s'appuyant sur des dialogues [Beaune 92a], [Beaune 92b] sont très prometteurs. En effet, cela permet de vérifier réellement si l'erreur provient du raisonnement du système ou des connaissances fournies par les experts. Toutefois, pour aboutir à un bon résultat de validation, l'utilisation des deux démarches (validation directe, évaluation des résultats) est nécessaire. Cela peut s'expliquer par le

problème qui se pose au niveau de la démarche à suivre pour déterminer les scénarios de validation et les jeux d'essais. En effet, il n'est pas possible de savoir exactement à quel moment il faut arrêter la procédure de test pour passer à une utilisation réelle du système. Aussi, l'expert ne peut pas déceler toutes les incomplétudes et les erreurs dans la Base de Connaissances par le seul moyen d'une consultation facilitée des connaissances acquises.

Pour les raisons que nous venons d'explicitier, la validation des connaissances dans le système EXPEAU intègre les deux démarches. D'une part, comme la connaissance apprise est représentée par un graphe (le Réseau de Dépendances Perçues), les experts du domaine ont la possibilité de valider et de corriger directement les connaissances en visualisant et en vérifiant les connaissances contenues dans ce graphe. D'autre part, le suivi continu des interventions sur des problèmes réels permet d'évaluer les résultats fournis par le système et de valider les connaissances en cas de nécessité [Senoune 94b].

#### 5.4.1 Détecter et corriger les incohérences en utilisant le RDP

De la même manière que l'utilisation directe du RDP permet de révéler des relations qui existent entre les classes de connaissances qui le constituent, elle permet aussi de détecter des incohérences [Ayel 86], [Ayel 90], [Beauvieux 88] dans ces connaissances. Ces incohérences peuvent se présenter sous différentes formes, à savoir :

- **incohérence dans une même classe d'Equivalences Perçues** : les descripteurs appartenant à une même classe d'Equivalences Perçues sont ceux qui apparaissent ensemble dans la description des exemples d'apprentissage. Si, par exemple, dans une même classe d'Equivalences Perçues, nous trouvons les deux descripteurs  $d_1 = (\text{compétence} = \text{eau\_pluviale})$  et  $d_2 = (\text{compétence} = \text{épuration})$ , ce qui signifie que les deux descripteurs  $d_1$  et  $d_2$  appartiennent en même temps à la description d'au moins un exemple d'apprentissage, nous en déduisons que l'ensemble d'apprentissage contient au moins un exemple dont la description est incorrecte, car l'opération d'épuration ne concerne pas les eaux pluviales,

- **incohérence entre deux classes d'Equivalences Perçues** : si une classe dans laquelle nous trouvons le descripteur  $d_1 = (\text{compétence} = \text{eau\_pluviale})$  est en relation de dépendance perçue avec une autre classe dans laquelle nous trouvons également le descripteur  $d_2 = (\text{compétence} = \text{épuration})$ , ce qui signifie, tout comme dans le premier cas, que tous les exemples d'apprentissage contenant le descripteur  $d_1$  contiennent aussi le descripteur  $d_2$  dans leur description. Nous en déduisons que l'ensemble d'apprentissage contient au moins un exemple dont la description est incorrecte,

- **incohérence dans un ensemble de classes d'Equivalences Perçues** : si une classe d'Equivalence Perçue est en relation de dépendance perçue avec deux autres classes  $CE_1, CE_2$  tel que le descripteur  $d_1 = (\text{compétence} = \text{eau\_pluviale}) \in CE_1$  et le descripteur  $d_2 = (\text{compétence} = \text{épuration}) \in CE_2$ , nous en déduisons l'existence d'au moins un exemple d'apprentissage dont la description est incorrecte. En effet, cette constatation met en évidence la présence d'au moins une observation dont la description contient les deux descripteurs incohérents  $d_1$  et  $d_2$ .

Pour éliminer ces incohérences du RDP, nous devons retrouver les exemples d'apprentissage qui les ont produites. Puisque les noeuds  $n_i$  qui contiennent les descripteurs incohérents sont localisés, il est possible de déduire l'ensemble  $E_i$  des exemples d'apprentissage dont la description contient ces descripteurs. En effet, l'ensemble des exemples d'apprentissage qui contiennent une formule (atomique ou complexe) associée à un noeud  $n_i$  est constitué par la couverture de ce noeud ; et donc  $E_i = \text{couv}(n_i)$ . Mais, dans notre cas, nous devons traiter les trois cas de figure :

Dans le premier cas (**incohérence dans une même classe d'Equivalences Perçues**), un seul noeud  $n_i$  du RDP correspondant à une classe d'Equivalences Perçues est concerné par l'existence de l'incohérence détectée. La correction va donc porter sur l'ensemble des exemples d'apprentissage qui sont couverts par l'étiquette de ce noeud. Cet ensemble est constitué par :  $E_i = \text{couv}(n_i)$ .

Dans le deuxième cas (**incohérence entre deux classes d'Equivalences Perçues**), deux noeuds  $n_i$  et  $n_j$  tel que ( $\text{étiquette}(n_i) \rightarrow \text{étiquette}(n_j)$ ) sont concernés par l'existence de

l'incohérence détectée. Puisque de la relation (étiquette( $n_i$ )  $\rightarrow$  étiquette( $n_j$ )), il en ressort que ( $\text{couv}(n_i) \subseteq \text{couv}(n_j)$ ), il suffit donc de s'intéresser à l'ensemble des exemples d'apprentissage qui sont couverts par l'étiquette du noeud  $n_j$ . Cet ensemble est constitué par :  $E_j = \text{couv}(n_j)$ .

Dans le troisième cas (**incohérence dans un ensemble de classes d'Equivalences Perçues**), deux noeuds  $n_i$ ,  $n_j$  tel que (étiquette( $n_k$ )  $\rightarrow$  étiquette( $n_i$ )) et (étiquette( $n_k$ )  $\rightarrow$  étiquette( $n_j$ )) sont concernés par l'existence de l'incohérence détectée. La correction va donc porter sur l'ensemble des exemples d'apprentissage qui sont couverts à la fois par l'étiquette du noeud  $n_i$  et par l'étiquette du noeud  $n_j$ . Cet ensemble est constitué par :  $E_{i \cap j} = \text{couv}(n_i) \cap \text{couv}(n_j)$ .

Une fois que l'ensemble  $E$  des exemples d'apprentissage ayant induit l'incohérence détectée est obtenu, nous demandons aux experts de corriger la description de chacun d'eux. Comme la description de chaque exemple d'apprentissage contient nécessairement un descripteur qui permet d'identifier l'expert qui l'a fourni, les exemples d'apprentissage concernés sont corrigés par les experts qui les ont fournis. La modification des descripteurs incohérents consiste à retirer du RDP les exemples ayant une description incorrecte, et de les réintroduire après modification des descripteurs en question.

#### 5.4.2 Suivi des interventions pour valider les connaissances

Le suivi des interventions consiste à traiter le retour d'information sur les résultats d'intervention des experts sur un cas de crise pour lequel ils ont été sélectionnés. Cela permet de juger la pertinence des résultats produits par le système par rapport à des situations réelles. Comme les connaissances apprises peuvent être bruitées, les résultats produits ne sont pas sûrs et par conséquent les décisions associées sont critiquables et peuvent être remises en cause [Sallantin 91]. Mais lorsque les résultats du système sont incorrects, nous ne pouvons pas savoir a priori si cela est dû à une description incorrecte des exemples d'apprentissage, ou au contraire que c'est le système qui a induit des connaissances incorrectes. Pour cela nous intégrons dans le système un mécanisme qui permet une interaction entre le système et son environnement.

Un résultat donné par le système peut donc être remis en cause, par un utilisateur, à travers un dialogue établi entre les deux parties. Le système peut toutefois réfuter cette critique en donnant une preuve de la justesse des résultats qu'il a produit [Lakatos 84].

Pour procéder à la validation des connaissances en exploitant les rapports de suivi des interventions sur des cas réels, il est nécessaire d'établir un cadre dans lequel vont se faire les échanges de données entre le système et les différents acteurs qui interviennent dans le projet [Sallantin 91], [Beaune 92a], [Gasarch 92].

Sur la figure ci-dessous nous décrivons le schéma de communication entre le système et l'environnement externe [senoune 94b] :

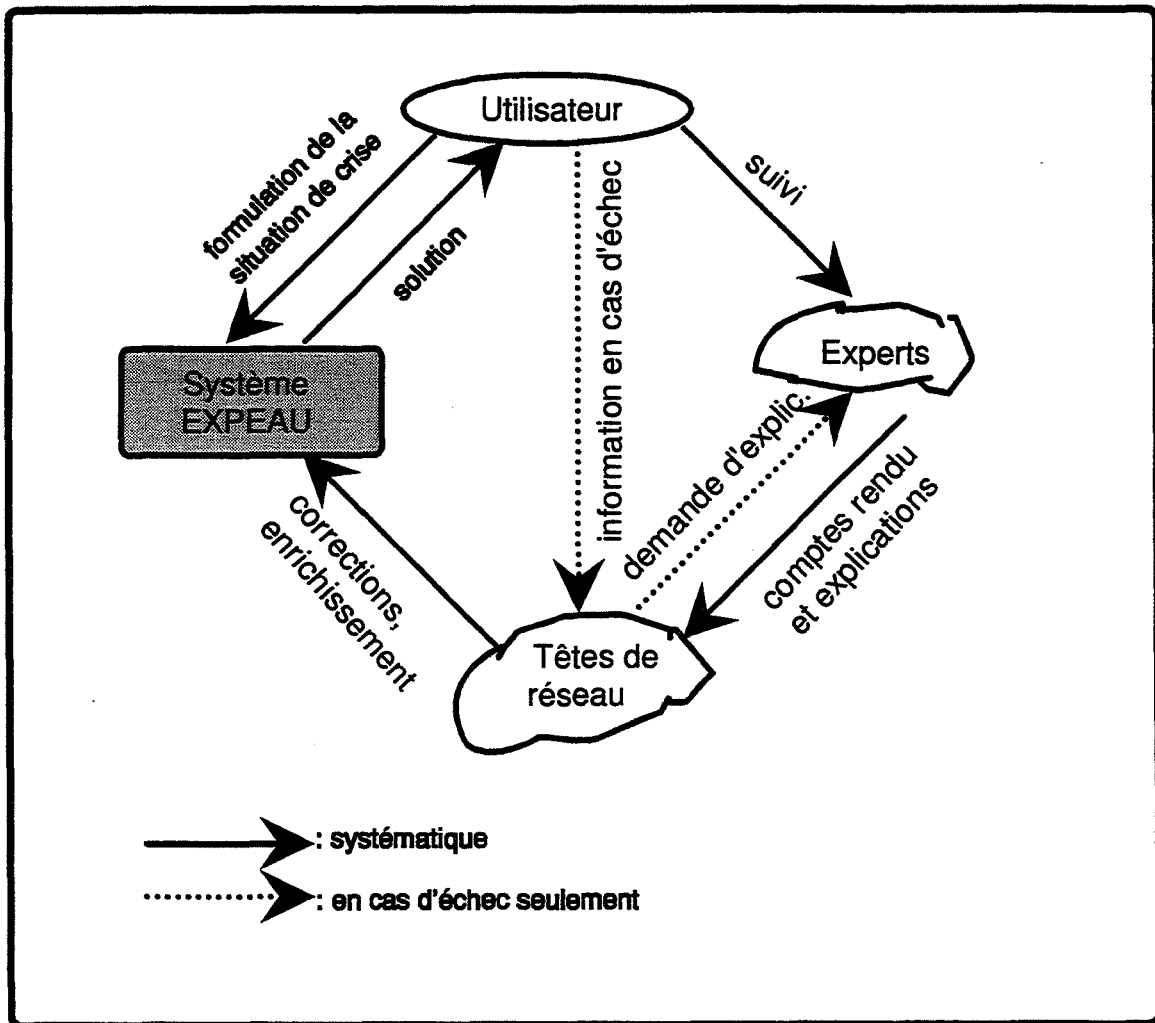


Figure 5.7 : Schéma de communication entre le système EXPEAU et son environnement

Ce schéma permet de faire des critiques au système pour améliorer incrémentalement les connaissances [Silverman 92]. Mais, un tel schéma ne peut être concrètement réalisable que si le projet d'acquisition de connaissances est organisé en accord avec ses acteurs. Dans notre cas d'application, des "têtes de réseau" participant au comité de pilotage du projet de développement de l'outil ont été définies. Une tête de réseau est une personne qui est reconnue experte dans le domaine d'application et qui représente un ensemble d'autres experts du même domaine. Les échanges d'information entre le système et son environnement se font de la manière suivante :

- l'utilisateur décrit une situation de crise en utilisant les éléments de connaissances appartenant au langage d'interrogation. Le système analyse le problème et sélectionne les experts appropriés. Il affiche un message indiquant l'échec de la recherche dans le cas où aucun expert ne correspond pour une intervention sur le problème décrit ou sur des problèmes similaires,
- en cas d'échec (aucun expert trouvé) l'utilisateur avise les têtes de réseau. Ces derniers tentent, en s'adressant aux experts, de trouver ce qui manque comme connaissances dans le système pour combler cet échec,
- dans le cas où des experts sont proposés, l'utilisateur se charge du suivi des interventions (les experts sont-ils intervenus ?, les experts ont-ils réussi leur intervention ?, ...) pour pouvoir actualiser le système,
- les experts établissent des comptes-rendus de leur intervention et les communiquent aux têtes de réseau,
- les experts donnent des explications sur demande des têtes de réseau à la suite d'un échec de recherche ou d'intervention,
- les têtes de réseau enrichissent le système en analysant les comptes rendus et les explications fournis par les experts,

## 5.5 CONCLUSION

Dans ce chapitre nous avons décrit les procédures d'utilisation des connaissances contenues dans le RDP ainsi que celles spécifiées dans l'environnement des observations. Un langage d'interrogation est construit pour permettre à des utilisateurs de formuler des requêtes correctes au système EXPEAU dans le cas d'une exploitation en déduction des connaissances.

Nous avons aussi décrit les procédures permettant de détecter des incohérences dans les connaissances contenues dans le RDP construit et nous avons montré comment ces incohérences peuvent être corrigées par les experts.

Le système EXPEAU est réalisé en Common LISP sur une station de travail SUN. Les principaux modules qui le composent sont :

- spécification de l'environnement d'observation,
- introduction des exemples d'apprentissage,
- construction du graphe de connaissances,
- utilisation des connaissances,
- validation des connaissances.





## *Conclusion et perspectives*

Ce travail se place dans le contexte général des Systèmes à Base de Connaissances. Nous nous sommes intéressés plus particulièrement au problème de l'acquisition de connaissances pour construire une Base de Connaissances dans le domaine de l'eau.

L'approche "Systèmes à Base de Connaissances" a été choisie pour permettre une représentation et une utilisation des connaissances adaptées à la description de situations de crise et à la sélection d'experts pouvant résoudre des problèmes engendrés par de telles situations.

Etant donné que le domaine de l'eau est caractérisé par des connaissances empiriques, évolutives et multidisciplinaires, le transfert de connaissances depuis les experts vers le système informatique a nécessité une technique d'apprentissage automatique. Ainsi, nous ne demandons pas aux experts de délivrer explicitement toutes leurs connaissances, mais de fournir des rapports sur les interventions qu'ils ont effectués sur des cas réels de crise. En effet, comme les experts dans le domaine de l'eau appartiennent à différentes disciplines et possèdent une connaissance empirique (leur savoir est construit dans la pratique au fil des interventions qu'ils effectuent sur des situations de crise), certains liens interdisciplinaires ne peuvent être découverts qu'en examinant les interventions réelles d'experts.

En plus du savoir faire associé aux interventions qu'ils ont effectuées, les experts possèdent une importante connaissance descriptive (concepts, relations entre les concepts,

taxinomies des concepts, langage de description des exemples d'apprentissage) et des connaissances stratégiques. Les connaissances descriptives du domaine permettent de représenter la sémantique des connaissances et les relations qui existent entre elles. Les connaissances stratégiques représentent les métaconnaissances, utilisées pour reformuler les descriptions de requêtes lors de la recherche de solutions par le système.

Pour recueillir ces deux types de connaissances (connaissances descriptives, connaissances stratégiques), nous avons adopté une méthodologie d'acquisition de connaissances qui s'appuie sur la philosophie de "KADS". Nous avons ainsi construit un langage de description adéquat pour permettre aux experts de décrire facilement les cas d'interventions qu'ils ont effectué. Le processus de construction de ce langage a nécessité l'analyse de documents sur le domaine et des rapports d'interventions de différents experts sur des cas de crise.

Après la phase d'acquisition et de construction de la Base de Connaissances, vient une autre phase d'utilisation et de validation des connaissances.

L'exploitation des connaissances acquises se fait de deux manières. D'une part, le graphe de connaissances construit peut être utilisé, de façon directe et interactive ; d'autre part, les connaissances acquises peuvent être utilisées en déduction. Ainsi, le système EXPEAU peut même être utilisé par des utilisateurs qui ne sont pas familiarisés aux structures de données ou qui ne savent pas exactement quelles sont les questions à poser.

La validation des connaissances acquises se fait aussi de deux manières différentes :

- les experts du domaine peuvent travailler sur le Réseau de Dépendances Perçues construit pour détecter des incohérences et des incomplétudes dans les connaissances. Les experts corrigent les connaissances erronées en modifiant les exemples d'apprentissage qui ont induit ces erreurs,

- le suivi des interventions qui consiste à traiter le retour d'information sur les résultats obtenus par les experts à propos d'une intervention sur un cas de crise, permet d'évaluer les résultats fournis par le système et de valider les connaissances en cas de nécessité.

En conclusion, nous pouvons affirmer que l'apport de cette thèse réside dans l'intégration d'une méthodologie d'acquisition de connaissances et d'une technique d'apprentissage pour la construction incrémentale d'une Base de Connaissances pour les situations de crise. Cette Base de Connaissances n'est pas utilisée dans le but de trouver des solutions à des problèmes donnés, mais pour sélectionner des experts pouvant résoudre ces problèmes. De plus, elle permet de révéler des relations existantes entre les connaissances du domaine qui ne sont pas évidentes a priori.

Un autre apport important de cette thèse consiste en l'utilisation interactive du Réseau de Dépendances Perçues construit pour la détection et la correction d'erreurs dans les connaissances qu'il contient.

Toutefois, la phase de validation du système peut être améliorée dans le but d'arriver à :

- déduire les sous domaines (les disciplines) qui ne sont pas couverts par les exemples d'apprentissage fournis au système,
- changer les stratégies de reformulation de requêtes en examinant les résultats obtenus par l'utilisation des stratégies existantes.



## *Références bibliographiques*

**[Ackley 85]** Ackley, D.H. Hinton, G.E. and Sejnowski, T.J. A learning algorithm for Boltzmann machines. *Cognitive Science*, 1985, Vol. 9, p.147-169.

**[Agapeyeff 88]** d'Agapeyeff, A. Practical elicitation for business applications. *Proceedings of the 8th International Workshop Expert Systems and Their Applications*, Avignon, 30 mai-3 juin 1988, Nanterre : EC2, 1988, Vol. 1, p. 625-642.

**[Albert 90]** Albert, P et Vogel, C. KOD-STATION : un environnement intégré pour le génie cognitif. *Génie Logiciel & Systèmes Experts*, 1990, N° 19, p. 28-30.

**[Amarel 68]** Amarel, S. On representing of Problems of Reasoning About Actions. *Machine Intelligence*, 1968, Vol. 3, p. 131-171.

**[Aussenac 89]** Aussenac, N. Conception d'une méthodologie et d'un outil d'acquisition de connaissances expertes. Thèse de doctorat : Université Paul Sabatier de Toulouse, 1989. 253 p.

**[Ayel 86]** Ayel, M. L'incohérence dans les bases de connaissances. *Colloque International d'Intelligence Artificielle*, Marseille, 1-5 décembre, 1986. p. 293-212.

**[Ayel 90]** Ayel, M. et Rousset, M.C. La cohérence dans les bases de connaissances. Toulouse : CEPADUES-EDITIONS, 1990. 106 p.

**[Bareiss 90]** Bareiss, R-R. Porter, B-W and Wier, C-C. Protos : An Exemplar-Based Learning Apprentice. *MACHINE LEARNING : An Artificial Intelligence Approach*. Edited by Y. Kodratoff. San Mateo (CA) : Morgan Kaufmann Publishers, 1990, Vol. III, p. 112-139.

**[Barr 81]** Barr, A et Feigenbam, E.A. *The Handbook of Artificial Intelligence*. Vol. 1. Los Altos (CA) : William Kaufmann, Inc, 1981. 409 p.

**[Beaune 92a]** Beaune, Ph. Acquisition interactive de la connaissance par un système télématique dans le domaine de l'assainissement de l'eau. Thèse de Doctorat : Université de Montpellier II, Sciences et techniques du Languedoc, 1992. 215 p.

**[Beaune 92b]** Beaune, Ph et Senoune, R. Acquisition et distribution de la connaissance dans le domaine de l'Environnement. Colloque Méthodes et Outils d'Aide à la Décision, Béjaia (Algérie), 15-17 Décembre, 1992. p. 427-433.

**[Beauvieux 88]** Beauvieux, A. Contrôler la cohérence d'une base de connaissances. Actes des huitièmes Journées Internationales : les Systèmes Experts et leurs applications, Avignon, 30 mai - 3 juin, 1988. p. 38-45.

**[Benkirane 90]** Benkirane, M. Dumas, Ph. Houriez, B et Millot, P. Aspects de la psychologie cognitive dans l'acquisition des connaissances. Convention IA 90 : Actes de la deuxième conférence européenne sur les techniques et les applications de l'Intelligence Artificielle en milieu industriel et de service, Paris, 15-18 janvier, 1990, Paris : Hermes, 1990. p. 607-621.

**[Bollinger 86]** Bollinger, T. Généralisation en apprentissage à partir d'exemples. Thèse de Doctorat : Université de Paris-Sud centre d'Orsay, 1986. 184 p.

**[Boose 86]** Boose, J-H. ETS : A System for the Transfer of Human Expertise. Knowledge Based Problem Solving. Edited by J. S. Kowalik. Englewood Cliff (New Jersey) : Prentice-Hall, 1986. p. 68-111.

**[Bouchet 89]** Bouchet, C. Brunet, E. and Anjewierden, A. SHELLEY : AN INTEGRATED WORKBENCH FOR KBS DEVELOPMENT. In : Proceedings of the 9th International Workshop Expert Systems and Their Applications, Avignon, 29 mai-2 juin 1989, Nanterre : EC2, 1989, Vol. 1, p. 303-315.

**[Brenker 85]** Brenker, J and Wielinga, B. KADS : Structured Knowledge Acquisition for Expert Systems. In : Proceedings of the 5th International Workshop Expert Systems and Their Applications, Avignon, mai 1985, Nanterre : EC2, 1985. p. 887-900.

**[Brunet 90]** Brunet, E. KADS & Merise : Vers une unification du génie cognitif et du génie logiciel. Génie Logiciel & Systèmes Experts, 1990, N° 19, p. 10-27.

**[Brunet 91]** Brunet, E. KADS : Méthode d'ingénierie de la connaissance. Génie Logiciel & Systèmes Experts, 1991, N°23, p. 24-34.

**[Burstein 86]** Burstein, M.H. Concept Formation by Incremental Analogical Reasoning and Debugging. Machine Learning : An Artificial Intelligence Approach. Edited by R. S. Michalski. Los Altos (CA) : Morgan Kaufmann Publishers, 1986, Vol. II, p. 351-369.

**[Carbonell 83]** Carbonell, J.G. Learning by analogy : Formulating and Generalizing Plans from Past Experience. Machine Learning : An Artificial Intelligence Approach. Edited by R. S. Michalski. Los Altos (CA) : Morgan Kaufmann Publishers, 1986, Vol. II, p. 137-162.

**[Carbonell 86]** Carbonell, J.G. Derivational Analogy : A Theory of Reconstructive Problem Solving and Expertise Acquisition. Machine Learning : An Artificial Intelligence Approach. Edited by R. S. Michalski. Los Altos (CA) : Morgan Kaufmann Publishers, 1986, Vol. II, p. 371-392.

**[Causse 92]** Causse, K et Canamero, D. La généralité en acquisition des connaissances. Actes des 3ièmes Journées d'Acquisition de Connaissances du PRC-IA, Dourdan, 14-16 Avril, 1992. p. 161-175.

**[Cavarero 86]** Cavarero, J-L et Biondi, J. Un modèle universel pour la représentation des phénomènes d'apprentissage. CIIAM 86 : Actes du 2<sup>e</sup> Colloque International d'Intelligence Artificielle, Marseille, 1-5 décembre, 1986. Paris : Hermes, 1986. p. 427-442.

**[Charniak 85]** Charniak, E. and McDermott, D. Introduction to Artificial Intelligence. Menlo Park, CA : Addison-Wesley Publishing Company, 1985. 701 p.

**[Chouraqui 85]** Chouraqui, E. Farreny, H. Kayser, D et Prade, H. Modélisation du raisonnement et de la connaissance. Technique et Science Informatiques, 1985, N° 4, p. 391-396.

**[Chouraqui 86]** Chouraqui, E. Le raisonnement analogique : sa problématique, ses principes. Journées Nationales sur l'Intelligence Artificielle, Aix-les-Bains, 20-21 Novembre, 1986. Toulouse : CEPADUES-EDITIONS, 1986. p. 107-117.



**[Clark 90]** Clark, P. A comparison of Rule and Exemplar-Based Learning Systems. *Machine Learning, Meta-Reasoning and Logics*. edited by P.B. Brazdil and K. Konolige. Boston : Kluwer Academic Publishers, 1990. p. 159-186.

**[Compagnon 91]** Compagnon, F. Gaudinat, B. Jouis, C. Rousseau, J-M et Tora, C. *METHODAC : une méthodologie pour l'acquisition et la modélisation des connaissances*. 8ème Congrès RFIA, Lyon, novembre 25-29, 1991, Paris : AFCET, 1991. p. 35-42.

**[Curtis 92]** Curtis, B. Marc, I.K. and Over, J. Process Modeling. *Communication of the ACM*, 1992, Vol. 35, Number 9, p. 75-89.

**[Desfray 92]** Desfray, P. *Ingénierie des objets : approche classe-relation, application à C++*. Paris : Masson, 1992. 230 p.

**[Dieng 90]** Dieng, R. *Méthodes et outils d'acquisition des connaissances*. Actes des journées ERGO-IA'90, Biarritz, 1990. p. 245-271.

**[Dietterich 84]** Dietterich, T-G and Michalski, R-S. A Comparative Review of Selected Methods for Learning from Examples. *Machine Learning : An Artificial Intelligence Approach*. Edited by R. S. Michalski. Palo Alto (CA) : Morgan Kaufmann Publishers, 1984, p. 41-81.

**[Dominé 88]** Dominé, C.H. *Techniques de l'Intelligence Artificielle : Un guide structuré*. Paris : Bordas, 1988. 310 p.

**[Dorrrough 86]** Dorrrough, D. C. ANALOGY : A PRIORI REQUIREMENTS FOR DEEP EXPERT SYSTEM APPLICATIONS. *Knowledge Based Problem Solving*. Edited by J. S. Kowalik. Englewood Cliff (New Jersey) : Prentice-Hall, 1986. p. 112-165.

**[Eshelman 86]** Eshelman, L. and McDermott. MOLE : A Knowledge Acquisition Tool That uses its Head. In : *proceedings of the National Conference on Artificial Intelligence : AAAI*, Philadelphia, PA, August 1986. p. 950-955.

**[Feigenbaum 89]** Feigenbaum, E-A. Barr, A et Cohen, P-R. *The Handbook of Artificial Intelligence*. Menlo Park (CA) : Addison-Wesley Company, Inc, 1989, Vol. IV. 699 p.

**[Feng 93]** Feng, X. and Weber, T.A. Knowledge Acquisition Advisor (KA<sup>2</sup>) : An Interactive Knowledge-Acquisition Tool for Expert System Development. *Engineering Applications of Artificial Intelligence*, 1993, Vol. 6, Number 6, p. 507-518.

[Fisher 87] Fisher, D-H. Conceptual Clustering, Learning from Examples, and Inferences. Proceedings of the Fourth International Workshop on MACHINE LEARNING. University of California, Irvine, June 22-25, 1987. Los Altos (CA) : Morgan Kaufmann Publishers, Inc, 1987. p. 38-49.

[Forgues 91] Forgues, B. La décision en situation de crise. Revue Française de Gestion, 1991, Novembre-Décembre, p. 39-45.

[Ganascia 87] Ganascia, J.G. AGAPE et CHARADE : deux techniques d'apprentissage symbolique appliquées à la construction de bases de connaissances. Thèse d'état : Université de PARIS-SUD, CENTRE D'ORSAY, 1987. 347 p.

[Ganascia 88] Ganascia, J.G. Charade : une sémantique cognitive pour les heuristiques d'apprentissage. Actes des huitièmes Journées Internationales : les Systèmes Experts et leurs applications, Avignon, 30 mai - 3 juin, 1988. p. 567-585.

[Gasarch 92] Gasarch, W. I. Smith, C. H. Learning via Queries. Journal of the Association for Computing Machinery, 1992, Vol. 39 Number 3, p. 649-674.

[Grundstein 88] Grundstein, M. de Bonnières, P et Pra, S. Les systèmes à base de connaissances : Systèmes experts pour l'entreprise. Paris : AFNOR, 1988. 261 p.

[Harmon 93] CASE-BASED REASONING. Edited by P. Harmon. N° 5. Airlington (MA) : Cutter Information Corp., 1993. 28 p.

[Hart 88] Hart, A. Acquisition de savoir pour les Systèmes Experts. Collection Sciences Cognitives. Paris : Masson, 1988. 142 p.

[Hayes-Roth 83] Hayes-Roth, F. Waterman, D. A. Lenat, D. Building Expert Systems. Reading (Massachusetts) : Addison-Wesley Publishing Company, Inc., 1983. 444 p.

[Hopfield 82] Hopfield, J.J. Neural Networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, USA, 1982, Vol. 79, p. 2554-2558.

[Jaulent 92] Jaulent, P. Génie Logiciel : les méthodes. Paris : Armand Colin, 1992. 294 p.

[**Kauffmann 85**] Kauffmann, H. Un outil d'analyse pour les langages de représentation de connaissances. 5ème Congrès RFIA, Grenoble, novembre 27-29, 1985, Paris : AFCET, 1985, Tome II. p. 1099-1112.

[**Kervern 91**] Kervern, G-Y. Rubise, P. L'archipel du danger : Introduction aux cyndyniques. Paris : Economica, 1991. 444 p.

[**Kietz 88**] Kietz, J.W. Incremental and Reversible Acquisition of Taxinomies. Berlin : Technical University Beerlin, 1988. 11 p. KIT-Report 66.

[**Kodratoff 86**] Kodratoff, Y. Leçons d'Apprentissage Symbolique Automatique. Toulouse : CEPADUES-EDITIONS, 1986. 191 p.

[**Kodratoff 87**] Kodratoff, Y. Science des explications et sciences des nombres. Journées "Symbolique-Numérique" pour l'Apprentissage de Connaissances à partir de Données, Paris, 8-9 décembre, 1987, Paris : LISE-CEREMADE, 1987. p. 7-20.

[**Kodratoff 91**] Kodratoff, Y et Diday, Y. Induction Symbolique et Numérique à partir de Données. Toulouse : CEPADUES-EDITIONS, 1991. 459 p.

[**Kodratoff 93**] Kodratoff, Y. Apprentissage de la Connaissance Experte par amélioration des explications fournies par le système. APPRENTISSAGE SYMBOLIQUE : une approche de l'intelligence artificielle. Edité par Y. Kodratoff. Toulouse : CEPADUES EDITIONS, 1993. Tome II, p. 355-387.

[**Kolodner 87**] Kolodner, J-L. Extending Problem Solver Capabilities Through Case-Based Inference. Proceedings of the Fourth International Workshop on MACHINE LEARNING, University of California, Irvine, June 22-27, 1987. Los Altos (CA) : Morgan Kaufmann Publishers, Inc. p. 167-178.

[**Krisch 93**] Krisch, Ph. Maesano, L et Rabaux, E. Open KADS : méthode & atelier pour la modélisation des connaissances. Génie Logiciel & Systèmes Experts, 1993, N° 31, p. 36-40.

[**Lagadec 86**] Lagadec, P. Stratégies de communication en situation de crise. Annales des Mines, 1986, Octobre-Novembre, p. 125-129.

[**Lagadec 91**] Lagadec, P. La gestion des crises : Outils de reflexion à l'usage des décideurs. Paris : Mc GRAW-HILL, 1991. 326 p.

**[Lakatos 84]** Lakatos, I. PREUVES ET REFUTATIONS : Essai sur la logique mathématique. Traduction de Balacheff, N & Laborde, J-M. Paris : HERMANN, 1984. 218 p.

**[Langley 84]** Langley, P. Bradshaw, G-L. Simon, A-H. Rediscovering chemistry with the BACON system. Machine Learning : An Artificial Intelligence Approach. Edited by R. S. Michalski. Palo Alto (CA) : Morgan Kaufmann Publishers, 1984, p. 307-329.

**[Langley 86]** Langley, P. Zytkow, J-M. Simon, A-H. Bradshaw, G-L. Search of regularities : four aspects in scientific discovery. Machine Learning : An Artificial Intelligence Approach. Edited by R. S. Michalski. Los Altos (CA) : Morgan Kaufmann Publishers, 1986, Vol. II, p. 425-469.

**[Lebowitz 86]** Lebowitz, M. Concept learning in a rich input domain : Generalization-based Memory. Machine Learning : An Artificial Intelligence Approach. Edited by R. S. Michalski. Los Altos (CA) : Morgan Kaufmann Publishers, 1986, Vol. II, p. 193-214.

**[Lebowitz 87]** Lebowitz, M. Experiments with Incremental Concept Formation : UNIMEM. Artificial Intelligence, 1987, Vol. 2, p. 103-138.

**[Lenat 82]** Lenat, D. The nature of heuristics. Artificial Intelligence, 1982, Vol. 19, N° 2, p. 184-249.

**[Levine 89]** Levine, P. et Pomerol, J.C. Systèmes interactifs d'aide à la décision et systèmes experts. Paris : Hermes, 1989. 335 p.

**[Lewis 94]** Lewis, M. Knowledge-Based Systems : Do they have a future ? Intelligent Tutoring Media, 1994, Vol. 5, Number 1, p. 31-34.

**[Lissandre 90]** Lissandre, M. Maîtriser SADT. Paris : Armand Colin, 1990. 219 p.

**[Manago 87]** Manago, M.V. and Kodratoff, Y. Noise and Knowledge Acquisition. Tenth International Joint Conference on Artificial Intelligence, Milan, 1987. p. 348-354.

**[Manago 89]** Manago, M. Contruyt, N. KATE : Un système d'apprentissage avec objets. Actes des quatrièmes journées Françaises de l'apprentissage, St-Malo, 22-24 Mai, 1989. Rennes : IRISA/INRIA, 1989. p. 121-135.

[**Marquis 89**] Marquis, M. Une étude formelle de la recherche d'explications fondée sur la logique. Actes des quatrièmes journées Françaises de l'apprentissage, St-Malo 22-24 Mai, 1989. Rennes : IRISA/INRIA, 1989. p. 137-151.

[**Michalski 86a**] Michalski, R-S. A Theory and Methodology of Inductive Learning. MACHINE LEARNING : An Artificial Intelligence Approach. Edited by R. S. Michalski. Los Altos (CA) : Morgan Kaufmann Publishers, 1986, Vol. I, p. 83-134.

[**Michalski 86b**] Michalski, R-S. UNDERSTANDING THE NATURE OF LEARNING : Issues and Research Direction. MACHINE LEARNING : An Artificial Intelligence Approach. Edited by R. S. Michalski. Los Altos (CA) : Morgan Kaufmann Publishers, 1986, Vol. II, p. 3-25.

[**Michalski 90**] Michalski, R-S. and Kodratoff, Y. Research in Machine Learning : Recent Progress, Classification of Methods and Future Directions. MACHINE LEARNING : An Artificial Intelligence Approach. Edited by Y. Kodratoff. San Mateo (CA) : Morgan Kaufmann Publishers, 1990, Vol. III, p. 3-30.

[**Mineau 90**] Mineau, G. Godin, R. Gecsei, J. La classification Symbolique : Une approche non-subjective. 5èmes Journées Françaises de l'Apprentissage, Lannion, 25-26 Avril, 1990. p. 169-189.

[**Mitchell 82**] Mitchell, T-M. Generalization as Search. Artificial Intelligence, 1982, Vol. 18, p. 203-226.

[**Mitchell 83**] Mitchell, T.M. Learning and Problem Solving. Proceedings of International Joint Conference on Artificial Intelligence, Karlsruhe, 1983. p. 1139-1151.

[**Morik 87**] Morik, K. Acquiring domain models. International Journal of Man-Machine Studies, 1987, Vol. 26, p. 93-104.

[**Neubert 93**] Neubert, S. Model Construction in MIKE (Model Based and Incremental Knowledge Engineering). Proceedings of 7th European Workshop, EKAW'93, Toulouse and Caylus, France, 1993, Berlin : Springer-Verlag, 1993. p. 200-219.

[**Newell 82**] Newell, A. The knowledge level. Artificial Intelligence Magazine, 1982, N° 18, p. 87-127.

- [Pitrat 90] Pitrat, J. Métaconnaissance : futur de l'intelligence artificielle. Paris : Hermès, 1990. 401 p.
- [Poisvert 93] Poisvert, R. Comprendre pour entreprendre : la méthode METADOC. Génie Logiciel & Systèmes Experts, 1993, N° 31, p. 51-57.
- [Pottier 86] Poittier, L. Généralisation d'exemples et de contre-exemples en calcul propositionnel. CIIAM 86 : Actes du 2<sup>e</sup> Colloque International d'Intelligence Artificielle, Marseille, 1-5 décembre, 1986. Paris : Hermes, 1986. p. 462-470.
- [Quillian 68] Quillian, M.R. Semantic memory. Semantic Information Processing. MINSKY, M. (ed). Cambridge : MIT press, 1968, p. 227-270.
- [Quinlan 86] Quinlan, J-R. Induction of Decision Trees. Machine Learning, 1986, Vol. 1, p. 81-106.
- [Quinlan 87] Quinlan, J-R. Decision Trees as Probabilistic Classifiers. In proceedings of 4th International Workshop on Machine Learning, California, 1987. Los Altos (CA) : Morgan Kaufmann Publishers, 1987. p. 31-37.
- [Quinlan 90] Quinlan, J-R. Probabilistic Decision Trees. MACHINE LEARNING : An Artificial Intelligence Approach. Edited by Y. Kodratoff. San Mateo (CA) : Morgan Kaufmann Publishers, 1990, Vol. III, p. 140-142.
- [Rasmussen 83] Rasmussen, J. Skills, rules and Knowledge : signals, signs, and symbols, and other distinctions in human performance models. IEEE Transactions on System, Man and Cybernetics, 1983, vol SMC-13, N° 3. p. 257-266.
- [Reitz 92] Reitz, Ph. Contribution à l'étude des environnements d'apprentissage : Conception, spécifications et prototypage. Thèse de Doctorat : Université Montpellier II - Sciences et Techniques du Languedoc, 1992. 216 p.
- [Risler 91] Risler, J.J., Talbot, A. et Vançon, J.J. Mise au point d'un système d'aide à la décision en cas d'alerte à la pollution des eaux. Congrès International : Innovation, Progrès Industriel et Environnement. Strasbourg , juin 4-6, 1991. p. 1-10.

[Rissland 93] Rissland, E.L. Skalak, D.B. Friedman, M.T. Case Retrieval through Multiple Indexing and Heuristic Search. Proceedings of International Joint Conference on Artificial Intelligence, Chambéry, 1993. Vol. 2. p. 902-908.

[Royer 90] Royer, J-C. MANDRIN : un système d'apprentissage pour l'aide au réglage d'un instrument. Thèse de Doctorat : Institut National Polytechnique de Grenoble, 1990. 156 p.

[Sabah 88] Sabah, G. L'intelligence artificielle et le langage : représentation des connaissances. Paris : Hermès. 1988. 352 p.

[Sabah 89] Sabah, G. L'intelligence artificielle et le langage : processus de compréhension. Paris : Hermès. 1989. 411 p.

[Sainte Marie 89] De Sainte Marie, C. Dépendance Perçue : une introduction. 7ème Congrès de Reconnaissance des Formes et Intelligence Artificielle, Paris, 1989. p. 1445-1459.

[Sainte Marie 90a] De Sainte Marie, C. Apprentissage et acquisition de la connaissance. Actes de la Convention IA, Paris, 1990. Paris : HERMES, 1990. p. 521-535.

[Sainte Marie 90b] De Sainte Marie, C. De l'observation à la connaissance empirique (en passant par la Dépendance Perçue). 5èmes Journées Françaises de l'Apprentissage, Lannion, 1990. p. 251-272.

[Sallantin 91] Sallantin, J. Szczeciniarz, J-J. Barboux, C. Lagrange, M-S et Renaud, M. Théories semi-empiriques : conceptualisation et illustrations. Revue d'intelligence artificielle, 1991, Vol. 5 - n° 1, p. 9-67.

[Samuel 59] Samuel, A. L. Some studies in machine learning using the game of checkers. IBM Journal R&D, 1959, Vol. 3, p. 211-229.

[Samuel 67] Samuel, A. L. Some studies in machine learning using the game of checkers II-recent progress. IBM Journal R&D, 1967, Vol. 11. p. 601-617.

[Senoune 92] Senoune, R. Graillot, D. et Chaix, Ph. Apport de l'intelligence artificielle pour la recherche d'experts en situation de crise, Hydroscopie, 1992, Juillet, n° 15. p. 2-3.

[Senoune 94a] Senoune, R. and Graillot, D. Use of Perceived Dependency to Build a Knowledge Base in Water Domain. 1994 Fourth Annual Mohawk Valley IEEE Dual Use

Technologies and Applications Conference, Utica/Rome (New York), May 23-26, 1994, Vol. II, p. 375-380.

[**Senoune 94b**] Senoune, R. and Grailot, D. MEDEXP'EAU : Knowledge-Based System for Crisis Situations in Water Domain. The Third Turkish Symposium on Artificial Intelligence and Neural Networks, Ankara, Turkey, June 22-24, 1994. p. 11-18.

[**Senoune 95**] Senoune, R. and Grailot, D. Learning From Examples to Build Information System. Advances in Computer Cybernetics. Edited by G. E. Lasker. Ontario : IIAS, 1995. Vol. II, p. 14-18.

[**Seroussi 89**] Seroussi, B et Boisvieux, J-F. Apprentissage et raisonnement par analogie dans le système ALEX. Actes des quatrièmes journées Françaises de l'apprentissage, St-Malo, 22-24 Mai 1989. p. 169-181.

[**Silverman 92**] Silverman, B-G. Building a Better Critic : Recent Empirical Results. IEEE Expert, April 1992, N° 4, p. 18-24.

[**Smith 92**] Smith, M-F-S. Donald, J-H. RULE INDUCTION : MACHINE LEARNING TECHNIQUES FOR DATA ANALYSIS, CLASSIFICATION AND KNOWLEDGE ELICITATION. Surrey, England : ERA Technology Limited, 1992, 156 p. Report n° 92-0581.

[**Thayse 91**] Thayse, A. Delastre, Ph. Hagelstein, J. Louis, J. Rifaut, A. Vauclair, M. Dubois, E. Lamsweerde, A-V et Van Der Linden, F. Approche logique de l'intelligence artificielle, vol. 4 : De l'apprentissage artificiel aux frontières de l'IA. Paris : Dunod, 1991. 358 p.

[**Tiberghien 94**] Tiberghien, G. Psychologie de la mémoire humaine. Sciences humaines, 1994, N° 43, p. 25-28.

[**Tong 92**] Tong, X. CARMEN : Une approche pour combler le fossé entre la modélisation de l'expertise et l'implémentation du SBC. Actes des 3èmes Journées d'Acquisition des Connaissances du PRC-IA, Dourdan, 14-16 Avril, 1992. p. 135-147.

[**Velichkovsky 92**] Velichkovsky, B-M. Organisation fonctionnelle de l'intelligence humaine. Revue internationale de systémique, 1992, Vol. 6, N° 3, p. 241-262.



[Vere 75] Vere, S.A. Induction of Concepts in the Predicate Calculus. Proceedings International Joint Conference on Artificial Intelligence, Tbilisi, 1975. p. 281-287.

[Vere 80] Vere, S.A. Multilevel Counterfactuals for Generalizations of Relational Concepts and Productions". Artificial Intelligence, 1980, vol. 14, p. 139-164.

[Vicat 93] Vicat, C. Busac, A. and Ganascia, J.G. CERISE : A CYCLIC APPROACH FOR KNOWLEDGE ACQUISITION. Proceedings of 7th European Workshop, EKAW'93, Toulouse and Caylus, France, 1993, Berlin : Springer-Verlag, 1993. p. 237-255.

[Vogel 88] Vogel, C. Génie Cognitif. Paris : Masson, 1988. 196 p.

[Vogel 89] Vogel, C. Comment qualifier un système à base de connaissances ?. Génie Logiciel & Systèmes Experts, 1989, N°16, p. 4-9.

**FOLIO ADMINISTRATIF**

**THESE SOUTENUE DEVANT L'INSTITUT DES SCIENCES APPLIQUEES DE LYON**

NOM : <b>SENOUNE</b> (avec précision du nom de jeune fille, le cas échéant)	DATE de SOUTENANCE <b>19 juin 1995</b>
Prénoms : <b>REDOUANE</b>	
TITRE : <b>Acquisition de Connaissances et Apprentissage Automatique : Contribution pour le développement incrémental d'un système à Base de Connaissances pour les situations de crise - application au domaine de l'eau -</b>	
NATURE : <b>Doctorat</b>	Numéro d'ordre : <b>95ISAL0044</b>
Formation doctorale : <b>Ingénierie Informatique</b>	
Cote B.I.U. -Lyon : <b>T 50/210/19 /.</b> et bis	CLASSE :
RESUME : <p>Cette thèse a pour objectif la réalisation d'un Système à Base de Connaissances pour les situations de crise. A travers le développement de ce système, deux principaux axes de recherche ont été entrepris : l'acquisition et la validation de connaissances.</p> <p>Pour l'étape d'acquisition de connaissances, nous avons intégré une méthodologie d'acquisition de connaissances et une technique d'apprentissage automatique. Dans un premier temps, la méthodologie d'acquisition de connaissances nous a permis de recueillir l'ensemble des connaissances descriptives et stratégiques du domaine, et de construire un langage de description des exemples d'apprentissage. Une technique d'apprentissage est ensuite utilisée pour construire incrémentalement un graphe de connaissances en utilisant des cas d'interventions sur des situations de crise fournis par les experts du domaine.</p> <p>Pour la phase d'exploitation du système, nous avons proposé deux procédures différentes. La première procédure consiste en l'utilisation interactive du graphe de connaissances construit et la deuxième procédure consiste en l'utilisation en déduction des connaissances contenues dans ce graphe.</p> <p>L'approche proposée pour la validation des connaissances s'appuie sur l'utilisation interactive du graphe de connaissances construit et sur un suivi des interventions des experts sur des cas de crise.</p>	
MOTS-CLES : <b>Base Connaissance, Acquisition, Apprentissage, Aide Décision, Validation, Crise</b>	
Laboratoire (s) de recherches : <b>Centre SIMADE, Ecole des Mines de Saint-Etienne</b>	
Directeur de thèse : <b>Monsieur Albert MATHON, Professeur</b>	
Président de jury :	
Composition du jury : <b>MM. J. KOULOUMDJIAN, A. FATHOLAHZADEH, J. QUINQUETON, E. CHOURAQUI, A. HOCINE, C. DE SAINTE MARIE, A. MATHON, D. GRILLOT</b>	





