



HAL
open science

Acquisition interactive de la connaissance par un système télématique dans le domaine de l'assainissement de l'eau

Philippe Beaune

► To cite this version:

Philippe Beaune. Acquisition interactive de la connaissance par un système télématique dans le domaine de l'assainissement de l'eau. Modélisation et simulation. Université Montpellier II - Sciences et Techniques du Languedoc, 1992. Français. NNT : 1992MON20004 . tel-00838502

HAL Id: tel-00838502

<https://theses.hal.science/tel-00838502>

Submitted on 25 Jun 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.

ACADEMIE DE MONTPELLIER

UNIVERSITE MONTPELLIER II

— SCIENCES ET TECHNIQUES DU LANGUEDOC —

THESE

présentée à l'UNIVERSITE MONTPELLIER II - SCIENCES ET TECHNIQUES DU LANGUEDOC
pour obtenir le DIPLOME DE DOCTORAT

SPECIALITE : informatique

Formation Doctorale : informatique

ACQUISITION INTERACTIVE
DE LA CONNAISSANCE
par un système télématique
dans le domaine de l'assainissement de l'eau

par

PHILIPPE BEAUNE

Soutenue le 16 janvier 1992 devant le jury composé de :

M. CHEIN	Michel, Professeur, LIRMM, Montpellier	Président
M. RECHENMANN	François, Directeur de Recherche, IMAG, Grenoble	Rapporteur
M. SALLANTIN	Jean, Directeur de Recherche, LIRMM, Montpellier	Rapporteur
M. QUINQUETON	Joël, Directeur de Recherche, LIRMM, Montpellier	Examineur
M. DE SAINTE MARIE	Christian, Ingénieur, 3ASI sàrl, Grenoble	Examineur
M. MATHON	Albert, Professeur, EMSE, Saint-Etienne	Examineur
M. GRILLOT	Didier, Maître de Recherche, EMSE, Saint-Etienne	Examineur

ACADEMIE DE MONTPELLIER

UNIVERSITE MONTPELLIER II

— SCIENCES ET TECHNIQUES DU LANGUEDOC —

THESE

présentée à l'UNIVERSITE MONTPELLIER II - SCIENCES ET TECHNIQUES DU LANGUEDOC
pour obtenir le DIPLOME DE DOCTORAT

SPECIALITE : informatique

Formation Doctorale : informatique

Ecole nationale supérieure
des mines de Saint-Etienne
SCIDEM / ESPACE FAURIEL
158 cours Fauriel
42023 Saint-Etienne cedex 2

ACQUISITION INTERACTIVE
DE LA CONNAISSANCE
par un système télématique
dans le domaine de l'assainissement de l'eau

par

PHILIPPE BEAUNE

Soutenu le 16 janvier 1992 devant le jury composé de :

M. CHEIN	Michel, Professeur, LIRMM, Montpellier	Président
M. RECHENMANN	François, Directeur de Recherche, IMAG, Grenoble	Rapporteur
M. SALLANTIN	Jean, Directeur de Recherche, LIRMM, Montpellier	Rapporteur
M. QUINQUETON	Joël, Directeur de Recherche, LIRMM, Montpellier	Examineur
M. DE SAINTE MARIE	Christian, Ingénieur, 3ASI sàrl, Grenoble	Examineur
M. MATHON	Albert, Professeur, EMSE, Saint-Etienne	Examineur
M. GRAILLOT	Didier, Maître de Recherche, EMSE, Saint-Etienne	Examineur



Remerciements

Je remercie Monsieur Michel CHEIN, Professeur au Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, Monsieur François RECHENMANN, Directeur de Recherche au laboratoire ARTEMIS-IMAG de Grenoble, et Monsieur Jean SALLANTIN, Directeur de Recherche au Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, d'avoir accepté de faire partie du jury et d'avoir examiné attentivement mon travail.

Mes remerciements vont également à Monsieur Joël QUINQUETON, Directeur de Recherche au Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, et à Monsieur Didier GRAILLOT, Maître de Recherche à l'Ecole des Mines de Saint-Etienne, qui ont encadré ce long travail, l'un d'un point de vue informatique, l'autre d'un point de vue applicatif.

J'adresse une vive reconnaissance à Monsieur Albert MATHON, Professeur à l'Ecole des Mines de Saint-Etienne, pour m'avoir accueilli au sein du département qu'il dirige, pour la confiance qu'il m'a constamment témoignée, et pour avoir mis à ma disposition les nombreux moyens nécessaires à la réalisation de ce travail.

L'accomplissement de ce travail doit aussi beaucoup à Monsieur Chrétien DE SAINTE MARIE, qui a su me communiquer sa passion pour la recherche scientifique, et sa connaissance de cette jeune discipline qu'est l'Apprentissage Symbolique Automatique. En espérant que notre collaboration puisse se poursuivre et soit encore fructueuse, je tiens à le remercier très vivement.

Je remercie chaleureusement les membres du Département Stratégie du Développement de l'Ecole des Mines de Saint-Etienne, pour leur participation de près ou de loin à ce travail, ainsi que les membres du Département Informatique Appliquée de l'Ecole des Mines de Saint-Etienne qui ont toujours su répondre avec gentillesse à mes sollicitations.

Je remercie tout particulièrement Serge et Annie, mes beaux-parents, qui ont lu, avec un soin extrême, ce mémoire pour y déceler les fautes de frappe et d'orthographe; ainsi que Françoise, ma femme, pour son indispensable soutien moral et pour la patience dont elle a su faire preuve durant l'élaboration et la rédaction de ce travail.

" Pour mon catéchumène, c'est une révélation. Il s'est aperçu soudain que le travail géologique ne consiste nullement, comme on le croit souvent, à ramasser au petit bonheur des échantillons de roches ou de fossiles, mais à interpréter les aspects du sol, à en saisir l'architecture, à en reconstituer l'histoire. Une explication, certes, provisoire, fragmentaire, maladroite, mais lourde de promesses et principe des synthèses futures. "

(Théodore Monod - *Méharées*)

" Il est très difficile de transmettre une expérience vécue, et les chemins de la recherche de la vérité passent par l'expérience, qui peut être mortelle, de l'erreur. "

(Edgar Morin - *Science avec conscience*)

Sommaire

Sommaire	7
Liste des figures	13
Résumé.....	15
Abstract	17
Introduction.....	19
Chapitre 1	
La gestion des projets d'aménagement en eau.....	23
1.1 La pollution des eaux.....	24
1.1.1 Enjeux.....	25
1.1.2 Solutions.....	26
1.2 La recherche appliquée en hydrologie.....	27
1.2.1 La connaissance des phénomènes physiques.....	28
1.2.2 Les outils d'aide à la décision.....	29
1.3 L'intégration des outils	31
1.3.1 La simulation	31
1.3.1.1 Les simulateurs MISE.....	31
1.3.1.2 Les simulateurs PROMISE.....	32
1.3.2 La Station de Travail Intelligente	33
1.3.2.1 L'opérationnalité de la Station de Travail	34
1.3.2.2 L'intégration des outils existants.....	36
1.3.2.3 Le recueil des connaissances	37
1.4 Particularités de l'assainissement autonome des eaux usées.....	39
1.4.1 Une préoccupation croissante.....	39
1.4.2 Un savoir-faire dispersé.....	39
1.4.3 Une législation naissante	40
1.4.4 Les traces de l'expertise	41
1.4.5 La complexité des solutions.....	41
1.5 Conclusion	42

Chapitre 2

Les outils d'acquisition de la connaissance	45
2.1 Pourquoi acquérir de la connaissance ?.....	46
2.2 Le goulot d'étranglement.....	47
2.3 La méthode empirique.....	48
2.4 Les méthodes analytiques.....	49
2.4.1 KADS	50
2.4.2 KOD.....	52
2.4.3 MACAO.....	53
2.4.4 Les limites des méthodes analytiques.....	54
2.5 L'automatisation de l'acquisition.....	56
2.5.1 Un modèle simple d'apprentissage automatique.....	56
2.5.2 Classifications des systèmes d'apprentissage.....	58
2.5.2.1 Apprentissage numérique ou symbolique.....	58
2.5.2.2 Apprentissage déductif ou inductif	59
2.5.2.3 Classification selon la stratégie.....	60
2.5.2.4 Classification selon le type de connaissance acquise	62
2.5.2.5 Apprentissage par détection de similarités ou par recherche d'explications	64
2.5.3 L'apprentissage par détections de similarités	64
2.5.3.1 La connaissance sur le domaine.....	65
2.5.3.2 La prise en compte des données erronées.....	68
2.5.3.3 Incrémentalités	69
2.6 Quelques algorithmes d'apprentissage.....	71
2.6.1 L'algorithme d'élimination des candidats	72
2.6.2 ID3.....	73
2.6.3 COBWEB.....	74
2.6.4 UNIMEM	75
2.6.5 CHARADE	77
2.6.6 Les RDP.....	78
2.6.6.1 La notion de dépendance.....	78
2.6.6.2 Définitions.....	79
2.6.6.3 Représentation de la Dépendance Perçue : le RDP.....	80
2.6.6.4 Apprentissage automatique et RDP.....	81
2.7 Conclusion	81

Chapitre 3

Conception de la Station de Travail Intelligente de l'aménageur en eau.....	85
3.1 La Station de Travail : un outil de communication.....	86
3.1.1 Fonctions de la Station de Travail.....	86
3.1.2 Les explications de la Station de Travail.....	88
3.1.3 Les niveaux de la connaissance experte	90
3.1.3.1 La connaissance de stratégie.....	90
3.1.3.2 La connaissance de surface.....	91
3.1.3.3 La connaissance statique.....	92

3.1.4 Acquisition interactive de la connaissance.....	93
3.2 Conceptualisation des transferts de connaissance.....	94
3.2.1 Les Théories Semi-Empiriques (TSE).....	94
3.2.2 L'environnement d'apprentissage de Reitz.....	96
3.3 Analyse des transferts de connaissance.....	97
3.3.1 Le système télématique.....	98
3.3.1.1 Les interlocuteurs du système télématique.....	98
3.3.1.2 Les dialogues avec le système télématique.....	100
3.3.1.3 L'acquisition de la connaissance par Minitel.....	103
3.3.1.4 La redistribution de la connaissance par Minitel.....	103
3.3.2 Le simulateur.....	104
3.3.2.1 Formalisation des dialogues avec le simulateur.....	104
3.3.2.2 Les perturbations.....	106
3.3.3 La Station de Travail.....	106
3.4 Architecture de la Station de Travail.....	108
3.4.1 Représentation de la connaissance.....	108
3.4.2 Description de la Station de Travail.....	109
3.5 Conclusion.....	111

Chapitre 4

Moïse : un système télématique d'acquisition interactive de la connaissance à base de RDP.....	115
4.1 Réalisation d'un système expert télématique.....	116
4.1.1 Architecture.....	116
4.1.2 Interface utilisateur.....	118
4.1.3 Exploitation des consultations.....	119
4.1.4 Bilan de l'expérience.....	122
4.2 Choix des RDP.....	123
4.3 Utilisation d'un RDP.....	124
4.3.1 Représentation de la connaissance.....	125
4.3.1.1 Une base d'exemples pédagogique.....	126
4.3.1.2 Contenu du RDP.....	127
4.3.1.3 Lecture du RDP.....	128
4.3.2 Intégration d'un nouvel exemple.....	130
4.3.2.1 L'algorithme d'entretien incrémental d'un RDP.....	130
4.3.2.2 Exemple d'intégration d'un exemple à un RDP.....	133
4.3.2.3 Formules atomiques et formules complexes.....	135
4.3.2.4 La fonction de généralisation.....	136
4.3.2.5 La connaissance sur le domaine.....	136
4.3.2.6 La prise en compte des données erronées.....	137
4.3.2.7 Incrémentalités.....	138
4.3.3 Initialisation du système.....	140
4.3.4 Consultation du graphe.....	141
4.3.4.1 Chaînage avant.....	142
4.3.4.2 Chaînage arrière.....	143
4.3.4.3 Chaînage bi-directionnel.....	144

4.4 Modes d'interactions	145
4.4.1 Interface consultant.....	145
4.4.1.1 La caractérisation d'une solution.....	146
4.4.1.2 Interprétation de l'arrêt d'une consultation en chaînage avant.....	146
4.4.1.3 Interprétation de l'arrêt d'une consultation en chaînage arrière.....	147
4.4.1.4 Recherche d'une preuve.....	148
4.4.2 Interface expert.....	149
4.4.2.1 Description d'un problème et de sa solution, par l'expert.....	149
4.4.2.2 Recherche d'une preuve.....	150
4.4.2.3 Traitement d'un contre-exemple.....	150
4.4.2.4 Evolution du domaine et facteurs cachés.....	151
4.4.2.5 Recherche d'un problème par le système.....	154
4.4.2.6 Validité, pertinence et cohérence du RDP.....	155
4.4.3 L'accès multi-voie au système.....	156
4.5 Description fonctionnelle du système télématique	157
4.6 Conclusion	159

Chapitre 5

Moïse : application à l'assainissement autonome des eaux

usées	161
5.1 La relation fondatrice	162
5.2 Le langage de description du domaine	162
5.2.1 L'expertise en assainissement autonome des eaux usées.....	163
5.2.2 Le langage d'observation.....	164
5.2.2.1 Hétérogénéité de la parcelle étudiée.....	165
5.2.2.2 Caractérisation symbolique et numérique d'un même paramètre.....	165
5.2.2.3 Caractérisation multiple d'un même paramètre.....	166
5.2.2.4 La régionalisation de la connaissance experte.....	167
5.2.3 Le langage de décision.....	167
5.2.4 Le langage intermédiaire.....	168
5.2.5 Sémantiques de la Dépendance Perçue.....	169
5.2.6 Incomplétude des expertises.....	169
5.3 Définition du langage initial du système	170
5.3.1 Les opérateurs.....	170
5.3.1.1 Les opérateurs booléens.....	170
5.3.1.2 Les opérateurs d'affectation.....	171
5.3.2 Le langage d'observation.....	171
5.3.2.1 La pente du terrain.....	171
5.3.2.2 Le sol.....	172
5.3.2.3 Le substratum.....	174
5.3.2.4 La nappe aquifère.....	174
5.3.2.5 La présence d'un captage d'eau.....	175
5.3.2.6 L'existence d'un point de rejet superficiel.....	175
5.3.2.7 La localisation géographique de l'expertise.....	176
5.3.3 Le langage des conclusions intermédiaires.....	176
5.3.4 Le langage des réponses.....	177
5.3.5 Le langage des précisions.....	178
5.4 La pertinence des conseils fournis	178
5.5 Conclusion	179

Conclusion.....	181
Bibliographie	185
Annexe A	
Intégration d'un exemple à un RDP.....	193
A.1 Création du nouveau nœud :.....	194
A.2 Parcours topologique du RDP :.....	194
A.3 Vérification de l'existence d'un exemple identique :.....	198
Annexe B	
La base d'expertises.....	199
Annexe C	
Intégration d'un exemple à un RDP.....	211
C.1 Le RDP issu de la base d'expertises :.....	211
C.2 Correspondance symbolique/numérique :.....	213
C.3 Règle d'applicabilité d'une filière :.....	215

Liste des figures

fig. 1.1 :	principales agglomérations accueillant des équipes de recherche hydrologiques (d'après [Dubreuil 90]).....	27
fig. 1.2 :	les étapes d'un projet d'aménagement pour un bureau d'études	35
fig. 2.1 :	le goulot d'étranglement de l'acquisition de la connaissance	47
fig. 2.2 :	l'acquisition de la connaissance par prototypage rapide.....	49
fig. 2.3 :	KADS : le cycle de vie d'un système expert.....	51
fig. 2.4 :	les trois modèles de KOD.....	53
fig. 2.5 :	un modèle général d'apprentissage automatique	57
fig. 2.6 :	deux scènes à généraliser ([Ganascia 88])	65
fig. 2.7 :	les espaces de versions de Mitchell	72
fig. 3.1 :	architecture sommaire de la Station de Travail.....	87
fig. 3.2 :	taxinomie des connaissances de Pierce (d'après [Sallantin & al. 91]).....	95
fig. 3.3 :	environnement d'apprentissage ([Reitz 92]).....	97
fig. 3.4 :	intentions des différents interlocuteurs du système télématique (d'après [Beaune-Graillet 90]).....	99
fig. 3.5 :	formalisation des dialogues avec le système télématique.....	101
fig. 3.6 :	formalisation des dialogues avec le simulateur-AIC.....	104

fig. 3.7 :	formalisation des dialogues avec le simulateur-EIAO	105
fig. 3.8 :	formalisation des dialogues avec la Station de Travail.....	107
fig. 3.9 :	architecture de la Station de Travail Intelligente de l'aménageur en eau	110
fig. 3.10 :	la plate-forme de simulation et ses différents emplois	112
fig. 4.1 :	architecture de la maquette de serveur expert	117
fig. 4.2 :	répartition des types de réponses du serveur pendant la période d'essai	119
fig. 4.3 :	répartition des conseils fournis par le serveur pendant la période d'essai.....	121
fig. 4.4 :	un exemple de RDP.....	126
fig. 4.5 :	l'algorithme d'entretien incrémental d'un RDP (d'après [Sainte Marie 89])....	132
fig. 4.6 :	mise à jour des liens lors de l'entretien incrémental d'un RDP (d'après [Sainte Marie 89])	133
fig. 4.7 :	un nouvel exemple est intégré au RDP	134
fig. 4.8 :	description fonctionnelle de Moïse.....	158
fig. 5.1 :	principe de l'assainissement individuel ([Cres 89]).....	163
fig. A.1 :	RDP après examen de n1	194
fig. A.2 :	RDP après examen de n3	195
fig. A.3 :	RDP après examen de n5	196
fig. A.4 :	RDP après examen de n8	196
fig. A.5 :	RDP après examen de n6	197
fig. A.6 :	RDP après examen de n9	197
fig. C.1 :	le RDP issu de la base d'expertises.....	212
fig. C.2 :	le RDP représentant les valeurs de perméabilité.....	214

Résumé

La réalisation d'un outil d'aide à la conduite de projets, dans le domaine de l'aménagement en eau, nécessite l'étude du transfert de la connaissance, depuis les experts du domaine, vers l'outil informatique. L'architecture informatique de cette Station de Travail Intelligente est présentée, avec comme paradigme, celui des Systèmes Experts de Seconde Génération : plusieurs niveaux de connaissance apparaissent qui nécessitent plusieurs formes d'acquisition de la connaissance.

Pour l'acquisition de la connaissance profonde, un simulateur de projets est utilisé; et un système télématique, à base d'apprentissage automatique à partir d'exemples, permet de traiter l'acquisition de la connaissance de surface. Cet apprentissage est effectué au moyen de la construction incrémentale d'un Réseau de Dépendances Perçues.

Néanmoins, nous n'exigeons pas du système télématique qu'il apprenne seulement à partir des exemples de problèmes résolus et fournis par les experts. Le système doit aussi être capable de dialoguer avec les experts de façon à faire valider la connaissance apprise. Ceci permet d'éviter l'introduction de certaines données erronées, tout en autorisant les changements de la base de connaissance dus à une éventuelle évolution du domaine (innovation technologique, évolution réglementaire, ...).

Mots clés

apprentissage symbolique automatique, acquisition de la connaissance, gestion de projets, simulation de projets, aménagement en eau, assainissement des eaux usées.

Abstract

The aim of this work is to build a water project management tool, which involves the study of expert knowledge transfer to a computer system. The architecture of this Intelligent Workstation is introduced with the paradigm of Second Generation Expert Systems : different knowledge levels, which need several ways of knowledge acquisition, appear.

On one hand, a project simulator is used for deep knowledge acquisition; and, on the other hand, a telematics system, based on machine learning from examples, is used for shallow knowledge acquisition. This machine learning is done by incrementally constructing a Perceived Dependencies Network.

Nevertheless, we don't expect the system to learn only from examples of resolved problems supplied by experts. The system also has to be able to hold a dialogue with experts to validate learned knowledge. This permits to prevent erroneous data introduction, while allowing knowledge base changes due to possible field evolutions (technological innovation, statutory evolution, ...).

Key words

machine learning, knowledge acquisition, project management, project simulation, water management, waste water treatment.

Introduction

La conduite des projets d'aménagement (alimentation en eau potable, assainissement des eaux usées, stockage de déchets industriels, ...) nécessite une rigueur de plus en plus grande pour faire face à la dérive des coûts, et fait intervenir un grand nombre de disciplines scientifiques (géologie, hydrobiologie, hydrogéologie, économie, mathématiques, ...).

Nous étudions donc la réalisation d'une Station de Travail Intelligente, outil d'aide à la décision destiné aux Bureaux d'Etudes. Cette Station de Travail n'aura pas pour vocation de décider à la place des ingénieurs, mais de les aider à prendre des décisions : elle devra donc être capable de fournir des conseils, et surtout de pouvoir justifier ces conseils. En cela, elle constitue un Système à Base de Connaissance traditionnel, c'est à dire, un système contenant de la connaissance et capable de délivrer cette connaissance à un utilisateur.

Le présent travail montre qu'une telle considération des Systèmes à Base de Connaissance n'est pas suffisante, et ne permet pas toujours de résoudre le problème de la constitution de la Base de Connaissance. Sont donc présentées les méthodes actuelles d'acquisition de la connaissance (les méthodes analytiques et celles à base d'apprentissage automatique) de façon à faire apparaître leurs limites, par rapport au domaine qui nous intéresse : l'aménagement en eau.

Ce domaine possède des particularités qui compliquent considérablement le transfert de la connaissance : il est constitué d'un savoir-faire non formalisé, dispersé géographiquement et dans plusieurs corps de métiers, et capable d'évolutions. Après avoir étudié les possibilités offertes par les méthodes analytiques (qui font intervenir un cogniticien), nous envisageons les méthodes d'apprentissage automatique. Ces méthodes, lorsqu'elles sont capables de traiter le problème du bruit dans les données fournies par les experts, ne savent le faire que par le biais d'un traitement statistique de l'information. Les conseils fournis, sur la base d'un tel apprentissage, ne pourront qu'être probabilistes, et donc interprétables uniquement par des experts.

Nous proposons donc une architecture de la Station de Travail qui repose à la fois sur l'étude de la connaissance experte, et sur l'étude des interactions de notre système avec, d'une part, les utilisateurs finaux, qui consultent le système, mais aussi, d'autre part, les experts, qui détiennent la connaissance. L'étude de la connaissance experte s'inspire des modèles cognitifs des méthodes analytiques d'acquisition de la connaissance et du paradigme des Systèmes Experts de Seconde Génération (SESG). Nous proposons ainsi trois types de connaissances (connaissance profonde, connaissance de surface et connaissance statique). L'étude des interactions est faite sur la base d'une conceptualisation des dialogues entre un apprenant et un apprenti. Pour chaque type de connaissance, nous approfondissons l'étude des transferts de connaissance, et notamment l'acquisition, par la Station de Travail, de la connaissance des experts.

Ainsi, pour la connaissance de surface relative à l'assainissement autonome des eaux usées, nous proposons un système télématique qui nous permet de joindre facilement les experts. Pour cette application, nous utilisons le fait que l'acquisition de la connaissance est incrémentale, et que par conséquent l'expert peut avoir accès à tout moment à la connaissance déjà apprise. La présence de bruit peut donc être traitée par un dialogue entre le système et l'expert, sur la base d'une comparaison entre l'information fournie par l'expert et la connaissance acquise, de façon à expliciter les causes de conflit entre cette nouvelle information et la connaissance déjà apprise. Nous montrons donc que l'acquisition de la connaissance à base d'apprentissage automatique n'est pas une entrave à la résolution des conflits, dus à l'introduction d'éventuelles informations bruitées, lorsqu'elle s'accompagne d'un dialogue entre le système et l'expert.

Une telle acquisition interactive de la connaissance experte demande un effort de réflexion, de la part de l'expert, mais permet de mieux modéliser la connaissance experte, dès lors qu'elle est

appliquée à un domaine où l'on a un besoin impératif d'une décision sûre, destinée à une personne incapable d'en faire une interprétation poussée.

Ce mémoire comporte cinq chapitres. Le premier, "*La gestion des projets d'aménagement en eau*", présente l'environnement, l'utilité et l'opérationnalité de notre projet de réalisation d'une Station de Travail Intelligente de l'aménageur en eau. Il conclut sur l'importance toute particulière du transfert de la connaissance, depuis les experts vers le système informatique, pour que l'opérationnalité de notre Station de Travail soit garantie.

Le deuxième chapitre, "*Les outils d'acquisition de la connaissance*", présente un état de l'art en matière d'acquisition de la connaissance, avec deux grandes orientations : d'un part les méthodes analytiques qui font intervenir un intermédiaire humain (le cognicien) et, d'autre part, l'apprentissage automatique. Quelques algorithmes d'apprentissage automatique sont présentés en fin de chapitre.

Le troisième chapitre, "*Conception de la Station de Travail Intelligente de l'aménageur en eau*", détaille l'architecture informatique de la Station de Travail, en considérant ce système non pas seulement comme un système contenant et délivrant de la connaissance, mais comme un outil de communication entre les experts et les utilisateurs finaux.

Le quatrième chapitre, "*Moïse : un système télématique d'acquisition interactive de la connaissance à base de RDP*", est consacré à l'un des outils prévus pour alimenter la base de connaissance de la Station de Travail : le système télématique. Ce chapitre montre que les Réseaux de Dépendances Perçues (RDP) permettent les dialogues prévus au chapitre précédent entre le système d'apprentissage et ses interlocuteurs.

Enfin, le cinquième chapitre, "*Moïse : application à l'assainissement autonome des eaux usées*", présente le début de réalisation de notre système, et un essai d'apprentissage sur la base de quelques exemples d'expertises en assainissement autonome des eaux usées.

Les résultats de cet essai se trouvent en annexe de ce mémoire, accompagnés de quelques expertises brutes utilisées.

Une autre annexe présente l'intégration d'un nouveau cas à un RDP, selon l'algorithme d'entretien incrémental, présenté au quatrième chapitre.

Chapitre 1

La gestion des projets d'aménagement en eau

"Avant la révolution industrielle, en amont, la nature dispensait une eau pure; en aval, les hommes rejetaient une eau polluée, à charge pour cette nature de digérer, de dégrader les polluants. Mais avec la civilisation industrielle, et surtout avec la croissance des populations, l'épuration naturelle n'a plus suffi" ([Brodhag 90]). Désormais, si l'on veut disposer d'une eau de qualité suffisante pour être consommable (même après traitement de l'eau captée), ou si l'on ne veut pas dégrader irrémédiablement le milieu récepteur, on ne peut plus se contenter de rejeter l'eau usée, telle quelle, dans la nature; il nous faut la préparer de façon à ce que la nature puisse la digérer. C'est cette préparation de l'eau à être recevable par un milieu naturel (ou même réutilisable directement) qui constitue l'assainissement des eaux usées.

La problématique de cet assainissement peut être appréhendée de nombreuses façons :

- sous l'angle du projet industriel : on distingue alors différentes phases : la collecte des eaux usées, son prétraitement, son traitement et son évacuation. On distingue aussi différents acteurs du projet dont le financeur, le maître d'œuvre, le maître d'ouvrage, l'utilisateur, etc.

- sous l'angle du contexte du projet : sommes-nous en milieu urbain ou bien en milieu rural ? s'agit-il d'étudier un assainissement autonome ou bien un assainissement collectif ? voulons-nous concevoir un dispositif d'assainissement neuf ou bien en réhabiliter un ancien ?
- sous l'angle des disciplines scientifiques qu'elle fait intervenir. On peut citer, par exemple : l'hydrologie (étude des cycles de l'eau), la pédologie (étude des propriétés du sol), l'hydrogéologie (étude des interactions entre l'eau et la géologie), l'hydraulique (étude des écoulements d'eau), l'hydrobiologie (étude des organismes vivant dans l'eau), etc.
- sous l'angle économique et social : quels sont les investissements qui seront permis ? quel sera le coût d'exploitation ? désirerons-nous réutiliser les eaux épurées ?
- sous l'angle administratif et juridique : des réglementations techniques contraignent le problème, et des démarches auprès de certaines administrations sont prévues par des textes de loi, hors considération de l'aspect financier du projet.
- sous l'angle de la pollution : on peut étudier les caractéristiques intrinsèques de l'eau (ses qualités chimiques et bactériologiques par exemple), ou bien les caractéristiques du milieu récepteur (protection de certaines zones sensibles).
- etc.

Les différents points de vue énumérés ci-dessus - la liste n'est pas exhaustive - montrent toute la complexité du problème de l'assainissement des eaux usées. Dans ce chapitre, après avoir présenté les enjeux de la pollution de l'eau, nous allons examiner comment ce problème peut se résoudre (quels outils pour faire quoi ?), et nous proposerons une intégration des modes de résolution, afin que ce problème puisse être appréhendé de manière synthétique.

1.1 LA POLLUTION DES EAUX

La pollution des eaux revêt deux formes : les pollutions événementielles (ponctuelles, très fortes, concentrées, médiatiques) et les pollutions chroniques, beaucoup plus insidieuses car régulières et moins visibles.

L'acuité du problème de cette deuxième forme de pollution des eaux s'est révélée au grand public lors des derniers étés secs (1988-1990). En effet, en période de sécheresse, les pollutions chroniques créent des situations spectaculaires (développement très important d'algues, par exemple) car les polluants sont moins dilués. La prise de conscience a alors été brutale, et s'est concrétisée par un battage publicitaire sur les lessives avec ou sans phosphates. Pourtant les phosphates présents dans certaines lessives ne sont pas les seules origines de phosphore dans les eaux. On peut citer, par exemple, les déjections animales lorsqu'elles sont

mal épandues. Mais l'apport le plus déterminant est d'origine domestique : chacun d'entre nous émet quotidiennement près de 4 g de phosphore (soit 12 g si on l'exprime en poids de phosphates - PO_4), dont la moitié ne provient pas des lessives ([Delisle-Lefevre 91]). Supprimer les phosphates des lessives ne résoudra donc pas entièrement le problème (mais ce n'est pas une raison pour remettre en cause cette suppression !), il faut quand même assainir les eaux usées.

Parallèlement, mais accentué par cette prise de conscience, un nouveau marché de la dépollution a vu le jour, et se traduit aujourd'hui par la multiplication de mesures en faveur de la qualité de l'eau.

1.1.1 Enjeux

Ce nouveau marché de la dépollution se révèle à différents niveaux de décision :

- au niveau politique : le 12 juin 1991, Mme Edith Cresson, Premier Ministre, annonce un quasi-doublément des dépenses des Agences de l'Eau en faveur de la dépollution (81 milliards de francs sur cinq ans). De façon plus générale, afin d'atteindre les objectifs des nouvelles normes de la CEE (taux de dépollution = 90%), les dépenses françaises d'investissement en matière d'assainissement et d'épuration des eaux usées sont évaluées dans une fourchette de 135 à 170 milliards de francs sur les dix ans à venir (Le Monde du 16/17 juin 1991). Ce qui correspond à un doublement par rapport au rythme actuel.

- au niveau industriel : créations de cellules "environnement", au sein même d'entreprises. Ces cellules sont chargées, entre autres, d'étudier la dépollution des eaux industrielles. Récemment, à Saint-Etienne, a même été créé un "Pôle de l'Eau", regroupant des industriels et des chercheurs, qui vise au renforcement de la mobilisation des partenaires de l'épuration des eaux. Mais le traitement n'est pas le seul aspect pris en compte : on commence à voir aussi surgir le concept d'"écobilan" des systèmes de production, qui consiste à faire intervenir le coût "déchets" dans la gestion de la production. Mme Edith Cresson va même jusqu'à affirmer que *"la bataille de l'environnement se remportera sur le front industriel"*.

- au niveau des collectivités locales : des départements et des communes, de plus en plus nombreux, créent leur propre "Service Environnement" et embauchent des spécialistes de ce domaine.

La prise de conscience est donc très large : de Monsieur Tout le Monde au dirigeant d'entreprise, en passant par les élus et donc les administrations. L'engagement de notre société dans le problème de la pollution de l'eau est total. Mais comment résoudre ce problème ? Quelles solutions sont offertes ?

1.1.2 Solutions

Les solutions pour résoudre le problème de la pollution de l'eau dépendent du type de pollution. Pour simplifier, on peut reprendre les deux formes de pollutions énoncées ci-dessus.

D'une part, des pollutions événementielles : ces pollutions, souvent accidentelles, sont très fortes et ponctuelles dans le temps et dans l'espace. Le principal travail consiste en leur prévention, et en la prévention de l'extension de leurs nuisances.

D'autre part, des pollutions chroniques : on distingue deux sous-formes de ces pollutions. Celles qui sont concentrées et celles qui sont diffuses. Là aussi deux remèdes : prévenir et traiter. Ci-dessous nous n'examinons que le traitement de la pollution.

Pour les pollutions concentrées, lorsqu'elles sont identifiées (émissaire d'un égout, rejets d'une usine, ...), un traitement consiste en la construction d'une station d'épuration adaptée aux types de polluants présents dans l'effluent.

En revanche, pour les pollutions diffuses (rejets d'eaux usées d'une habitation isolée, par exemple) il n'est plus question, pour des raisons économiques, d'investir lourdement dans des canalisations pour rejoindre une station d'épuration, ni d'en construire une localement. Il faut donc faire appel à des techniques alternatives, moins coûteuses, moins lourdes à mettre en place et à en entretenir.

Entre ces extrêmes, se situe le problème le plus crucial. Quelle est la limite pour décider de l'installation d'une station d'épuration ? A partir de quelle limite faut-il que l'usager prenne en charge directement l'exploitation du dispositif d'assainissement ? Jusqu'à quelle limite un exploitant peut-il augmenter la taxe, afin d'étendre l'envergure de la collecte ?

Toutes ces questions n'ont pas de réponse simple. Comme nous l'avons vu en introduction de ce chapitre, le problème de l'assainissement des eaux usées comporte de multiples facettes, et traverse un grand nombre de disciplines scientifiques. Toutefois, pour aider à prendre une décision, car il faut bien en prendre une, des recherches sont effectuées qui débouchent sur des outils toujours plus nombreux, comme nous allons le voir ci-dessous.

1.2 LA RECHERCHE APPLIQUEE EN HYDROLOGIE

Un rapport de l'ORSTOM ([Dubreuil 90] - commandé par le Ministère de l'Environnement et celui de la Recherche) d'évaluation du potentiel français en hydrologie, donne une définition de l'hydrologie qui distingue trois champs : la recherche, l'ingénierie et l'opérationnel. L'hydrologie y est définie par "*l'étude du cycle hydrologique dans sa partie relative aux terres émergées et aux eaux des continents, des processus physiques, chimiques et biologiques les concernant, de leur rapport avec le climat et avec les autres facteurs physiques et géographiques et des interrelations entre eaux de surface et eaux souterraines*".

Dans la recherche hydrologique, nous trouvons les "*recherches relatives à l'établissement de modèles mathématiques et physiques représentant les systèmes hydrologiques concernant eaux de surface et souterraines*". L'ingénierie hydrologique est comprise comme "*l'emploi des connaissances hydrologiques*". Et l'hydrologie opérationnelle est constituée notamment des "*activités relatives au développement des réseaux hydrologiques, à l'amélioration du système de collecte et de traitement des données*".

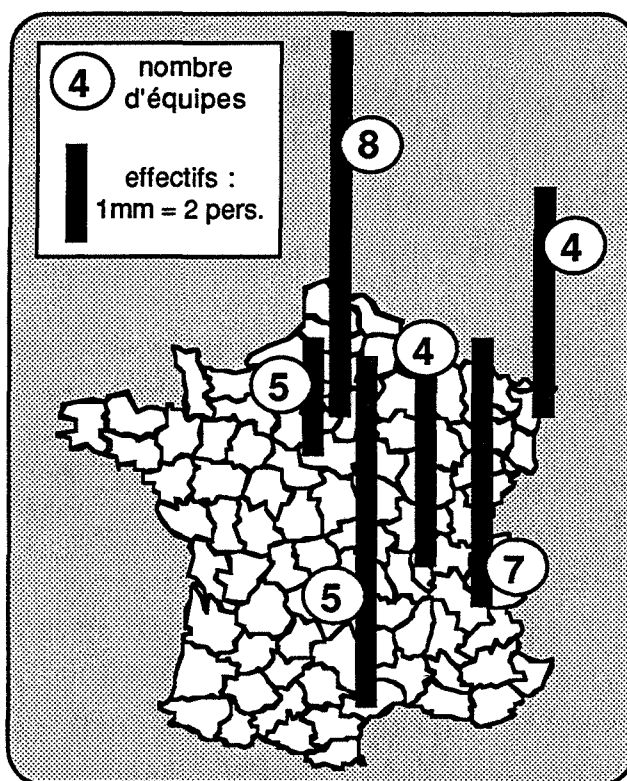


fig. 1.1 : principales agglomérations accueillant des équipes de recherche hydrologique (d'après [Dubreuil 90])

Comme nous pouvons le constater, l'hydrologie est transdisciplinaire, et regroupe aussi bien des préoccupations théoriques que pratiques. La recherche appliquée en hydrologie regroupe aussi ces préoccupations. De plus, comme le montre [Dubreuil 90], elle est dispersée (plusieurs dizaines d'équipes de recherche recensées) géographiquement, et thématiquement puisque le CNRS ne comporte pas de section interdisciplinaire traitant l'ensemble du cycle de l'eau (cf. figure 1.1). Notons toutefois que la prise de conscience évoquée ci-dessus, a aussi des répercussions dans le domaine de la recherche : le Plan National pour l'Environnement prévoit de multiplier par quatre d'ici à dix ans la dépense de recherche publique et privée consacrée à l'environnement, et prévoit aussi de renforcer ses moyens de coordination et de coopération ([Lalonde 91]).

Ci-dessous, nous passons en revue d'une part la recherche hydrologique (sous le titre "La connaissance des phénomènes physiques"), et d'autre part l'ingénierie hydrologique (sous le titre "Les outils d'aide à la décision") en montrant dans quelle mesure chacune d'elles peut être opérationnelle. Enfin, au paragraphe suivant, après avoir présenté une expérience originale de simulation des projets d'aménagement en eau, nous proposerons un outil d'intégration de la plupart de ces travaux qui devrait permettre d'accroître leur opérationnalité.

1.2.1 La connaissance des phénomènes physiques

Les activités de recherche en hydrologie sont principalement centrées sur la connaissance des phénomènes physiques qui interviennent dans le cycle hydrologique.

Une activité importante de cette recherche est la modélisation. Elle consiste en la conception de modèles comportementaux de phénomènes physiques ou chimiques. Les principaux domaines étudiés sont :

- les écoulements souterrains (nappes aquifères),
- les écoulements superficiels (rivières, canaux, ...),
- la dispersion des polluants dans les milieux ci-dessus cités (nappes, rivières, ...),
- ainsi que l'évolution de la qualité des eaux dans les dispositifs d'épuration (filtre à sable, par exemple).

La modélisation peut aussi être mathématique. Il s'agit alors de résoudre des équations différentielles qui sont des combinaisons d'équations de continuité et d'équations de conservation.

L'utilisation de ces modèles est limitée par des temps de calcul souvent prohibitifs, des interfaces utilisateur peu conviviales, et même parfois par l'indisponibilité des données pour les

alimenter. Il est alors fait appel à des modèles simplifiés, qui tentent de représenter l'essentiel de la réalité.

Une autre branche de la recherche en hydrologie est l'étude expérimentale de phénomènes physiques et chimiques. Nous pouvons citer, à titre d'exemple : le test de nouveaux dispositifs autonomes d'épuration des eaux usées, ou bien l'étude de l'infiltration des effluents en terme de temps de séjour dans le dispositif, en terme de vitesse d'écoulement, en terme de perméabilité du matériau utilisé dans le dispositif, ...

Ces recherches donnent naissance à de nouveaux traitements des eaux, qui seront utilisés après habilitation par les autorités compétentes et dans un contexte bien spécifié.

Mais ces recherches ne peuvent pas être déconnectées de la réalité sociale, et font donc intervenir des outils d'aide à la décision (simulation, recherche opérationnelle, analyse multicritère, systèmes experts, ...) pour prendre en compte, notamment, des paramètres économiques.

1.2.2 Les outils d'aide à la décision

Tous ces outils ont été étudiés dans [Graillet 86]. Nous y trouvons, outre les modèles évoqués ci-dessus :

- des modèles d'optimisation qui ont l'avantage de permettre à un décideur de choisir entre plusieurs solutions, mais en considérant des contraintes qui ne reflètent pas toujours la réalité. De plus il est très difficile d'y définir une fonction objectif à optimiser. Ils ne sont donc utilisés que pour des problèmes très simples, et ne sont donc opérationnels que dans des parties ponctuelles des projets hydrologiques.

- la recherche opérationnelle qui est un outil de gestion industrielle, traditionnel dans le domaine de l'eau. Il nécessite la fixation d'un critère d'optimisation (souvent une fonction économique qu'il s'agit de minimiser en fonctions de contraintes imposées). Dans le cas où les contraintes et la fonction objectif sont linéaires, il s'agit d'un problème de programmation linéaire. Elle est utilisée notamment pour l'optimisation technico-économique d'un réseau de distribution d'eau. D'autres méthodes sont applicables si les contraintes ne sont pas linéaires. C'est le cas, par exemple, de la programmation quadratique, qui peut servir à l'optimisation de l'emplacement des captages d'eau sur rivières. Pour certains projets d'aménagement en eau la nature particulièrement séquentielle des processus de décision permet d'utiliser les méthodes de la

programmation dynamique, pour optimiser par exemple le coût d'installation d'un réseau d'irrigation.

- l'analyse économétrique qui est une technique statistique qui peut permettre, entre autres, d'évaluer la demande en eau potable. Pour cela elle fournit une équation de cette demande qui permet de calculer des coefficients d'élasticité de cette demande par rapport à des facteurs explicatifs.

- l'étude de rentabilité qui est souvent effectuée pour évaluer la faisabilité de mesures correctives, à la fin de projets très importants.

- le modèle PERT (Program Evaluation and Review Technique) qui permet au responsable du projet de mieux maîtriser le déroulement du programme.

- l'analyse multicritère qui permet d'évaluer un projet sur des critères autres que ceux des coûts et bénéfices.

- de nombreuses bases de données, dont ECOTHEK relative à l'environnement, BRUIT et DECHETS qui sont les plus importantes.

- et des systèmes experts. Dans [Cres 89], quelques systèmes français sont présentés :

- Apogée, développé au CERGRENE, sur les réseaux d'assainissement,
- Hydroexpert sur l'implantation de forages (hydraulique villageoise) en Afrique,
- un système d'optimisation du fonctionnement d'une station d'épuration développé par OTV (Omnium de Traitement et de Valorisation),
- un système pour le suivi de la qualité du traitement d'eau potable réalisé par la CGE (Compagnie Générale des Eaux),
- et bien sûr, Moïse, présenté au chapitre 4 de ce mémoire, sur le choix d'une filière d'assainissement autonome des eaux usées.

De cette liste, non exhaustive, nous retiendrons surtout que ces outils sont très nombreux et divers, et qu'en conséquence le problème actuel d'un décideur est de choisir le bon outil, au bon moment dans le déroulement du projet, et surtout l'outil le plus adapté au problème qu'il veut résoudre. Nous allons donc présenter ci-dessous une expérience originale d'intégration de tous ces outils ([Graillet 86], [Cres 89]), dans un simulateur à vocation pédagogique, puis nous proposerons une autre version de cet outil, à vocation industrielle.

1.3 L'INTEGRATION DES OUTILS

Le département Stratégie du Développement de l'Ecole des Mines de Saint-Etienne a pour mission, entre autres, d'assurer un enseignement des sciences de l'eau. Cette matière, principalement axée sur la gestion des ressources en eau, nécessite un enseignement préalable de nombreuses notions technologiques, et fait appel à des notions de mathématique, de physique, de chimie, de géologie, de météorologie. Malheureusement, le nombre d'heures imparties à ce cours a fortement diminué ces dernières années, et désormais ce ne sont plus des notions techniques qui sont enseignées, mais plutôt la gestion globale d'un projet d'aménagement en eau. En effet, il n'est plus envisageable, à l'heure actuelle, d'enseigner tous les rudiments techniques qui interviennent dans un projet, pour former un chef de projet. Il est bien plus préférable de former l'élève ingénieur à la conduite de projet de façon plus générale, en lui apprenant à utiliser tous les outils qui peuvent s'offrir à lui, c'est à dire à bien cerner l'applicabilité de tel ou tel type d'outil, et à savoir quand l'utiliser.

1.3.1 La simulation

C'est ainsi, que pour l'enseignement des sciences de l'eau, les élèves ingénieurs de l'Ecole des Mines de Saint-Etienne utilisent désormais des simulateurs de projets réels, pour se former à la conduite de projets d'aménagement en eau. Ce nouvel enseignement fait appel aux techniques d'Enseignement Assisté par Ordinateur (EAO) et utilise un simulateur nommé "Modèle Intégré de Stratégie de l'Eau" (MISE).

1.3.1.1 Les simulateurs MISE

Les simulateurs de la famille MISE, développés sous l'impulsion du Professeur Philippe Davoine, responsable de l'enseignement des sciences de l'eau à l'Ecole des Mines de Saint-Etienne, ont pour objet de mettre un apprenant en condition de chef de projet, et de le faire réagir à des situations critiques.

Une session MISE dure environ une semaine, pendant laquelle l'apprenant doit d'abord répondre à un appel d'offre en fournissant un devis. Ensuite, lorsque sa proposition a été retenue (après avis du coordinateur de la session), l'apprenant doit mettre en œuvre son projet, en respectant son devis. Puis, comme c'est le cas dans les projets réels d'aménagement en eau (alimentation en eau potable, irrigation ou assainissement) l'élève doit proposer un cadre juridique et financier pour l'exploitation de l'aménagement.

Puisque l'apprenant n'est pas forcément un technicien du domaine concerné, certaines notions lui manquent pour évaluer les travaux à effectuer. Pour cela, des informations générales lui sont fournies par le logiciel MISE, qui lui permettent de faire cette évaluation. Ce sont notamment, des adresses d'organismes qui fournissent des renseignements d'ordre sociologique (la démographie, ...), topographique (plan d'occupation des sols), etc. Ce sont aussi des règles de calculs simples (pour prévoir un échéancier par exemple), ou des modèles mathématiques complexes (pour simuler une campagne de mesures sur le terrain par exemple).

Ensuite vient la phase la plus délicate : la réalisation des travaux. Pour rendre le projet plus vivant, et plus conforme à la réalité, des incidents sont créés qui font que l'apprenant doit réagir de façon à redresser une situation qui peut se dégrader par rapport aux prévisions. Pour cela, des experts du domaine assistent aux sessions de formation, et lorsqu'un apprenant ne sait pas comment agir, l'expert le conseille. Pour être bien guidé, l'apprenant doit être convaincu que la décision est pertinente, et ainsi s'instaure un dialogue entre l'expert et le novice, qui réalise un véritable transfert de connaissance. C'est cette phase des sessions MISE, montrant que toute une connaissance experte est ainsi explicitée, qui a donné l'idée à ses concepteurs d'essayer de "piéger" cette connaissance pour, dans un premier temps améliorer le fonctionnement des simulateurs, et dans un deuxième temps, réaliser une Station de Travail Intelligente de l'aménageur en eau.

Mais pour cela, il a d'abord fallu faire évoluer la structure informatique de MISE car les différents modules de MISE ne sont pas reliés informatiquement. Le lien est assuré par le coordinateur. Ainsi, on perd toute l'information contenue dans l'enchaînement des diverses étapes d'un projet. De plus, les aléas, qui viennent perturber le bon déroulement du projet, ne prennent pas en compte le passé du projet ([Cres 89]).

1.3.1.2 Les simulateurs PROMISE

La deuxième génération des simulateurs MISE a été baptisée PROMISE. L'architecture de ces systèmes, développée par François Noël Cres ([Cres 89]), répond aux critiques formulées ci-dessus, et adopte un style de programmation plus convivial permettant une mise à jour plus aisée du logiciel.

Ce style de programmation, emprunté à l'Intelligence Artificielle, utilise un formalisme proche de celui des règles de production. Pour toute action possible de la part de l'apprenant, une unité de connaissance est associée qui contient :

- la caractérisation de l'objectif, c'est à dire son coût moyen prévu (financier et temporel),
- l'analyse du passé, c'est à dire un ensemble de connaissances qui lie le passé du projet à une pondération des coûts prévus, pouvant aller jusqu'à une impossibilité de réalisation de l'action, et
- le contenu de l'action à réaliser.

Ainsi, le système PROMISE se comporte comme une sorte de système d'exploitation du simulateur : il est en quelque sorte le coordinateur qui active telle ou telle phase du projet, en tenant compte du passé. La réalisation d'une action peut être soit un programme extérieur à PROMISE, pourvu que PROMISE en recueille le résultat, ou bien une simple imputation sur le budget et une avancée dans le calendrier.

Les deux grands avantages des simulateurs PROMISE sont :

- leur plus grande facilité de programmation, ce qui permet à presque n'importe quel non-informaticien de fabriquer son simulateur à partir de l'architecture de PROMISE, et aussi
- l'enchaînement centralisé de toutes les étapes et de toutes les phases du projet, ce qui, à terme, permettra de greffer un module d'apprentissage ayant pour but de "piéger" la connaissance relative à la conduite de projet.

Mais ces simulateurs ont encore une lacune qu'il faudra combler si l'on désire qu'ils deviennent de véritables systèmes pédagogiques : il leur manque un module d'explications intelligentes. Une condition sine qua non à l'obtention d'une explication intelligente est de séparer la connaissance en plusieurs niveaux ([Kassel 88], [Kassel 89]). On atteint alors des systèmes dits de seconde génération; nous détaillons ce concept, dans le chapitre 3, pour l'architecture de la Station de Travail.

L'architecture de cette Station de Travail sera proche de celle d'un système d'Enseignement Intelligemment Assisté par Ordinateur (EIAO), qui constituera la troisième génération de simulateurs MISE.

1.3.2 La Station de Travail Intelligente

Les simulateurs MISE, et encore plus les simulateurs PROMISE, de par leurs structures, pourraient avoir un autre rôle que celui de système d'Enseignement Assisté par Ordinateur. Ils pourraient permettre d'effectuer des simulations de projets non encore réalisés afin d'étudier les différents choix qui se présentent aux ingénieurs qui conduisent le projet.

En effet, face à la dérive des coûts de développement, il devient nécessaire de gérer les projets d'aménagement en eau de plus en plus efficacement. La simulation de projets peut contribuer à cette gestion plus efficace en permettant de tester diverses solutions.

L'objectif de notre équipe de recherche, au sein du Département Stratégie du Développement de l'Ecole des Mines de Saint-Etienne, est donc de réaliser un outil pour contenir tous les outils qui sont à la disposition de l'aménageur en eau, mais pas de proposer un outil de plus, qui viendrait s'ajouter aux autres ! Au contraire, il s'agit de réaliser une boîte à outil qui intégrerait tous les outils déjà existants, et qui guiderait l'aménageur en eau vers les outils adaptés au problème qu'il veut résoudre.

1.3.2.1 L'opérationnalité de la Station de Travail

L'actuelle prise de conscience de la gravité des problèmes de l'eau, nous l'avons vu, accentue considérablement la croissance du marché de la dépollution. Plus généralement, on observe un intérêt croissant pour les études relatives aux projets d'aménagement ayant un impact sur l'environnement naturel. Pour notre part, nous ne considérons, pour l'instant, que les projets d'aménagements en eau c'est à dire les aménagements liés au cycle hydrologique continental. Parmi ces aménagements, l'expérience de notre équipe de recherche porte essentiellement sur l'irrigation, l'alimentation en eau potable et l'assainissement des eaux usées. Nous limiterons donc notre travail à ces trois domaines, que nous regrouperons, peut-être abusivement, sous le terme "aménagement en eau".

Notre objectif est donc de proposer une Station de Travail aux décideurs intervenant dans les projets d'aménagement en eau. Le but de cette Station de Travail est de les aider dans leur décision en les guidant lors de la gestion de leurs projets.

Ces décideurs sont de différents types selon le moment où ils interviennent dans le projet. On peut, pour résumer, distinguer deux niveaux de décision : le niveau collectivité territoriale (de l'Etat au syndicat intercommunal ou même la commune) et le niveau ingénierie.

Les décideurs du premier niveau interviennent lors de l'élaboration du projet, et notamment dans le financement de celui-ci. Nous les appellerons les "gestionnaires". A ce niveau, les décisions sont très complexes et doivent satisfaire plusieurs objectifs, parfois conflictuels entre eux. Pour les caractériser globalement, on peut dire qu'elles sont politiques.

Le niveau ingénierie est occupé par les ingénieurs, les bureaux d'études, les sociétés distributrices d'eau et certaines administrations de l'Etat. Nous regrouperons, par la suite, ces décideurs sous le terme "bureau d'études". Les décisions, à ce niveau, sont d'une tout autre nature : elles sont techniques. Elles consistent à proposer un devis pour l'étude de faisabilité, puis si la proposition est retenue, à étudier la faisabilité technique du projet et enfin, si le projet est accepté, à conduire sa réalisation.

En fait, les deux niveaux sont très imbriqués puisque la décision des gestionnaires dépend du travail effectué par le bureau d'études, et réciproquement la réalisation technique du projet dépend de la décision des gestionnaires.

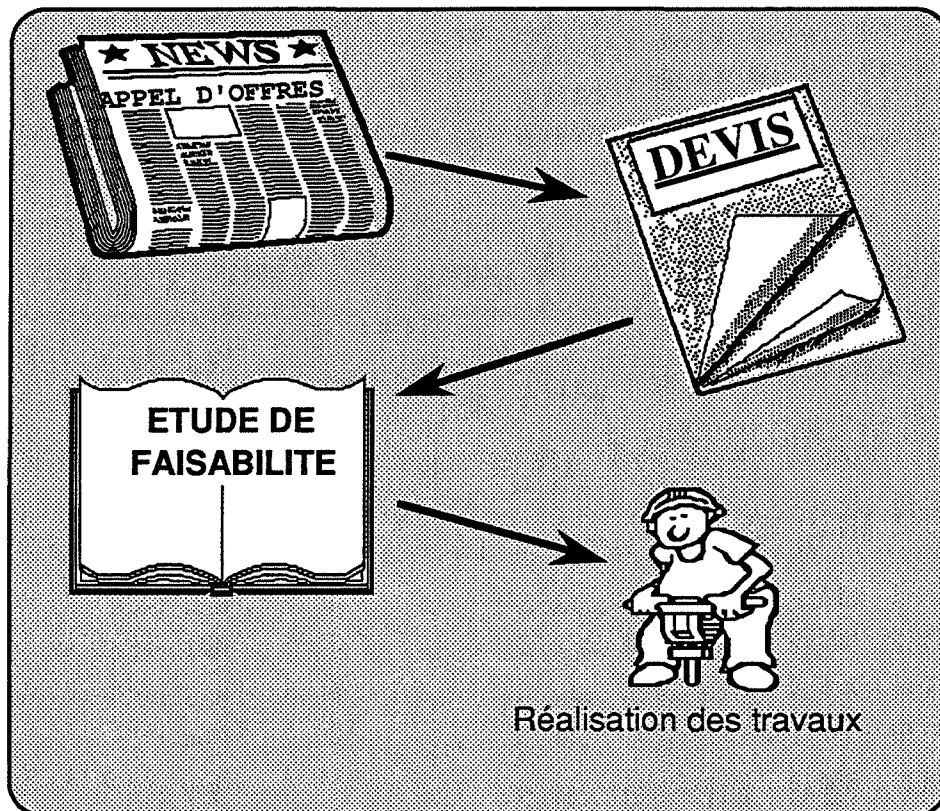


fig. 1.2 : les étapes d'un projet d'aménagement pour un bureau d'études

Les décisions qui nous intéressent, dans le cadre de la Station de Travail, sont celles du niveau ingénierie (cf. figure 1.2) : celles qui permettent de proposer aux gestionnaires un devis pour l'étude de faisabilité, celles qui interviennent pendant l'étude de faisabilité, et celles qui

permettent de conduire la réalisation du projet une fois qu'il est décidé au niveau des gestionnaires.

Notre Station de Travail s'adresse donc en priorité aux bureaux d'études. Son opérationnalité sera assurée par le fait qu'un grand nombre de connaissances est exigée dans la maîtrise efficace des projets d'aménagement en eau, et que les bureaux d'études sont dans l'impossibilité de les regrouper toutes.

Mais la Station de Travail n'aura pas pour objectif de résoudre le problème à la place des ingénieurs du bureau d'études : elle devra guider leur démarche en les aidant à caractériser les problèmes à résoudre et en les orientant vers les outils les mieux adaptés, c'est à dire les plus opérationnels, et en leur proposant des informations pertinentes. La Station de Travail ne fera que conseiller les ingénieurs, afin que ceux-ci puissent prendre des décisions; pour cela, la Station de Travail devra être capable de justifier ses conseils, car, en aucun cas, un ingénieur digne de ce nom n'accepterait de prendre une décision sur le simple conseil obscur d'une machine. Pour que l'ingénieur puisse accepter les conseils d'une machine, il faut qu'il connaisse, et qu'il comprenne le mécanisme qui a fourni le conseil. Nous reviendrons souvent, par la suite, sur cet aspect très important de l'explicabilité des outils d'aide à la décision.

1.3.2.2 L'intégration des outils existants

L'objet de la Station de Travail est donc de regrouper les outils déjà existants, mais aussi de leur associer un domaine d'opérationnalité, et d'intégrer tout cela dans la démarche globale de la gestion d'un projet (établissement du devis, étude de la faisabilité et conduite de la réalisation).

Pour établir un devis, il faut caractériser le projet de façon à évaluer le coût de l'étude de faisabilité. C'est à dire qu'il faut détailler les objectifs généraux, présents dans l'appel d'offres publié par les gestionnaires, afin de déterminer quelles seront les études à réaliser pour pouvoir proposer des solutions au problème soumis. Ces études peuvent être très variées : études sociologiques, campagnes de mesures sur le terrain, recherches d'informations, etc. Chacune de ces études a un rôle bien précis qu'il s'agit de déterminer, pour le situer dans un contexte d'opérationnalité. Les connaissances relatives à toutes ces études devront être contenues dans la Station de Travail.

Pour étudier la faisabilité du projet, défini par l'appel d'offres, des études ont été prévues qui conduiront le bureau d'études à des conclusions, lesquelles conclusions conduiront à des

solutions. Ce sont ces solutions qui seront proposées aux gestionnaires. Tous les mécanismes, qui concourent à l'établissement d'une solution, devront être présents dans la Station de Travail.

Pour conduire la réalisation de la solution retenue par les gestionnaires, des outils de gestion de projets sont utilisables, des études complémentaires sont à réaliser, des réajustements des prévisions sont parfois nécessaires. En bref, de nombreuses décisions sont à prendre et les connaissances relatives à ces décisions devront être contenues dans la Station de Travail.

L'intégration des outils existants dans une structure unique, la Station de Travail, ne consiste donc pas seulement en la juxtaposition de ces outils. En fait, cette intégration consiste surtout en la définition de l'applicabilité de chacun. Et l'opérationnalité de la Station de Travail obtenue résultera à la fois de la pertinence des conseils fournis, et de ses capacités à montrer cette pertinence pour convaincre l'ingénieur du bureau d'études, car celui-ci, à chaque étape du projet, devra convaincre à son tour les gestionnaires du projet.

1.3.2.3 Le recueil des connaissances

La construction de la Station de Travail consiste donc d'une part en la réalisation d'une structure informatique (décrite au chapitre 3 de ce mémoire) capable d'accueillir tous les outils existants, et capable de répondre aux exigences exposées ci-dessus, et, d'autre part, en un recueil des connaissances intervenant dans les différentes prises de décision.

Comme nous l'avons vu, l'opérationnalité de la Station de Travail réside à la fois dans l'exactitude du conseil qu'elle fournit, et dans sa capacité à pouvoir justifier ce conseil. Le premier point ne sera garanti que si on localise la connaissance nécessaire et que l'on parvient à la transférer dans le système sans trop la dégrader. Quant au deuxième point, il sera assuré par la structure informatique retenue, et aussi par la localisation de la connaissance et la qualité de son transfert.

En ce qui concerne la localisation de la connaissance, on peut opérer une dichotomie traditionnelle entre la connaissance des spécialistes (ceux qui ont une connaissance essentiellement théorique, les universitaires par exemple) et la connaissance des experts (ceux dont la connaissance repose surtout sur l'expérience et s'insère dans une dynamique de décision).

Dans le domaine de l'eau, les spécialistes appartiennent à de nombreuses disciplines scientifiques (cf. paragraphe 1.2) mais n'interviennent que très peu dans les projets d'aménagement en eau. Leur connaissance se limite à la réalisation de nouveaux outils, sans toujours participer à la définition de l'opérationnalité de ceux-ci.

Les experts, en revanche, sont ceux qui interviennent le plus dans ces projets. Ils utilisent à la fois une connaissance empirique (une connaissance experte) et une connaissance théorique. Leur connaissance théorique leur est utile pour choisir tel ou tel outil, lorsqu'ils en connaissent d'utilisables, et partout où ils n'ont aucune connaissance théorique, puisqu'il leur faut de toutes façons prendre une décision, ils utilisent une connaissance purement empirique. Ce sera d'ailleurs souvent le cas, car ils ne peuvent pas avoir de connaissances théoriques sur de nombreux domaines scientifiques.

Les experts sont essentiellement dans les bureaux d'études en environnement. Ce sont aussi, parfois, les maîtres d'ouvrage (pour l'assainissement des eaux usées par exemple), des employés de Syndicats Intercommunaux (pour l'alimentation en eau potable par exemple) ou ceux des Services Environnement des collectivités locales. Les experts sont aussi dans les administrations (les Directions Départementales des Affaires Sanitaires et Sociales par exemple) où ils valident les projets d'un point de vue réglementaire, et où ils font face, aussi, aux problèmes engendrés par les dysfonctionnements des dispositifs. Ce sont notamment les hydrogéologues agréés. Certains experts, enfin, se trouvent dans les Agences de l'Eau (anciennes Agences Financières de Bassin) où ils étudient les projets pour éventuellement les co-financer.

La connaissance nécessaire à la conduite d'un projet d'aménagement en eau étant localisée, reste le problème crucial, désormais bien identifié, du transfert de la connaissance.

Dans le chapitre 2 de ce mémoire nous exposons ce problème du transfert de la connaissance, et présentons divers moyens de le résoudre.

Toutefois, afin de restreindre l'étendue de nos investigations, sans pour autant simplifier la problématique générale, nous nous limiterons, par la suite, au domaine spécifique de l'assainissement autonome des eaux usées en tant que partie de la Station de Travail. Dans le paragraphe suivant, nous montrons la complexité et la richesse de ce domaine.

1.4 PARTICULARITES DE L'ASSAINISSEMENT AUTONOME DES EAUX USEES

Nous avons vu, en introduction de ce chapitre, la multitude des points de vue qui permettent de considérer le problème de l'assainissement des eaux usées. Certains de ces points de vue (contexte du projet et environnement économique et social) font apparaître qu'il n'existe pas une seule solution économique et technique pour traiter les eaux usées. Si l'on se trouve en milieu urbain, la solution coûteuse de la station d'épuration pourra être retenue et supportée financièrement par un nombre relativement grand de citoyens. Tandis qu'en milieu rural (pour de petits lotissements ou hameaux isolés, et a fortiori pour les bâtiments isolés), la construction d'une station d'épuration, ou le raccordement à une station déjà existante, sont en général d'un coût trop élevé par rapport à la quantité d'eau à assainir. Il faut donc faire appel à d'autres solutions techniques, dites alternatives : celles de l'assainissement autonome des eaux usées.

1.4.1 Une préoccupation croissante

A l'heure actuelle, en France, le taux global d'assainissement des eaux usées est de 36%, pour seulement 28% en milieu rural. A titre de comparaison, en Allemagne, le taux global s'élève à 70%. Un gros effort est donc consenti par les pouvoirs publics pour atteindre les objectifs énoncés dans le cadre de la CEE : 90% d'eaux usées assainies ! En milieu urbain, cet effort sera techniquement facile à atteindre car la pollution est concentrée et les solutions sont maîtrisées.

En revanche, en milieu rural, il s'agit de résoudre un très grand nombre de cas. Une étude de marché, réalisée pour notre équipe en 1988, révélait que 200 000 systèmes d'assainissement autonome étaient construits ou réhabilités chaque année en France. Suite à la récente prise de conscience, ce chiffre est à réviser nettement à la hausse.

Le domaine que nous choisissons est donc très intéressant du point de vue opérationnel. De plus, comme nous allons le voir, il est très représentatif de la complexité et de la diversité des sciences de l'eau en général.

1.4.2 Un savoir-faire dispersé

En effet, la mise en place de technologies d'assainissement autonome fait appel à un large champ de connaissances :

- pédologie,
- géologie,

- hydrogéologie,
- chimie, etc.

et subit de fortes contraintes d'ordre social :

- quelle est la démographie du lieu étudié ?
- quel est le taux d'équipements sanitaires des habitants de la zone ?
- quelle est leur motivation à assainir leurs eaux usées ? etc.

De plus, une autre contrainte vient s'ajouter à celles-ci : il s'agit de l'occupation de la parcelle qui accueillera le dispositif d'assainissement. Un dispositif peut convenir aux qualités du terrain et de l'eau à assainir, ainsi qu'à la quantité des effluents, mais peut se révéler trop encombrant pour la parcelle. Il faut alors envisager d'autres dispositifs, plus coûteux et moins encombrants. Dans le cas d'une habitation isolée, où souvent la parcelle de terrain hébergeant l'habitation sera aussi destinée à accueillir le dispositif d'assainissement, il faut alors trouver un compromis entre la suppression de certaines fonctions de la parcelle (aire de jeux, potager, ...) et l'augmentation du coût du dispositif.

Parallèlement à la dispersion de ce domaine dans de nombreuses disciplines, on constate aussi une dispersion géographique de l'expertise. Chaque région, chaque département, a ses habitudes et son savoir-faire. Ceci est principalement dû aux diversités géologiques qui font que certains dispositifs sont très peu adaptés en moyenne, donc peu connus, et par conséquent très rarement construits, même lorsque, exceptionnellement, ils conviennent. Cette dispersion est aussi due à la nature de la connaissance très empirique de ce domaine : elle repose essentiellement sur l'expérience, donc elle est particulière à chaque expert. Ceci est accentué par le fait que les experts sont rarement appelés à se rencontrer. Par exemple, une demande d'agrément d'un dispositif n'est examinée que par un seul expert. Un expert n'est donc jamais confronté à l'avis d'un autre expert.

Une autre raison importante de cette dispersion de la connaissance est l'absence de toute réglementation précise en la matière.

1.4.3 Une législation naissante

La réglementation dans ce domaine existe sous forme de DTU (Documents Techniques d'Urbanisme). Pour l'assainissement des eaux usées, la DTU date de mai 1991 seulement.

De toutes façons, une réglementation ne peut pas prendre en compte tous les paramètres, ni toutes les particularités régionales, ni toutes les solutions possibles. Cette prise en compte entraînerait un accroissement beaucoup trop important du nombre de cas à considérer.

La connaissance experte, dans ce domaine, restera donc l'élément de base de la prise de décision, la réglementation ne pouvant servir que de référence.

1.4.4 Les traces de l'expertise

Outre la DTU, qui peut nous fournir des premiers éléments d'études sur l'état de l'art, c'est à dire sur le type de connaissance mise en œuvre en matière d'assainissement autonome des eaux usées, il existe deux autres traces concrètes de l'expertise : les rapports des hydrogéologues agréés et les formulaires DDASS.

Les rapports des hydrogéologues agréés sont intéressants car ils reflètent des cas qui n'ont pas pu être résolus de façon simple, où des cas pour lesquels il y a dysfonctionnement d'un dispositif installé. Ils contiennent donc une connaissance très importante. Par exemple, pour les cas qui n'ont pas pu être résolus de façon simple, ils contiennent des combinaisons de paramètres que l'on ne rencontre pas ordinairement, et qui, dans ces cas-là, empêchent de préconiser les dispositifs habituels. De même, pour les dysfonctionnements, ils permettent de dégager des cas où, a priori, on aurait pu installer tel dispositif, mais qui en fait fonctionne mal car on n'a pas pris en compte telle considération. De plus, ces rapports comportent une expertise déjà expliquée, qui est donc plus facile à saisir.

Quant aux formulaires des DDASS, liés aux demandes d'autorisation des permis de construire, ils sont intéressants car ils nous permettront d'avoir une grande base d'exemples pour tester notre système.

1.4.5 La complexité des solutions

La dernière particularité du domaine choisi, qui fait que ce domaine est représentatif des sciences de l'eau en général, est le nombre important de solutions d'assainissement autonome et de paramètres à prendre en compte.

Lors de la construction de la maquette du système Moïse (cf. paragraphe 4.1), nous avons retenu quatre types de filières (l'épandage en tranchées, le filtre à sable, le terre d'infiltration et le filtre bactérien percolateur). Ces quatre types de filières permettent, selon les aménagements qui y sont associés, d'engendrer une quinzaine de filières.

Toutes ces filières ne prennent pas en compte les mêmes paramètres. Au total, nous avons répertorié neuf paramètres pour caractériser la parcelle devant accueillir le système d'assainissement. Mais nous n'avons pas pris en compte la notion de place disponible sur la parcelle (certains systèmes d'assainissement sont plus volumineux que d'autres), ce qui aurait nécessité une description thématique de la parcelle pour trouver un emplacement adéquat. De même, nous n'avons pas pris en compte des paramètres divers du type "existe-t-il un chemin faisant au moins telle largeur permettant d'acheminer tel matériau ou tel matériel ?". Nous n'avons pas non plus considéré l'aspect maintenance du système, ni l'aspect économique : dans certains cas ne vaut-il pas mieux s'associer à son voisin (si c'est un ami !) pour étudier un dispositif commun, ou acheter une autre parcelle plus apte à recevoir un dispositif moins cher ?

En fait, il est pratiquement impossible d'imaginer a priori tous les paramètres qui pourraient intervenir dans ce genre d'étude. Seules les expériences vécues par les nombreux experts pourront nous permettre de bâtir une connaissance générale de ce domaine.

1.5 CONCLUSION

Pour mener à bien les projets d'aménagements en eau, des outils existent qui aident les bureaux d'études, mais ces outils sont en grand nombre et ont chacun un domaine d'applicabilité restreint. De plus, les projets d'aménagements en eau font intervenir un grand nombre de disciplines que les bureaux d'études ne peuvent pas maîtriser toutes.

Pour aider à une gestion plus efficace des projets d'aménagement en eau, nous proposons donc la réalisation d'une Station de Travail Intelligente, véritable boîte à outils, qui guiderait l'aménageur en eau dans la succession de ces prises de décision. Pour que cette Station de Travail soit opérationnelle, il faut que, d'une part, le conseil qu'elle fournit soit exact, et que, d'autre part, elle soit capable d'expliquer ce conseil.

La première contrainte, l'exactitude du conseil fourni, sera satisfaite par une localisation de l'expertise et surtout par un transfert de cette connaissance experte vers la Station de Travail. Il faudra aussi que le transfert ne dégrade pas trop cette connaissance.

La seconde contrainte, l'explicabilité, sera satisfaite par la structure informatique adoptée et présentée au chapitre 3. Mais la valeur des explications dépendra surtout de la localisation de la connaissance et de son transfert.

On constate donc que le transfert de la connaissance, depuis les experts, vers la Station de Travail, constitue un point névralgique de la construction de notre outil. Ce transfert sera

d'autant plus difficile à effectuer que la connaissance en sciences de l'eau est dispersée (géographiquement, dans diverses disciplines scientifiques et dans divers corps de métiers). De plus, cette connaissance est essentiellement empirique et elle n'est pas stabilisée. En effet, des évolutions technologiques et réglementaires font évoluer ce domaine.

Ce problème (l'acquisition de la connaissance) a déjà été bien identifié dans le cadre de la construction des Systèmes Experts. Dans le chapitre suivant, nous présentons quelques solutions qui permettent de le résoudre, et ensuite nous étudierons comment ces solutions peuvent satisfaire nos contraintes : les contraintes liées au domaine (la connaissance experte est dispersée et évolutive), ainsi que les contraintes liées aux exigences d'opérationnalité de la Station de Travail Intelligente (l'exactitude du conseil et la capacité à expliquer ce conseil).

Chapitre 2

Les outils d'acquisition de la connaissance

Le problème de l'acquisition de la connaissance pour les systèmes experts (ou plus généralement pour les systèmes à base de connaissance) a souvent été présenté comme "le goulot d'étranglement" de la réalisation de ces systèmes. Après avoir présenté l'enjeu de l'acquisition de la connaissance et les premières tentatives de constructions de systèmes experts, nous détaillons, dans ce chapitre, les deux grandes familles de méthodes actuellement utilisées pour effectuer le transfert de la connaissance depuis le monde réel vers le système artificiel.

Dans un premier temps nous examinons les méthodes analytiques qui font intervenir un intermédiaire, le cogniticien, ou ingénieur de la connaissance. Pour cela, nous détaillons trois méthodes : KADS, KOD, et MACAO, car elles prennent en compte tout le processus d'acquisition de la connaissance. De plus, ces méthodes nous apportent un enseignement très riche : elles nous proposent chacune un modèle de la connaissance experte.

Ensuite nous examinons une deuxième voie : l'acquisition de la connaissance à base d'Apprentissage Automatique. Ces méthodes d'acquisition, qui se passent de l'intervention du cogniticien, sans pour autant le remplacer, posent de nombreux problèmes et aucune d'entre

elles ne permet de les résoudre tous. Pour bien cerner tous ces problèmes, nous détaillons plusieurs classifications de ces méthodes, et nous présentons quelques algorithmes d'apprentissage.

2.1 POURQUOI ACQUERIR DE LA CONNAISSANCE ?

Avant d'aborder le problème de l'acquisition de la connaissance, il semble bon de rappeler ce que l'on entend habituellement par *système expert* et en quoi ce type de logiciels pose des problèmes particuliers, non encore entrevus en informatique traditionnelle.

Un système expert est un logiciel d'aide à la décision, dont on attend qu'il reproduise un comportement d'expert, lorsque la connaissance du domaine de l'expert est a priori peu structurée (pour des discussions plus détaillées de la définition d'un expert et d'un système expert, le lecteur pourra consulter [Farreny 85], [Farreny-Ghallab 87], [Ermine 89], ou [Hart 88] par exemple).

L'approche retenue pour modéliser cette connaissance, est celle des systèmes à base de connaissance (dont font partie les systèmes à base de règles). Les systèmes à base de connaissance sont des logiciels qui se distinguent organiquement des logiciels traditionnels par la séparation que l'on y effectue entre la connaissance utilisée et l'algorithmique mise en œuvre pour manipuler cette connaissance. Typiquement, l'organisation d'un système à base de connaissance se fait autour de trois composantes principales :

- une *base de connaissance* qui contient une représentation de la connaissance mise en œuvre,
- un *mécanisme d'inférence* qui contient les procédures de manipulation de cette connaissance,
- et une *interface utilisateur* qui permet la consultation de la base de connaissance via le mécanisme d'inférence.

La construction d'un système expert consiste donc, d'une part, en la modélisation de la connaissance d'un expert (ou de plusieurs experts), et, d'autre part, en la modélisation du mécanisme de manipulation de cette connaissance. Comme dans toute modélisation informatique, il s'agit de passer de la perception que l'on a d'un système réel, à sa représentation informatique plus ou moins simplifiée. En informatique traditionnelle, la bonne perception que l'on a du système réel, nous permet d'en construire un modèle. C'est le cas des systèmes réels bien formalisés (un plan comptable, par exemple). C'est aussi le cas des systèmes physiques pour lesquels on peut améliorer la perception que l'on en a par l'analyse de chaque composante du système et de ses liens avec les autres composantes (un atelier de production, par exemple).

Mais dans le cas d'un système cognitif (la résolution de problèmes, par exemple), la perception que l'on a du système n'est que globale, et tous les essais de formalisation sont encore restés vains (pour s'en convaincre le lecteur pourra parcourir [Centre Royaumont 79]). Et pourtant on va en tenter la modélisation.

2.2 LE GOULOT D'ETRANGLEMENT

La plupart des systèmes experts opérationnels reposent sur la logique formelle et les règles de production ([Hart 88]). Pour ses systèmes, on a donc un moteur d'inférence (à base de Modus Ponens) et un cadre pour exprimer la connaissance. Il ne reste donc plus qu'à décrire la connaissance de l'expert sous forme de règles de production. En fait le plus gros du travail reste encore à faire : c'est ce qu'on appelle communément "le goulot d'étranglement" du transfert de la connaissance.

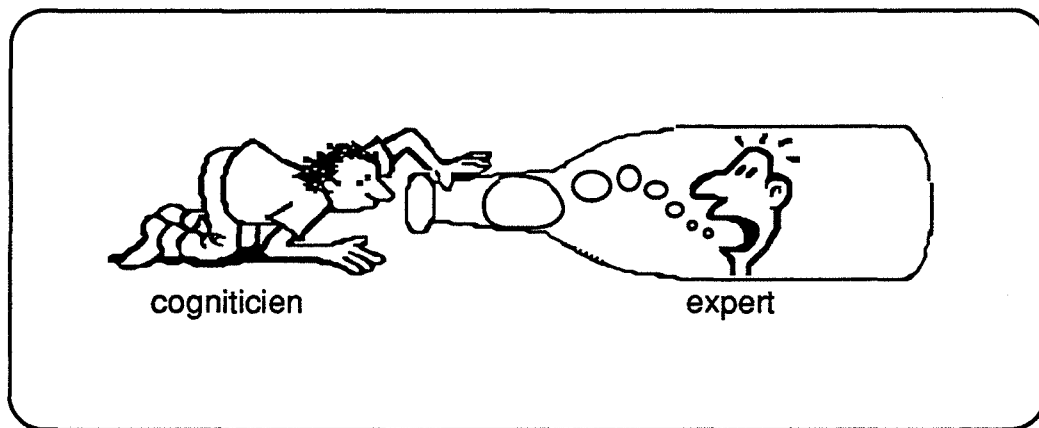


fig. 2.1 : le goulot d'étranglement de l'acquisition de la connaissance

En effet, il n'est pas envisageable d'imposer à l'expert l'utilisation du langage du système informatique pour décrire explicitement toute sa connaissance, au risque de voir la connaissance ainsi fournie fortement biaisée, voire même erronée. D'un autre côté, l'informaticien, développeur du système, n'est pas forcément apte à interroger efficacement l'expert, et là aussi, la connaissance extraite risque d'être biaisée ou erronée.

Il convient donc de se pencher sur le difficile problème du transfert de la connaissance, depuis son détenteur humain vers le système informatique cible. Une voie pour réaliser ce transfert, est

d'intercaler un humain entre les deux : un spécialiste de ce transfert, dit "cogniticien". Cette phase de transfert *"peut durer des mois voire des années, et compte-tenu de la complexité des connaissances de l'expert qui parvient difficilement à expliciter ses processus mentaux, la connaissance extraite risque souvent d'être inexacte, incomplète, voire inconsistante"* ([Dieng 90]).

De plus certaines applications rendent difficile l'intervention du dit cogniticien, notamment celles où la connaissance experte évolue, et/ou celles où elle est dispersée chez plusieurs experts. En effet dans ces cas, le cogniticien aurait un trop grand nombre d'interventions à effectuer. Ce qui nous oblige à explorer une autre voie, dans laquelle on peut se passer de ce spécialiste de la connaissance : l'automatisation de l'acquisition de la connaissance, ou apprentissage automatique.

2.3 LA METHODE EMPIRIQUE

Les premiers développements de systèmes experts se sont faits sans aucune méthode précise quant à l'acquisition du savoir des experts. Le principal souci des développeurs était alors le développement du moteur d'inférence. Les cogniticiens (qui étaient parfois les informaticiens eux-mêmes) procédaient incrémentalement, par prototypage rapide, en entrant dans le cycle "extraction de la connaissance — modélisation de la connaissance extraite — validation de la base de connaissance" jusqu'à ce que la base de connaissance soit correcte.

L'extraction de la connaissance se fait avec diverses techniques (souvent inspirées de travaux en psychologie) que l'on retrouvera dans les méthodes analytiques. Ces techniques sont entre autres ([Dieng 90]) :

- l'interview, plus ou moins guidée par le cogniticien qui peut éventuellement user de la reformulation, c'est à dire exprimer avec ses propres mots ce que l'expert vient de dire.
- l'analyse de protocoles, qui consiste à enregistrer l'expert résolvant un problème; l'expert doit réfléchir à haute voix. Puis ces enregistrements sont décodés et interprétés.
- l'observation directe, qui repose sur l'enregistrement vidéo de l'expert.
- l'utilisation d'un questionnaire de façon à contrôler l'expression de l'expert.
- la verbalisation rétrospective, où l'expert commente un cas déjà traité.
- l'introspection, qui consiste à faire résoudre par l'expert un cas typique imaginaire.
- la technique du "magicien d'Oz", où l'expert simule le comportement du futur système.
- le brainwriting, lorsque plusieurs experts cherchent à formaliser leur expertise.

- les grilles-répertoires : le cogniticien demande à l'expert de comparer des triplets d'objets, puis le cogniticien, par des calculs statistiques, construit un arbre des ressemblances où sont représentés tous les objets, et l'expert doit valider cet arbre ([Aussenac 89]).

Il n'est bien sûr pas question d'utiliser toutes ces techniques. Il faut choisir les plus adaptées selon le domaine d'expertise, selon le nombre d'experts, selon leur disponibilité, selon leur coopération, etc.

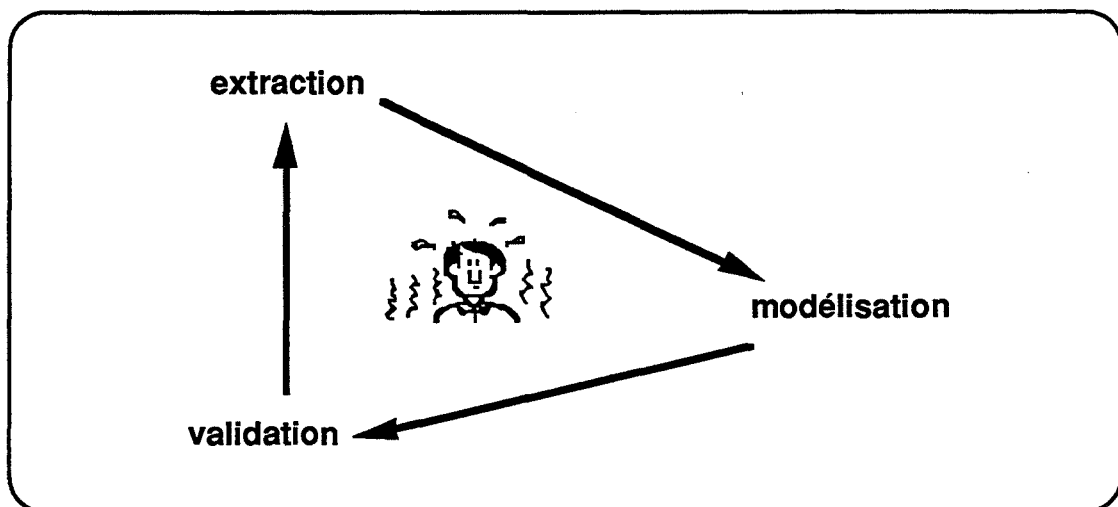


fig. 2.2 : l'acquisition de la connaissance par prototypage rapide

La modélisation de la connaissance extraite est une reformulation de ce que le cogniticien a perçu, dans le formalisme choisi pour la base de connaissance.

La validation de la base de connaissance pose le plus gros problème : il faut d'une part étudier la cohérence et la complétude de la base, puis son adéquation par rapport à l'utilisation qu'en feront les utilisateurs finaux du système.

La méthode par prototypage rapide a été abandonnée au profit de méthodes plus structurées (les méthodes analytiques), qui offrent un plus grand espoir de voir aboutir un projet de construction d'une base de connaissance.

2.4 LES METHODES ANALYTIQUES

Les méthodes analytiques apportent une amélioration très importante par rapport à la méthode par prototypage rapide décrite ci-dessus. Malheureusement, un grand nombre d'entre elles

n'intervient que partiellement dans le processus qui va du monde réel jusqu'au modèle informatique : certaines sont spécialisées dans un type de tâches précis, d'autres sont focalisées dans un domaine d'application donné, d'autres encore n'aident le cogniticien que lors d'une étape seulement du cycle de vie du système expert.

En revanche, certaines méthodes (KADS, KOD et MACAO par exemple) sont suffisamment générales pour pouvoir intervenir dans n'importe quel type de domaine, et prennent en compte tout le processus d'acquisition de la connaissance, depuis le monde réel jusqu'au modèle informatique. Ces méthodes proposent des modèles de connaissance qui guideront le cogniticien dans son effort d'interprétation de la connaissance extraite, et donc dans son effort d'extraction de la connaissance.

Ces modèles de connaissance sont des modèles intermédiaires qui permettent de passer de l'expression de la connaissance par l'expert, à la formalisation informatique de cette même connaissance. Ils fournissent un cadre de travail au cogniticien (ce qui lui permet d'interpréter, presque en direct, le discours de l'expert) et par conséquent le guident dans son travail d'extraction.

De plus, l'introduction d'une étape intermédiaire permet une validation moins tardive de la connaissance extraite. En effet, les modèles de connaissance proposés par ces méthodes sont plus proches de la connaissance de l'expert que ne le sont les formalisations informatiques de cette même connaissance. Ainsi l'enchaînement des cycles "extraction de la connaissance - modélisation de la connaissance extraite - validation de la base de connaissance" est considérablement raccourci, car les phases de modélisation et de validation reposent sur un formalisme plus adapté que ne le sont les formalismes informatiques.

2.4.1 KADS

KADS (*Knowledge Acquisition and Design Support*), inspirée du génie logiciel, est une méthode issue de plusieurs projets Esprit, auxquels ont participé, entre autres, l'Université d'Amsterdam, STC Technology Ltd, SCICON Ltd, Cap Gemini Innovation et SCS GmbH. Cette méthode consiste en la construction successive de plusieurs modèles : une phase d'analyse des données précède la conception et l'implémentation du système expert ([Hickman & al. 89]). Cette méthode est assortie d'un atelier logiciel qui guide toute la démarche grâce à une structure de type hypertexte ([Dieng 90]).

La phase d'analyse permet d'identifier le domaine, les besoins et les contraintes. Deux axes, dans cette phase, sont distingués : l'axe externe qui tient compte de l'environnement

informatique, et l'axe interne qui prend en compte le savoir faire des experts. La phase d'analyse débouche sur une documentation des besoins externes et sur un **modèle conceptuel** de l'expertise.

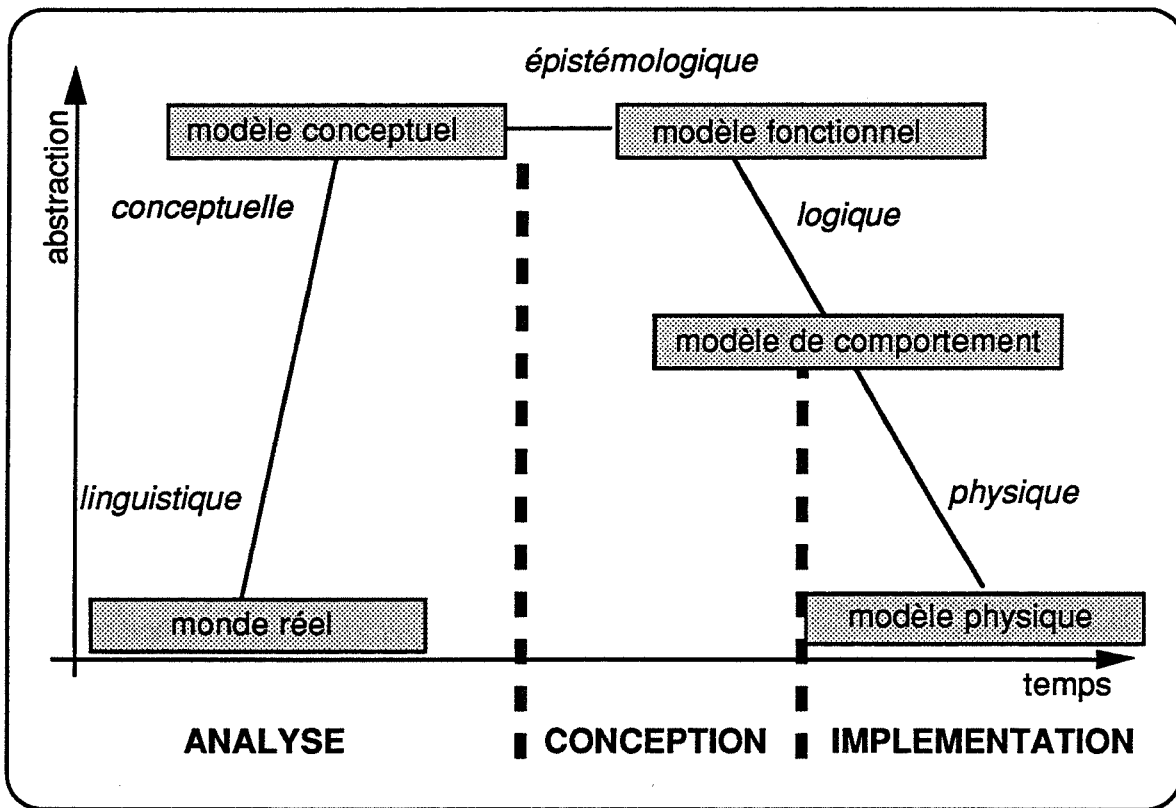


fig 2.3 : KADS : le cycle de vie d'un système expert

La phase de conception est similaire à celle développée en informatique classique (analyse fonctionnelle : description des fonctions que devra contenir le système), et débouche sur le **modèle fonctionnel**. Ce dernier est ensuite exploité (analyse organique : comment sera réalisée chaque fonction) pour donner le **modèle de comportement**.

Puis vient la phase d'implémentation, qui consiste en un codage du modèle de comportement.

La phase d'analyse (c'est à dire d'extraction de la connaissance) est facilitée par l'emploi d'un langage de modélisation conceptuelle (KCML), grâce à plusieurs niveaux conceptuels :

- niveau domaine, qui permet de décrire la connaissance statique en terme de concepts,
- niveau inférence, qui contient la méta-connaissance sur le niveau domaine,
- niveau tâche, qui permet d'explicitier l'utilisation des structures d'inférence du niveau supérieur,

- niveau stratégique, qui contient les plans de résolution du problème.

Comme nous pouvons le constater, cette méthode s'attache à détailler les différentes phases d'un projet de construction d'un système à base de connaissance. De plus elle propose un modèle de connaissance qui guide le cognicien dans sa démarche d'extraction de la connaissance. Mais aucun lien n'est proposé entre le modèle fonctionnel et le modèle de comportement : aucune correspondance n'est faite entre les différents niveaux de connaissance et les différents outils de programmation actuellement disponibles. C'est cette lacune que KOD se propose de combler en proposant un autre modèle de connaissance.

2.4.2 KOD

La méthode KOD (*Knowledge Oriented Design*), développée par CISI Ingénierie et présentée dans [Vogel 88], s'inspire de la linguistique et de l'anthropologie cognitive. Elle s'accompagne d'un outil logiciel (la K-Station) qui fournit des outils informatiques garantissant la mise en œuvre des différentes étapes de la méthode, indépendamment de l'implémentation ([Albert-Vogel 90]). Cette méthode repose sur trois modèles de connaissance (le modèle pratique, le modèle cognitif et le modèle informatique) et sur trois paradigmes (l'être, le faire et le dire) du langage.

Lors de l'étude de la base d'énoncés (transcription des entretiens), le cognicien s'efforce de faire apparaître des formes statiques (les taxèmes), des formes causatives (les actèmes) et des formes déclaratives (les inférences). Ces dernières formes doivent contenir l'intention et l'interprétation de l'expert. Ces différentes formes du discours de l'expert sont ensuite explicitées en taxinomies, actinomies et schémas, puis validées avec l'expert. D'éventuels retours vers les entretiens peuvent alors être effectués. Lorsque la base de connaissance est au point, le cognicien, ou l'informaticien, peut alors passer au codage : les taxinomies sont traduites en objets et en classes d'objets, les actinomies en méthodes, et les règles sont programmées avec un langage déclaratif.

Comme KADS, cette méthode détaille les différentes phases du projet de construction d'un système à base de connaissance, et propose un modèle de connaissance de l'expert, avec l'avantage que ce modèle est mis directement en parallèle avec des outils informatiques : la programmation par les objets, la programmation par les acteurs et la programmation par la logique. Mais en revanche, KOD ne propose rien pour distinguer les différents niveaux de

connaissance de l'expert, ce qui peut nuire à l'explicabilité du système ainsi construit (cf. chapitre 3).

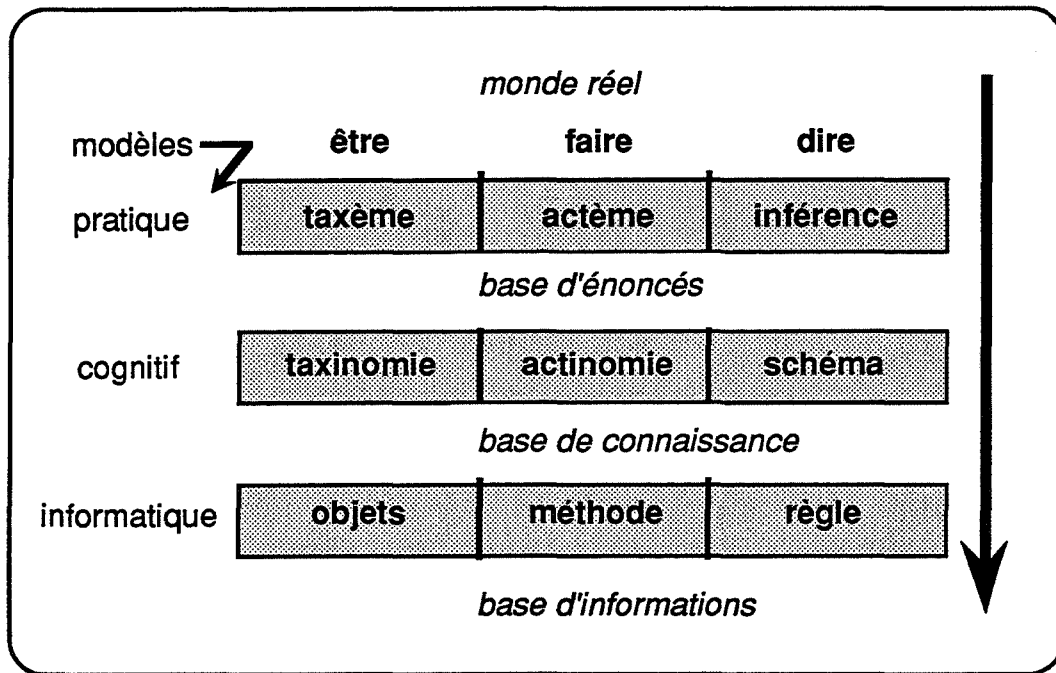


fig 2.4 : les trois modèles de KOD

La méthode suivante (MACAO) est à rapprocher de KADS, en ce sens qu'elle distingue différents niveaux de connaissance, et qu'elle possède le même inconvénient de ne pas proposer de parallèle entre ces niveaux et des outils de programmation. Néanmoins elle possède l'originalité de permettre la validation de la connaissance extraite directement par l'expert, grâce à une formalisation de cette connaissance sous forme de graphes.

2.4.3 MACAO

La méthode d'acquisition des connaissances MACAO (Méthode d'Acquisition des Connaissances Assistée par Ordinateur) guide l'ensemble du transfert d'expertise et s'appuie sur un logiciel d'aide à la construction de la base de connaissance grâce à un modèle cognitif.

Le modèle cognitif est à base de ([Dieng 90]) :

- schémas empiriques (ou procédures) décrivant la connaissance utilisée pour résoudre certaines parties du problème, et de
- schémas conceptuels, plus abstraits, décrivant les stratégies de choix de procédures.

Ces schémas sont composés d'unités de raisonnement appelées unités fonctionnelles. Les unités fonctionnelles qui permettent de décider du choix d'une procédure sont appelées déterminants.

La méthode MACAO, décrite dans [Aussenac 89], se déroule en quatre étapes : identification de l'expertise, recueil de protocoles, analyse des protocoles, et validation des connaissances structurées par l'expert.

L'identification de l'expertise a pour but de positionner les limites et les structures des zones de compétence. Elle permet de mettre en confiance l'expert et d'éviter l'oubli de toute une partie de l'expertise. Cette étape se compose de dialogues expert-cogniticien et d'observations de l'expert "expertisant". Les divers problèmes sont organisés en catégories par l'expert, au moyen de techniques grilles-répertoires.

Le recueil de protocoles consiste à recueillir des connaissances directement liées à la mise en œuvre de l'expertise. L'expert est invité à résoudre des problèmes identifiés dans la première étape. Durant cette étape, l'expert doit entrer dans le logiciel MACAO toutes les informations qu'il traite, tout en raisonnant à haute voix. Le raisonnement oral de l'expert est enregistré.

L'analyse des protocoles est faite par le cogniticien grâce au modèle cognitif présent dans le logiciel MACAO (unités fonctionnelles, déterminants, procédures, etc.). Cette analyse se fait en trois temps : analyse des protocoles de chaque problème, puis analyse transversale des problèmes de chaque catégorie, et enfin analyse globale. Ces analyses sont traduites sous forme de graphes de connaissance pour être présentées à l'expert. Le logiciel MACAO guide aussi le cogniticien en comparant les structures internes.

La validation des connaissances structurées est effectuée directement par l'expert sur le logiciel MACAO.

Contrairement à KOD, cette méthode ne vise pas une simulation du fonctionnement cognitif de l'expert, mais cherche à atteindre le niveau de performances de l'expert, en utilisant le modèle cognitif seulement comme point d'entrée "dans" les connaissances de l'expert et comme ensemble de points de repères pour les recueillir ([Aussenac 89]).

2.4.4 Les limites des méthodes analytiques

Les méthodes que nous venons de voir, n'ont pas toutes les mêmes caractéristiques. KOD par exemple convient bien pour les applications du type Système Expert de Première Génération, c'est à dire des applications où seulement des connaissances de surface sont nécessaires (voir le chapitre 3 pour une présentation des Systèmes Experts de Seconde Génération (SESG), et leurs

avantages). KADS et MACAO répondent beaucoup mieux aux applications de type SESG, mais en revanche elles n'apportent pas de solution immédiate quant au passage du modèle cognitif au modèle informatique. On peut bien sûr s'inspirer de KOD pour effectuer ce passage, mais ce n'est pas facile. Par exemple, si on essaie de combiner KADS et KOD, on s'aperçoit que le niveau domaine peut s'exprimer sous formes de taxèmes, et donc se programmer avec des langages orientés objets. De même avec le niveau inférence qui peut s'associer à la programmation par la logique. Mais en ce qui concerne les niveaux tâche et stratégique, il n'y a plus de correspondance exacte.

Bien que ces méthodes soient très riches du point de vue des modèles de connaissance - modèles que nous utiliserons, plus loin, dans ce travail - elles possèdent néanmoins une limitation très grave quant à leur applicabilité. Dans les domaines où de nombreux experts sont à consulter, ou bien dans les domaines où la connaissance évolue, et a fortiori dans les domaines qui combinent ces deux caractéristiques, ces méthodes se révèlent impuissantes car le cognicien y joue un rôle pendant toute la phase de développement du système.

Cela nous amène tout naturellement à examiner des méthodes d'acquisition de la connaissance où l'on peut se passer du cognicien. Parmi celles-ci se trouve l'Analyse de Données.

Le but de l'Analyse de Données est d'extraire des informations utiles d'un grand nombre de données, selon un critère fixé. Les résultats qu'elle fournit sont sous forme d'équations ou de matrices, et nécessitent donc une interprétation humaine pour être exploitable par un logiciel du type Système à Base de Connaissance. En effet, le critère d'explicabilité caractéristique des Systèmes à Base de Connaissance, ne peut être obtenu par un traitement automatique des résultats bruts de l'Analyse de Données, car ce qui lie les informations qui forment ce résultat, est issu d'un calcul numérique compliqué.

Par exemple, en analyse discriminante, on cherchera la règle de décision la plus courte qui classe le plus grand nombre possible d'exemples. On obtiendra, par exemple, un ensemble de n descripteurs qui classent correctement 90% des exemples fournis. Mais on ne se souciera pas de savoir quels sont les descripteurs qui font que 10% des exemples ne sont pas pris en compte par la règle de décision. On ne peut donc pas comprendre directement pourquoi les n descripteurs classent presque totalement les exemples fournis. Pour obtenir cette compréhension, il faut revenir sur la sémantique des descripteurs.

L'Analyse de données possède donc le même inconvénient que les méthodes analytiques d'acquisition de la connaissance pour les Systèmes à Base de Connaissance : elle fait intervenir un intermédiaire humain.

Dans le paragraphe suivant, nous examinons donc une autre voie : l'Apprentissage Automatique, qui permet d'acquérir une connaissance sans intermédiaire humain.

2.5 L'AUTOMATISATION DE L'ACQUISITION

Un bref historique de "l'apprentissage pour les ordinateurs" est donné dans [Kodratoff 86] : il remonte à la fin des années 50 avec le *perceptron* de Rosenblatt et le joueur de dames de Samuel, pour ensuite arriver aux années 60 où l'on se préoccupa alors d'apprentissage symbolique, orienté vers l'acquisition de concepts, de connaissances structurées. Dans les années 70, une nouvelle approche de l'apprentissage vit le jour : l'approche "intelligence artificielle" qui vise l'explicabilité dans la base de connaissance formée.

Le défaut du joueur de dames de Samuel, résidait dans le fait que l'apprentissage dans ce programme consistait en un ajustement de coefficients pondérant toute une série de paramètres : toute la connaissance était en fait contenue dans la définition de ces paramètres, plus que dans les valeurs numériques associées ([Kodratoff 86]). Le réel apprentissage était donc fait par Samuel lui-même dans son choix des paramètres.

Ensuite, dans les essais d'apprentissage symbolique des années 60, les systèmes capables d'apprentissage contenaient en fait *implicitement* une grande quantité d'informations. MetaDendral, par exemple, est capable d'engendrer des règles expliquant la spectrométrie de masse, mais il est dédié à ce problème ([Carbonell & al. 84]). Cette information, spécifique au domaine d'application, n'est donc pas accessible. L'approche "Intelligence Artificielle" vise à expliciter toute cette information pour ensuite pouvoir l'utiliser dans les mécanismes d'explication, voire même la modifier. C'est cette voie-là que nous allons explorer dans les chapitres qui suivent, et que l'on nomme "Apprentissage Symbolique Automatique" (en anglais : *Machine Learning*).

2.5.1 Un modèle simple d'apprentissage automatique

Tentons en premier lieu de définir ce que peut être "l'apprentissage automatique".

Une définition simpliste, proposée par Simon, et reprise par Cohen et Feigenbaum ([Cohen-Feigenbaum 82]) est "tout processus par lequel un système améliore sa performance". Mais cette définition est beaucoup trop générale, comme l'atteste la célèbre phrase de Tonimoto :

*" Clearly wine improve with time,
but nobody would call such an improvement learning "*
(Steve Tonimoto in [Michalski 86])

Cohen et Feigenbaum proposent également de définir l'apprentissage automatique comme étant un processus permettant "l'acquisition d'une connaissance explicite" ([Cohen 82]). Nous préférons cette dernière formule, car à notre connaissance, aucune autre meilleure définition synthétique n'existe pour l'instant.

Afin de nous doter d'un vocabulaire sans équivoque, nous adopterons aussi le modèle général de l'apprentissage automatique de Cohen et Feigenbaum ([Cohen 82]) (cf. figure 2.5), qui, bien que primitif, nous permettra par la suite de présenter quelques classifications des systèmes d'apprentissage.

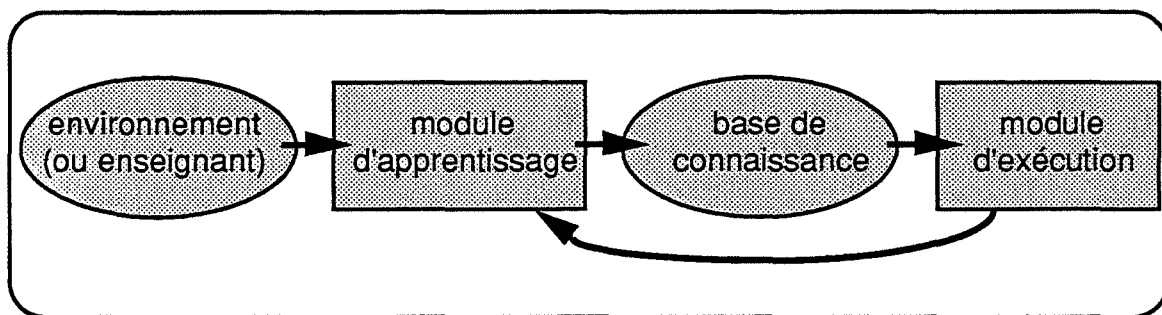


fig. 2.5 : un modèle général d'apprentissage automatique

L'environnement (ou enseignant) : c'est le facteur le plus important dans la conception du système d'apprentissage, par le niveau et la qualité des informations qu'il fournira. Plus l'information sera de bas niveau (i.e. très spécifique) plus le module d'apprentissage devra la généraliser en ignorant des détails, pour obtenir des descriptions utilisables par le module d'exécution dans un large éventail de situations. Inversement, si l'information fournie est trop générale (de trop haut niveau), le module d'apprentissage devra la spécialiser pour que le module d'exécution puisse l'utiliser dans des cas particuliers. La qualité de l'information aura un effet sur la complexité du module d'apprentissage (exemples classés ou fournis sans ordre, données erronées ou non, ...).

Le module d'apprentissage : il utilise les informations fournies par l'environnement (ou l'enseignant) pour améliorer la base de connaissance. Laquelle base de connaissance est utilisée par le module d'exécution (le système expert par exemple) pour accomplir sa tâche.

Eventuellement, l'information obtenue durant l'exécution peut être retournée au module d'apprentissage.

Pour bien percevoir l'éventail des difficultés que soulève la réalisation de ce dernier module, et pour ensuite situer notre problème particulier, nous examinons ci-dessous quelques classifications d'algorithmes d'apprentissage automatique. De plus, ces différentes classifications nous amèneront à définir plusieurs concepts qui nous seront utiles par la suite.

2.5.2 Classifications des systèmes d'apprentissage

Le premier critère que nous allons examiner pour classer les systèmes d'apprentissage est la distinction entre apprentissage numérique et apprentissage symbolique, puis nous évoquerons les différences entre apprentissage déductif et apprentissage inductif. Ensuite nous emprunterons à Carbonell, Michalski et Mitchell, deux autres axes de classification : classification selon la stratégie d'apprentissage et classification selon le type de connaissance acquise ([Carbonell & al. 84]). Enfin, pour terminer, le cinquième axe que nous examinerons est la distinction récemment admise entre apprentissage par détection de similarités et apprentissage par recherche d'explications.

2.5.2.1 Apprentissage numérique ou symbolique

La distinction entre ces deux apprentissages se situe au niveau des types d'opérations mises en œuvre dans l'algorithme d'apprentissage. L'apprentissage numérique repose essentiellement sur des opérations de type numérique (l'Analyse de Données par exemple), alors que l'apprentissage symbolique repose essentiellement sur des opérations de type logique.

L'approche numérique (Analyse de Données, Réseaux de Neurones ...) vise à faire apparaître des descripteurs qui sont les meilleurs relativement à un critère. Pour cela, ces méthodes numérisent les données et effectuent ensuite des opérations sur ces données numérisées. L'inconvénient majeur de ces méthodes est, d'une part, l'appauvrissement considérable des données initiales lors de leur traduction en nombres et, d'autre part, que la sémantique des opérations numériques peut être très éloignée de celle des données initiales, avant numérisation. Mais, en revanche, les méthodes numériques sont généralement efficaces et résistantes aux données erronées.

Inversement, l'approche symbolique est moins efficace, et notamment peu résistante aux données erronées, mais elle opère sur une meilleure traduction des données initiales : les méthodes symboliques effectuent des opérations de généralisations sur une formalisation logique des données, qui permet de mieux prendre en compte leur sémantique et leur structuration. Néanmoins, dans les applications pratiques, des calculs numériques sont souvent nécessaires, lors du processus de sélection des hypothèses, notamment lorsqu'il faut prendre en compte d'éventuelles données erronées ou incomplètes. C'est typiquement le cas de Charade (cf. § 2.6.5) qui effectue un traitement statistique des données pour sélectionner des règles incertaines. C'est aussi le cas de Plage ([Boucheron & al. 86]) qui utilise une "logique majoritaire", à base de traitements statistiques. ID3 (cf. § 2.6.2) utilise aussi une fonction numérique pour construire des arbres de décisions de faible profondeur. Etc... En général, un critère numérique, pour la sélection des hypothèses, produit de meilleurs résultats qu'un critère logique, en évitant la sur-spécialisation ([Gascuel-Caroux 91]).

Dans la suite de ce chapitre, nous ne considérerons que l'approche symbolique, c'est à dire uniquement des méthodes dont l'essentiel du processus d'apprentissage est à base d'opérations logiques. Cette approche constitue ce que l'on nomme l'Apprentissage Symbolique Automatique (ASA).

Au sein de l'ASA, on distingue généralement plusieurs types d'apprentissage selon le degré d'induction qu'ils sont capables d'effectuer. Avant de passer en revue ces types d'apprentissage, examinons ce qu'est l'apprentissage déductif, en opposition à l'apprentissage inductif.

2.5.2.2 Apprentissage déductif ou inductif

Un apprentissage est dit déductif lorsque l'environnement donne explicitement toute l'information nécessaire pour la construction ou le raffinement de la base de connaissance. Le module d'apprentissage effectue alors des inférences déductives, qui préservent la véracité, et fournit ses conclusions à la base de connaissance. Cela inclut la reformulation, la compilation, la création de macro-opérateurs, le "caching", l'agglomération, etc...

L'apprentissage est qualifié d'inductif lorsque l'information qui est fournie par l'environnement peut parfois l'être sous une forme implicite. Par exemple, si le module d'apprentissage fait intervenir une généralisation de l'information fournie par l'environnement et une sélection du résultat le plus plausible, alors on est en présence d'apprentissage inductif ([Michalski 86]).

Le but de l'apprentissage inductif étant de fournir la base de connaissance en descriptions de concepts, ces descriptions peuvent être ([Diettrich 83]) :

- *caractéristiques* : il s'agit de décrire une classe d'objets, souvent par rapport à d'autres classes d'objets,
- *discriminantes* : il s'agit de décrire une classe d'objets uniquement dans le contexte d'un ensemble fixe de plusieurs autres classes,
- *taxinomiques* : il s'agit de décrire une classe d'objets qui partage cette classe en sous-classes.

Néanmoins, Michalski remarque que les systèmes d'Intelligence Artificielle actuels ne sont en fait que déductifs, puisqu'ils ne sont capables que de dresser des conclusions à partir de la connaissance qu'on leur a fournie; et qu'ils ne sont pas capables d'acquérir ou d'engendrer d'eux-mêmes une nouvelle connaissance ([Michalski 86]). Des systèmes qui seraient capables de cela seraient alors qualifiés d'"inventifs", mais on ne doit pas s'attendre à rencontrer bientôt de tels systèmes ([Kodratoff 86]).

Nous admettons, quant à nous, qu'un algorithme d'apprentissage est inductif lorsqu'il est capable d'effectuer des généralisations : c'est à dire lorsqu'à partir de cas, il peut engendrer des lois plus générales. En fait, tous les algorithmes ne sont pas purement déductifs ou purement inductifs, comme le montre la classification ci-dessous.

2.5.2.3 Classification selon la stratégie

Cette classification, qui détaille la précédente, est faite selon la quantité d'inférences que le module d'apprentissage effectue sur les informations qui lui sont fournies. Dans tous les cas, le module d'apprentissage transforme une information fournie par l'environnement (ou l'enseignant) en une nouvelle information directement utilisable par le module d'exécution. Les systèmes d'apprentissage sont présentés ci-dessous selon l'ordre croissant du nombre d'inférences que doit effectuer le module d'apprentissage ([Carbonell & al. 84], [Cohen 82], [Michalski 86]).

Apprentissage par cœur :

Ce type d'apprentissage est aussi appelé "apprentissage par mise en œuvre directe" ([Lesaffre 89]). L'information fournie par l'environnement est directement utilisable par le module d'exécution. Aucune inférence ou autre transformation de la connaissance n'est requise de la part du module d'apprentissage.

Apprentissage par instruction :

L'information fournie est trop vague et doit être transformée pour être lisible par le module d'apprentissage. Pour Cohen et Feigenbaum, le module d'apprentissage doit même détecter les données manquantes et le cas échéant demander plus de détails ([Cohen 82]). Les transformations sont la sélection et la reformulation.

Apprentissage par analogie :

Il consiste en l'utilisation d'une connaissance relative à un concept (ou une tâche) donné(e) pour l'identification d'un autre concept (ou l'exécution d'une autre tâche) qui présente de fortes similarités avec le concept (ou la tâche) précédent(e). Cette apprentissage est à la fois inductif et déductif.

Apprentissage à partir d'exemples :

C'est le type même de l'apprentissage inductif : à partir d'exemples (et éventuellement de contre-exemples) d'un concept, fournis par l'environnement, le module d'apprentissage construit des descriptions plus générales de ce concept. En d'autres termes, à partir d'une définition en *extension* d'un concept, le module d'apprentissage en fournit une description en *intention* (i.e. en *compréhension*).

Apprentissage à partir d'observations :

Il s'agit d'un autre apprentissage inductif qui nécessite encore plus d'inférences que l'apprentissage à partir d'exemples car il n'est pas supervisé par l'enseignant : contrairement à l'apprentissage à partir d'exemples, aucune classe n'est fournie et le module d'apprentissage doit regrouper ses observations et "découvrir" les descriptions de ces regroupements. Cet apprentissage comprend la classification conceptuelle que l'on retrouve en analyse de données.

Synthèse :

Les algorithmes d'apprentissage purement déductifs ne sont guère intéressants pour résoudre le problème du goulot d'étranglement de l'acquisition de la connaissance. En effet, ils supposent que l'environnement fournit explicitement toute l'information nécessaire pour construire la base de connaissance. Or c'est justement cette explicitation de la connaissance qui constitue le goulot d'étranglement. Nous nous intéresserons donc uniquement aux algorithmes qui sont capables d'effectuer des opérations inductives.

Parmi ceux-ci, nous trouvons l'apprentissage par analogie, l'apprentissage à partir d'exemples, et l'apprentissage à partir d'observations. En fait, ce sont trois paradigmes d'apprentissage, plutôt que trois classes.

Par exemple, un apprentissage ne sera jamais purement par analogie, car cela supposerait que l'on connaisse a priori tous les types de concepts que le système rencontrera. De même, dans un apprentissage à partir d'exemples, le fait de vouloir former des concepts intermédiaires, entre les paramètres qui caractérisent les exemples et les concepts fournis par l'environnement, rapproche cet apprentissage de celui à partir d'observations. Il faut donc repérer l'opération la plus importante dans l'algorithme d'apprentissage, pour pouvoir qualifier celui-ci, mais ne pas exclure les autres formes d'induction.

Pour notre application (assainissement des eaux usées), c'est un algorithme à base d'apprentissage à partir d'exemples qui est le plus adapté. Il semble en effet assez facile de demander à un expert de fournir des cas qu'il a résolus sous la forme d'un couple (caractéristiques de la parcelle, solution préconisée), et de former, à partir de ces cas, des descriptions plus générales de l'applicabilité des systèmes d'assainissement.

Un autre critère important, pour le module d'apprentissage, est le formalisme avec lequel on désire coder la base de connaissance. Ce critère est important, car de lui dépendra en partie la complexité du module d'apprentissage.

2.5.2.4 Classification selon le type de connaissance acquise

La complexité du module d'apprentissage dépend non seulement de l'information fournie par l'environnement (forme et qualité de cette information), mais dépend aussi grandement de la forme que doit avoir l'information à fournir au module d'exécution. Certains formalismes, par leur expressivité plus grande, permettront une meilleure explicitation de la connaissance, mais en revanche nécessiteront des algorithmes plus puissants. La base de connaissance peut s'exprimer en utilisant différentes représentations ([Carbonell & al. 84]), dont les principales sont :

Expressions algébriques :

Apprendre, dans ce contexte, consiste alors à ajuster des paramètres ou coefficients numériques dans des expressions algébriques. C'est le cas du perceptron de Rosenblatt ou du joueur de dames de Samuel. On ne peut alors pas prétendre à une approche Intelligence Artificielle de l'apprentissage car le module d'exécution peut difficilement fournir des explications sur son comportement. L'apprentissage réside en fait dans l'invention de ces paramètres ou coefficients plus que dans leur valeur numérique ([Kodratoff 86]).

Règles de production :

Il s'agit de paires 'condition-action' ($C \Rightarrow A$) où C est un ensemble de conditions et A une séquence d'actions. A cause de leur simplicité et de leur interprétation aisée, elles sont très largement utilisées pour la représentation des connaissances dans les systèmes d'apprentissage. Les quatre opérations de base que l'on peut leur appliquer sont la création, la généralisation, la spécialisation et la composition. Leurs performances dépendent du formalisme choisi pour exprimer chaque partie (C et A).

Graphes et réseaux :

Ils consistent en nœuds qui représentent des objets, des concepts, des événements, et en liens entre ces nœuds qui représentent leurs interrelations. L'interprétation de ces structures dépend uniquement du programme qui les manipule, il n'y a aucune convention. Leurs performances dépendent du formalisme choisi pour décrire les nœuds et les liens, ainsi que de la sémantique qu'on attribue à ces nœuds et à ces liens. Ils peuvent être, entre autres :

- des arbres de décisions. Les nœuds d'un arbre de décision correspondent aux attributs des objets sélectionnés, et les branches aux valeurs alternatives prédéterminées de ces attributs. Les feuilles de l'arbre correspondent aux ensembles d'objets d'une même classe. Cette forme de représentation de la connaissance ne permet pas de représenter des concepts intermédiaires et limite donc fortement l'explicabilité du module d'exécution.

- des "frames". C'est une structure de données qui inclue des informations déclaratives et procédurales. Par exemple, dans le programme AM de Lenat, un concept est représenté sous forme d'un "frame" qui contient la définition du concept, ses exemples positifs ou négatifs connus, ses liens avec d'autres concepts qui sont des spécialisations et des généralisations, sa validité, et quelques autres choses ([Cohen 82]).

- des taxinomies. Une description taxinomique est la description d'une classe d'objets qui partage cette classe en sous-classes. L'apprentissage par observations (notamment la classification conceptuelle) peut se ramener à une recherche de "clés d'identification". Une représentation de la connaissance sous forme de taxinomies facilite alors la tâche d'apprentissage.

Synthèse :

Mis à part les expressions algébriques, en ASA on trouve tous les autres formalismes. Les règles de production ont l'avantage de pouvoir être utilisées directement par un module d'exécution du type "moteur d'inférences". Mais elles ne permettent pas d'explicitier la sémantique de la relation d'implication. Est-ce une relation de cause à effet, une relation de précedence, une relation de généralisation, ... ?

En revanche, les réseaux et les graphes, par leur grande généralité, permettent d'explicitier les liens qui existent entre les nœuds, et permettent donc de faire cohabiter plusieurs types de liens.

Une dernière et récente distinction, au sein des algorithmes d'ASA, permet de mettre l'accent sur un point très important : la disponibilité éventuelle d'une théorie sur le domaine.

2.5.2.5 Apprentissage par détection de similarités ou par recherche d'explications

Pendant l'International Workshop on Machine Learning de 1985 a été admise la distinction, au sein des apprentissages inductifs, entre "apprentissage par détection de similarités" (ou SBL, i.e. Similarity Based Learning) et "apprentissage par recherche d'explications" (ou EBL, i.e. Explanation Based Learning - [Kodratoff 86]). En SBL, on apprend en repérant des similarités entre les exemples et des dissimilarités entre les exemples et les contre-exemples, alors qu'en EBL on apprend à partir d'explications dérivées de l'analyse d'un exemple ou d'un contre-exemple, et grâce à une théorie du domaine.

L'EBL s'appliquera donc particulièrement bien aux domaines où la connaissance experte repose, au moins en partie, sur une théorie bien formalisée. C'est le cas par exemple pour les mathématiques, la physique, la médecine, ... Ce n'est pas le cas pour le domaine qui nous intéresse : un problème d'aménagement, et plus particulièrement l'assainissement des eaux usées.

Nous nous situons donc dans le cas de l'apprentissage par détections de similarités, et d'autre part, comme nous l'avons vu au § 2.5.2.3, dans le cadre d'un apprentissage centré sur l'apprentissage à partir d'exemples. Nous allons donc, dans les pages qui suivent, détailler les problèmes particuliers que soulève ce type d'apprentissage.

2.5.3 L'apprentissage par détections de similarités

L'apprentissage par détections de similarités (SBL) est un apprentissage inductif. Rappelons que l'apprentissage inductif (qui regroupe l'apprentissage par analogie, l'apprentissage à partir d'exemples et l'apprentissage à partir d'observations) est caractérisé par le fait qu'il est capable d'effectuer des opérations inductives, c'est à dire des généralisations.

Le principal problème des algorithmes inductifs, est de savoir limiter quantitativement la production des descriptions de concepts, et notamment lors des opérations de généralisation. En effet, supposons que nous observons les deux scènes décrites par la figure 2.6.

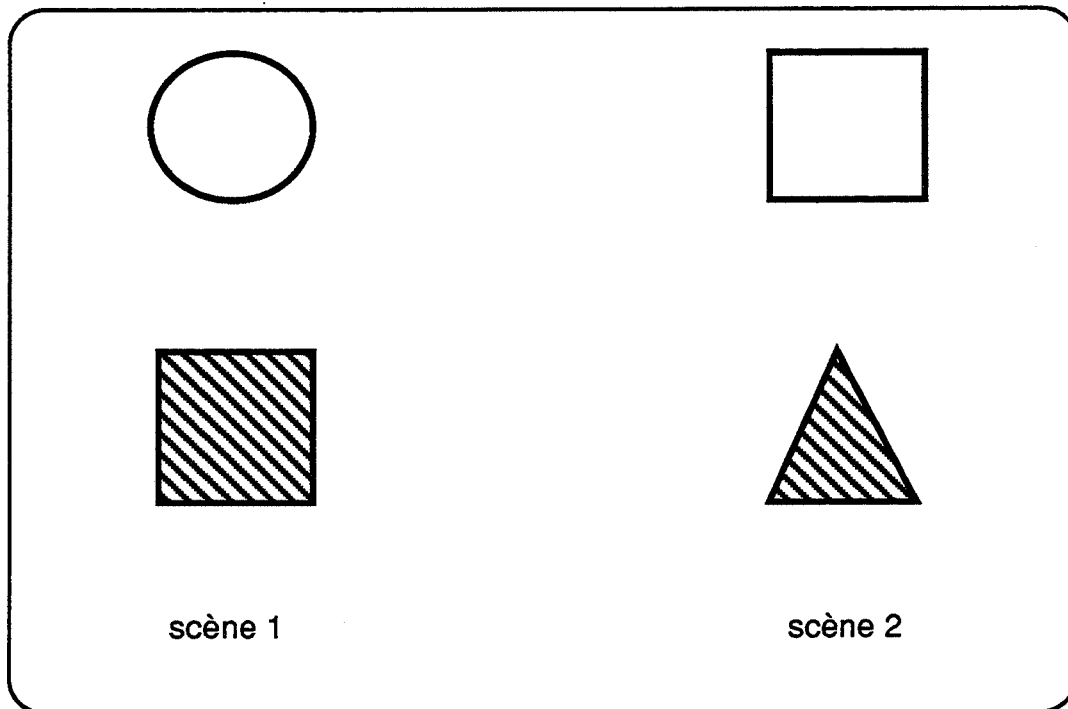


fig. 2.6 : deux scènes à généraliser ([Ganascia 88])

Même pour un cas aussi simple, le nombre de descriptions possibles, pour caractériser ces deux scènes, est très grand. On peut proposer : il y a deux objets l'un sur l'autre, l'objet du dessous est un polygone, il y a deux objets dont un carré, il y a un polygone rayé, etc. Le problème qui se pose alors est de choisir quelles sont les descriptions pertinentes. Il faut donc limiter l'espace des généralisés possibles, au moyen de contraintes que l'on regroupe sous le concept de "connaissance sur le domaine".

2.5.3.1 La connaissance sur le domaine

"Connaissance sur le domaine" est le terme généralement employé pour traduire "background knowledge". On le trouve aussi traduit par "connaissance a priori" ou "connaissance de fond". C'est un ensemble d'informations qui a pour conséquence de restreindre la production des descriptions de concepts. Cette connaissance est exprimée, notamment, dans le choix des règles de généralisation et de spécialisation, dans le langage de description des exemples et dans celui de la base de connaissance. Mais elle peut aussi être contenue dans la définition de différents critères (tels que critère de consistance, ou critère de préférence). On convient généralement de scinder la connaissance sur le domaine en deux ([Russel-Grossof 87]) :

- d'une part le critère de consistance, que nous examinons ci-dessous, et

- d'autre part toutes les autres formes de connaissance qui permettent de restreindre le nombre de descriptions, que l'on regroupe sous le terme de "learning bias".

Complétude, discrimination et consistance :

Les avis divergent quant à la définition de ces trois concepts. On appellera

- exemple positif d'un concept, un exemple appartenant à ce concept, et
- exemple négatif d'un concept, un exemple n'appartenant pas à ce concept, c'est à dire un contre-exemple de ce concept.

Mitchell et Utgoff définissent une description consistante comme étant une description qui recouvre tous les exemples positifs et aucun exemple négatif ([Mitchell 82], [Utgoff 86]).

Pour Michalski la condition de complétude est remplie lorsque la description d'un concept recouvre tous les exemples positifs de ce concept, et la condition de consistance lorsque la description du concept ne recouvre aucun exemple négatif ([Michalski 83]). Kodratoff adopte les mêmes définitions que Michalski mais remplace le terme "consistance" par "discrimination" ([Kodratoff 86]), qui reconnaissons-le, est plus évocateur du concept décrit.

Nous adoptons donc la terminologie suivante : en reprenant les termes et définitions de Kodratoff (complétude et discrimination), nous définissons la consistance comme étant la conjonction de la complétude et de la discrimination. Nous retrouvons en fait la définition de la consistance selon Mitchell et Utgoff. Ce critère de consistance permet de limiter raisonnablement le nombre de descriptions engendrées : le moins qu'on puisse attendre de la description d'un concept, c'est bien qu'elle couvre les exemples positifs et qu'elle ne couvre pas les exemples négatifs.

D'autres auteurs utilisent des critères différents. Pottier, par exemple, utilise un critère de cohérence qui impose, en plus, aux contre-exemples d'un concept de décrire la négation du concept ([Nicolas 87], [Pottier 86]). Ce critère est plus fort que le nôtre et risque d'éliminer des descriptions de concepts dont la négation n'a pas de sens. Nicolas, quant à lui, utilise un critère plus faible que le nôtre, où il permet (mais n'oblige pas) à la négation du concept considéré d'être consistante avec les contre-exemples.

Toutes les autres connaissances sur le domaine, qui permettent de limiter le nombre de descriptions engendrées par le module d'apprentissage, sont regroupées dans le concept de "learning bias", que l'on traduira par "biais inductif", ou plus simplement par "biais".

"Learning bias" :

Le "*learning bias*" ou "biais inductif" est la connaissance sur le domaine, non contenue dans le critère de consistance et qui permet de restreindre le nombre de descriptions engendrées par le module d'apprentissage. Nous verrons à la fin de ce chapitre, avec le système CHARADE, qu'il est possible de représenter explicitement certaines formes du biais inductif. Cela permet alors de pouvoir paramétrer le module d'apprentissage et de l'adapter au domaine sur lequel on veut apprendre. Dans CHARADE, le biais inductif peut revêtir trois formes explicites : le formalisme de représentation de la connaissance, le langage de description des entrées du module d'apprentissage, le langage de description de la connaissance acquise, et enfin des contraintes syntaxico-sémantiques.

Le biais inductif peut aussi se trouver dans des critères permettant d'évaluer les descriptions engendrées et d'en éliminer certaines. Michalski appelle cet ensemble de critères les critères de préférence et propose ([Michalski 83]) :

- la simplicité pour la compréhension humaine, mesurée par exemple par le nombre de descripteurs et le nombre d'opérateurs utilisés dans la description,
- le degré de correspondance entre les observations et la description (degré de généralisation par exemple),
- le coût (en temps de calcul) pour mesurer les valeurs des descripteurs dans la description,
- le coût (en temps de calcul) pour évaluer la description,
- la mémoire nécessaire pour stocker la description,
- la quantité d'informations nécessaire pour coder la description en utilisant des opérateurs prédéfinis.

L'approche Intelligence Artificielle de l'apprentissage automatique vise à expliciter le plus possible les connaissances utilisées dans les algorithmes. Ceci est notamment vrai pour le biais inductif, composante très importante du module d'apprentissage. Nous nous efforcerons donc, dans notre système, d'expliciter au maximum ce biais, de façon à ce que le dialogue entre l'expert et notre système soit le plus limpide possible ... pour l'expert !

Les deux autres grandes caractéristiques des systèmes d'apprentissage inductif, et donc a fortiori des systèmes d'apprentissage par détection de similarités, est leur aptitude à acquérir une "bonne" connaissance malgré l'introduction de données erronées, et leur aptitude à améliorer incrémentalement la base de connaissance. La première de ces caractéristiques, que nous examinons ci-dessous, est souvent appelée "résistance au bruit".

2.5.3.2 La prise en compte des données erronées

La plupart des systèmes d'apprentissage inductif supposent que chaque information soumise au module d'apprentissage ne comporte aucune erreur. C'est à dire qu'ils supposent que l'enseignant est bienveillant et précis lorsqu'il fournit ses informations. Ces suppositions sont trop restrictives pour des applications réelles, et désormais les systèmes d'apprentissage tentent d'explorer la possibilité d'exemples mal étiquetés ou non étiquetés (dans le cas de l'apprentissage à partir d'exemples), de données mal décrites (certains attributs non spécifiés par exemple), ou d'erreurs de mesure sur des valeurs d'attribut ([Carbonell 89]).

L'approche traditionnelle pour traiter le bruit (i.e. les données erronées) consiste à associer des coefficients numériques aux descriptions des concepts (voir CHARADE par exemple). Cette approche ne résout pas tous les problèmes relatifs au bruit et engendre elle-même d'autres problèmes. Par exemple un expert humain aura du mal à se retrouver dans un grand ensemble de telles règles et, par conséquent, à les valider ([Manago-Kodratoff 87]).

Il convient donc de chercher à prévenir le bruit, plutôt qu'à le traiter.

Pour prévenir le bruit, tout d'abord, le choix du langage de description des exemples doit être fait soigneusement afin que les exemples puissent être exprimés sans ambiguïté : de nombreux problèmes relatifs au bruit pourront être résolus en améliorant ce langage.

Une autre source de bruit est le manque d'exemples, qui fera que, dans le cas, par exemple, des Version Spaces de Mitchell (cf. § 2.6.1), les ensembles S et G ne se rejoindront pas.

En résumé, le bruit peut revêtir trois formes ([Manago 87], [Ganascia 88]) :

- informations erronées,
- manque d'information et mauvais langage de description,
- données inexactes, ou difficilement observables (donc difficilement décrivables),

et pour mieux prévenir le bruit, Manago et Kodratoff préconisent entre autres de prévoir des concepts intermédiaires judicieux (entre les instances et les concepts finaux recherchés) et de collecter les exemples au moyen de questionnaires. Gams, quant à lui, souligne l'intérêt des règles de confirmation dans la prévention du bruit : plusieurs sources d'information sont consultées et leurs avis sont comparés.

Pour notre application, et grâce aux Réseaux de Dépendance Perçue, nous adopterons une méthode qui consiste à consulter plusieurs experts, et à dialoguer avec chacun d'eux, de façon à leur faire prendre conscience des conséquences de l'introduction de leurs exemples, et ainsi à les faire réfléchir sur la validité de leurs exemples. Pour cela on introduira des concepts intermédiaires de façon à mieux décomposer le travail du système.

Le problème des données bruitées est plus difficile à résoudre dans le cas de méthodes d'apprentissage dirigées par les données, que dans celles dirigées par le modèle.

Les méthodes dirigées par les données (en anglais : "*bottom-up*" ou "*data driven*") consistent à raffiner progressivement des descriptions au fur et à mesure que des informations sont fournies, tandis que les méthodes dirigées par le modèle (en anglais : "*top-down*" ou "*model driven*") consistent à engendrer des descriptions possibles et d'en choisir quelques-unes qui sont les meilleures en les testant sur l'ensemble des informations fournies.

Les méthodes dirigées par les données ont donc de la difficulté à s'affranchir du problème des données bruitées, puisqu'elles recherchent la consistance avec tous les exemples fournis. Afin de résoudre ce problème, il faut affaiblir la condition de consistance.

Par exemple, pour les Versions Spaces (cf. § 2.6.1), Mitchell propose de maintenir plusieurs S et plusieurs G : S₀ est consistant avec tous les exemples, S₁ avec tous sauf 1, S₂ avec tous sauf 2, etc., de même pour G₀, G₁, G₂, etc., et lorsque G₀ et S₀ se croisent, on en conclut qu'il n'existe pas de concept dont la description soit consistante avec tous les exemples, et on passe à (S₁,G₁), et ainsi de suite ([Mitchell 82]). L'inconvénient de cette méthode est qu'elle nécessite beaucoup de mémoire pour stocker tous les S et G nécessaires ([Cohen 82]).

En résumé, d'un côté, les méthodes dirigées par le modèle peuvent résister aux données erronées, mais, en contrepartie, elles obligent à parcourir, à chaque étape, toutes les descriptions possibles et ensuite les soumettre à tous les exemples ([Diettrich 83]). De l'autre côté, les méthodes dirigées par les données peuvent apprendre incrémentalement ([Cohen 82]), mais ont plus de difficulté de s'affranchir des données bruitées.

Cette distinction met l'accent sur la difficulté de l'incrémentalité de l'apprentissage. Nous allons examiner cette dernière caractéristique importante des systèmes d'apprentissage inductif, indispensable pour certains cas d'application.

2.5.3.3 Incrémentalités

L'incrémentalité du module d'apprentissage peut être abordée de deux façons : soit d'un point de vue externe, soit d'un point de vue interne; selon que l'on s'intéresse à l'utilisation du système d'apprentissage, ou à son fonctionnement.

Incrémentalités d'utilisation :

D'un point de vue externe, on considère que l'apprentissage est incrémental s'il se fait au fur et à mesure que les informations sont fournies au système, c'est à dire lorsque le système

d'apprentissage fournit une nouvelle base de connaissance à chaque fois qu'une information lui est soumise. Dans le cas contraire, le module d'apprentissage n'est déclenché que lorsque toutes les informations ont été fournies.

L'apprentissage est forcément incrémental dans certains cas, notamment lorsque le système reçoit continuellement des informations (il n'est donc pas possible d'attendre de les avoir toutes), ce qui est le cas lorsque la connaissance du domaine d'application évolue et que l'on désire que le système suive cette évolution au fur et à mesure que de nouvelles situations lui sont présentées.

L'apprentissage incrémental peut alors se faire de deux façons : soit le système oublie (i.e. il ne mémorise pas) l'information lorsqu'elle est traitée, soit il réapprend à chaque fois à partir de toutes les informations déjà fournies ("learning from scratch", [Lebowitz 86]).

L'avantage du premier cas de figure est que le système n'aura pas à organiser la mémorisation d'un grand nombre de situations. En revanche, le traitement d'une situation s'accompagne forcément d'une perte d'information (la seule façon de ne perdre aucune information est de mémoriser la situation dans sa totalité), et l'on ne pourra pas revenir à cette situation si le besoin s'en fait sentir. Cela peut poser des problèmes lorsque les informations que le système reçoit sont susceptibles d'être bruitées, ou lorsque le domaine d'application est évolutif.

Le deuxième cas de figure ne possède pas ces inconvénients puisque l'on conserve à la fois l'information agglomérée et l'information initiale dans sa totalité. En revanche, dans le cas où les informations fournies sont très nombreuses, il peut y avoir un problème de mémorisation, et de temps de calcul.

Incrémentalités de fonctionnement :

D'un point de vue interne, on s'attache au fonctionnement même du module d'apprentissage, et on considère que l'apprentissage sans oubli et à partir de tous les exemples ("*learning from scratch* ") n'est pas incrémental, car rien ne le distingue, d'un point de vue interne, de l'apprentissage non-incrémental. Sainte Marie distingue alors trois sortes d'incrémentalités ([Sainte Marie 90a]) : l'incrémentalité de construction, l'incrémentalité d'usage, et l'incrémentalité de représentation, qui mettent l'accent sur les différents avantages de l'incrémentalité :

Incrémentalité de construction : une nouvelle situation est intégrée à la base de connaissance par des modifications locales, sans en nécessiter la reconstruction complète, d'où un gain de temps de calcul.

Incrémentalité d'usage : la forme et les propriétés de la connaissance acquise et de la connaissance sur le domaine sont indépendantes de leur contenu, ce qui permet d'avoir une

base de connaissance utilisable à tout moment, et le module d'exécution peut donc commencer à travailler pendant que l'acquisition continue à se faire. Ceci est particulièrement utile lorsque le domaine d'application évolue.

Incrémentalité de représentation : le langage de représentation de la connaissance peut être modifié sans nécessiter la reconstruction de la base de connaissance. Ceci peut se révéler utile, notamment lorsqu'un nouveau descripteur est à prendre en compte.

Par la suite, lorsque nous utiliserons le terme "incrémental", ce sera en référence à l'incrémentalité de fonctionnement (point de vue interne). Si besoin est, nous préciserons de quelle incrémentalité il s'agit (de construction, d'usage ou de représentation).

Après avoir parcouru les principales caractéristiques des systèmes d'apprentissage inductif, et plus particulièrement les systèmes d'apprentissage par détections de similarités, nous allons examiner quelques-uns de ces algorithmes.

2.6 QUELQUES ALGORITHMES D'APPRENTISSAGE

Nous présentons, dans ce paragraphe, six algorithmes d'apprentissage par détection de similarités.

Le premier, l'algorithme d'élimination des candidats ([Mitchell 78], [Mitchell 82]) est un algorithme très simple d'apprentissage incrémental à partir d'exemples : il est présent ici à titre d'introduction. Ensuite nous présentons ID3 ([Quinlan 86]), un autre algorithme d'apprentissage à partir d'exemples, qui permet de construire des arbres de décision. Les deux suivants, COBWEB ([Fisher 87]) et UNIMEM ([Lebowitz 87]) sont des algorithmes d'apprentissage à partir d'observations : ils sont présentés ici pour la façon qu'ils ont d'organiser incrémentalement la mémorisation des concepts appris. Ensuite nous décrivons le système CHARADE ([Ganascia 88]) qui permet d'engendrer, par la détection de régularités empiriques sur l'ensemble d'apprentissage, des systèmes de règles (certaines ou incertaines) directement utilisables par un module d'exécution, en explicitant une partie du "*learning bias*". Enfin, les Réseaux de Dépendance Perçue ([Sainte Marie 89], [Sainte Marie 91]) sont une façon d'organiser la connaissance acquise qui permet un algorithme d'apprentissage incrémental. Cet algorithme est présenté au chapitre 4. Il possède l'avantage, d'une part, de fabriquer et entretenir incrémentalement la connaissance sous la forme d'un réseau directement utilisable par le module d'exécution, et d'autre part, d'autoriser un dialogue avec l'expert qui permettra de prévenir l'introduction de données erronées.

2.6.1 L'algorithme d'élimination des candidats

Mitchell a développé une méthode d'apprentissage incrémental à partir d'exemples (l'algorithme d'élimination des candidats, *Candidate Elimination Algorithm*, [Mitchell 78]) qui utilise les espaces de version (*Version Spaces*). Cette méthode repose sur l'existence d'un ordre partiel dans l'espace des descriptions de concepts.

Soient D1 et D2 deux descriptions d'un même concept, D1 est plus spécifique (ou, de façon équivalente, moins générale) que D2 si et seulement si D1 recouvre un sous-ensemble des instances recouvertes par D2. Cela nous définit une relation d'ordre (partiel en général) sur les descriptions d'un concept.

La description la plus générale est la description nulle (celle qui ne contient aucune restriction, et qui décrit donc tout). Les descriptions les plus spécifiques sont les exemples d'apprentissage eux-mêmes, représentés dans le langage d'expression des descriptions de concepts.

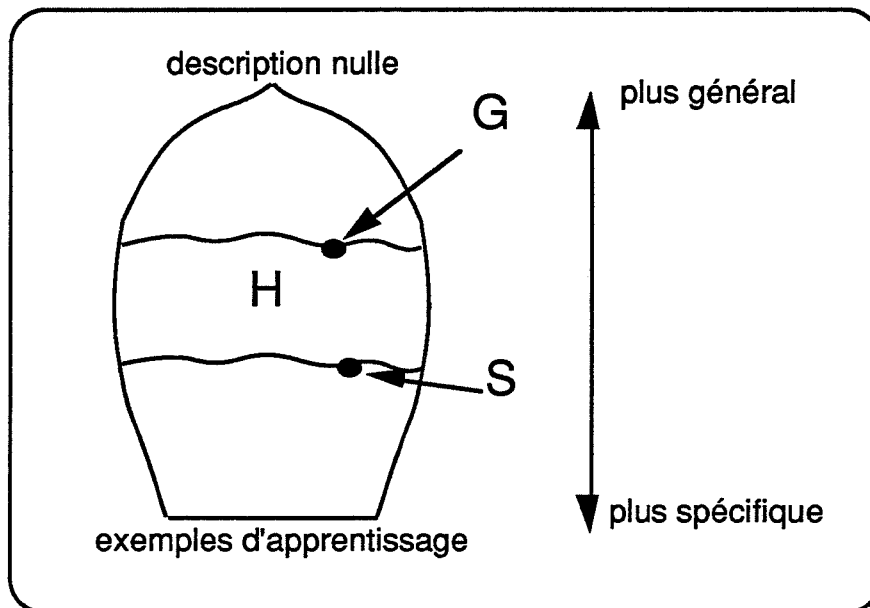


fig. 2.7 : Les espaces de versions de Mitchell

Mitchell a montré l'intérêt de cet ordre partiel pour représenter de façon compacte l'ensemble H des hypothèses plausibles. En effet, un ensemble de points dans un ordre partiel peut-être représenté par ces éléments les plus généraux et les plus spécifiques.

Ainsi l'ensemble H peut-être représenté par deux sous-ensembles (fig. 2.7) : l'ensemble G des éléments les plus généraux de H, et l'ensemble S des éléments les plus spécifiques de H. L'ensemble H des hypothèses plausibles est appelé l'espace des versions. Initialement H est l'espace entier de tous les concepts possibles (G contient la description nulle et S contient le premier exemple positif). Puis, au fur et à mesure que les exemples sont fournis, des concepts candidats sont éliminés de H. C'est l'algorithme d'apprentissage par élimination des candidats : un exemple positif force la généralisation (des descriptions très spécifiques sont enlevées de H : G est privé des concepts qui ne couvrent pas cet exemple positif, et les éléments de S sont généralisés le moins possible de façon à couvrir ce nouvel exemple positif); et inversement un exemple négatif force la spécialisation (des descriptions très générales sont enlevées de H : S est privé des concepts qui couvrent cet exemple négatif, et les éléments de G sont spécialisés le moins possible de façon à ne plus couvrir ce nouvel exemple négatif).

L'applicabilité de cet algorithme est limitée par la confiance dans les exemples d'apprentissage (présence de données erronées) et par le danger d'obtenir des ensembles S et G beaucoup trop grands (notamment lorsque le langage de description des concepts est trop complexe).

Une autre limitation est son impossibilité à apprendre des concepts disjonctifs, car ces concepts empêchent la généralisation : S ne contiendrait que la disjonction des exemples positifs reçus (sauf si le langage de description des concepts prévoit un nombre maximum de disjonctions dans une description).

Deux autres méthodes sont proches des espaces de versions : *interference matching* ([Hayes-Roth - Mac Dermott 77]), et *maximal unifying generalization* ([Vere 75]).

Nicolas, quant à lui, a repris l'algorithme d'élimination des candidats, et a décrit la généralisation sous un formalisme logique, dans un système nommé ALLY ([Nicolas 87]).

2.6.2 ID3

ID3 (*Induction of Decision Trees* – [Quinlan 86]), est un système inductif capable de fabriquer un arbre de décision à partir d'exemples, sans aucune limitation de domaine. A partir d'un ensemble d'exemples (objets décrits sous la forme d'un vecteur d'attributs, où chaque attribut prend une valeur dans un ensemble, de préférence petit, de valeurs discrètes mutuellement exclusives), et de deux classes mutuellement exclusives (par exemple : P pour exemples positifs et N exemples négatifs d'un concept que l'on cherche à décrire), le système recherche une règle

de classification qui permette de déterminer la classe d'un objet quelconque à partir des valeurs de ses attributs.

Pour cela, l'ensemble d'exemples est divisé en deux : un échantillon de base dont les éléments sont choisis au hasard pour construire l'arbre de décision, et les éléments restants qui sont utilisés pour tester l'arbre engendré. Si le test est positif, on en reste là; sinon une partie des mal-classés est ajoutée à l'échantillon de base pour former un nouvel ensemble d'apprentissage, afin de réitérer l'opération.

La procédure de construction de l'arbre de décision est la suivante : si tous les éléments de l'ensemble d'apprentissage n'appartiennent pas à la même classe : choisir un attribut A qui, selon son nombre de modalités, permettra de diviser l'ensemble d'apprentissage en sous-ensembles. Chaque sous-ensemble est formé des éléments de l'ensemble d'apprentissage qui prennent la même valeur pour l'attribut A. Puis il faut recommencer cette procédure pour chaque sous-ensemble obtenu, jusqu'à ce que l'on obtienne uniquement des ensembles ne contenant que des éléments d'une même classe. Pour représenter ce partitionnement de l'ensemble d'apprentissage, on utilise un arbre dont les feuilles sont les ensembles obtenus, et sont étiquetées par le nom de la classe correspondante. Chaque nœud non terminal de cet arbre est étiqueté par un attribut choisi durant le processus de construction; et les arcs, issus d'un nœud N, portent les différentes valeurs de l'attribut étiquetant le nœud N.

A chaque étape de la construction de l'arbre, se pose le problème du choix d'un attribut : ce choix se fait grâce à une fonction numérique à base d'entropie, notion inspirée de la théorie de l'information. Afin que l'arbre soit le plus lisible possible, cette fonction vise à en diminuer la profondeur : pour cela elle indique l'attribut qui minimisera le nombre de tests nécessaires pour classer un exemple.

Malgré cela, les arbres, lorsqu'ils sont trop grands, sont difficiles à lire et ne font pas apparaître de concepts intermédiaires. De plus, ils ne permettent pas de représenter de liens entre les attributs ou valeurs d'attributs : ils ne peuvent donc utiliser aucune connaissance sur le domaine. Enfin, ID3 ne permet pas d'apprentissage incrémental. Ci-dessous sont présentés deux algorithmes (COBWEB et UNIMEM) qui permettent un apprentissage incrémental, et qui de plus prennent en compte les données bruitées.

2.6.3 COBWEB

COBWEB ([Fischer 87]) est un système d'apprentissage incrémental à partir d'observations (découverte de classes) dont le but est de résumer des données sous une forme compréhensible, par la formation de concepts.

COBWEB incorpore incrémentalement des objets dans un arbre de classification où chaque nœud est un concept probabiliste qui représente une classe d'objets. Cette incorporation consiste à classer l'objet en descendant l'arbre de classification selon un chemin indiqué par une mesure heuristique : "l'utilité catégorielle", qui prend en compte les similarités intra-classes et les dissimilarités inter-classes sous forme de produit de probabilités conditionnelles. A chaque étape du parcours de l'arbre, cette heuristique indique aussi l'opération à déclencher et sur quel(s) nœud(s) la déclencher. Les opérations disponibles sont :

- placer un objet dans une classe déjà existante : pour cela COBWEB essaie chaque nœud du niveau courant et retient le meilleur au sens de l'utilité catégorielle en évaluant à chaque fois la partition qui serait obtenue si l'on incluait l'objet au nœud essayé.

- créer une nouvelle classe : la qualité de la partition obtenue en plaçant le nouvel objet dans la meilleure classe déjà existante retenue à l'opération précédente est comparée avec celle de la partition obtenue en créant une classe supplémentaire ne contenant que ce nouvel objet.

- regrouper deux classes en une seule : pour diminuer la combinatoire de ce choix, les deux nœuds qui sont testés sont les deux meilleurs pour héberger le nouvel objet (première opération). Les deux nœuds regroupés deviennent des fils du nœud ainsi créé.

- scinder un nœud en plusieurs : cela revient à supprimer un nœud et à remonter ses fils à son niveau. Le seul nœud testé est celui retenu à la première opération.

Les deux premières opérations rendent l'algorithme très sensible à l'ordre d'arrivée des exemples. Les deux dernières diminuent cette sensibilité grâce à leurs effets inverses et permettent une sorte de "remontée" dans l'algorithme général qui est du type "*hill climbing*".

CLASSIT ([Gennari-Langley-Fisher 89]), un successeur de COBWEB, n'enregistre pas des probabilités mais des distributions normales sous la forme d'une moyenne et d'un écart type.

2.6.4 UNIMEM

UNIMEM (*UNiversal MEMory* – [Lebowitz 87]), est un système d'apprentissage incrémental à partir d'observations (découverte de classes) qui est capable de traiter des domaines où les exemples sont très nombreux. Les descriptions fournies sont pragmatiques : elles ne décrivent pas exactement tous les exemples, ce qui permet d'appliquer UNIMEM à des problèmes

concrets où les données peuvent être incomplètes ou inconsistantes. Le but de ce système est de découvrir des liens entre concepts.

La connaissance acquise et apprise est stockée dans une mémoire à long terme organisée en hiérarchie de concepts (GBM : *Generalization Based Memory*), par recherche de similarités entre les exemples et l'abstraction de ces similarités. Chaque nœud de la hiérarchie représente un concept, étiqueté par un vecteur attribut/valeurs (qui est l'abstraction d'exemples, ou de parties d'exemples) et auquel sont éventuellement rattachés des exemples qui n'ont pu encore être rapprochés pour constituer un sous-nœud après abstraction. Chaque nœud est relié à un nœud père qui représente un concept plus général.

L'ajout d'un nouvel exemple dans la hiérarchie se déroule en deux étapes :

- sont d'abord recherchés tous les nœuds les plus spécifiques qui sont "proches" du nouvel exemple;
- puis, pour chacun de ces nœuds, une mise à jour est effectuée pour contenir le nouvel exemple : si un autre exemple, précédemment relié au nœud considéré, est suffisamment "proche", un sous-nœud est créé en généralisant ces deux exemples, sinon le nouvel exemple est simplement relié au nœud dans l'attente d'un autre exemple "proche".

On constate donc qu'un exemple peut être stocké à plusieurs endroits de la hiérarchie : cela signifie que les concepts découverts ne sont pas forcément disjoints.

On constate aussi qu'UNIMEM fabrique une description sur la base de deux exemples uniquement. Cela peut créer des descriptions sur-spécifiques notamment dans les domaines où les exemples sont décrits par un très grand nombre de caractéristiques. Pour y remédier, il est alors possible d'intégrer un système de gestion de coefficients de confiance attachés à chaque caractéristique de chaque nœud de la hiérarchie. De même, lorsqu'un concept est trop général (il ne fournit plus assez d'informations), on peut décider d'effacer le nœud correspondant, au risque de perdre certains exemples qui lui sont attachés.

ADECLU ([Decaestecker 89a], [Decaestecker 89b]), part d'UNIMEM et utilise un critère d'adéquation inspiré de COBWEB. Pour chaque concept, deux règles sont maintenues :

- l'ensemble des sous-concepts forme une partition du concept père (exclusion mutuelle),
- les caractéristiques d'un concept sont celles qui n'apparaissent pas chez le concept père.

De plus la représentation des connaissances qui est choisie dans ADECLU fait intervenir une pondération de chaque caractéristique de concept qui est maintenue incrémentalement.

L'inconvénient majeur des ces algorithmes (UNIMEM, COBWEB et leurs successeurs) est qu'ils contiennent implicitement une importante partie de la connaissance sur le domaine (proximité des exemples pour UNIMEM et utilité catégorielle pour COBWEB). CHARADE, le système présenté ci-dessous, tente de résoudre ce problème.

2.6.5 CHARADE

Le système Charade ([Ganascia 88]) permet de construire des bases de connaissances par la recherche de régularités empiriques présentes sur l'ensemble d'apprentissage. Pour cela, le système a besoin d'un langage de description contenant un ensemble d'attributs typés et d'axiomes exprimant la sémantique du domaine, d'un ensemble d'exemples, et de la description des fonctionnalités du système expert dont on veut construire la base de connaissance.

L'ensemble des parties de l'ensemble d'apprentissage et l'ensemble des conjonctions de descripteurs sont représentés à l'aide de deux treillis booléens. Une régularité empirique correspond à une corrélation observée sur l'ensemble d'apprentissage : si tous les exemples de l'ensemble d'apprentissage qui possèdent le descripteur d_1 possèdent aussi le descripteur d_2 , on peut induire que d_1 implique d_2 sur l'ensemble d'apprentissage, et construire la règle $d_1 \Rightarrow d_2$.

Ganascia définit donc deux fonctions D et C. La fonction C va du treillis des descripteurs vers le treillis des exemples : elle associe à chaque conjonction de descripteurs l'ensemble des exemples couverts par cette description. La fonction D fait correspondre à chaque sous-ensemble de l'ensemble d'apprentissage l'ensemble des descripteurs présents dans la description de tous les exemples. Ainsi, étant donnée une conjonction de descripteurs S, C(S) fournit l'ensemble des exemples couverts par S, et D(C(S)) fournit les descripteurs communs à tous les exemples de C(S). Par exemple, si $D(C(d_1 \& d_2)) = d_1 \& d_2 \& d_3$ alors on détecte une régularité empirique entre $d_1 \& d_2$ et d_3 , et on construit la règle suivante : $d_1 \& d_2 \Rightarrow d_3$. Cette technique détecte les régularités certaines. Pour les régularités incertaines, Ganascia utilise les probabilités conditionnelles : si $\Pr(d_i/d_j)$ est la probabilité que le descripteur d_i soit présent dans la description d'un exemple lorsque la conjonction d_j y est, alors on définit un coefficient de vraisemblance $cv(d_i/d_j)$ par $cv(d_i/d_j) = (2 * \Pr(d_i/d_j) - 1)$ qui viendra pondérer la règle $d_i \Rightarrow d_j$.

Cependant, il n'est pas question de détecter toutes les régularités présentes sur l'ensemble d'apprentissage, et Charade permet, grâce aux propriétés de treillis, et aux propriétés liées à l'implication logique, de ne pas engendrer certaines règles inutiles. Par exemple, $d_1 \& d_2 \Rightarrow d_2$ est une règle inutile. Charade n'examinera donc pas les descriptions du type $(a \& b \& \dots)$ lorsque $a \Rightarrow b$. De même, si $d_1 \Rightarrow d_2$ et $d_2 \Rightarrow d_3$, alors $d_1 \Rightarrow d_3$ est inutile, etc.

De plus, le but du système de règle est pris en compte. Par exemple, pour un système de classification, si C_1, \dots, C_n sont les classes, chaque fois qu'une règle du type $a \Rightarrow C_i$ est engendrée, il est inutile d'examiner les descriptions du type $(a \& \dots)$ qui sont plus générales que a .

Plus généralement, Charade accepte toutes les propriétés P telles que, lorsque la description d est inutile en vertu de P , alors $(d \& \dots)$ est inutile en vertu de cette même propriété ([Ganascia 88]).

Le langage de description de Charade accepte des descripteurs linéaires, nominaux et structurés, et permet ainsi d'expliciter des caractéristiques qui peuvent exister entre certains descripteurs afin de les prendre en compte dans l'apprentissage.

L'avantage principal de CHARADE est de pouvoir engendrer des connaissances incertaines, et ainsi de pouvoir prendre en compte des données éventuellement bruitées. Mais ceci est fait au prix d'un apprentissage forcément non-incrémental. Les Réseaux de Dépendance Perçue (RDP), présentés ci-dessous, permettent, quant à eux, un apprentissage incrémental.

2.6.6 Les RDP

Les Réseaux de Dépendance Perçue (RDP – [Sainte Marie 89], [Sainte Marie 90a], [Sainte Marie 90b], [Sainte Marie 91]), proposent un formalisme qui permet de représenter, sous forme de graphe, un ensemble de Dépendances Perçues. De plus ce graphe peut s'entretenir incrémentalement.

2.6.6.1 La notion de dépendance

La notion de *Dépendance Perçue* est une notion que l'être humain utilise intuitivement de façon courante. L'exemple, désormais banal, pour illustrer cette notion est celui des oiseaux :

Si tous les oiseaux que j'ai pu observer avaient des plumes, et que j'ai aussi observé que tous les oiseaux et tous les avions volaient, j'en conclurais qu'il y a une **dépendance** du fait d'avoir des plumes au fait de voler. Et je percevrai cette dépendance tant que je n'aurai pas connaissance de l'existence des autruches ou des manchots.

Tant que mon expérience n'intègre pas les autruches ni les manchots, la possession de plumes est un moyen de prédire la capacité de voler.

2.6.6.2 Définitions

Définition de la Relation de Dépendance Perçue :

Soient : \mathcal{A} et \mathcal{D} deux ensembles,
 \mathcal{R} une relation dans $\mathcal{A} \times \mathcal{D}$, et
 \mathcal{F}_a et \mathcal{F}_b deux éléments de \mathcal{D} ,
 \mathcal{F}_a et \mathcal{F}_b sont en relation de *Dépendance Perçue* si et seulement si :
 $\exists o_1 \in \mathcal{A}, \forall o_i \in \mathcal{A}, o_1 \mathcal{R} \mathcal{F}_a \wedge (o_i \mathcal{R} \mathcal{F}_a \Rightarrow o_i \mathcal{R} \mathcal{F}_b)$

On notera le fait que \mathcal{F}_a et \mathcal{F}_b sont en relation de Dépendance Perçue par : $\mathcal{F}_a \rightarrow \mathcal{F}_b$.
 \mathcal{R} est appelée la relation fondatrice.

Définition de la Relation d'Equivalence Perçue :

\mathcal{F}_a et \mathcal{F}_b sont en relation d'Equivalence Perçue (et on notera $\mathcal{F}_a \leftrightarrow \mathcal{F}_b$)
 si et seulement si : $\mathcal{F}_a \rightarrow \mathcal{F}_b$ et $\mathcal{F}_b \rightarrow \mathcal{F}_a$.

Définition de la relation fondatrice :

Une définition de la relation fondatrice, en logique des prédicats du 1^{er} ordre, est, pour un langage de description \mathcal{L} :

soit \mathcal{D} l'ensemble des formules bien formées \mathcal{F} du langage de description \mathcal{L} ,
 et soit \mathcal{A} l'ensemble des descriptions des cas d'observation ($\mathcal{D} \supseteq \mathcal{A}$) :
 la relation fondatrice \mathcal{R} est définie par :
 $\mathcal{d} \in \mathcal{A}$ est en relation par \mathcal{R} avec $\mathcal{F} \in \mathcal{D}$
 si et seulement si $(\mathcal{d} \vdash \mathcal{F})$, (\vdash étant le symbole de la dérivation)
 c'est à dire
 si et seulement si \mathcal{F} est une description du cas décrit par \mathcal{d} , ou encore
 si et seulement si \mathcal{F} est plus générale que la description \mathcal{d} .

Ainsi, on obtient : \mathcal{F}_a et \mathcal{F}_b sont en relation de *Dépendance Perçue* si et seulement si :

- il existe un élément \mathcal{d}_1 de \mathcal{A} qui est plus spécifique que \mathcal{F}_a , et si
- tout élément \mathcal{d}_i de \mathcal{A} qui est plus spécifique que \mathcal{F}_a , est aussi plus spécifique que \mathcal{F}_b .

Ce qui revient à dire : \mathcal{F}_α et \mathcal{F}_β sont en relation de *Dépendance Perçue* si et seulement si l'ensemble des descriptions de cas d'observation plus spécifiques que \mathcal{F}_α est non vide et inclus dans l'ensemble des descriptions de cas d'observation plus spécifique que \mathcal{F}_β . On reconnaît là une définition introduite par Ganascia : les *Dépendances Perçues* sont en fait identiques aux régularités empiriques exactes.

La relation de *Dépendance Perçue* peut aussi se formuler par : "*j'ai déjà observé le fait \mathcal{F}_α , donc je sais qu'il est pertinent, et chaque fois que je l'ai observé, j'ai aussi observé le fait \mathcal{F}_β ; tout se passe donc comme si \mathcal{F}_α dépendait de \mathcal{F}_β* " ([Sainte Marie 90b]).

2.6.6.3 Représentation de la *Dépendance Perçue* : le RDP

Indépendamment des caractéristiques propres de la relation fondatrice et de l'ensemble d'observation, la relation de *Dépendance Perçue* est un préordre (du fait de la réflexivité et de la transitivité de l'implication présente dans la définition de la relation de *Dépendance Perçue*). En conséquence, la relation d'*Equivalence Perçue* est une relation d'équivalence. Donc la relation de *Dépendance Perçue* induit un ordre sur l'ensemble des classes d'*Equivalence Perçue* ($\mathcal{D} / \leftrightarrow$). Le squelette de cette relation d'ordre induite peut être représenté par un graphe orienté simple et sans circuit : ce graphe est appelé le **Réseau de Dépendances Perçues** (RDP – [Sainte Marie 89]). C'est la représentation la plus économique de la relation de *Dépendance Perçue*.

Un RDP est donc défini par le couple $(V_{RDP}, \mathcal{E}_{RDP})$ où :

- V_{RDP} est un ensemble de nœuds étiquetés par les éléments de $\mathcal{D} / \leftrightarrow$, et
- \mathcal{E}_{RDP} est un ensemble de paires de nœuds de V_{RDP} (i.e. d'arcs orientés) tel qu'une paire de nœuds de V_{RDP} appartient à \mathcal{E}_{RDP} si et seulement si elle appartient au squelette de la relation d'ordre induite sur $\mathcal{D} / \leftrightarrow$ par la relation de *Dépendance Perçue*.

Ainsi :

- deux formules \mathcal{F}_α et \mathcal{F}_β seront attribuées à un même nœud du RDP si $\mathcal{F}_\alpha \leftrightarrow \mathcal{F}_\beta$, c'est à dire si \mathcal{F}_α a été observée chaque fois que \mathcal{F}_β l'était, et inversement,
- et deux formules \mathcal{F}_α et \mathcal{F}_β seront attribuées à deux nœuds distincts n_a et n_b tels qu'il existe un chemin de n_a à n_b si $\mathcal{F}_\alpha \rightarrow \mathcal{F}_\beta$, c'est à dire si \mathcal{F}_α a été observée chaque fois que \mathcal{F}_β l'était, mais la réciproque n'étant pas vraie dans ce cas là.

2.6.6.4 Apprentissage automatique et RDP

La question qui nous intéresse est de savoir comment un Réseau de Dépendances Perçues peut nous permettre d'acquérir une connaissance à partir d'exemples. Nous nous limiterons au cas où le formalisme d'expression de la connaissance est un formalisme logique (ce qui convient bien à l'usage que l'on aura de la connaissance acquise : un système expert). Le formalisme que nous choisirons sera au plus la logique des prédicats du 1^{er} ordre. Dans ce cadre-là, on peut choisir comme relation fondatrice, celle présentée ci-dessus (la relation de dérivation : $\mathcal{d} \mathcal{R} \mathcal{F}$ ssi $\mathcal{d} \vdash \mathcal{F}$). Cette relation fondatrice étant réflexive, l'ensemble des descriptions des cas d'observation \mathcal{A} est inclus dans son image par \mathcal{R} : $\mathcal{R}(\mathcal{A}) \supseteq \mathcal{A}$ ([Sainte Marie 89]). De plus, puisque la relation de dérivation est un préordre, alors la relation de Dépendance Perçue contient sa relation fondatrice ([Sainte Marie 90b]), c'est à dire que lorsque que deux descriptions sont en relation par \mathcal{R} , alors elles le sont aussi par la Dépendance Perçue.

Etant donné cela, Sainte Marie montre que " $\mathcal{F}_a \rightarrow \mathcal{F}_b$ " signifie que "dans \mathcal{A} , tout se passe comme si $\mathcal{F}_a \vdash \mathcal{F}_b$ " ([Sainte Marie 90b]). Ainsi, lorsque \mathcal{F}_a sera en relation de Dépendance Perçue avec \mathcal{F}_b , cela signifiera que d'après la connaissance empirique du système (ce que contient \mathcal{A}), il semble que \mathcal{F}_b soit dérivable de \mathcal{F}_a .

La Dépendance Perçue permet donc, à partir d'exemples, de représenter des lois générales qui seraient vraies si l'univers était restreint à ces exemples. Chacune de ces lois est donc considérée comme vraie jusqu'à l'arrivée d'un exemple la contredisant. De plus toutes les lois établies sont fondées sur au moins un cas d'observation, ce que dit la clause existentielle de la définition de la Dépendance Perçue.

Mais il ne suffit pas d'être capable de représenter la connaissance acquise, encore faut-il être capable de l'acquérir. Pour cela un algorithme est proposé dans [Sainte Marie 89] et repris dans [Royer 90] qui permet la construction et surtout l'entretien incrémental d'un RDP : cet algorithme utilise une fonction de généralisation qui permet de fabriquer, à partir d'un élément de \mathcal{D} , des descriptions qui lui sont plus générales. La qualité des lois établies dépendra de cette fonction de généralisation. Cet algorithme est présenté au chapitre 4.

2.7 CONCLUSION

Les méthodes analytiques que nous avons examinées (KADS, KOD et MACAO) sont intéressantes car elles proposent des modèles cognitifs de l'expert, mais elles possèdent l'inconvénient majeur d'introduire un intermédiaire entre la connaissance de l'expert et celle du

système artificiel, d'où une perte d'information dans les transferts successifs expert – cognitif, cognitif – informaticien et informaticien – système artificiel (il semble toutefois plus efficace d'introduire cet intermédiaire, plutôt que de mettre directement en relation l'expert et l'informaticien).

De plus, les méthodes analytiques peuvent ne pas convenir à certaines applications, notamment dans les domaines où l'expertise est dispersée chez plusieurs experts (ce qui est souvent le cas, puisque par définition la connaissance experte est empirique, et par conséquent propre à chaque expert), et les domaines dans lesquels l'expertise évolue. Lorsque ces deux caractéristiques sont conjointement présentes, les méthodes analytiques sont absolument impraticables. En effet, en admettant qu'un cognitif parvienne à rencontrer un ensemble significatif d'experts du domaine, il lui faudrait ensuite rencontrer tous ces experts continuellement, de façon à ce que la base de connaissance construite suive l'évolution du domaine.

Une caractéristique importante des logiciels qui accompagnent ces trois méthodes, est l'absence de tout mécanisme d'induction. L'aide qu'apporte ces logiciels est purement de nature descriptive : ils aident le cognitif (éventuellement l'expert, dans un deuxième temps) dans l'explicitation de la connaissance en proposant un cadre, et en vérifiant des cohérences ou des complétudes.

L'apprentissage automatique permet l'acquisition de la connaissance, et se distingue des méthodes analytiques par le fait qu'il vise à l'acquisition de la connaissance sans l'intermédiaire du cognitif, c'est à dire en mettant directement en relation l'expert et le système, pendant toute la phase d'extraction de la connaissance.

Parmi les méthodes d'apprentissage automatique, celles qui ne possèdent pas de mécanisme inductif (apprentissage par cœur ou apprentissage par instruction) ne sont guère intéressantes, car elles supposent que l'environnement fournit une connaissance déjà explicitée, alors que c'est principalement l'explicitation de la connaissance qui constitue le goulot d'étranglement de l'acquisition de la connaissance. Aussi ne nous intéresserons-nous qu'aux méthodes d'apprentissage automatique reposant sur des mécanismes inductifs.

Parmi les méthodes présentées ci-dessus, celles qui permettent une construction incrémentale de la base de connaissance sont particulièrement intéressantes pour les domaines où la connaissance évolue.

ID3 ne possède pas cette caractéristique, et de plus le formalisme avec lequel ID3 fournit la connaissance s'adapte peu aux systèmes experts. En effet, ID3 ne permet pas la représentation de concepts intermédiaires, qui sont indispensables pour satisfaire le critère d'explicabilité des systèmes experts.

COBWEB et UNIMEM sont intéressants pour les domaines où une connaissance imprécise est souhaitée. Mais ils ont l'inconvénient de contenir implicitement une importante connaissance sur le domaine, dans leurs fonctions heuristiques (utilité catégorielle pour COBWEB, ou proximité des exemples pour UNIMEM).

CHARADE vise à résoudre ce problème en explicitant le plus possible le biais inductif. De plus CHARADE permet, si cela est souhaité, de découvrir des règles incertaines, mais malheureusement au prix d'un apprentissage non-incrémental.

Les RDP se rapprochent de CHARADE (les Dépendances Perçues sont identiques aux régularités empiriques exactes) mais ne permettent pas d'apprendre des règles incertaines. Ils possèdent néanmoins l'avantage de permettre un apprentissage incrémental.

Par rapport à UNIMEM et COBWEB, les RDP possèdent l'avantage de ne pas contenir implicitement une importante connaissance sur le domaine. En effet, dans UNIMEM cette connaissance implicite se trouve dans la fonction mathématique qui permet d'évaluer la distance entre exemples et/ou concepts, et permet ainsi d'acquérir une connaissance incertaine. Il en est de même pour COBWEB avec l'utilité catégorielle. Puisque les RDP ne représentent que la connaissance certaine, il est bien évident qu'ils ne contiendront pas cette connaissance implicite.

Mais l'apprentissage automatique, à lui seul, ne permet pas de remplacer le cognicien, car il ne consiste qu'en un monologue entre l'expert et le système artificiel : l'expert fournit des informations au système, et ce dernier doit en extraire des connaissances utiles.

En fait, UNIMEM et COBWEB conviendront parfaitement pour un apprentissage incrémental pour lequel on désire obtenir des descriptions incertaines, mais ils ne permettront pas un dialogue très riche entre le système et l'expert. En revanche, les RDP conviendront très bien dans les cas où l'on désire que l'acquisition de la connaissance soit réalisée au travers d'un dialogue système/expert.

C'est cette dernière méthode que nous choisissons pour notre application, car elle nous permet un apprentissage automatique incrémental à partir d'exemples, et surtout un dialogue entre l'expert et le système. Ce dialogue a pour but de faire valider, par l'expert lui-même, la connaissance acquise. Nous nous passons ainsi du cognicien, et nous traitons le problème des données erronées sans avoir recours à des traitements statistiques des exemples fournis.

Chapitre 3

Conception de la Station de Travail Intelligente de l'aménageur en eau

Dans le premier chapitre, nous avons montré l'intérêt de proposer un système intégrant tous les outils susceptibles d'être utilisés par un aménageur en eau, et avons surtout mis en évidence le point névralgique de la construction de ce système : l'acquisition de la connaissance, qui est d'autant plus difficile à réaliser que la connaissance de ce domaine est dispersée et évolutive.

Ensuite, dans le deuxième chapitre, nous avons montré l'inadéquation des outils actuels (méthodes analytiques ou apprentissage automatique) à résoudre tous nos problèmes d'acquisition de la connaissance pour notre Station de Travail. Cela provient des présupposés que font ces méthodes : les méthodes analytiques font l'hypothèse que le transfert de la connaissance sera une traduction effectuée par un cogniticien, et l'apprentissage automatique suppose que toute la connaissance à acquérir est contenue dans les exemples fournis par les experts.

Dans ce chapitre, nous concevons donc l'architecture informatique de cette Station de Travail Intelligente de l'aménageur en eau, non seulement en tant qu'outil intégrant tous les services nécessaires à la conduite d'un projet d'aménagement, mais aussi en tant qu'outil de communication entre experts et utilisateurs. Pour cela, en nous inspirant des modèles cognitifs des méthodes analytiques d'acquisition de la connaissance, et aussi de l'architecture des Systèmes Experts de Seconde Génération (SESG), nous définissons trois types de connaissances, mis en œuvre lors de la conduite des projets d'aménagement. Nous utilisons ensuite une conceptualisation des transferts de connaissances (à base de Théories Semi-Empiriques), qui est décrite dans le modèle d'apprentissage de Reitz, et pour chaque type de connaissance défini, nous approfondissons l'analyse des transferts de connaissances.

Les deux derniers chapitres de ce mémoire étant exclusivement consacrés à l'outil télématique permettant l'acquisition des connaissances de surface, nous détaillons plus particulièrement ce point-ci dans les lignes qui suivent.

3.1 LA STATION DE TRAVAIL : UN OUTIL DE COMMUNICATION

Dans un premier temps, notre Station de Travail pourra servir aux Bureaux d'Etudes pour répondre aux appels d'offres, en permettant à ceux-ci d'étendre leur domaine de compétence pour fournir des devis sur l'ensemble du projet. Dans un deuxième temps, lorsque le système aura un degré d'utilisation plus élevé, il pourra être utilisé pour n'importe quelle phase du projet (études de faisabilité et conduite de la réalisation des travaux), en prenant en compte chaque composante économique, industrielle, écologique voire sociale de l'environnement du projet.

3.1.1 Fonctions de la Station de Travail

De [Graillet 86] il ressort que cette Station de Travail devra permettre de répondre aux appels d'offres, et permettre aussi l'assistance à la conduite de projets d'aménagement en eau, pour les domaines suivants : alimentation en eau potable, irrigation et assainissement. De façon plus précise, il devra permettre :

- l'organisation et la planification des tâches d'un projet,
- l'assistance au lancement et au suivi de ces tâches,
- l'analyse des effets des modifications du projet,
- la proposition de solutions de rechanges, et surtout

- l'explication des conseils fournis.

Ce dernier point est très important : en aucun cas le logiciel ne décide. Il ne fait que guider ou conseiller un décideur. Et pour que le décideur soit bien conseillé, il faut que le conseil qui lui est fourni, soit expliqué jusqu'à un niveau de détail admissible par le décideur. C'est l'approche Intelligence Artificielle de l'aide à la décision.

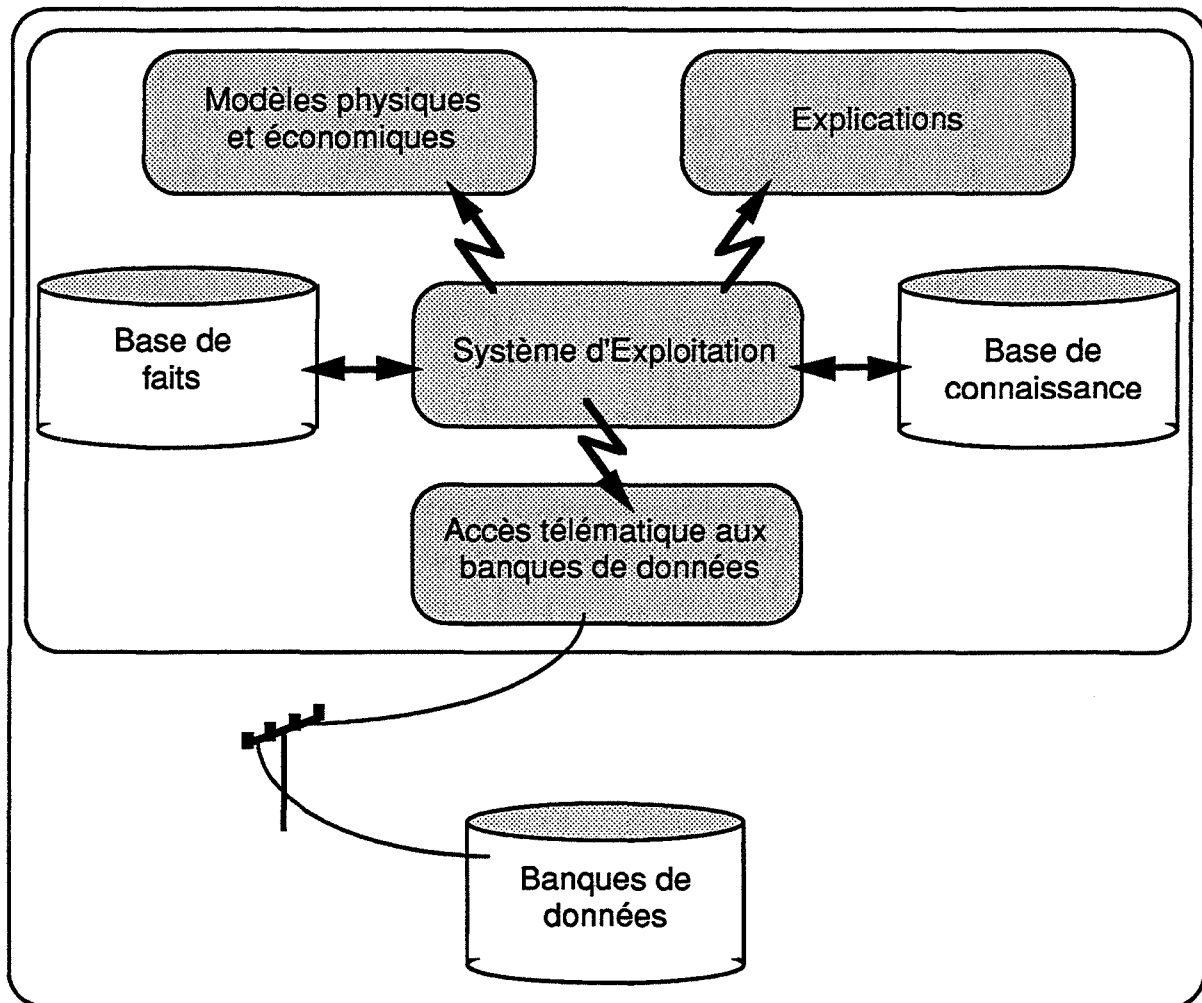


fig. 3.1 : architecture sommaire de la Station de Travail

Les différents composants de la Station de Travail seront donc (cf. figure 3.1) :

- un module d'accès à des banques de données télématiques, pour les informations concernant l'hydrogéologie, la géographie, la sociologie, les réseaux (eau, gaz, électricité, routes, ...), etc...,

- un module comprenant des modèles physiques et économiques,
- une base de connaissance sur la gestion de projets d'aménagement en eau,
- une base de faits contenant toutes les informations connues du projet en cours de traitement,
- un module d'explications, ayant pour rôle d'explicitier les conseils fournis, et
- un système d'exploitation (le cœur du système) capable d'enchaîner l'activation de chaque module et de gérer la base de connaissance et la base de faits.

Les accès aux banques de données télématiques devront être automatisés de façon à ce que leur utilisation soit transparente, et paramétrables de façon à ce que l'utilisateur de la Station de Travail puisse facilement modifier ces accès au fur et à mesure de l'évolution des serveurs.

Les modèles physiques et économiques seront les mêmes que ceux déjà présents dans les simulateurs MISE ou PROMISE. Il s'agit de modèles mathématiques utilisés pour la prévision du comportement hydrodynamique des nappes aquifères, pour le calcul des réseaux de distribution; il s'agit aussi de programmes de recherche opérationnelle utilisés pour l'optimisation technico-économique des tracés de réseaux, de logiciels destinés à l'interprétation des données géophysiques, etc...

Avant d'approfondir l'analyse de la Station de Travail Intelligente de l'aménageur en eau, nous devons examiner une contrainte très importante de cette station : l'explicitabilité, car elle va conditionner cette architecture.

3.1.2 Les explications de la Station de Travail

Dans tout Système à Base de Connaissance, l'explication a un rôle primordial. Kassel montre les limites de l'explicitabilité des Systèmes Experts classiques (dits de Première Génération), qui ne reposent que sur des connaissances heuristiques : la seule explication possible est la trace fidèle du raisonnement. Ils sont incapables de synthétiser leur raisonnement pour en faire ressortir les étapes importantes, parce qu'ils n'ont pas explicitement, dans leur base de connaissance, la stratégie employée ([Kassel 88], [Kassel 89]).

C'est aussi ce que nous suggèrent les modèles de connaissance de KADS et MACAO (cf. chapitre 2) qui nous indiquent très clairement que plusieurs niveaux de connaissances interviennent dans une expertise. C'est cette distinction entre plusieurs niveaux de connaissances qui constitue le concept de **Système Expert de Seconde Génération (SESG)**.

Dans ces systèmes, on s'efforce d'explicitier encore plus la connaissance de l'expert, en séparant une connaissance de surface (connaissance heuristique propre à une tâche précise) d'une connaissance profonde (connaissance qui permet de décrire le domaine). Pour cette dernière, Steels propose une dichotomie supplémentaire : la connaissance profonde est composée ([Steels 88]) :

- d'une connaissance statique : connaissance décrivant la nature des différents éléments présents dans la connaissance de surface. Par exemple, dans un Système Expert de diagnostic, sont manipulés des symptômes, des dysfonctionnements, des remèdes, ainsi qu'une architecture du système à diagnostiquer (Steels emploie le terme de théories du domaine pour nommer cette connaissance. Nous préférons la désigner sous le terme "connaissance statique" car elle est purement descriptive du domaine, et ne fait pas forcément référence à une quelconque théorie),

- et d'une connaissance de stratégie qui décrit des plans de résolution d'un problème général du domaine.

Les avantages d'une telle approche sont très importants. En premier lieu, elle constitue un pas en avant dans la recherche d'explicitation de la connaissance, un des objectifs de l'IA. Cela a pour conséquence que :

- de meilleures explications sont alors envisageables car le système possède une meilleure compréhension de son raisonnement et de la nature de ses connaissances, et que,
- comme cela nous est suggéré par les modèles cognitifs des méthodes comme KADS ou MACAO, ils permettent une meilleure acquisition des connaissances.

De plus, une limite très grave des premiers Systèmes Experts, à base uniquement de connaissance de surface, est leur impossibilité à détecter qu'ils ne savent pas résoudre un problème. Ainsi, comme le font remarquer David, Krivine et Simmons (préface de [SESG 89]), leurs performances se dégradent brutalement aux frontières de leur expertise. Les SESG, grâce à l'explicitation de la connaissance profonde du domaine d'expertise, peuvent résoudre ce problème, ...

... avec une limite, toutefois : les SESG ne peuvent pas détecter les cas où ils ne savent pas détecter s'ils savent ou non résoudre un problème. Mais il suffit alors d'introduire une troisième génération de Systèmes Experts, chose que nous ne ferons pas, par crainte de nous aventurer

dans les "*Brins d'une Guirlande Eternelle*" (version française du "Eternal Golden Braid" de Douglas R. Hofstadter, [Hofstadter 79]).

3.1.3 Les niveaux de la connaissance experte

Nous avons donc défini trois niveaux de connaissances expertes : la connaissance de stratégie, la connaissance de surface, et la connaissance statique. Voyons comment ces trois niveaux peuvent permettre la description de la connaissance experte mise en œuvre lors de la conduite d'un projet d'aménagement en eau.

3.1.3.1 La connaissance de stratégie

La connaissance de stratégie est la connaissance qui spécifie la résolution globale d'un problème général du domaine. C'est une connaissance experte, c'est à dire qui repose essentiellement sur l'expérience.

Dans le modèle cognitif de KADS (cf. chapitre 2), cette connaissance est partagée dans deux niveaux conceptuels (les niveaux tâche et stratégique). Le niveau stratégique décrit le plan de résolution du problème posé et contrôle le niveau tâche. Le niveau tâche décrit les étapes du raisonnement (buts de la tâche) et la manière d'atteindre ces buts. Dans la conduite d'un projet d'aménagement en eau, le problème se décompose bien en diverses étapes, qui se décomposent elles-mêmes en sous-étapes, etc..., et qui, à un certain niveau, correspondent aux "tâches" du modèle cognitif de KADS.

Par exemple, pour l'étude de faisabilité d'un projet d'assainissement, on peut facilement trouver les grandes étapes du niveau stratégique : des modalités juridiques et administratives sont à respecter pour construire préalablement le financement de l'opération, puis vient la phase d'étude du projet qui aboutira à la définition de plusieurs schémas d'assainissement et au dimensionnement des divers ouvrages. Ces différents schémas seront proposés aux gestionnaires avant d'entreprendre les travaux. Des stratégies similaires sont décelables dans les projets d'irrigation et d'alimentation en eau potable. Ainsi, il est possible de décrire une stratégie globale de résolution d'un problème d'aménagement en eau. L'explicitation de cette stratégie permettra ensuite de fournir des explications très synthétiques sur le déroulement d'un projet.

Au niveau tâche, on spécifie comment sera parcourue une structure d'inférence pour atteindre un but ([Brunet 91]).

Par exemple, le diagnostic d'un dysfonctionnement peut demander la mise en œuvre d'un raisonnement hypothético-déductif sur des règles du type :

dysfonctionnement => symptôme ,

puis la proposition des remèdes peut nécessiter un raisonnement déductif en chaînage avant sur des règles du type :

dysfonctionnement => remède .

Pour les sciences de l'eau, cette connaissance est mise en œuvre lors de la conduite globale d'un projet d'aménagement. Nous choisissons donc de l'acquérir grâce à un simulateur qui permet de mettre un expert en situation de conduite d'un projet d'aménagement en eau.

3.1.3.2 La connaissance de surface

La connaissance de surface est la connaissance heuristique propre à la résolution d'un problème précis, indécomposable en sous-problèmes plus élémentaires. C'est typiquement une connaissance experte, c'est à dire une connaissance de l'expert qui repose essentiellement sur son expérience.

La connaissance de surface correspond au niveau "inférence" du modèle cognitif de KADS. En matière d'assainissement autonome des eaux usées, ce sera, par exemple, la préconisation du filtre bactérien percolateur seulement lorsque toutes les autres filières ne sont pas satisfaisantes, car il est très délicat à installer, et il est donc sujet à de plus fréquents dysfonctionnements. Elle relie donc heuristiquement des éléments de la connaissance statique pour résoudre une tâche précise. Ces liens sont les raccourcis de raisonnement que l'expert utilise. Par exemple, en matière de diagnostic, une connaissance de surface relie directement un dysfonctionnement à un symptôme et à un remède.

Cette connaissance peut être modularisée (c'est d'ailleurs là un des intérêts des SESG) en plusieurs bases de connaissance, l'activation de chacune d'entre elles étant décidée par la connaissance de stratégie. Ainsi on intègre, à l'intérieur de la Station de Travail, plusieurs Systèmes Experts de première génération. Pour chacun d'entre eux, la base de connaissance est de taille réduite et donc facilement gérable. Le partage de la connaissance de surface entre les différentes bases se fait thématiquement. Par exemple, une base pour le diagnostic des dysfonctionnements des systèmes d'assainissement autonome, une autre pour ceux des stations

d'épuration, une autre encore pour le choix d'un système d'assainissement, une autre pour la réhabilitation, etc...

Pour notre Station de Travail, la connaissance de surface est une connaissance particulière à une tâche donnée, indécomposable en tâches plus élémentaires, et qui ne se réduit pas à l'appel d'un modèle physique ou économique.

Dans un premier temps, comme nous l'avons indiqué au chapitre 1, nous nous intéressons uniquement à la tâche particulière de l'assainissement autonome des eaux usées. Cette connaissance est caractérisée par sa grande dispersion (au niveau géographique, dans diverses disciplines scientifiques et dans différents corps de métiers) et par son évolution (réglementaire et technologique). Ces deux caractéristiques nous font opter pour un système télématique qui nous permettra de joindre facilement un grand nombre d'experts.

3.1.3.3 La connaissance statique

La connaissance statique est la connaissance qui décrit la sémantique du domaine, et correspond au niveau "domaine" de KADS. Cette connaissance est à rapprocher de la "connaissance a priori" que nous avons évoquée au chapitre 2 : elle permet de restreindre l'espace de recherche lors d'opérations d'induction, en imposant une cohérence sémantique aux règles de la connaissance de surface. C'est ce que fait le système ADELE, exposé dans un article intitulé "Une aide à l'acquisition des connaissances de surface à partir de connaissances profondes" ([Reynaud 88]).

La connaissance statique est utilisée lors de la résolution globale d'un problème. Par exemple, dans l'étude de faisabilité d'un projet d'assainissement des eaux usées, des modalités juridiques et administratives sont utilisées pour proposer un montage financier de l'opération. L'expert, chargé de ce problème, a besoin de ces modalités, qui seront décrites dans la connaissance statique.

De même, la manipulation de la connaissance de surface utilise aussi la connaissance statique. Pour continuer le même exemple, une fois choisi le type d'assainissement le plus adapté, il faut orienter le décideur vers un système d'assainissement particulier. S'il s'agit d'un assainissement autonome, une base de connaissance de surface dédiée à ce problème sera activée; cette base de connaissance aura besoin de connaître, par exemple, les caractéristiques d'encombrement des systèmes d'assainissement autonome afin de vérifier si celui qui a été

retenu est bien installable sur la parcelle retenue. Cet encombrement sera décrit dans la connaissance statique.

Puisqu'elle est utilisée à la fois par la connaissance de stratégie, et par la connaissance de surface, nous choisissons de la scinder en deux parties :

- une partie relative à la conduite générale du projet : nous la nommerons "connaissance statique générale", et
- une partie relative à la résolution heuristique des tâches : nous la nommerons "connaissance statique spécifique".

La connaissance statique générale pourra être acquise avec le simulateur en même temps que la connaissance de stratégie, et la connaissance statique spécifique avec le système télématique, en même temps que la connaissance de surface.

3.1.4 Acquisition interactive de la connaissance

Le problème, maintenant, et conformément à ce que nous avons vu au chapitre 2, est de construire informatiquement toutes ces connaissances. Il ne suffit pas, en effet, de pouvoir représenter informatiquement des connaissances, encore faut-il pouvoir les acquérir.

Traditionnellement, les Systèmes à Bases de Connaissance sont considérés comme des systèmes contenant de la connaissance, et leur architecture est donc étudiée en ne tenant compte, principalement, que d'un seul transfert de connaissance : le système doit dialoguer avec l'utilisateur final pour lui délivrer sa connaissance. En revanche, le transfert de la connaissance des experts vers le Système à Base de Connaissance est considéré uniquement comme un monologue où le système reçoit de la connaissance :

- avec les méthodes analytiques, il s'agit d'un monologue entre l'informaticien (parfois le cognicien) et le système, et
- avec les méthodes à base d'apprentissage automatique, il s'agit d'un monologue entre le(s) expert(s) et le système.

Toutefois, certains systèmes, comme ETS, AQUINAS, EAR, DILEM ([Boy – Faller – Sallantin 88]) ou MACAO ([Aussenac 89]), qui présentent la connaissance acquise sous forme d'arbres ou de graphes, permettent à l'expert un contrôle visuel de la connaissance acquise. Un autre système, LEGAL, va plus loin en utilisant un mécanisme d'argumentation pour justifier les choix effectués lors de la phase d'apprentissage ([Liquière – Mephu Nguifo, 90]). Mais ces

systèmes ne font que de l'édition de connaissance, l'acquisition restant toujours à la charge de l'expert ([Boy – Faller – Sallantin 88]).

Pour notre Station de Travail, nous allons donc considérer l'acquisition de connaissance comme un processus dialogique entre le système et les experts : nous considérons notre Station de Travail non seulement comme un système contenant de la connaissance, mais aussi comme un système capable de partager cette connaissance. Notre Station de Travail est donc étudiée en tant qu'outil de communication entre les experts et les utilisateurs finaux : nous considérons à la fois les transferts de connaissance entre le système et les utilisateurs finaux, et les transferts de connaissance entre le système et les experts. Pour mener à bien cette étude nous allons utiliser le modèle d'apprentissage de Reitz, qui permet de conceptualiser les dialogues qui interviennent lors d'un processus d'apprentissage.

3.2 CONCEPTUALISATION DES TRANSFERTS DE CONNAISSANCE

Le modèle d'apprentissage que nous présentons dans ce paragraphe est emprunté à [Reitz 92]. Il a pour objet d'étudier les dialogues qui interviennent autour d'un apprenti, que celui-ci soit un système face à des humains, ou un humain face à un système d'EAO ou d'aide à la découverte. Ce modèle nous permettra donc de préciser, de façon unifiée, les interactions entre nos systèmes (système télématique et simulateur) et leurs interlocuteurs. Il sera aussi utilisable dans de futurs travaux concernant la troisième génération des simulateurs MISE.

Avant de présenter ce modèle d'apprentissage, nous présentons les Théories Semi-Empiriques (développées par Jean Sallantin et son équipe au LIRMM) qui ont pour but de conceptualiser une forme de connaissance susceptible d'être élaborée et critiquée par des humains comme par des machines ([Sallantin & al., 91]). Cette conceptualisation nous servira à décrire de façon précise les dialogues entre les différents acteurs présents dans le modèle de Reitz.

3.2.1 Les Théories Semi-Empiriques (TSE)

Les **TSE** sont une forme de représentation des connaissances non complètement axiomatique (elles ne s'appuient pas sur des connaissances initiales), ni empirique (elle ne dépend pas de l'analyse des données mais se développe sur leur étude). L'objectif est d'offrir un cadre conceptuel à un système dont on veut qu'il construise une théorie d'un domaine, au moyen

d'un dialogue fructueux avec son utilisateur. Cette théorie s'exprimera sous la forme de conjectures, issues de l'étude des données offertes au système. La Théorie Semi-Empirique progresse par preuves et réfutations.

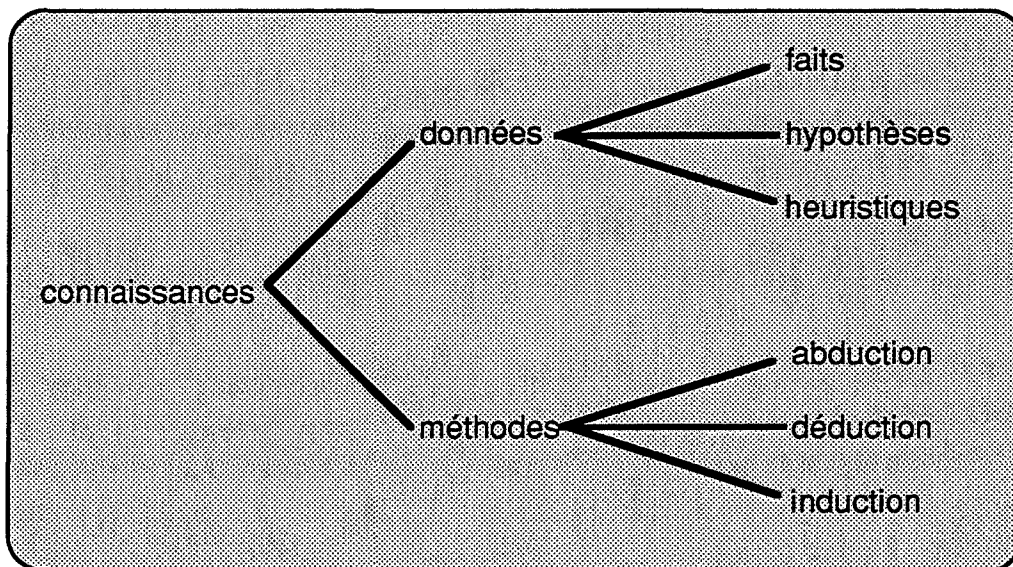


fig. 3.2 : taxinomie des connaissances de Pierce (d'après [Sallantin & al. 91])

Pour représenter les connaissances, la taxinomie de Pierce a été retenue (cf. figure 3.2) : les **données** sont les connaissances qui peuvent évoluer, et les **méthodes** les connaissances qui font évoluer les données.

Les **faits** sont des énoncés dont on peut estimer la validité. Cette validité s'exprime sous forme de croyances. Sallantin & al. proposent trois croyances : *justification* (J), *contestation* (C) ou *silence* (S). On les formalise par un triplet (énoncé, domaine, croyance) où le domaine est l'ensemble sur lequel s'estime la croyance de l'énoncé.

Les **hypothèses** sont des relations entre faits, supposées pertinentes. Elles peuvent soit énoncer une condition que doit respecter un ensemble de faits (elle décrivent alors des **objets**), soit formuler une expression que doit respecter un ensemble d'objets (elles décrivent alors des **concepts**). Une **conjecture** est une hypothèse formulée après examen d'un ensemble d'exemples.

Un **exemple** d'un concept est un objet qui vérifie la conjecture caractérisant le concept (il appartient à l'extension du concept), et un **contre-exemple** un objet qui ne vérifie pas la conjecture (il appartient à la contre-extension du concept).

Les **heuristiques** sont des règles de calcul de faits nouveaux, et des règles de calcul d'hypothèses nouvelles.

Les **méthodes** sont la déduction, l'abduction et l'induction :

- la déduction tire des conséquences de données reconnues acquises, sous forme de faits affirmés,
- l'abduction propose de nouveaux faits à confirmer, et
- l'induction propose de nouvelles hypothèses et de nouvelles heuristiques, à confirmer aussi.

Les critères d'évaluation d'un raisonnement sont la validité et la pertinence. La **validité** est une évaluation des faits produits par applications d'hypothèses. La **pertinence** est une évaluation de la manière dont les hypothèses produisent des faits. L'objectif de l'apprentissage est d'augmenter, par dialogue, la validité des faits et la pertinence des hypothèses produits par le système.

Les termes et concepts suivants sont inspirés de [Lakatos 84].

Lors du dialogue, des preuves et des réfutations sont échangées. Une **preuve** est la décomposition d'une conjecture en lemmes. Un **lemme** est une conjonction de faits caractéristique des exemples satisfaisant la conjecture. La **réfutation** d'un lemme est la production d'un contre-exemple. L'analyse de la preuve est une recherche de réfutations des lemmes de la preuve. Une preuve acceptée est une preuve pour laquelle on ne connaît pas de réfutation de ses lemmes.

L'intérêt des TSE est qu'elles nous offrent une conceptualisation d'un dialogue entre un système et un humain lors d'un processus d'apprentissage. Munis de ces notions, nous allons pouvoir étudier l'environnement d'apprentissage de Reitz.

3.2.2 L'environnement d'apprentissage de Reitz

Ce modèle (cf. figure 3.3) a pour objet d'étudier les dialogues qui interviennent autour d'un appreni. Nous allons expliciter ces différents dialogues avec les termes introduits dans le paragraphe précédent.

- Un **problème** et une **solution** sont des faits.
- Le **Client** soumet un problème à l'**Apprenti** qui lui renvoie une solution.
- L'**Apprenti** soumet une preuve au **Maître** qui lui renvoie une critique.
- Une **critique** est la réfutation d'un lemme de la preuve, ou bien l'acceptation de la preuve.

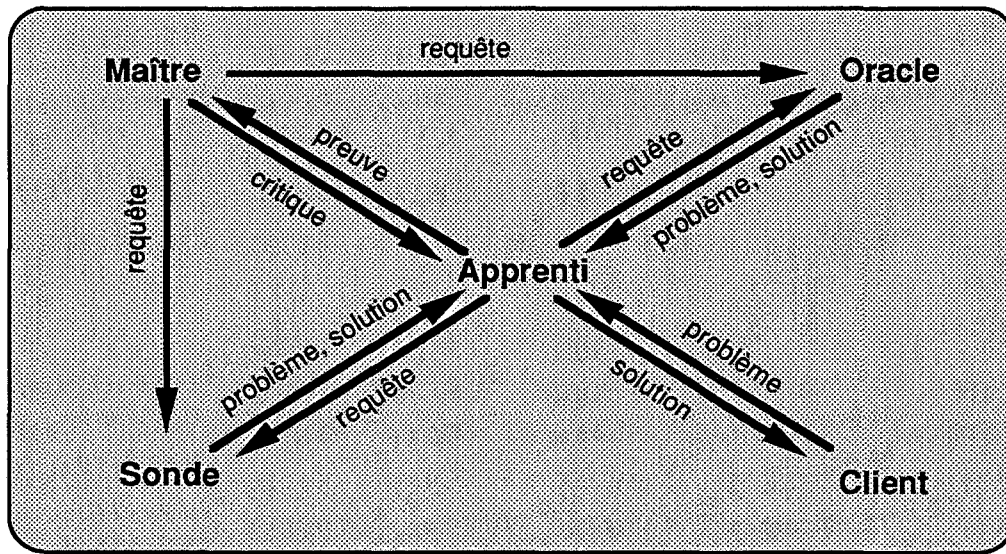


fig. 3.3 : environnement d'apprentissage ([Reitz 92])

- L'**Oracle** et la **Sonde** fournissent des couples (problème, solution) à l'**Apprenti**, suite à sa requête ou suite à la requête du **Maître**. L'Oracle et la Sonde se différencient par le traitement que fait l'Apprenti à réception des informations :
- lorsque la **Sonde** fournit des informations, l'Apprenti modifie sa connaissance jusqu'à ce que le Maître accepte la preuve de son accord ou de son désaccord avec la Sonde,
- tandis que lorsque l'**Oracle** fournit des informations, l'Apprenti modifie sa connaissance pour être ensuite capable de fournir la même solution pour le même problème.

Le fait d'avoir structuré la connaissance en plusieurs niveaux (connaissance de stratégie, connaissance statique et connaissance de surface) va nous permettre d'envisager l'acquisition de chacune de ces connaissances séparément.

3.3 ANALYSE DES TRANSFERTS DE CONNAISSANCE

Au paragraphe 3.1.3, nous avons défini trois niveaux de connaissance : pour chacun d'entre eux, nous avons prévu un outil d'acquisition :

- pour la connaissance de surface et la connaissance statique spécifique, un système télématique permet de joindre facilement un grand nombre d'experts,

- pour la connaissance de stratégie et la connaissance statique générale, un simulateur permet de mettre un expert en conditions quasi-réelles de conduite d'un projet d'aménagement en eau.

3.3.1 Le système télématique

Afin d'acquérir la connaissance de surface et la connaissance statique spécifique, particulières à une tâche donnée, nous prévoyons d'interroger les experts du domaine au moyen d'un service télématique.

Pour l'instant, nous nous intéressons à une partie seulement de l'aménagement en eau : l'assainissement autonome des eaux usées. Au chapitre 1, nous avons montré l'importance de ce problème, et le fait que la connaissance relative à ce domaine est particulièrement représentative de la connaissance utilisée en aménagement en eau.

3.3.1.1 Les interlocuteurs du système télématique

Nous construisons donc un système télématique d'acquisition interactive de la connaissance, appliqué à l'assainissement autonome des eaux usées.

L'opérationnalité d'un tel système repose essentiellement sur la motivation des détenteurs actuels de la connaissance à collaborer à cette entreprise. Pour cela, nous avons contacté les différents acteurs de ce domaine, et avons élaboré un schéma de faisabilité du système : son rôle ne sera pas uniquement de recueillir la connaissance pour alimenter la Station de Travail Intelligente. Pour que le projet soit viable, il faut aussi que cette connaissance soit redistribuée afin que tous les acteurs du domaine aient un intérêt dans le projet.

La figure 3.4 présente les différents interlocuteurs du système télématique, ainsi que leurs intentions respectives.

Le rôle des Agences de l'Eau (anciennes Agences Financières de Bassin) est, entre autres, d'améliorer l'assainissement des eaux usées. Elles sont donc intéressées par un outil efficace d'aide au choix d'un système d'assainissement. Mais elles doivent valider cet outil. Cette validation est double : elles doivent valider la connaissance diffusée, par une validation de la

façon dont est acquise cette connaissance, et elles doivent aussi valider la façon dont l'aide au choix est fournie. En retour, si elles sont satisfaites du processus (les contacts que nous avons eus avec celle du Bassin de la Loire nous font présager que ce sera le cas), ces Agences recommanderont le système à leurs interlocuteurs (administrations, collectivités locales, entreprises, ...).

mode acteurs	consultation	acquisition
agences de bassin	validation	validation
affaires sanitaires	résolution de problèmes en série	prise en compte des spécificités régionales
hydrogéolo- gues agréés		traitement des cas particuliers
cabinets d'études	rentabilité rapidité	
particuliers	autonomie prix	

fig. 3.4 : intentions des différents interlocuteurs du système télématique
(d'après [Beaune – Graillet 90])

Les Directions Départementales des Affaires Sanitaires et Sociales ont pour rôle de vérifier la construction ou la réhabilitation de tous les systèmes d'assainissement (conformité à la législation et garantie de bon fonctionnement). Elles effectuent ces vérifications elles-mêmes, ou bien, lorsqu'il s'agit de cas trop particuliers, elles font appel aux hydrogéologues agréés. Si ces cas particuliers, une fois résolus, sont fournis au système, les DDASS sauront ensuite les résoudre par simple consultation du système, et feront donc de moins en moins appel aux hydrogéologues agréés.

Les cabinets d'études n'ont pas toujours toutes les compétences pour résoudre les problèmes d'assainissement autonome des eaux usées. Ce système leur facilitera la tâche et leur garantira

que leurs études seront acceptées directement par la DDASS, même en dehors de leur zone de travail habituelle.

Enfin, les particuliers, lorsqu'ils sont aptes à effectuer les mesures nécessaires sur le terrain, pourront bâtir leur dossier eux-mêmes et à moindre coût, avec la garantie qu'il sera accepté par la DDASS.

On constate donc qu'un tel système peut avoir une utilité sociale immédiate. Ceci nous garantit son opérationnalité et nous permettra d'alimenter, pour la Station de Travail, la base de connaissance spécifique à l'assainissement autonome des eaux usées.

3.3.1.2 Les dialogues avec le système télématique

Dans le modèle d'apprentissage que nous avons décrit ci-dessus (cf. § 3.2), il est immédiat que l'**Apprenti** est le système télématique, qui cherche à se construire une théorie de façon à servir un **Client**, c'est à dire le consultant du système. Nous ajoutons toutefois une réponse possible de l'**Apprenti** au **Client** : la preuve. En effet, notre système est un système d'aide à la décision, il doit donc convaincre son Client du bien fondé de son conseil. Le but de l'apprentissage est alors que notre système se construise une théorie qui lui permette de fournir des solutions valides et des preuves pertinentes. Nous définirons ces deux notions ci-dessous.

Il est tout aussi immédiat que, pour notre cas d'application, l'expert jouera les rôles de **Maître** et de **Sonde**. Nous ne pouvons pas considérer l'expert comme un éventuel **Oracle**, car nous supposons que l'expert peut se tromper (ne serait-ce qu'en commettant une faute de frappe lors de la saisie de ses informations). En effet, une caractéristique de l'**Oracle** est que l'on ne peut pas contester sa solution, et qu'en conséquence l'**Apprenti** doit modifier sa théorie sans en référer au **Maître**. Pour notre application, nous ne pouvons pas prendre ce risque et considérerons toujours l'expert comme une **Sonde**, c'est à dire avec méfiance.

Là aussi, nous modifions un peu le dialogue entre l'expert et le système : lorsque l'expert fournit un couple (problème, solution), c'est forcément de sa propre initiative (nous éliminons la requête). De plus, nous prévoyons que le système puisse engendrer de lui même un problème pour lequel il a envie de connaître la réponse de l'expert : ce dernier fournit alors une solution.

Le dialogue système/consulteur est un but du système que nous cherchons à construire. Pour cela, nous avons besoin d'extraire la connaissance des experts et la représenter dans le système. Cette extraction est réalisée par le dialogue système/expert. L'expert jouera le rôle de **Maître** lorsque le système lui soumettra des preuves qu'il devra critiquer. Lorsque l'expert fournira un

couple (problème, solution), le système considérera l'expert comme une Sonde : il modifiera sa connaissance jusqu'à ce que la preuve de son accord ou de son désaccord soit acceptée par l'expert-Maître. Lorsqu'à la fin, c'est à dire lorsque l'expert-Maître accepte la preuve :

- si cette preuve est la preuve d'un accord avec le couple (problème, solution) fourni par l'expert-Sonde, on peut considérer que la solution n'est pas fautive, et donc que le couple (problème, solution) est un exemple, c'est à dire un couple tel qu'en aurait fourni un Oracle.

- si cette preuve est la preuve d'un désaccord avec le couple (problème, solution) fourni par l'expert-Sonde, ce couple est considéré comme contre-exemple : l'expert a simplement voulu tester le système.

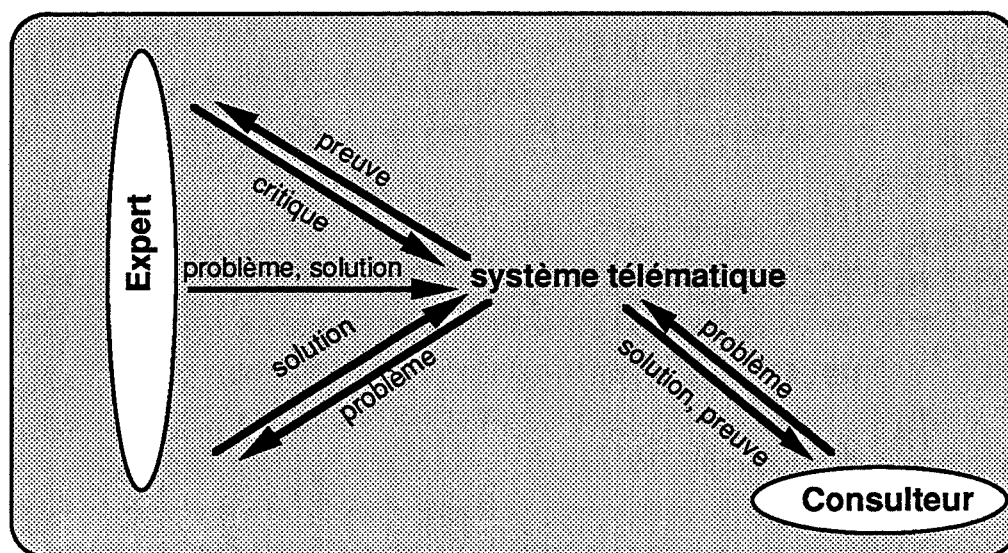


fig. 3.5 : formalisation des dialogues avec le système télématique

Reitz affirme que l'Apprenti ne doit fournir une réponse à son Client, que lorsque sa réponse est valide vis à vis de son Oracle, et pertinente vis à vis de son Maître ([Reitz 92]). Pour cela, il prévoit que les dialogues Apprenti/Client et Apprenti/Maître soient simultanés de façon à ce que l'Apprenti puisse interroger son Maître avant de répondre à son Client. Pour notre application, ce n'est pas possible, car nous n'avons pas toujours un Maître (i.e. un expert) disponible immédiatement : les deux dialogues du système-Apprenti, avec l'expert-Maître d'une part, et avec le consulteur-Client d'autre part, sont indépendants dans le temps. En conséquence, l'Apprenti doit répondre à son Client avant d'avoir un échange avec un Maître. Il nous faut

donc trouver un autre moyen pour que l'Apprenti fournisse des réponses valides et pertinentes au Client.

En ce qui concerne la validité de la solution que l'Apprenti fournit au Client, nous ne pouvons pas faire référence à l'Oracle, comme le fait Reitz, puisque, dans notre cas, nous n'avons pas d'Oracle. Nous pouvons, néanmoins, faire référence aux couples (problème, solution) fournis par l'expert-Sonde, et qui sont des exemples.

Nous dirons que la théorie de l'Apprenti est valide si elle permet de retrouver toutes les solutions aux problèmes des couples (problème, solution) qui sont des exemples. Et nous posons qu'une solution à un problème, fournie par l'Apprenti au Client, est valide si la théorie de l'Apprenti est valide.

Mais se pose alors le problème de l'évolution du domaine (dans le monde réel). Nous désirons, bien évidemment, que la théorie de l'Apprenti suive aussi cette évolution. Ainsi, un exemple à un instant donné, ne gardera pas forcément toujours son statut d'exemple. Nous ne pouvons donc que considérer la validité d'une théorie à un instant donné.

En nous situant dans le cadre d'un apprentissage incrémental à partir d'exemples (ce qui est notre cas), nous considérons donc qu'à partir d'une théorie valide et d'un couple (P, S) qui est un exemple, on construit une nouvelle théorie qui est valide si et seulement si elle permet de retrouver la solution S au problème P, ainsi que tous les exemples précédemment appris et qui ne sont pas remis en cause explicitement par l'expert-Sonde. Nous admettons que la théorie initiale de l'Apprenti est valide; entre autres, la théorie "vide" (celle qui permet à l'Apprenti de répondre uniquement par le silence) est valide. La validité des réponses de l'Apprenti est donc garantie par l'algorithme de construction incrémentale de la théorie.

Pour la pertinence de la preuve, fournie par l'Apprenti au Client, le même problème se pose. Nous posons qu'une preuve, fournie par un apprenti dont la théorie est pertinente, est pertinente. Pour cela, nous considérons que la théorie de l'Apprenti est pertinente, à un instant donné, lorsque qu'elle est construite à partir d'une théorie pertinente et d'une preuve dont la critique par le Maître est l'acceptation, et lorsque toutes les preuves que le Maître refuse, ne peuvent plus être engendrées par cette théorie. Nous admettons que la théorie initiale de l'Apprenti est pertinente; entre autres, la théorie "vide" (celle qui permet à l'Apprenti de répondre uniquement par le silence, et donc de ne fournir qu'une seule preuve : "je me tais

parce que je ne sais rien") est pertinente. La pertinence des réponses de l'Apprenti est donc garantie par l'algorithme de construction incrémentale de la théorie.

3.3.1.3 L'acquisition de la connaissance par Minitel

L'acquisition de la connaissance de surface est donc effectuée par notre système télématique au moyen de dialogues système/expert. Ces dialogues peuvent prendre plusieurs formes :

- dans la forme la plus simple de dialogue, un expert nous fournit un cas qu'il a résolu, formellement un couple (problème, solution). Pour cela, il décrit son problème comme s'il était un simple consultant du système (éventuellement, il modifie la connaissance statique spécifique en ajoutant de nouveaux descripteurs) et fournit ensuite sa solution; le système répond alors par une preuve de son accord ou de son désaccord. Puis l'expert critique cette preuve soit en réfutant un lemme de la preuve, soit en acceptant la preuve. Dans le cas d'une réfutation, le système modifie sa base de connaissance pour fournir une nouvelle preuve tenant compte de la réfutation, et ainsi de suite jusqu'à ce que l'expert accepte la preuve du système. A la fin, lorsqu'aucune preuve n'est plus réfutée, le système intègre le couple (problème, solution) si toutes les preuves sont des preuves d'accord avec l'expert — i.e. le couple (problème, solution) est un exemple.

- dans l'autre forme de dialogue, c'est le système qui a repéré une lacune dans sa base de connaissance. Il profite alors qu'un expert soit connecté au système pour lui soumettre un problème. L'expert fournit donc une solution. Ensuite un dialogue similaire à celui décrit ci-dessus s'engage : le système prouve et l'expert réfute ou accepte.

3.3.1.4 La redistribution de la connaissance par Minitel

Parallèlement à cette acquisition, le système peut être consulté. Pour cela un consultant décrit son problème; en retour, le système lui fournit une solution. Si le consultant le désire, le système se justifie en fournissant une preuve de sa solution.

On pourra aussi, si cela rend l'outil plus ergonomique, prévoir que le système fournisse des conclusions intermédiaires dès qu'il le peut, au fur et à mesure que le consultant caractérise son problème. Le consultant pourra alors aussi demander une preuve d'une conclusion intermédiaire.

3.3.2 Le simulateur

Comme pour le système télématique, le simulateur n'aura pas pour unique mission d'acquérir de la connaissance. Sa mission sera aussi pédagogique, et il sera utilisé pour l'enseignement des sciences de l'eau.

Sa structure sera la même que celle de la Station de Travail Intelligente, avec toutefois deux modules supplémentaires : un module d'acquisition de la connaissance, et un module de gestion des perturbations.

3.3.2.1 Formalisation des dialogues avec le simulateur

Pour l'étude préliminaire du module d'acquisition, nous pouvons formaliser les dialogues entre le simulateur de projets et ses utilisateurs. Pour cela nous devons considérer deux situations différentes : le simulateur en temps que système d'Acquisition Interactive de la Connaissance (AIC), et le simulateur en temps que système d'Enseignement Intelligemment Assisté par Ordinateur (EIAO).

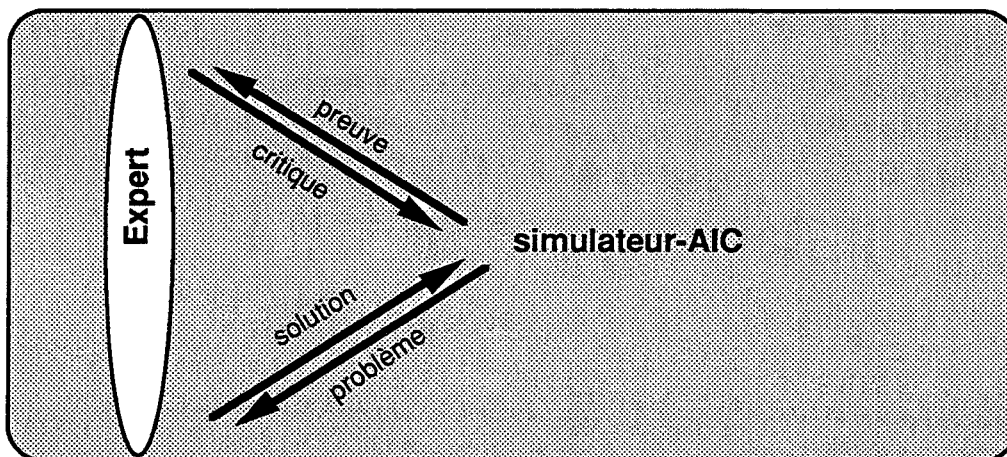


fig. 3.6 : formalisation des dialogues avec le simulateur-AIC

Le simulateur-AIC :

Dans la première situation (cf. figure 3.6), le simulateur-AIC a pour mission d'acquérir la connaissance de stratégie et la connaissance statique générale.

Dans cette situation, le simulateur-AIC est un **Apprenti** qui se trouve face à un expert. L'expert joue le rôle de **Sonde** : l'Apprenti lui propose un contexte d'avancement du projet et l'expert-Sonde prend une décision en fonction de ce contexte. Le contexte est le problème à résoudre, et la décision est la solution du problème. L'expert est aussi le **Maître** car il doit critiquer les preuves du simulateur-AIC. En effet, après réponse de l'expert-Sonde, un échange de preuves et de critiques a lieu, jusqu'à ce que l'expert-Maître accepte la preuve.

Le simulateur-EIAO :

Dans la seconde situation (cf. figure 3.7), le simulateur est un système d'Enseignement Intelligemment Assisté par Ordinateur, et l'utilisateur est un élève.

L'élève joue le rôle de l'**Apprenti**, et le simulateur joue les rôles du **Maître** et du **Client**. Il n'y a, dans ce cas, aucun **Oracle**. Le seul qui pourrait jouer ce rôle serait le simulateur-EIAO, mais l'élève n'a aucun intérêt de se voir fournir des problèmes résolus et non expliqués (pédagogie oblige !). On pourra éventuellement envisager que le simulateur joue le rôle de **Sonde**, en proposant des problèmes résolus à l'élève qui devra émettre une preuve de son accord ou de son désaccord avec la solution.

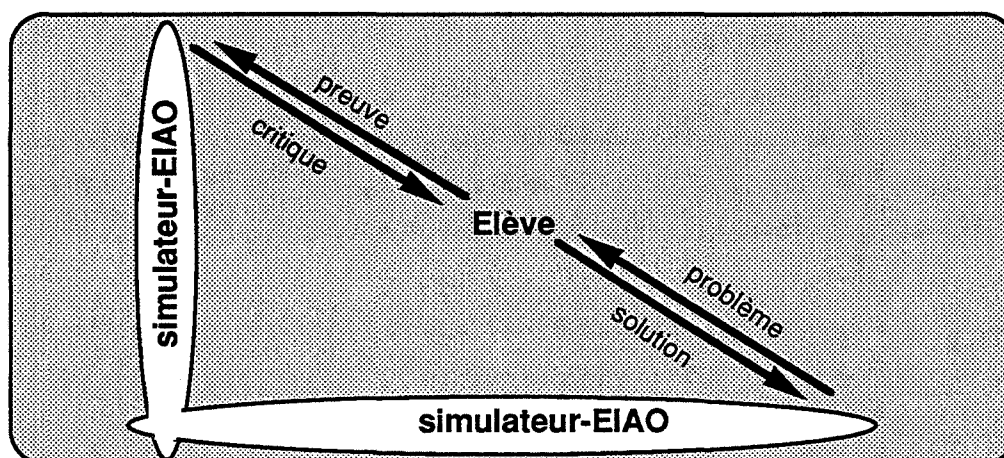


fig. 3.7 : formalisation des dialogues avec le simulateur-EIAO

Le simulateur-EIAO, dans le rôle du **Maître**, doit critiquer les preuves fournies par l'élève. Dans le rôle du **Client**, le simulateur-EIAO soumet des problèmes non résolus à l'utilisateur, au fur et à mesure que le projet avance.

De son côté, l'élève est donc amené à fournir au simulateur-EIAO soit des solutions, soit des preuves de solutions.

3.3.2.2 Les perturbations

Le simulateur a donc deux missions : un système d'Acquisition Interactive de la Connaissance, et un système d'Enseignement Intelligemment Assisté par Ordinateur.

Pour ces deux missions, un module supplémentaire est nécessaire : un module de création des perturbations.

Dans le cas du système d'EIAO, ces perturbations ont pour but de rendre la simulation plus attractive et plus conforme à la réalité. En effet, dans un projet réel, lorsque l'aménageur en eau prend une décision, cette décision se concrétise par une réalisation, et cette réalisation n'est pas toujours exactement conforme à ce qu'attendait l'aménageur. Cet aspect de la simulation des projets d'aménagement en eau a déjà été étudié dans [Graillot 86] pour le simulateur MISE, et dans [Cres 89] pour le simulateur PROMISE.

C'est par exemple prévoir 10 journées pour le creusement d'une tranchée (qui n'en nécessite que 9), et perturber cette prévision en "créant" une panne grave de la pelleuse d'une durée de 3 jours, ce qui retarde tout le planning et demande sa réorganisation. Sans ces perturbations, le montage d'un projet sera "trop facile" et donnerait à l'élève une vision quelque peu idéale de la gestion d'un projet d'aménagement en eau.

Dans le cas du système d'Acquisition Interactive de la Connaissance, ces perturbations ont le même but : rendre la simulation plus proche de la réalité, de façon à mettre l'expert dans une situation la plus proche possible de celle qu'il vit dans son travail.

Dans le paragraphe suivant, nous allons examiner la mise à jour de la connaissance de la Station de Travail Intelligente. Pour cela nous allons formaliser les dialogues entre la machine et son utilisateur (l'exploitant).

3.3.3 La Station de Travail

Cette mise à jour intervient notamment, nous l'avons vu au paragraphe 3.1, pour la modification des accès aux banques de données télématiques.

Plus généralement, cette mise à jour est une actualisation de la Station de Travail, par son exploitant. C'est aussi un apprentissage, que l'on peut rendre automatique. Dans le cas de la

modification des accès aux banques de données télématiques, il ne s'agit que d'un apprentissage par instruction, donc purement déductif. Mais dans certains cas, il peut s'agir d'un apprentissage inductif, notamment lorsque l'exploitant de la Station de Travail conteste un conseil émis par elle. Nous allons donc formaliser le dialogue entre la Station de Travail et son exploitant.

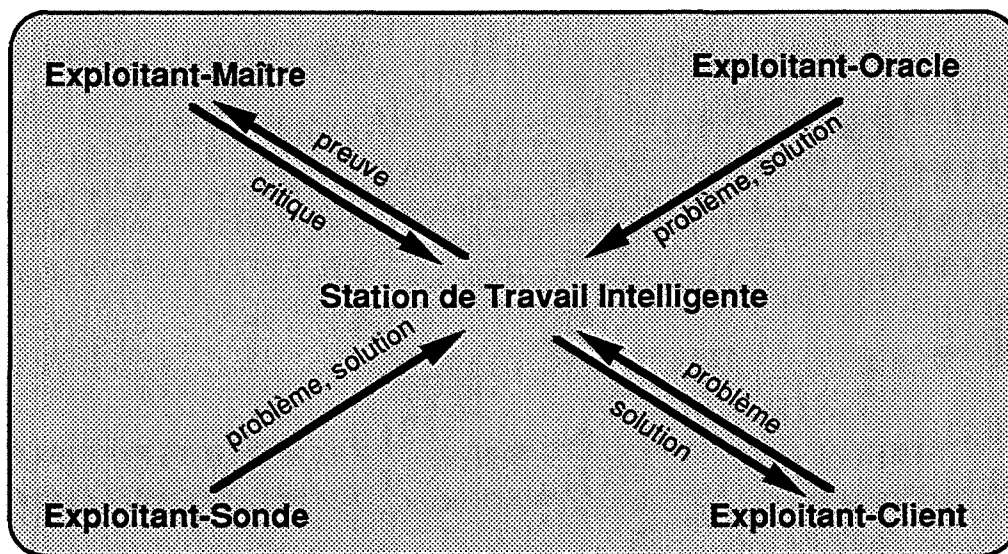


fig. 3.8 : formalisation des dialogues avec la Station de Travail

A chaque étape du projet en cours d'études, l'exploitant reçoit de la Station de Travail des conseils de prise de décision (avec des preuves de ces conseils). Eventuellement, il peut critiquer ces preuves de façon à opter pour une décision précise, qu'il fournit alors à la Station de Travail. Pour reprendre l'environnement d'apprentissage décrit en figure 3.3, on peut dire que la machine est l'Apprenti, et que son exploitant joue les rôles de Sonde, Maître, Oracle et Client (cf. figure 3.8) :

- Client car il soumet des problèmes à la machine qui lui renvoie des solutions (exploitation ordinaire),
- Sonde car il peut imposer une solution à la machine et désirer que la machine "comprenne" cette solution, par le biais d'échanges de preuves et de critiques,
- Maître car alors il est amené à critiquer les preuves de la machine, et enfin
- Oracle car il peut ne pas être disponible, parfois, pour critiquer les preuves; il désire alors seulement que la machine "admette" sa solution.

Dans ce cas, il n'y a donc plus de requête de la machine : elle est esclave et l'utilisateur prend toutes les initiatives. Le dialogue commence donc toujours par l'envoi d'un problème par l'utilisateur-Client. Puis l'utilisateur se comporte, à sa guise, en tant que Maître, Sonde ou Oracle.

3.4 ARCHITECTURE DE LA STATION DE TRAVAIL

Pour étudier la Station de Travail Intelligente de l'aménageur en eau, nous avons scindé la connaissance experte en trois niveaux, et avons examiné ses différents modes d'acquisition. Cela nous a amené à proposer trois nouveaux outils :

- un système télématique d'acquisition interactive de la connaissance qui est étudié dans le chapitre suivant,
- un système d'Enseignement Intelligemment Assisté par Ordinateur, et
- un système d'Acquisition Interactive de la Connaissance.

Ces deux derniers outils partagent un grand nombre de fonctionnalités avec la Station de Travail Intelligente de l'aménageur en eau : nous allons donc proposer une architecture commune à ces trois systèmes.

3.4.1 Représentation de la connaissance

Les trois paradigmes du langage, adoptés par Vogel pour KOD (cf. chapitre 2), nous proposent trois formes de représentation de la connaissance : l'*être* conduit aux objets, le *faire* conduit aux méthodes et le *dire* conduit aux règles de productions.

Nous allons reprendre ces trois formes de représentation de la connaissance, parallèlement à l'étude des différents types de connaissances :

- la connaissance de surface d'une part, et d'autre part la connaissance profonde contenant :
- la connaissance statique, c'est à dire la connaissance décrivant la sémantique du domaine, et
- la connaissance de stratégie, qui décrit la résolution générale d'un problème du domaine.

La connaissance de stratégie traduit le comportement actif de l'expert (le *faire*). Elle peut donc s'exprimer sous forme d'actinomie et être représentée à l'aide de méthodes (au sens des scripts d'acteurs). Les acteurs, dans ce cas, sont des problèmes et des sous-problèmes.

Résoudre un problème revient alors à envoyer un message à l'acteur sachant résoudre ce problème. Cet acteur, à son tour, après exécution de sa méthode (son script), fera apparaître un ou plusieurs autres problèmes (sous-problèmes du problème initial) et enverra un ou plusieurs messages aux acteurs sachant résoudre ces sous-problèmes. Ainsi de suite jusqu'à atteindre des acteurs qui savent résoudre leur problème sans faire appel à d'autres acteurs.

Certains acteurs pourront éventuellement faire appel à des procédures externes (modèles physiques, banques de données, etc...) pour résoudre leur problème. Ces procédures externes peuvent aussi être représentées sous forme d'acteurs. C'est le choix que nous ferons afin de simplifier la tâche du Système d'Exploitation de la Station de Travail.

La connaissance statique, dans les paradigmes proposés par KOD, correspond à l'*être* (elle traduit le comportement descriptif de l'expert). Elle peut donc être représentée informatiquement avec un langage de classes.

La connaissance de surface sera représentée sous forme de règle de production. En effet, elle correspond bien au mode déclaratif du discours de l'expert présenté dans la méthodologie KOD. Ces bases de connaissance (qui sont donc des bases de règles) seront manipulées par un moteur d'inférence capable de plusieurs formes de raisonnements.

Puisque ce moteur d'inférence joue le même rôle que les procédures externes (résoudre un problème élémentaire), ce moteur d'inférence pourra être représenté de la même façon que les procédures externes : sous la forme d'un acteur.

3.4.2 Description de la Station de Travail

Nous choisissons donc les formalismes de représentation de la connaissance suivants :

- acteurs pour la connaissance de stratégie,
- classes pour la connaissance statique, et
- règles de production pour la connaissance de surface.

On a aussi partagé la connaissance de surface en plusieurs bases de règles, distinctes et indépendantes, et la connaissance statique en connaissance statique générale et connaissance statique spécifique.

De plus, et par souci d'unification des mécanismes du Système d'Exploitation, les procédures externes et le moteur d'inférence chargé de la gestion de la connaissance de surface seront représentés sous la forme d'acteurs. En ce qui concerne le module d'explications, nous choisissons aussi de le représenter sous la forme d'un acteur, toujours par souci d'unification des mécanismes du Système d'Exploitation.

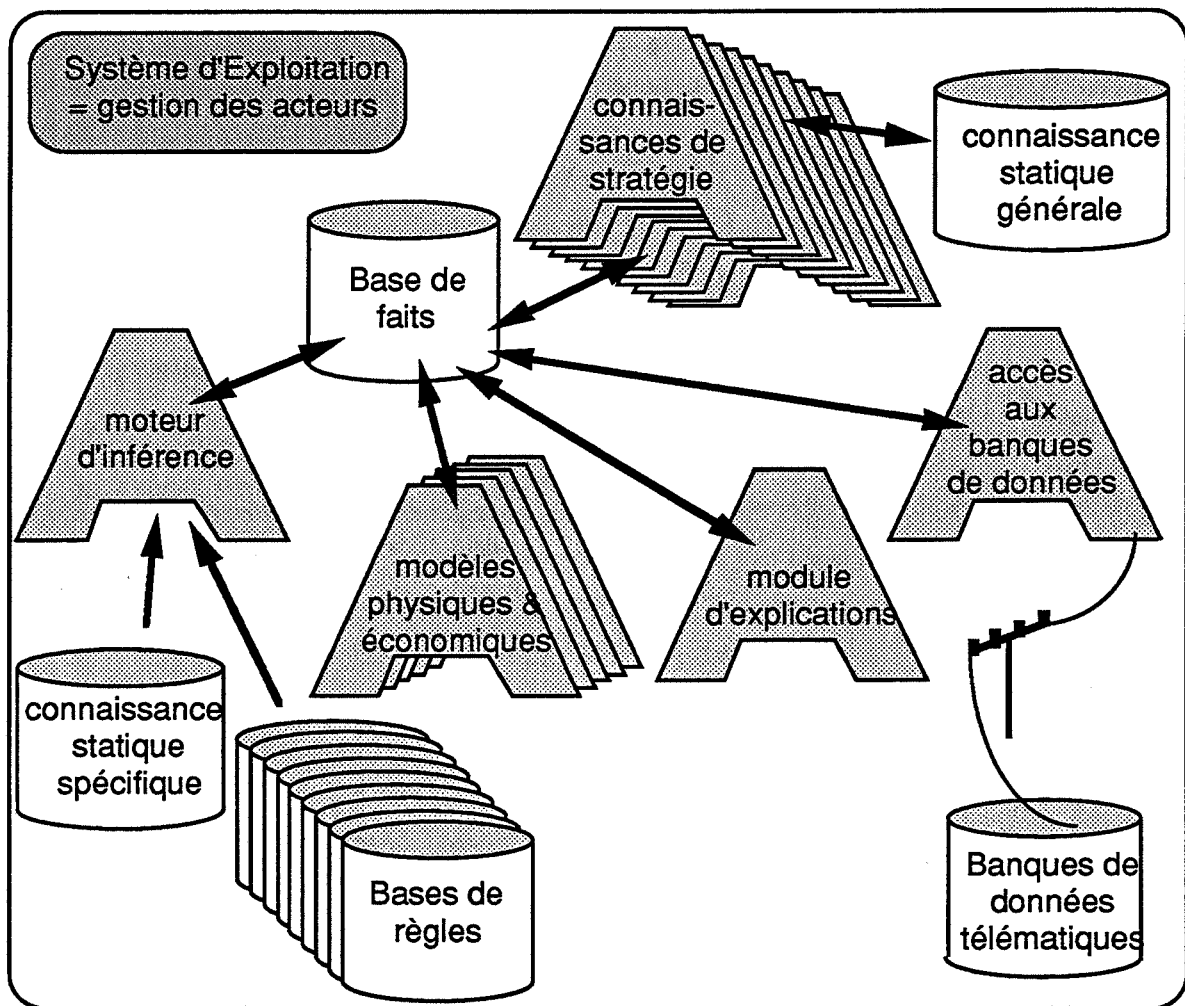


fig. 3.9 : architecture de la Station de Travail Intelligente de l'aménageur en eau

Ce dont a besoin ce dernier module est une trace des raisonnements effectués. Cette trace peut être laissée dans la base de faits : chaque fait caractérisant le projet en cours d'étude sera mémorisé avec son origine. Cette trace des faits peut être organisée en arborescence. La racine

de l'arbre sera "résoudre un problème d'aménagement en eau" et expliquer consistera en un parcours progressif de cet arbre.

Si le fait est l'instantiation d'un paramètre, l'explication consiste en un rappel de l'origine de l'instantiation (fourni par l'utilisateur, par une banque de données, par un modèle, etc...)

Si le fait est une décision prise par un acteur, l'explication se trouve dans le nom de l'acteur. Par exemple "choisir un système d'assainissement", "trouver le nombre d'habitants de la commune", "évaluer le coût du projet", "déterminer la date au plus tôt de la fin de telle tâche", etc...

On peut aussi prévoir un module d'explications plus "intelligent". Il pourrait reposer sur un modèle de l'utilisateur, et fournir des explications dont la profondeur correspondrait mieux au niveau de l'utilisateur. En effet, un utilisateur expérimenté n'a pas besoin de tous les détails du raisonnement, les grandes étapes lui suffisent. Inversement, le débutant attend qu'on lui détaille tout le raisonnement pour mieux le comprendre.

Le Système d'Exploitation aura donc pour unique rôle la gestion de tous les acteurs, et l'enregistrement, dans la base de faits, de toutes les informations concernant le projet en cours d'études sera assuré par chacun des acteurs.

Tout ceci nous amène à préciser la description fonctionnelle proposée au paragraphe 3.1.1. Le cœur de la Station de Travail est désormais la gestion de tous les acteurs (cf. figure 3.9).

3.5 CONCLUSION

Dans ce chapitre, nous avons étudié l'architecture informatique de la Station de Travail Intelligente de l'aménageur en eau, et avons précisé comment les différentes formes de connaissances de cette Station peuvent être acquises.

Pour cela, nous avons défini deux nouveaux outils : un système télématique d'une part, et un simulateur de projets d'autre part.

En ce qui concerne le simulateur de projets, nous avons vu qu'il pouvait avoir deux missions : celle d'enseigner et celle d'acquérir de la connaissance. Dans les deux cas, sa structure informatique sera en grande partie celle de la Station de Travail. Les différences se situent au niveau de l'interface avec l'utilisateur. La structure présentée en figure 3.9 peut donc être considérée comme une "plate-forme informatique de simulation" sur laquelle viennent se greffer

différents modules d'interface-utilisateur, selon l'emploi que l'on veut faire de la simulation (cf. figure 3.10) :

- pour l'acquisition de la connaissance, la machine doit pouvoir exposer des problèmes et prouver son accord ou son désaccord avec les solutions proposées par l'expert; elle doit, de plus, être capable d'apprentissage pour modifier sa connaissance de façon à devenir d'accord avec l'expert,
- pour l'enseignement assisté par ordinateur, il faut que la machine puisse formuler des problèmes et critiquer des preuves,
- pour l'exploitation de la Station de Travail, la machine doit pouvoir proposer des solutions, et fournir des preuves; de plus, elle doit être capable d'apprentissage pour actualiser sa connaissance selon les désirs de son exploitant.

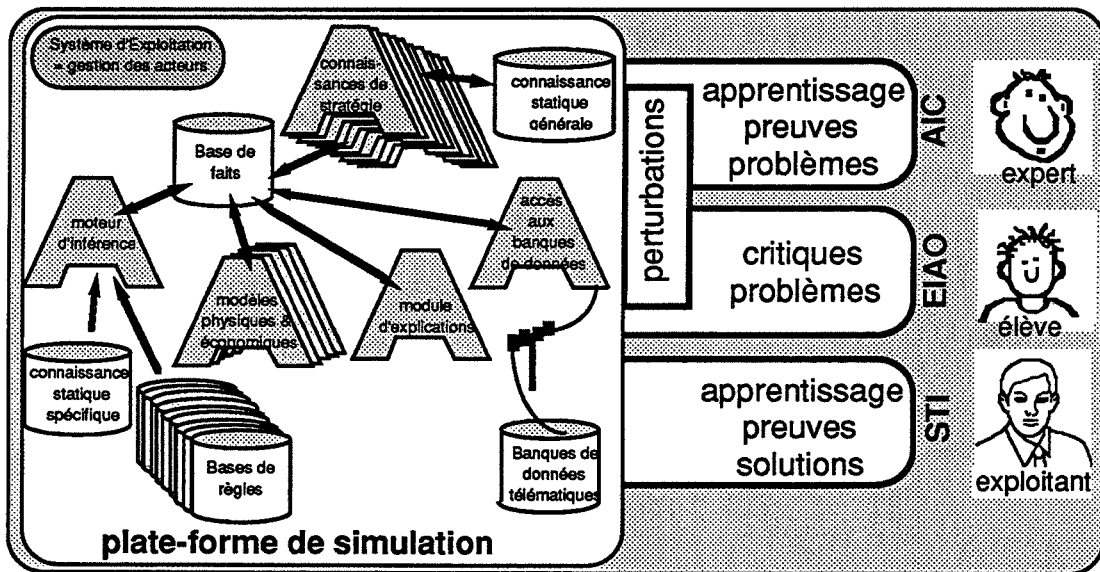


fig. 3.10 : la plate-forme de simulation et ses différents emplois

Une étude plus approfondie de la plate-forme de simulation et des différentes interfaces-utilisateurs sera réalisée ultérieurement dans notre laboratoire. Notamment, les divers modules d'acquisition de la connaissance devront être détaillés.

En ce qui concerne le système télématique d'acquisition interactive de la connaissance, nous lui avons défini deux missions :

- acquérir la connaissance pour alimenter la connaissance de surface de la plate-forme de simulation, et
- redistribuer la connaissance acquise; cette deuxième mission a pour objet de rendre possible la réalisation du système télématique.

Dans les chapitres qui suivent, nous étudions en détail ce système télématique.

Chapitre 4

Moïse : un système télématique d'acquisition interactive de la connaissance à base de RDP

Afin de pouvoir contacter les principaux acteurs des projets d'assainissement autonome des eaux usées, nous avons réalisé une maquette de serveur expert (par *serveur expert*, nous entendons un "Système Expert implanté sur un serveur télématique accessible par Minitel®"). Cette maquette de serveur expert n'est pas dotée de l'élément principal de notre étude : le module d'acquisition de la connaissance, mais elle nous a permis d'observer la réaction des utilisateurs du système (en phase de consultation). Ce sont ces observations qui nous ont aidées, entre autres, à caractériser le domaine d'étude. Un autre objectif de cette réalisation (en 1987-1988) était de prouver aux acteurs de ces projets, qu'un système expert était réalisable dans leur domaine, et qu'il pouvait être accessible par Minitel®.

Dans ce chapitre, après avoir décrit notre maquette de serveur expert, nous présentons le système Moïse, système télématique d'acquisition interactive de la connaissance. Ce système

réalise l'acquisition de la connaissance par apprentissage automatique à partir d'exemples et par dialogue avec l'expert. Après avoir justifié le choix de représentation de la connaissance avec les Réseaux de Dépendances Perçues (RDP), nous présentons l'algorithme de base (implanté dans le logiciel EDGE) qui permet la fabrication et l'entretien incrémental du RDP, ainsi que les dialogues prévus au chapitre précédent.

4.1 REALISATION D'UN SYSTEME EXPERT TELEMATIQUE

En 1987, au début de notre travail de recherche, les Systèmes Experts accessibles par Minitel® étaient relativement rares (au niveau mondial : 50 réalisations significatives, d'après [Delmas – Vernet 88]), et leur réalisation nécessitait a priori l'écriture complète d'un moteur d'inférence. En effet, les seuls Générateurs de Systèmes Experts performants du marché étaient des systèmes très fermés, et il n'était pas envisageable de déporter l'interface utilisateur, prévue sur l'écran-clavier de l'ordinateur, vers une sortie série connectée à un modem.

D'un autre côté, les premiers logiciels serveurs sur micro-ordinateur, étaient essentiellement prévus pour gérer des ensembles de pages écrans et des messageries. Au mieux, ils pouvaient être interfacés à un tableur ou à un Système de Gestion de Bases de Données. Quant aux Sociétés de Services Informatiques spécialisées en télématique, qui auraient pu héberger notre serveur expert, elles ne disposaient pas de *shell* de système expert.

Notre objectif limité (réaliser une maquette de serveur expert, pour convaincre les différents acteurs des projets d'assainissement autonome des eaux usées, du bien-fondé de notre projet de système télématique d'acquisition de la connaissance) rendait prohibitif l'écriture complète soit d'un logiciel serveur, soit d'un moteur d'inférence (pour nous, ou pour une Société de Services Informatiques). Nous avons donc opté pour un moyen terme, et avons choisi un logiciel serveur (Hostel®) légèrement ouvert sur l'extérieur, et un langage de programmation (prolog), pouvant être utilisé comme un Générateur de Système Expert suffisamment ouvert sur les divers éléments du matériel.

4.1.1 Architecture

L'avantage du logiciel serveur Hostel® est qu'il peut commander l'exécution d'un processus externe. Puisque Hostel® est implanté sur MSDOS®, système d'exploitation monotâche, son fonctionnement repose sur l'utilisation soit d'un sur-système d'exploitation (DoubleDos®) qui

permet l'utilisation simultanée de deux tâches MSDOS[®], soit sur l'utilisation d'un autre système d'exploitation (PCMOS[®]) qui est multitâche et dont chaque tâche est compatible avec MSDOS[®].

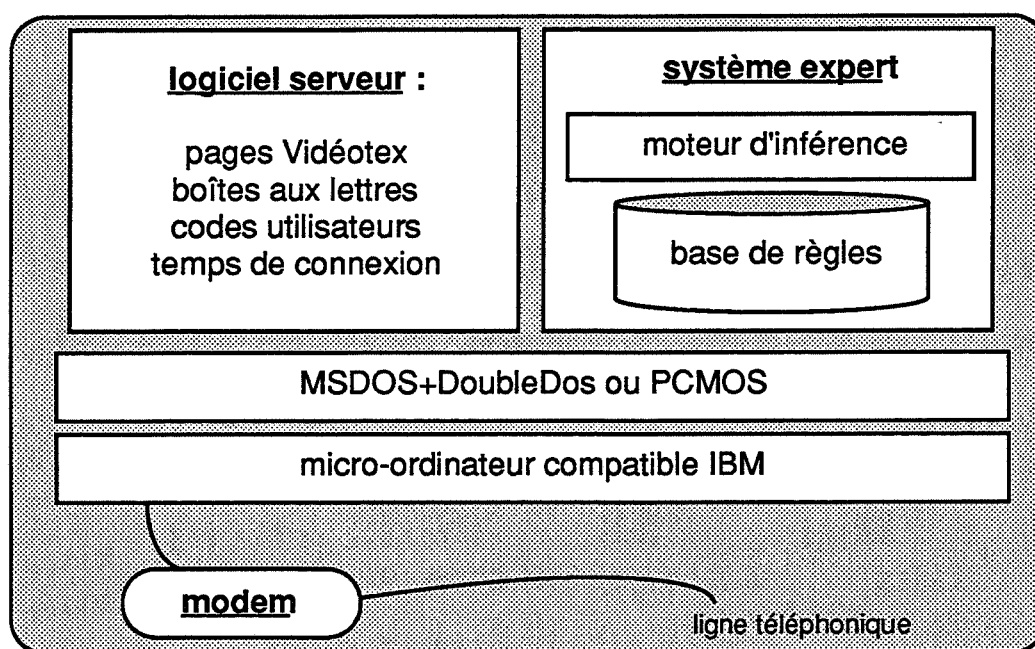


fig. 4.1 : architecture de la maquette de serveur expert

Dans un premier temps, nous avons utilisé DoubleDos[®] : l'inconvénient majeur de cette solution est que MSDOS[®] ne peut pas gérer au-delà de 640 Koctets de RAM, et donc que DoubleDos[®] ne peut que partager ces 640 Koctets en deux. Une fois retranchés MSDOS[®], DoubleDos[®] et Hostel[®], il reste moins de 300 Koctets pour le Système Expert. Pour la première version de la maquette de serveur expert, nous avons réussi, au détriment de la lisibilité du programme, à réduire la taille de notre système expert.

Dans un deuxième temps, nous avons utilisé PCMOS[®], qui offre l'avantage de permettre l'exécution de plusieurs processus pouvant occuper chacun jusqu'à 580 Koctets de RAM.

L'architecture de la maquette de serveur expert (cf. figure 4.1) est donc composée de deux tâches :

- le logiciel serveur gère les pages Vidéotex[®] dans lesquelles viendront s'inscrire des messages issus de l'exécution de l'autre tâche. Il gère aussi les boîtes aux lettres, les noms des utilisateurs et les statistiques sur les temps de connexion.

- le système expert est composé d'un moteur d'inférence (ordre 1 en chaînage arrière) à base de prolog, d'une base de règles, et d'un module "mouchard" qui enregistre, dans le détail, toutes les consultations du système.

4.1.2 Interface utilisateur

La base de règles de la maquette du serveur expert (écrite à partir de la connaissance des hydrogéologues de notre équipe) permet la description, par 9 paramètres, de la parcelle destinée à accueillir le système d'assainissement, pour un choix parmi 14 techniques d'assainissement autonome des eaux usées.

Au début d'une session de consultation du système, l'utilisateur a le choix entre deux modes de consultation :

- recherche d'une solution : l'utilisateur décrit la parcelle de terrain étudiée et le système lui retourne un conseil,
- ou bien vérification d'une technique d'assainissement : l'utilisateur fait le choix d'une technique parmi celles proposées par le système, puis celui-ci le questionne pour vérifier si la parcelle convient ou non à cette technique.

Dans les deux modes (déduction et vérification), la description de la parcelle est guidée par le système qui demande les valeurs, ou les fourchettes de valeurs, des paramètres dont il a besoin pour pouvoir porter un jugement. A tout moment l'utilisateur dispose d'une aide en ligne qui lui explique la signification du paramètre et comment l'évaluer, et peut interrompre la session pour se rendre dans la messagerie et nous laisser ses remarques puis reprendre l'expertise où elle en était.

En fin de session :

- le système, en mode déduction, peut conseiller une technique d'assainissement (en la décrivant avec des schémas, et en justifiant son choix); en mode vérification, le système indique si la technique choisie au départ est convenable ou non (en se justifiant); ou bien
- il peut expliquer pourquoi l'assainissement autonome est déconseillé; en dernier recours
- il peut avouer son ignorance, et conseiller à l'utilisateur de se reconnecter plus tard, lorsque les paramètres de sa parcelle auront été étudiés par notre équipe.

Ensuite l'utilisateur peut, s'il le désire, modifier l'un des paramètres entrés afin d'étudier la sensibilité du conseil par rapport à ce paramètre (paramètre dont la valeur est mal connue, par

exemple); ou bien il peut, tout en gardant les mêmes paramètres, choisir une autre technique d'assainissement, et vérifier si elle convient.

Enfin, si l'utilisateur est en mode vérification, il peut demander au système, en fin de session, s'il existe une meilleure solution que celle qu'il avait choisie au préalable. Le système passe alors automatiquement en mode déduction.

4.1.3 Exploitation des consultations

La maquette du serveur expert a été mise en service sur le réseau téléphonique commuté (RTC) pendant plusieurs semaines au printemps 88, à l'usage de plusieurs professionnels de la télématique ou de l'assainissement, afin de recueillir des critiques sur notre système.

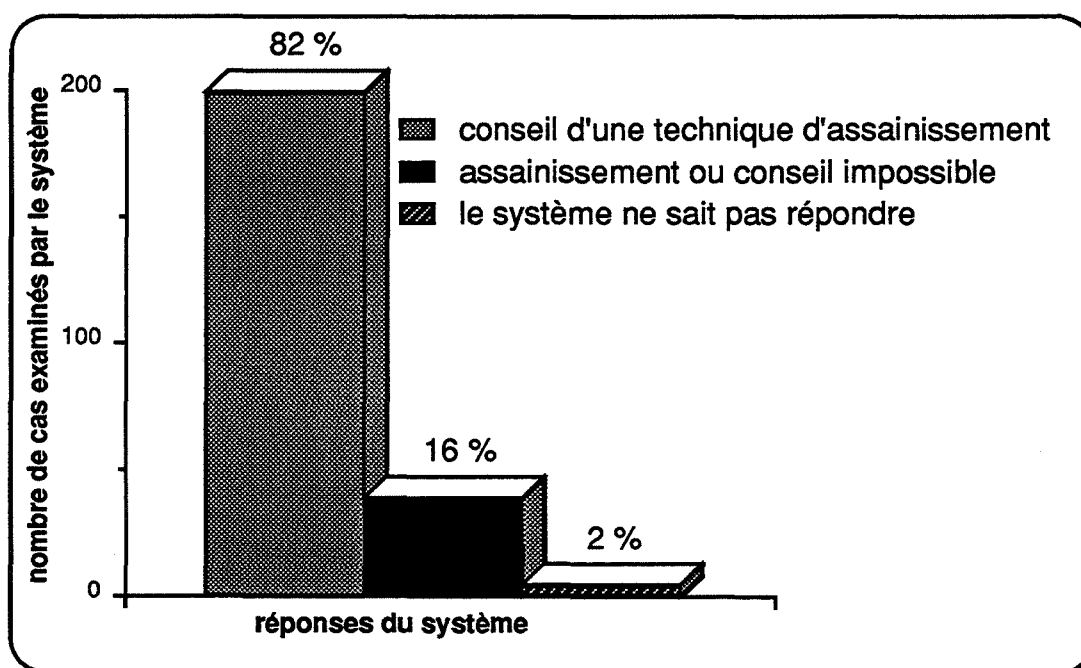


fig. 4.2 : répartition des types de réponses du serveur pendant la période d'essai

Nous avons ainsi recueilli **242 sessions d'expertise** (déduction ou vérification) se répartissant selon la figure 4.2, pour 47 connexions dont 8 venant de personnes n'exerçant pas dans le domaine de l'assainissement des eaux usées.

Les deux colonnes de droite de la figure 4.2 représentent tous les cas où le serveur ne conseille pas de système d'assainissement :

- pour la colonne de droite (2% des cas), il s'agit du cas où le serveur *n'a pas su* interpréter les paramètres. Une étude de ces paramètres a permis aux hydrogéologues de notre équipe d'enrichir la base de règles.
- pour la colonne centrale (16% des cas), il s'agit soit d'une *impossibilité* de déduire un conseil (il manque un paramètre important), soit d'une *impossibilité* d'adapter un système d'assainissement sans faire une étude plus approfondie du site, mais le serveur a su expliquer pourquoi. Cette colonne se répartit ainsi :
 - 9 % des cas où il manque un paramètre important de la parcelle. Si l'utilisateur du système ne sait pas déterminer la valeur de ce paramètre, une étude doit être faite par un bureau d'études,
 - 50 % où l'impossibilité d'adapter un assainissement autonome sans étude approfondie provient de la valeur extrême d'un seul paramètre : par exemple "*pente du site trop forte*", ou bien "*puits de captage trop proche*", ces cas sont facilement explicables,
 - 41 % où l'impossibilité provient d'une certaine combinaison des paramètres, par exemple : "*si le rejet superficiel est impossible, si la nappe n'est pas profonde, et si la perméabilité du sol est forte, alors risque de pollution de la nappe par les effluents*". Ces cas nous montrent l'importance des concepts intermédiaires (ici *risque de pollution de la nappe par les effluents*) pour que le système puisse fournir des explications.

La colonne de gauche de la figure 4.2, qui représente 82% des cas examinés par le serveur pendant la période de tests, est détaillée par la figure 4.3 (les 14 colonnes correspondent aux 14 systèmes d'assainissement prévus par la base de règles de la maquette du serveur expert). Nous y remarquons que les différents systèmes d'assainissement n'ont pas été consultés de façon uniforme : 8 systèmes (sur 14) représentent 87 % des sessions.

Le système le plus consulté (25% des cas) est "*l'épandage en tranchées parallèles à la pente du site*" : c'est en effet le système le plus fréquemment mis en oeuvre : il correspond à un type de contexte largement répandu. Ceci est confirmé par le fait que, pour 75%, il a été examiné en mode déduction, c'est-à-dire qu'il a été conseillé suite à la description d'une parcelle. Les 25 % restant (mode vérification) correspondent souvent à des personnes non expertes en assainissement des eaux usées et curieuses de connaître ce système et ses limites.

En revanche, le système n° 14, le "*filtre bactérien percolateur avec rejet des effluents dans un puits d'infiltration*" n'a été consulté que dans 1% des cas seulement. Pour ces systèmes très peu souvent installés, il nous faut prévoir que le serveur puisse interroger, de lui-même, les experts, lorsqu'ils sont connectés, de façon à avoir un nombre d'exemples suffisant pour tous les systèmes d'assainissement.

L'appel aux explications des paramètres (aide en ligne) a bien sûr été utilisé par les non-experts de l'assainissement, mais aussi par les professionnels du domaine, ce qui montre a priori le manque d'unanimité sur les concepts à prendre en compte en matière d'assainissement. Par exemple, les hydrogéologues de notre équipe ont fait le choix de considérer la "perméabilité du sol" et la "perméabilité du sous-sol" pour ne pas égarer l'utilisateur dans un grand nombre de mesures physiques, alors que certains experts préfèrent parler de "perméabilité en fonction de la profondeur (1 m, 2 m, ...)" car les concepts de sol et sous-sol ne sont pas clairement définis.

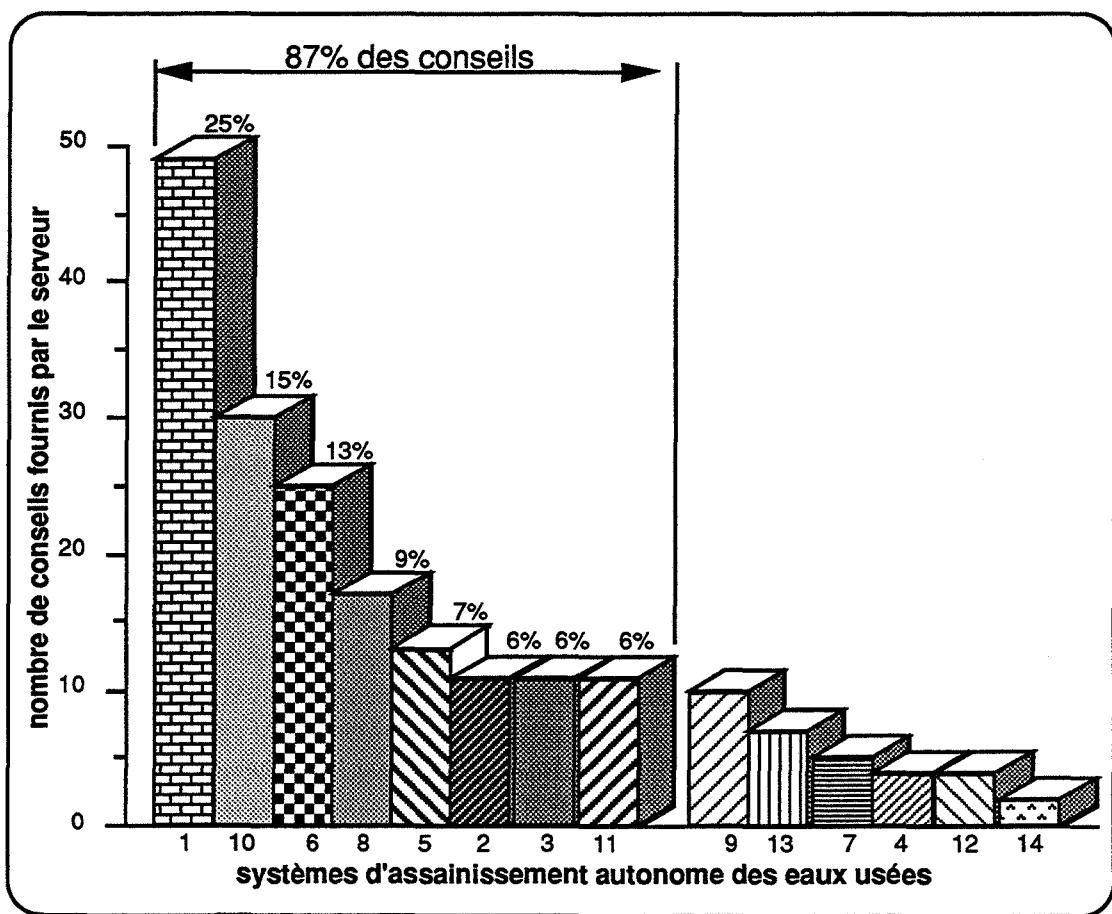


fig. 4.3 : répartition des conseils fournis par le serveur pendant la période d'essai

Pour recueillir des critiques sur notre système, comme nous l'avons dit ci-dessus, nous mettons à la disposition de l'utilisateur une messagerie accessible à tout moment. En plus de ces critiques "volontaires", nous avons à notre disposition une trace écrite et précise de toutes les sessions, qui nous est d'une grande utilité pour "débuguer" la base de connaissance.

A titre d'exemple, elle nous a permis de découvrir une lacune dans notre base de connaissance : à quatre reprises, et avec trois utilisateurs différents, le serveur expert n'a pas fourni de conseil (colonne de droite de la figure 4.2) sur une même combinaison des paramètres. Une fois cette combinaison analysée, nos hydrogéologues ont pu ajouter une règle dans la base. Il s'agissait en fait d'une impossibilité d'adapter un assainissement individuel, mais encore fallait-il indiquer pourquoi.

Autre exemple : dans le cas où la pente de la parcelle est nulle, le serveur conseillait, si les autres paramètres le permettaient, le système "*épandage en tranchées parallèles à la pente*". Nous avons donc traité à part le cas où la pente est nulle.

Les messages "volontaires" des utilisateurs ont porté principalement sur la présentation du serveur : ces commentaires nous permettront de perfectionner la version définitive, capable d'apprentissage automatique.

Les rares messages, relatifs au domaine d'application, ont porté :

- sur des paramètres que nous ne prenions pas en compte, et
- sur la méfiance de certains experts du domaine de l'assainissement, qui ont réagi en affirmant qu'un tel système ne pouvait pas fonctionner sans prendre en compte les spécificités locales, ce qui confirmait notre présomption, mais aucun ne nous a fourni de cas en contradiction avec le jugement porté par le serveur expert.

4.1.4 Bilan de l'expérience

La mise en service de cette maquette, outre les divers renseignements qu'elle nous a apportés sur la base de connaissance et sur l'interface utilisateur, nous a permis de contacter un grand nombre d'experts en assainissement. Notamment, nous avons ainsi été en relation avec l'Agence Financière de Bassin Loire-Bretagne, et avec certains services du Ministère des Affaires Sanitaires et Sociales (dont plusieurs Directions Départementales et la Direction Régionale Rhône-Alpes).

Ces contacts ont été les plus fructueux, car nous avons reçu un accueil très favorable pour notre projet, et l'assurance d'une aide future lorsqu'il faudra convaincre un grand nombre d'experts de nous fournir leur expertise :

- les différents services du Ministère des Affaires Sanitaires et Sociales nous permettront d'avoir accès à l'expertise dispersée dans toutes les Directions Départementales, et
- les Agences de l'Eau (anciennes Agences Financières de Bassin) nous permettront de contacter des experts de nombreuses entreprises privées.

De plus, au niveau de la caractérisation du domaine de l'assainissement autonome des eaux usées, cette expérience nous a confirmé que :

- la connaissance est dispersée géographiquement et dans divers corps de métiers,
- que chaque département a ses propres habitudes, et qu'une uniformisation de la connaissance experte n'est pas envisageable,
- qu'une très grande diversité de paramètres sont à prendre en compte pour effectuer une expertise,
- que certains systèmes d'assainissement, délicats à mettre en œuvre, sont très peu utilisés, sauf lorsqu'aucune autre solution n'est envisageable. Pour ces systèmes il faut que le serveur soit capable d'interroger de lui-même les experts pour que la base d'apprentissage soit suffisamment étoffée.

Les choix informatiques qui avaient été faits pour la maquette, décrite ci-dessus, ne sont pas définitifs. Pour le système télématique d'acquisition interactive de la connaissance, ces choix doivent être reconsidérés. Ce système, décrit dans la suite de ce chapitre, est baptisé "Moïse" : il ne s'agit pas d'un acronyme, mais tout simplement de la reprise du nom du premier système réalisé, dans notre équipe, pour les sciences de l'eau (*MISE*), auquel nous avons rajouté l'*O*. Mais cette appellation ne doit toutefois pas laisser penser que le système soit capable de prophétiser !

4.2 CHOIX DES RDP

Comme nous l'avons vu en conclusion du chapitre 2 et au paragraphe 3.1.4, les systèmes à base de connaissance sont traditionnellement étudiés en tant que systèmes contenant de la connaissance et capables de délivrer cette connaissance à un utilisateur final du système. Le transfert de connaissances entre le système et les experts n'est alors considéré que sous la forme d'un monologue (soit avec l'apprentissage automatique, soit avec une méthode analytique).

Nous pensons que les deux types d'approches soulèvent deux points importants pour réaliser un bon transfert de connaissance :

- d'un côté, avec l'apprentissage automatique, on vise à supprimer les intermédiaires entre l'expert et la machine pour ne pas détériorer l'information transférée,
- de l'autre côté, avec les méthodes analytiques, on dialogue avec l'expert de façon à lui faire expliciter le plus possible sa connaissance.

Nous choisissons donc de reprendre ces deux points dans notre système, et nous optons pour un dialogue direct entre l'expert et la machine de façon à obtenir une acquisition interactive de la connaissance.

Nous avons donc spécifié, au paragraphe 3.3.1, les dialogues entre le système télématique et ses utilisateurs (aussi bien les consultants du système que les experts détenteurs de la connaissance).

Ces dialogues nécessitent que le système soit capable de fournir des preuves sous forme d'enchaînements de lemmes, et qu'il soit capable de modifier sa connaissance pour prendre en compte les exemples et les réfutations de lemmes. Il faut donc que le système soit doté d'un module d'apprentissage incrémental (pour prendre en compte les exemples et les réfutations) et qu'il soit capable de manipuler des relations (les preuves manipulent des conjectures, qui sont des relations). Avec les Réseaux de Dépendances Perçues, nous pouvons réaliser ces dialogues, car ils sont orientés vers la formation de relations entre formules du langage de description choisi, et permettent une construction incrémentale de ces relations. En effet, les autres algorithmes présentés au chapitre 2, soit ne permettent pas un apprentissage incrémental (Charade, ID3), soit sont orientés vers la formation de concepts probabilistes (COBWEB, UNIMEM).

Ces derniers ne nous conviennent pas car notre système doit posséder une connaissance qui permet de résoudre tous les cas fournis par les experts, et non pas la plupart de ceux-ci (choix de la validité de la théorie de l'Apprenti fait au § 3.3.1.2).

Afin de pouvoir détailler ces dialogues, nous présentons ci-dessous les RDP : ce qu'ils contiennent et comment on peut les construire.

4.3 UTILISATION D'UN RDP

A la fin du chapitre 2, nous avons vu que la relation de Dépendance Perçue pouvait être représentée par un graphe appelé Réseau de Dépendances Perçues (RDP). Un RDP est défini par un couple (V_{RDP}, E_{RDP}) où :

- V_{RDP} est un ensemble de nœuds étiquetés par les éléments de l'ensemble des classes d'Equivalence Perçue,

- \mathcal{E}_{RDP} est un ensemble de paires de nœuds de V_{RDP} (i.e. d'arcs orientés) tel qu'une paire de nœuds (n_a, n_b) de V_{RDP} appartient à \mathcal{E}_{RDP} si et seulement si les éléments de la classe d'Equivalence Perçue étiquetant n_a sont en relation de Dépendance Perçue avec les éléments de la classe d'Equivalence Perçue étiquetant n_b .

C'est à dire qu'à l'intérieur d'un nœud, toutes les formules sont en relation d'Equivalence Perçue, et qu'un arc entre deux nœuds représente une relation de Dépendance Perçue.

4.3.1 Représentation de la connaissance

Un RDP permet de représenter toutes les relations de Dépendance Perçue et toutes les relations d'Equivalence Perçue que l'on peut construire entre des formules du langage de description choisi.

Le problème qui se pose est de choisir quelles formules du langage on va considérer dans le RDP. Nous nous limiterons au cas où le formalisme d'expression de la connaissance est un formalisme logique (ce qui convient bien à l'usage que l'on aura de la connaissance acquise : un système expert). Le formalisme que nous choisirons sera au plus la logique des prédicats du 1^{er} ordre. Dans ce cadre-là, et avec une interprétation des formules bien formées du langage de description, nous pouvons choisir, comme relation fondatrice, l'implication logique.

Ainsi, les seules formules qui nous intéressent, sont celles fondées sur au moins un exemple (ce que dit la clause existentielle de la définition de la Dépendance Perçue), c'est à dire les formules \mathcal{F} pour lesquelles il existe un exemple dont la description est \mathcal{d} et tel que $\mathcal{d} \vdash \mathcal{F}$. C'est à dire les formules plus générales que la description d'au moins un exemple.

Parmi celles-ci, seules les formules atomiques sont représentées dans le RDP. En effet, si le langage de description est muni de la disjonction et de la conjonction, toutes les formules complexes peuvent être retrouvées dans un RDP ne contenant explicitement que des formules atomiques ([Sainte Marie 90a]). Nous verrons comment retrouver ces formules complexes dans le § 4.3.1.3.

Nous ne considérons donc que des formules atomiques fondées sur au moins un exemple. Parmi celles-ci, seules nous intéressent celles qui permettent d'effectuer un regroupement d'exemples. En effet, le but de notre système est bien de découvrir tous les regroupements possibles d'exemples, car ils sont susceptibles de signifier un concept du domaine étudié.

Pour examiner comment la connaissance est représentée par un RDP, nous allons considérer un cas très simple, qui n'a aucune valeur dans son domaine d'application : l'assainissement autonome des eaux usées.

4.3.1.1 Une base d'exemples pédagogique

Afin de constituer le langage de description, nous prenons quatre descripteurs :

- **filière** est un descripteur à un attribut nominal, avec comme extension : (*épanchage en tranchées, filtre à sable*), et

- **pente**, **épaisseur**, et **perméabilité** sont des descripteurs à un attribut linéaire, avec comme extension ordonnée (*très faible, faible, moyenne, forte, très forte*),

et comme connecteurs : la conjonction et la disjonction.

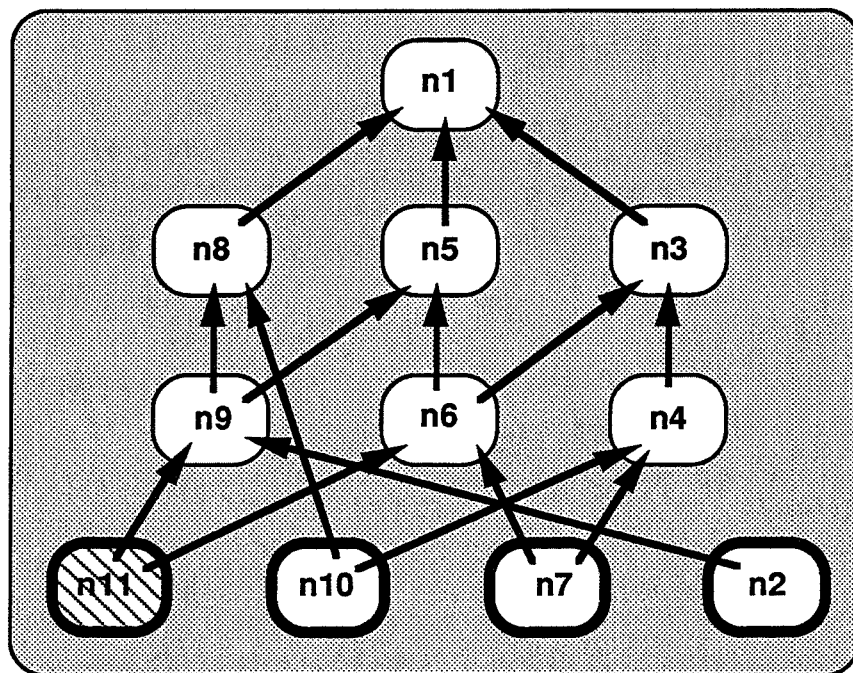


fig. 4.4 : un exemple de RDP

Ce qui nous permet de fixer les quatre exemples suivants :

- 1) **filière** = *épanchage en tranchées* et **pente** = *faible* et **épaisseur** = *moyenne* et **perméabilité** = *moyenne*

- 2) **filière = épandage en tranchées** et **pen**te = faible et **épaisseur = forte** et **perméabilité = moyenne**
- 3) **filière = filtre à sable** et **pen**te = moyenne et **épaisseur = très faible** et **perméabilité = moyenne**
- 4) **filière = filtre à sable** et **pen**te = moyenne et **épaisseur = moyenne** et **perméabilité = très forte**

Avec ce langage de description et ces quatre exemples, nous obtenons le RDP représenté en figure 4.4.

4.3.1.2 Contenu du RDP

Ce RDP est constitué de 11 nœuds et 15 arcs. Les nœuds ont pour contenu :

- n1 : **filière = quelconque**
faible ≤ **pen**te ≤ **moyenne**
très faible ≤ **épaisseur** ≤ **forte**
moyenne ≤ **perméabilité** ≤ **très forte**
- n8 : **perméabilité = moyenne**
- n5 : **moyenne** ≤ **épaisseur** ≤ **forte**
- n3 : **très faible** ≤ **épaisseur** ≤ **moyenne**
- n9 : **filière = épandage en tranchées**
pente = **faible**
- n6 : **épaisseur = moyenne**
- n4 : **filière = filtre à sable**
pente = **moyenne**
- n11 : -
- n10 : **épaisseur = très faible**
- n7 : **perméabilité = très forte**
- n2 : **épaisseur = forte**

Les nœuds n11, n10, n7 et n2 ont un contour plus gras pour signifier qu'ils représentent chacun un exemple. Mais il ne contiennent pas explicitement toute la description de l'exemple qu'il représente, car certaines formules de cette description se trouvent aussi dans d'autres nœuds. Par souci d'économie de représentation, nous ne dupliquons pas ces formules, car il est possible de les retrouver.

Par exemple, le nœud n10 représente l'exemple 3 (c'est le seul exemple dans lequel **épaisseur = très faible**). Pour retrouver le reste de sa description, il faut aller chercher les formules les plus spécifiques qui se trouvent dans les nœuds successeurs de n10, à savoir n8, n4, n3 et n1. Dans n8 nous trouvons une valeur pour la perméabilité; dans n4, nous trouvons une valeur pour la filière et une pour la pente; dans n3 nous trouvons une valeur pour l'épaisseur qui est plus générale que celle présente dans n10, nous ne la retenons donc pas; et ainsi de suite jusqu'à n1.

A l'extrême, un nœud peut être vide. Mais il s'agit alors obligatoirement d'une feuille, sinon il n'aurait pas été créé. C'est le cas du nœud n11 qui ne contient rien (nous le distinguons par un fond hachuré). Il est conservé uniquement pour mémoriser l'existence d'un exemple à cet emplacement, ce qui est indispensable, comme nous le verrons ci-dessous. Le fait qu'il soit vide signifie seulement que chaque formule de la description de l'exemple qu'il désigne (l'exemple 1) est plus générale (au sens large) qu'une formule présente dans la description d'un autre exemple, et a donc été déportée dans un nœud plus haut.

De façon plus générale, chaque nœud représente un ensemble d'exemples. Et tous les regroupements d'exemples sont représentés par la généralisation la plus spécifique de leurs formules. Si deux exemples n'ont rien en commun, alors leur regroupement est représenté par la racine du RDP : la racine du RDP représente la tautologie.

Le nœud n9 représente les exemples représentés par n11 et n2, c'est à dire les exemples 1 et 2. Le nœud n5 représente les exemples 1, 2 et 4. Le nœud n1 représente tous les exemples, c'est à dire toute la connaissance représentée dans le RDP.

4.3.1.3 Lecture du RDP

Les connaissances les plus simples à lire dans un RDP sont les classes d'Equivalence Perçue. Ainsi, dans notre RDP de la figure 4.4, nous pouvons lire que "la filière est l'épandage en tranchées" est équivalent à "la pente est faible" (contenu du nœud n9). En effet, si nous consultons la base d'exemples que nous avons utilisée, nous remarquons facilement que dans chaque exemple où il y a **filière = épandage en tranchées** (les exemples 1 et 2), il y a aussi **pente = faible**; et inversement, dans chaque exemple où il y a **pente = faible**; il y a aussi **filière = épandage en tranchées**. Au vu de cette base d'exemples, il semble donc qu'il soit équivalent d'affirmer **filière = épandage en tranchées** et **pente = faible**.

Une autre équivalence est présente dans le nœud n4 entre "la filière est le filtre à sable" et "la pente est moyenne". Nous retrouvons cette équivalence dans les exemples 3 et 4.

Les formules présentes dans le nœud n1 sont les plus générales. En effet, "la pente est comprise entre faible et moyenne" est une généralisation de tous les exemples de notre base. Il en va de même pour les trois autres formules.

En ce qui concerne les Dépendances Perçues, elles se trouvent dans les arcs du RDP. Par exemple, la liaison du nœud n9 vers le nœud n8 nous indique, qu'au vu de notre base d'exemples il semble que "la perméabilité est moyenne" est impliquée par "la pente est faible". En effet, si nous consultons les 4 exemples de la base, lorsque "la pente est faible" (dans les exemples 1 et 2), nous trouvons aussi "la perméabilité est moyenne"; mais cette fois la réciproque n'est plus vraie : dans l'exemple 3, "la perméabilité est moyenne" alors que "la pente est moyenne".

De même, "l'épaisseur est forte" (nœud n2) semble impliquer "la pente est faible" (nœud n9), et, par transitivité, semble impliquer aussi "la perméabilité est moyenne" (nœud n8).

Comme nous le prévoyions ci-dessus, seules les formules atomiques sont représentées à l'intérieur des nœuds du graphe. Mais les formules complexes peuvent être construites. Par exemple, en observant les exemples de la base, nous pouvons affirmer que "**épaisseur = moyenne** et **perméabilité = moyenne**" semblent impliquer que "**filière = épandage en tranchées**". Nous pouvons retrouver ce résultat dans le RDP. En effet, la conjonction de deux formules atomiques se trouve dans les nœuds qui sont les prédécesseurs les plus généraux, communs aux deux nœuds contenant les formules atomiques, s'il en existe, sinon la conjonction n'existe pas sur la base d'apprentissage. Il suffit ensuite de regarder le contenu des successeurs de ces nœuds pour savoir ce que semble impliquer la conjonction étudiée.

Si la conjonction recherchée appartient à une classe d'équivalence contenant au moins une formule atomique ou couvrant un seul exemple, alors la classe d'équivalence est représentée par un nœud dans le RDP, et cette conjonction se trouve dans cet unique nœud. Mais si cette classe d'équivalence ne contient aucune formule atomique ou ne couvre pas un unique exemple, alors elle n'est pas représentée explicitement dans le RDP, et la conjonction est localisée, dans le RDP, dans plusieurs nœuds qui représentent les classes d'équivalences contenant les formules atomiques les plus générales qui sont en Dépendance Perçue avec cette conjonction.

Ainsi, la conjonction citée ci-dessus se trouve dans les premiers prédécesseurs communs à n8 et n6, c'est à dire n11. Ensuite, nous regardons les successeurs de n11, entre autres n9 qui contient la formule que nous recherchions.

En ce qui concerne les formules disjonctives, le procédé est similaire. La disjonction de deux formules atomiques se trouve dans les nœuds qui sont les successeurs les plus spécifiques, communs aux deux nœuds contenant les formules atomiques. Par exemple, la disjonction "épaisseur = très faible ou perméabilité = très forte" se trouve dans les premiers successeurs communs de n10 et n7, c'est à dire n4. Nous ne retenons pas n3 et n1 qui sont plus "généraux" que n4. Le nœud n4 contient notamment "pente = moyenne". Nous pouvons donc affirmer qu'il semble qu'il y ait équivalence logique entre "épaisseur = très faible ou perméabilité = très forte" et "pente = moyenne". Nous retrouvons aisément ce résultat dans la base d'exemples.

Nous savons donc représenter une connaissance empirique au moyen d'un RDP, lorsque le langage de description est la logique des prédicats du 1^{er} ordre. Voyons maintenant comment un RDP peut être construit.

4.3.2 Intégration d'un nouvel exemple

L'algorithme présenté dans [Sainte Marie 89], et repris dans [Royer 90], permet une construction incrémentale d'un RDP (l'incrémentalité de construction est discutée au § 2.5.3.3). Avant de voir cet algorithme, précisons qu'au départ, lorsqu'aucun exemple n'est encore fourni, le RDP est constitué d'un seul nœud contenant la tautologie. En effet, sans aucune expérience, tout paraît vrai au système. Ensuite, chaque exemple présenté à l'algorithme permet au système de construire un nouveau RDP. Nous supposons donc posséder un RDP et vouloir lui intégrer un nouvel exemple que nous nommerons e.

4.3.2.1 L'algorithme d'entretien incrémental d'un RDP

Notations : f_n est l'ensemble des formules atomiques contenues dans le nœud n.

G est une fonction de généralisation :

Cette fonction de généralisation a deux arguments, qui sont deux ensembles de formules atomiques contenues dans deux nœuds du RDP, et elle retourne l'ensemble des formules atomiques les plus spécifiques généralisant les formules atomiques communes aux deux

arguments. Elle nécessite donc, pour chaque nature d'attributs, la définition d'une fonction de généralisation (\mathcal{g}) de deux formules atomiques, qui retourne la formule atomique la plus spécifique qui généralise les deux formules atomiques passées en arguments.

Par exemple, pour les attributs linéaires, nous pourrions choisir la fonction de fermeture d'un intervalle (closing interval, [Michalski 83]) :

$\mathcal{g}(d=a, d=b) = d \in [a, b]$, où a et b sont deux constantes. Ou, de façon plus générale :
 $\mathcal{g}(d \in [a, b], d \in [c, d]) = d \in [\min(a, c), \max(b, d)]$ où a, b, c et d sont des constantes.

Pour les attributs structurés, nous pourrions choisir la fonction d'ascension généralisante (climbing generalization tree, [Michalski 83]) :

$\mathcal{g}(d=a, d=b) = d=s$, où a, b et s sont des constantes, et s est le plus petit ascendant commun à a et b .

Enfin, pour les attributs nominaux, nous pourrions choisir la fonction de variabilisation :

$\mathcal{g}(d=a, d=b) = d=x$, où a et b sont des constantes distinctes, et x une variable. C'est celle que nous avons retenue pour les exemples de ce chapitre. Nous aurions pu aussi choisir la fonction d'ajout d'une alternative (adding alternative, [Michalski 83]) :

$\mathcal{g}(d=a, d=b) = d \in \{a, b\}$, où a et b sont des constantes distinctes, ou, de façon plus générale :
 $\mathcal{g}(d \in A, d \in B) = d \in (A \cup B)$, où A et B sont des ensembles de constantes.

L'algorithme (cf. figure 4.5) consiste à créer un nouveau nœud N contenant l'exemple à intégrer; puis à parcourir tout le RDP par ordre topologique (en largeur d'abord : n'examiner un nœud du RDP que si tous ses fils l'ont été au préalable). Chaque nœud visité, n , doit être comparé avec le nouveau nœud N : si n peut être généralisé pour contenir une partie de N , le faire en prenant soin :

- de retirer de N la partie qui vient d'être intégrée à n , et de relier N à n ,
- de modifier l'entourage de n de façon à conserver un RDP. Ces modifications de l'entourage de n concernent les liens autour de n , et le contenu de n .

Pour la procédure **mettre-à-jour- \mathcal{E}_{RDP}** (présentée en figure 4.6), nous avons besoin d'une fonction chemin définie par :

pour $(x, y) \in V_{RDP}^2$, $\text{chemin}(x, y) = \text{vrai}$ ssi il existe un chemin allant de x à y par des arcs de \mathcal{E}_{RDP} , sinon $\text{chemin}(x, y) = \text{faux}$.

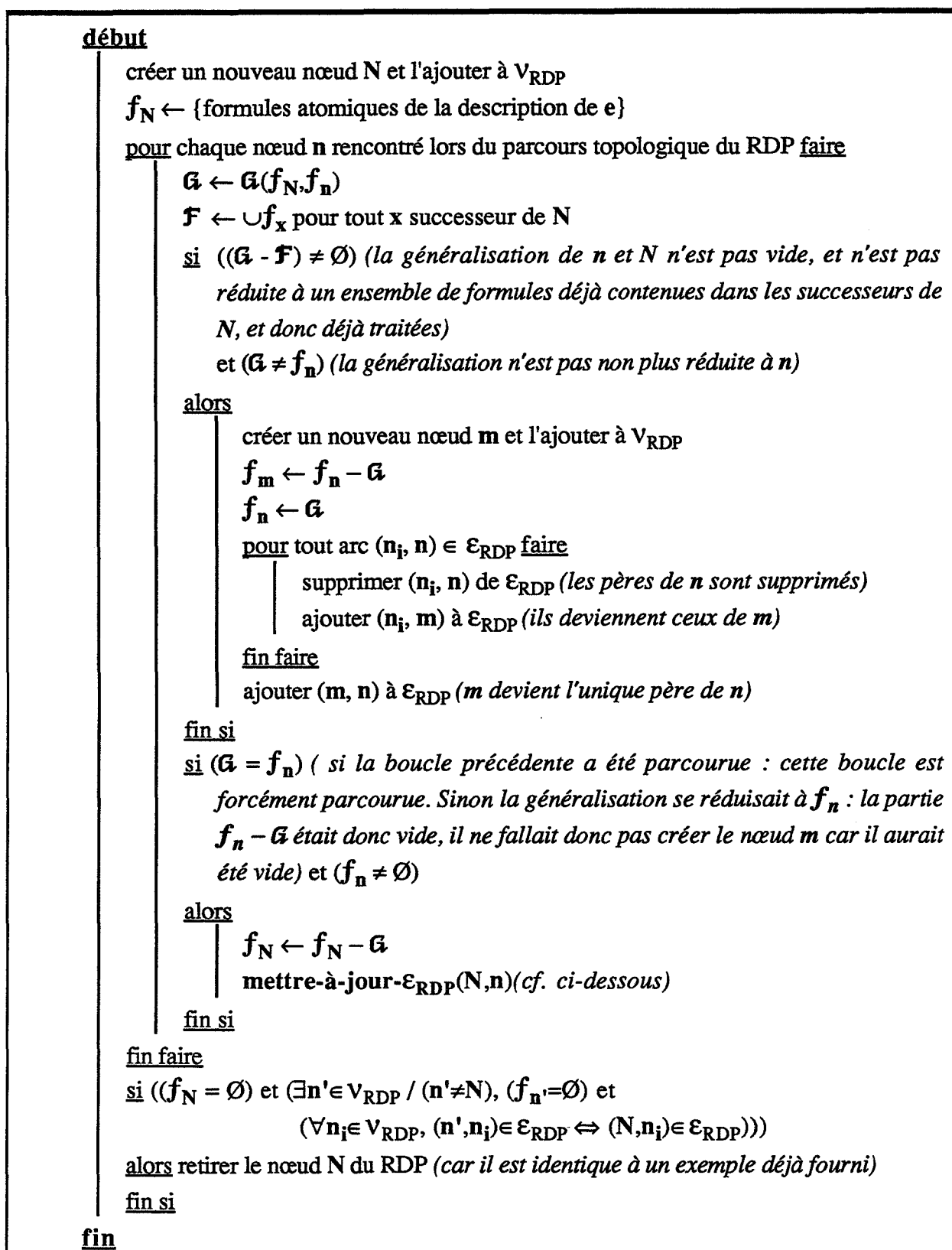


fig. 4.5 : l'algorithme d'entretien incrémental d'un RDP
(d'après [Sainte Marie 89])

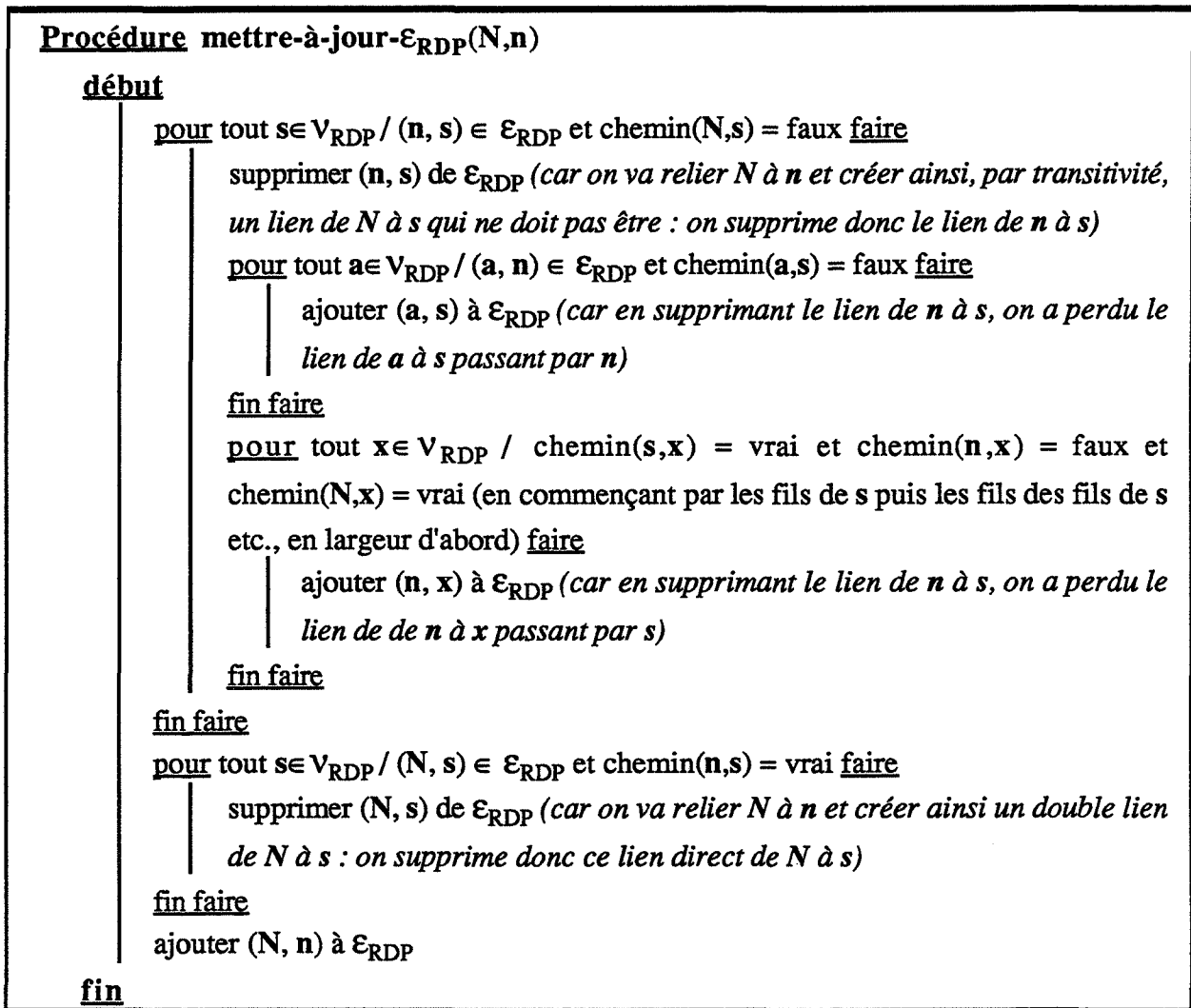


fig. 4.6 : mise à jour des liens lors de l'entretien incrémental d'un RDP
(d'après [Sainte Marie 89])

4.3.2.2 Exemple d'intégration d'un exemple à un RDP

En appliquant cet algorithme au RDP décrit en figure 4.4, et à l'exemple :

$e = ($ filière = *épannage en tranchées* et
 pente = *faible* et
 épaisseur = *moyenne* et
 perméabilité = *faible*),

nous obtenons le RDP de la figure 4.7. Le lecteur intéressé trouvera, en annexe de ce mémoire, le détail des opérations qui permettent cette construction.

En bref, les modifications sont :

- l'ajout de 3 nouveaux nœuds (n12, n13 et n14) dont les contenus sont :

- n13 : *moyenne* ≤ **perméabilité** ≤ *très forte*

- n14 : **perméabilité** = *moyenne*

- n12 : **perméabilité** = *faible*

- la modification des nœuds n1 et n8, qui contiennent désormais :

- n1 : **filière** = quelconque

faible ≤ **pente** ≤ *moyenne*

très faible ≤ **épaisseur** ≤ *moyenne*

faible ≤ **perméabilité** ≤ *très forte*

- n8 : *faible* ≤ **perméabilité** ≤ *moyenne*

- et la modification de certains arcs.

Le nœud n1 a été généralisé pour pouvoir contenir le nouvel exemple : la formule "**perméabilité** = *faible*" n'était en effet pas présente dans les quatre premiers exemples.

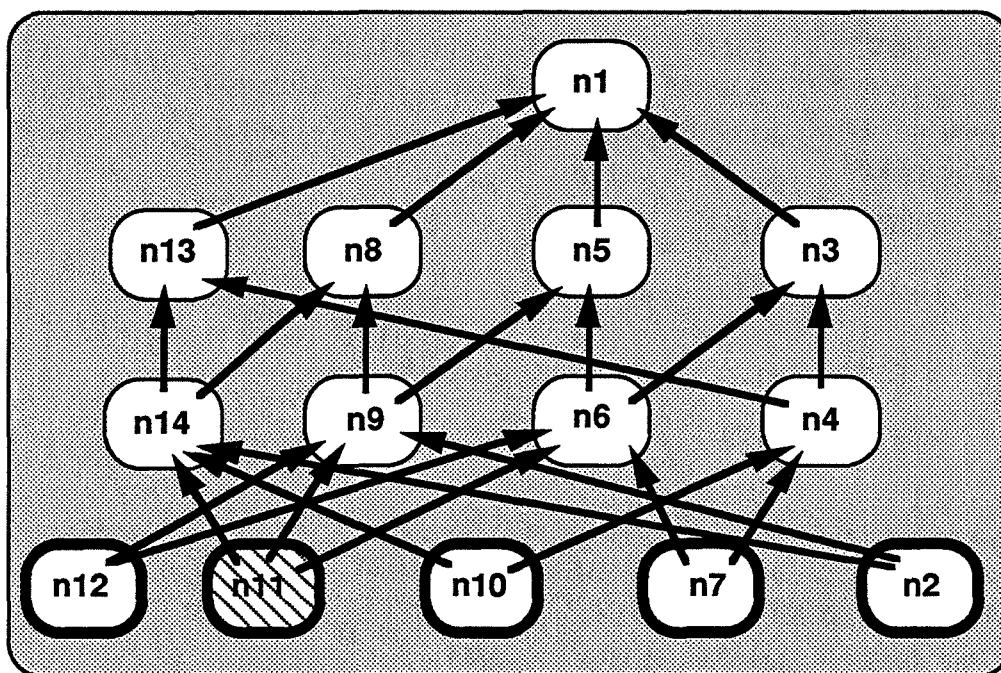


fig. 4.7 : un nouvel exemple est intégré au RDP

Le nœud n13 a été créé pour contenir la formule : "*moyenne* ≤ **perméabilité** ≤ *très forte*" qui était présente dans le nœud n1, et généralise donc les quatre premiers exemples.

Du fait de la généralisation du nœud n8, la formule : "**perméabilité = moyenne**" avait disparu : le nœud n14 a donc été créé pour la contenir, et pouvoir ainsi couvrir les trois premiers exemples.

Le nœud n12 a été créé au début de l'algorithme pour couvrir le nouvel exemple. A la fin il ne contient plus que la formule : "**perméabilité = faible**" qui est la seule que cet exemple contienne et qui ne soit pas commune avec un autre exemple.

4.3.2.3 Formules atomiques et formules complexes

Nous avons vu au § 4.3.1.3 que les formules conjonctives et les formules disjonctives n'avaient pas besoin d'être représentées explicitement dans le RDP. On peut montrer que cette non-représentation explicite ne porte pas préjudice au RDP construit incrémentalement par l'algorithme proposé. Sainte Marie montre que cela découle des propriétés de l'opérateur disjonctif et de l'opérateur conjonctif par rapport à la relation de dérivation (notre relation fondatrice de la Dépendance Perçue) ([Sainte Marie 90b], [Sainte Marie 91]).

En fait le logiciel EDGE permet aussi la détection et l'entretien incrémental de régularités plus complexes que celles examinées jusque là. Jusqu'à présent nous n'avons examiné que les régularités du type corrélation entre descripteurs avec restriction des domaines des attributs. Le logiciel EDGE permet de repérer aussi des relations entre les attributs eux-mêmes. Par exemple, il est possible de repérer des relations d'égalité ou d'inégalité entre attributs (d'autres relations peuvent aussi être introduites) qui ont été déclarées du même type (c'est à dire comparables).

Ainsi, à partir des exemples suivants :

- 1) **épaisseur du sol = 2m et profondeur de la nappe = 2m**
- 2) **épaisseur du sol = 1m et profondeur de la nappe = 1m**

le logiciel EDGE repérera que :

$$\mathbf{\text{profondeur de la nappe} = \text{épaisseur du sol} ,}$$

puisque les deux attributs sont d'un même type, que l'on pourrait nommer "distance". Dans le RDP, ceci est représenté par l'ajout de la formule :

$$\exists X \in [1,2] / \text{profondeur}(X) \wedge \text{épaisseur}(X)$$

dans le nœud adéquat.

Mais le traitement de ces régularités complique passablement l'algorithme, et, notamment, possède le gros inconvénient d'exiger des opérations sur le RDP qui ne sont alors plus locales

autour du nœud N créé dans l'algorithme présenté en figure 4.5. Mais l'incrémentalité de construction est quand même respectée, car tout le RDP n'est pas à reconstruire.

4.3.2.4 La fonction de généralisation

Puisque la fonction de généralisation \mathcal{G} retourne les généralisés les plus spécifiques, les concepts (le contenu des nœuds) ne sont représentés que par leur borne inférieure, leur borne supérieure étant implicitement représentée par les bornes inférieures des concepts immédiatement plus généraux (les nœuds directement en relation de Dépendance Perçue).

Ceci nous permet une économie de représentation, sans pour autant compliquer la procédure de consultation du RDP, comme nous le verrons dans un paragraphe ci-dessous.

Auparavant, nous allons examiner les différents critères qui permettent de caractériser les systèmes d'apprentissage, tels que nous les avons définis au § 2.5.3 : l'explicitation de la connaissance sur le domaine, la prise en compte des données erronées et l'incrémentalité de l'algorithme.

4.3.2.5 La connaissance sur le domaine

La connaissance sur le domaine est une connaissance qui a pour conséquence de restreindre la production des descriptions de concepts. Elle se trouve dans les règles de généralisation choisies, dans le langage de description adopté, et dans différents critères.

Une première restriction à la production des descriptions de concepts dans un RDP, est la condition de l'existence d'au moins un exemple qui fonde la description. Par exemple, dans le cas d'un système d'aide au dépannage d'automobiles, il ne semble y avoir aucun intérêt à s'encombrer de la description caractérisant les automobiles à 5 roues, tant qu'un tel véhicule n'existe pas. Cette restriction ne paraît donc pas abusive.

En ce qui concerne la fonction de généralisation, nous avons vu qu'elle ne retourne que des descriptions de concepts les plus spécifiques possibles. Cette restriction non plus ne pêche pas par excès car, implicitement, le RDP représente aussi la borne supérieure des concepts, et il existe un ordre partiel sur les descriptions de concepts. Les concepts peuvent donc être complètement décrits par leur borne inférieure et leur borne supérieure.

Pour ce qui est des définitions des fonctions de généralisation propres à chaque nature d'attribut (cf. § 4.3.2.1), il n'y a que pour les attributs nominaux que nous avons le choix. En fait nous

opterons pour la fonction de variabilisation, car la fonction d'ajout d'une alternative ne ferait que représenter explicitement une information déjà présente implicitement dans le RDP. En effet, $d \in \{a, b\} \Leftrightarrow (d=a \text{ ou } d=b)$, information présente implicitement dans les premiers nœuds communs et successeurs du nœud contenant $d=a$ et du nœud contenant $d=b$.

De plus, le logiciel EDGE permet la définition d'une fonction de généralisation propre à une nature d'attribut donnée. Ainsi, si on le désire, on peut définir des attributs ayant des structures complexes. Nous ne sommes donc pas limités par les trois fonctions de généralisation attachées aux trois natures prédéfinies d'attributs (nominal, linéaire et structuré). En fait, de tels attributs seraient des attributs nominaux avec une fonction de généralisation distincte de la fonction de variabilisation.

Enfin, le logiciel EDGE permet aussi de repérer des relations entre attributs de même type (égalité, inégalité, ou autre à définir). La fonction de généralisation nécessite donc une étude approfondie, relative au domaine d'application. Nous ferons cette étude pour notre domaine (l'assainissement autonome des eaux usées) au chapitre 5.

Le langage de description que nous choisirons sera au plus la logique des prédicats du 1^{er} ordre. Nous verrons au chapitre 5 que ce langage sera largement suffisant pour notre application.

En reprenant les définitions adoptées au § 2.5.3.1, nous remarquons que notre algorithme respecte le critère de consistance de Mitchell et Utgoff (i.e. complétude et discrimination). Le critère de consistance proposé par Nicolas restreint moins la production des descriptions de concepts, mais n'est pas intéressant pour les RDP, où l'on ne considère pas explicitement les négations de concepts.

Enfin, aucun critère de préférence n'est considéré dans les RDP, puisqu'on se limite à la représentation explicite des formules atomiques, en sachant que toutes les formules complexes sont représentées implicitement. Ce n'est qu'en phase de consultation du RDP, que l'on considère des formules complexes.

4.3.2.6 La prise en compte des données erronées

La deuxième caractéristique des systèmes d'apprentissage automatique est leur aptitude à prendre en compte la présence éventuelle de données erronées. L'algorithme d'entretien incrémental d'un RDP que nous avons choisi ne permet pas de traiter directement ces données, comme le fait, par exemple, CHARADE. Mais la représentation de la connaissance sous forme

de RDP permet de détecter, lors de leur traitement, des données inconsistantes avec la connaissance précédemment acquise.

En ce sens, nous pouvons affirmer que notre système, à base de RDP, prend en compte la présence éventuelle de données erronées. Nous verrons dans un paragraphe suivant comment se fait cette prise en compte.

4.3.2.7 Incrémentalités

La dernière caractéristique importante est l'incrémentalité de l'algorithme.

En l'état actuel de l'algorithme, l'incrémentalité de construction est respectée puisque l'intégration d'un nouvel exemple ne nécessite pas la reconstruction complète du RDP.

De même, l'incrémentalité d'usage est aussi respectée, puisque l'introduction d'un nouvel exemple permet de construire immédiatement un nouveau RDP, et comme nous le verrons dans le paragraphe suivant, un RDP est directement consultable, sans nécessiter de modifications.

En ce qui concerne l'incrémentalité de représentation, le problème est plus délicat. Nous ne considérerons que le cas où l'on désire ajouter un nouveau descripteur, ou en modifier un déjà présent. L'autre cas, la suppression d'un descripteur, ne semble pas avoir d'intérêt si ce descripteur a existé dans la description d'un exemple. Dans le cas où le descripteur n'est jamais apparu dans un exemple, sa suppression est immédiate puisqu'il n'apparaît alors pas dans le RDP.

L'ajout d'un descripteur pose peu de problèmes. En effet, lorsqu'un descripteur appartient au langage de description, mais n'apparaît dans aucun exemple, il n'apparaît alors pas dans le RDP. Un nouveau descripteur est donc similaire à un descripteur existant mais jamais utilisé. La seule difficulté réside dans l'ergonomie d'un tel ajout : il faut que le système soit capable de faire définir facilement ce nouveau descripteur à un expert quelconque.

La définition d'un descripteur consiste à nommer le prédicat qui représentera ce descripteur, et, pour chaque variable du prédicat (i.e. chaque attribut du descripteur), définir :

- sa nature (nominale, linéaire ou structurée),
- son domaine de définition (son extension ou sa fonction d'appartenance),
- éventuellement sa fonction de généralisation si elle doit être différente de celles prédéfinies,
- la possibilité ou l'impossibilité de généraliser cette variable,
- éventuellement le type (i.e. l'unité) pour permettre la détection de relations entre variables,

- pour une variable généralisable et linéaire : l'ordre,
- pour une variable généralisable et structurée : la structure.

Quant à la modification d'un descripteur, elle ne peut porter que sur ses attributs existants. Sinon ce serait l'ajout ou la suppression d'un attribut, et cela reviendrait à la suppression puis l'ajout d'un descripteur.

Nous devons donc examiner la modification d'un attribut d'un descripteur, et cela pour chaque nature d'attribut prédéfinie : nominale, linéaire et structurée. Si des attributs plus compliqués sont envisagés, il faudra alors examiner leur modification cas par cas. En fait de tels descripteurs seraient des descripteurs nominaux avec une fonction de généralisation spécifique, différente de la variabilisation. La spécificité de leur modification réside donc dans la modification de leur fonction de généralisation.

Les modifications qui respectent le critère d'incrémentalité de représentation sont :

- toute modification des attributs d'un descripteur jamais apparu dans les exemples,
- pour un attribut nominal défini en extension : ajout d'une nouvelle valeur, ou bien suppression d'une valeur jamais apparue dans les exemples,
- pour un attribut nominal défini en intention : modification de la fonction d'appartenance : le seul cas envisageable est l'extension du domaine sauf si cet attribut appartient à un descripteur qui n'est jamais apparu dans les exemples.
- pour un attribut linéaire, la modification du domaine de définition est identique à celle envisagée pour un attribut nominal, mais elle implique une modification de l'ordre :
- la modification de l'ordre d'un attribut linéaire ne demande aucune modification du RDP, s'il s'agit d'ajouter une ou plusieurs nouvelles valeurs ne modifiant pas l'ordre des valeurs précédemment présentes. Dans le cas contraire, une reconstruction complète du RDP est nécessaire.
- de même pour un attribut structuré : la modification du domaine de définition entraîne une modification de la structure :
- une modification de la structure d'un attribut structuré est identique à la modification de l'ordre d'un attribut linéaire.

En revanche, les modifications suivantes nécessitent la reconstruction du RDP, sauf si elles concernent l'attribut d'un descripteur absent de tous les exemples :

- modification de la nature d'un attribut,

- restriction du domaine de définition, car il faut éventuellement supprimer certains exemples, sauf pour les cas cités ci-dessus,
- modification de la fonction de généralisation,
- modification de la possibilité de généralisation, et
- l'ordre d'un attribut linéaire ou la structure d'un attribut structuré, hors les cas cités ci-dessus.

Dans notre système, nous exigerons que soit respectée l'incrémentalité de représentation, et nous ne permettrons donc que certaines modifications du langage de description.

Nous avons donc vu comment un RDP pouvait représenter la connaissance, et comment on pouvait faire évoluer cette connaissance. Voyons maintenant, quelles opérations sont nécessaires pour initialiser un système à base de RDP.

4.3.3 Initialisation du système

Le RDP en lui même ne nécessite aucune initialisation, si ce n'est que le premier exemple constitue, en un seul nœud, le premier RDP. Ensuite l'algorithme proposé est applicable incrémentalement aux exemples suivants. Ce qui n'empêche pas, si le domaine d'application le permet, et si cela semble souhaitable, de fabriquer, "à la main", un premier RDP contenant une connaissance dont on disposerait. Cela peut accélérer la constitution d'une base de connaissance prédictive du domaine.

Une autre initialisation est le choix de la relation fondatrice : nous avons déjà fait le choix de cette relation : la dérivation. Sainte Marie souligne l'importance de la sémantique attachée à cette relation fondatrice pour la clarté des explications (implication, prédiction, ... : [Sainte Marie 90a]).

La dernière initialisation est celle du langage de description. Son importance est relativisée par le fait que l'on sait faire évoluer ce langage (cf. paragraphe précédent). Il s'agit en fait de définir ce que De Morgan appelle l'Univers du Discours, c'est à dire le domaine de toutes les variables à considérer ([Habrias 88]). Pour nous, il s'agit de définir les descripteurs, leurs attributs, les caractéristiques des attributs, les opérateurs booléens (conjonction, disjonction, ...) et, éventuellement, les opérateurs reliant un attribut à une valeur (égalité, inégalité, appartenance à un ensemble ou un intervalle, ...).

De plus, il est possible, dans EDGE, de définir plusieurs sous-langages : le RDP mettra alors en relation ces différents langages. Pour ce qui concerne un système d'aide à la décision, deux sous-langages viennent immédiatement à l'esprit : le langage d'observation et le langage de décision; le but du système est de mettre en correspondance ces deux sous-langages de façon à pouvoir caractériser une décision à prendre à partir des observations.

Nous savons désormais faire naître, puis faire vivre un RDP. Il ne reste plus qu'à examiner comment utiliser la connaissance contenue dans un RDP.

4.3.4 Consultation du graphe

Le but d'un système à base de RDP est de faciliter l'acquisition d'une connaissance empirique, afin de pouvoir l'utiliser sur des cas différents que ceux décrits dans les exemples. Cette utilisation consiste donc en la consultation du graphe construit.

Plusieurs types d'applications peuvent être imaginés pour un tel système : dans notre cas, c'est la constitution d'une base de connaissance évolutive pour alimenter une Station de Travail, et plus précisément la base de connaissance d'un système expert, mais c'est aussi la construction d'un système expert télématique : dans les deux cas il s'agit donc de construire un système expert. On peut aussi envisager l'application des RDP dans des applications d'aide à la découverte, de reconnaissance des formes, etc.

Pour ce qui nous concerne, la construction d'un système expert, nous cherchons, en fait, à caractériser une décision, c'est à dire à mettre en relation des faits observables avec des faits caractérisant une décision. Pour cela, en phase d'initialisation, nous nous sommes définis deux sous-langages : le langage de description des décisions, et le langage de description des observations. Comme nous le verrons pour notre application, dans le chapitre 5, les deux sous-langages ne sont pas forcément disjoints.

Deux cas typiques peuvent être envisagés pour la consultation de la base de connaissance exprimée sous forme de RDP : soit le consultant fournit ses observations et demande une aide à la décision, soit il demande une aide à la décision et le système lui propose d'effectuer certaines observations. A partir de ces deux cas typiques, des panachages peuvent être effectués : par exemple le consultant possède quelques observations mais elles ne suffisent pas au système pour proposer une décision, le système guide alors le consultant pour qu'il complète ses observations.

Nous allons décrire uniquement les principes de base des mécanismes de consultation du RDP. Pour cela, nous considérons un graphe, que nous appelons Réseau complexe de Dépendances Perçues (RcDP) où sont représentées explicitement, en plus des formules atomiques représentées dans un RDP, toutes les formules conjonctives et disjonctives contenues implicitement dans un RDP. Les algorithmes permettant de prendre en compte efficacement ces formules dans un RDP sont décrites dans [Sainte Marie 91].

Pour décrire ces mécanismes, nous allons définir quelques fonctions opérant sur un RcDP :

- un nœud est éligible lorsqu'une formule du nœud est avérée, tandis que toutes les autres formules ne sont pas niées (elles sont avérées ou indéterminées).
- inversement, élire un nœud n revient à avérer toutes ses formules, et à propager la bonne nouvelle chez tous ses fils : il faut tous les élire.

- un nœud est rejetable lorsqu'une formule du nœud est niée, tandis que toutes les autres formules ne sont pas avérées (elles sont niées ou indéterminées).
- inversement, rejeter un nœud n revient à nier toutes ses formules, et à propager la mauvaise nouvelle chez tous ses pères : il faut tous les rejeter.

- un nœud est indéterminé lorsque toutes ses formules sont indéterminées.

- un nœud est incohérent lorsque, étant éligible on tente de le rejeter, ou, inversement, étant rejetable on tente de l'élire.

4.3.4.1 Chaînage avant

Nous allons prendre le premier cas typique : le consulteur fournit les faits observés, et demande une aide à la décision. Un mécanisme de consultation du RcDP est le suivant :

- 1) Avec le premier fait observé, le système parcourt le RcDP, à partir de la racine et en largeur d'abord, à la recherche du nœud le plus spécifique qui recouvre ce fait :
 - s'il existe, ce nœud est unique, car les formules ne sont pas dupliquées dans le RcDP. Soit n_1 ce nœud. Tous les faits possibles se trouvent désormais dans n_1 et ses successeurs. Soit $E = \{n_1\}$,
 - s'il n'existe pas, soit $E = \emptyset$.

2) Avec le deuxième fait, le système effectue la même opération qu'en 1) : si le nœud n_2 existe, alors $E \leftarrow E \cup \{n_2\}$. Ensuite, il s'agit de trouver la conjonction des deux faits : pour cela le système cherche le prédécesseur le plus général, commun aux deux nœuds n_1 et n_2 : s'il existe, soit c_2 ce nœud (ce nœud est unique car toutes les formules, même complexes, sont représentées explicitement dans le RcDP), et alors $E \leftarrow E \cup \{c_2\}$.

Tous les faits possibles se trouvent désormais dans les nœuds de E et leurs successeurs.

...

i) Avec le $i^{\text{ème}}$ fait, le système effectue la même opération qu'en 1) : si le nœud n_i existe, alors $E \leftarrow E \cup \{n_i\}$. Ensuite, il s'agit de trouver la conjonction des faits : pour chaque nœud e_k de E , tel que e_k ne soit pas plus général qu'un autre nœud de E , le système cherche le prédécesseur le plus général, commun aux deux nœuds e_k et n_i :

- s'il existe, soit $c_{i,k}$ ce nœud et $E \leftarrow E \cup \{c_{i,k}\}$

- sinon recommencer la même opération (recherche de conjonction) avec l'ensemble des successeurs immédiats de e_k dans E . Et ainsi de suite, récursivement.

Tous les faits possibles se trouvent désormais dans les nœuds de E et leurs successeurs.

...

Et ainsi de suite jusqu'à ce que tous les faits observés soient traités. Les réponses se trouvent alors dans les nœuds de E et leurs successeurs.

Au départ le système ne retient que la racine du RcDP : en effet, si le consultant ne fournit aucune observation, le système ne peut lui répondre que par les formules les plus générales du RcDP.

Ce mécanisme correspond à un fonctionnement en chaînage avant d'un moteur d'inférences d'un système à base de règles. Tel qu'il est décrit ici, il ne permet pas de détecter, ni de situer dans le RcDP, les incohérences dans les observations fournies par le consultant. Pour détecter toutes les incohérences, et en situer la plupart, il faut propager l'élection et le rejet des nœuds : à chaque nœud n visité lors du parcours du RcDP, il faut élire n s'il est éligible ou le rejeter s'il est rejetable.

4.3.4.2 Chaînage arrière

Pour le deuxième cas typique (le consultant demande une aide à la décision et le système lui propose d'effectuer certaines observations) nous allons adopter un mécanisme en chaînage arrière. Le mécanisme consiste donc en un parcours du RcDP à la recherche d'un nœud contenant une réponse, et en la recherche de l'élection de ce nœud. Pour élire un nœud on

cherche à élire l'un de ses prédécesseurs : c'est ce que fait la procédure réursive tester, décrite ci-dessous.

- tester un nœud n a pour but de chercher à l'élire ou le rejeter :

examiner les formules de n (éventuellement interroger le consultant) jusqu'à ce que l'une d'entre elles soit avérée ou niée ou qu'il n'y en ait plus à examiner

si le nœud n est éligible alors on l'élit

si le nœud n est rejetable alors on le rejette

tant que (le nœud n est indéterminé) et

(il existe un père indéterminé de n qui n'a pas été testé) faire

tester ce père de n

(appel récursif qui permet de tester tous les prédécesseurs de n)

fin tant que

Le mécanisme est :

- 1) Le système parcourt le RcDP par ordre topologique, à partir de la racine, à la recherche du premier nœud qui contient une réponse, et le teste.
- 2) Si le nœud est éligible : la réponse est fournie au consultant.
- 3) Si le nœud est rejetable ou indéterminé, le système continue le parcours du RcDP jusqu'au nœud suivant qui contient une réponse, le teste et va en 2).

Le lecteur notera que ce mécanisme peut détecter les incohérences des faits fournis par le consultant, et en situer certaines dans le RcDP. Cette détection se fait lors de la propagation de l'élection ou du rejet d'un nœud.

4.3.4.3 Chaînage bi-directionnel

Le dernier cas à examiner est celui où le consultant possède quelques observations mais elles ne suffisent pas au système pour proposer une décision : pour le savoir, le système procède par chaînage avant.

Le système procède ensuite par chaînage arrière (comme au § 4.3.4.2, mais avec une base de faits initiale non vide) et guide le consulteur pour qu'il complète ses observations.

4.4 MODES D'INTERACTIONS

Nous avons maintenant une connaissance suffisamment détaillée des RDP, de leur construction et de leur consultation, pour préciser les dialogues avec le système télématique, tels que décrits au chapitre précédent. Nous examinons donc, l'un après l'autre, chaque dialogue du système avec ses deux catégories d'utilisateur (le consulteur et l'expert) et nous montrons que ces dialogues sont bien réalisables avec un système à base de RDP.

4.4.1 Interface consulteur

D'après le schéma de la figure 3.5, nous pouvons prévoir 4 types d'échanges d'informations entre le consulteur et le système :

- du consulteur vers le système :
 - échange n°1 = description d'un problème,
- du système vers le consulteur :
 - échange n°2 = caractérisation d'une solution,
 - échange n°3 = preuve.

Nous adoptons le mécanisme le plus général, présenté au § 4.3.4.3, où le consulteur fournit en premier lieu les faits qu'il a déjà observés (échange n°1) : par un mécanisme de chaînage avant, le système peut éventuellement lui proposer une solution (échange n°2). Si le système n'a pas suffisamment de faits pour fournir une solution, alors un mécanisme de chaînage arrière est déclenché qui permet au système d'interroger le consulteur sur certains faits non encore observés jusqu'à ce que le système puisse fournir une solution (échange n°2). Il nous faut donc prévoir un échange supplémentaire :

- du système vers le consulteur :
 - échange n°4 = ensemble de descripteurs à instancier.

La description d'un problème, par le consulteur, ne pose aucun problème : en chaînage avant, le consulteur instancie certains descripteurs qu'il choisit parmi une liste, et en chaînage arrière, le consulteur instancie, s'il le désire, les descripteurs que lui présente le système.

Voyons donc, maintenant, les échanges d'informations du système vers le consulteur.

4.4.1.1 La caractérisation d'une solution

Les différentes caractérisations possibles d'une solution par le système sont :

- la description d'une solution,
- la détection d'une incohérence, et
- l'impossibilité de fournir une réponse.

La description d'une solution se fait au moyen du langage de décision (ou langage de réponse).

La consultation d'un RDP accorde une place très importante aux nœuds qui contiennent une réponse, c'est à dire aux nœuds qui contiennent une formule portant sur un prédicat appartenant au langage de décision. Néanmoins, toutes ces formules n'ont pas toutes le même intérêt. Si les généralisations sont permises sur ces prédicats, il se peut que la seule réponse possible soit d'une généralité telle qu'elle n'apporte aucune aide au consulteur. Il faudra donc, soit interdire la généralisation des prédicats du langage de décision, soit définir ce qu'est une réponse "intéressante" pour que les mécanismes de consultation puissent déterminer les nœuds qui contiennent effectivement une réponse. Supposons que cela soit réalisé en phase d'initialisation, et que le système sait déterminer les nœuds qui contiennent une réponse.

Il nous faut maintenant interpréter les différents états possibles d'un RDP, lorsqu'une consultation s'arrête.

4.4.1.2 Interprétation de l'arrêt d'une consultation en chaînage avant

Lorsqu'une consultation en chaînage avant est arrêtée :

- soit tous les faits fournis par le consulteur ont été traités, et alors deux cas peuvent se produire :
- le système a élu un ou plusieurs nœuds contenant une réponse, ou
- aucun nœud contenant une réponse n'est encore élu,
- soit l'arrêt s'est produit :

- sur détection d'un nœud incohérent, ou
- sur l'impossibilité de trouver un nœud couvrant le fait à traiter.

Lorsqu'en fin d'une consultation en chaînage avant, le système a élu un ou plusieurs nœuds contenant une réponse, le système fournit la (ou les) réponse(s) au consulteur. Si la théorie du système est valide (cf. § 3.3.1.2 et § 4.4.2.6), alors il n' y aura pas, parmi ces réponses, de réponses contradictoires.

En revanche, si aucun nœud contenant une réponse n'est encore élu, deux cas peuvent se produire :

- ou bien il reste des nœuds, contenant une réponse, qui sont encore indéterminés, et alors le passage en chaînage arrière permettra de les tester,
- ou bien tous les nœuds, contenant une réponse, sont rejetables : le système ne sait pas répondre car la conjonction des faits est en dehors de son domaine de connaissance.

Donc, lorsque la consultation s'arrête alors que tous les faits ont été traités, soit le système a élu un ou plusieurs nœuds contenant une réponse, soit il manque des faits pour trouver une réponse et il faut passer en chaînage arrière, soit le cas à traiter est hors de son domaine de connaissance.

Si la consultation s'arrête en cours de traitement des faits fournis par le consulteur :

- un nœud incohérent a été détecté : cela situe de façon précise une incohérence, ou bien
- le système ne trouve pas de nœud couvrant le fait en cours de traitement : ce fait est en dehors du domaine de connaissance du système.

Donc, dans tous les cas, le système peut interpréter de façon précise l'arrêt de la consultation en chaînage avant.

4.4.1.3 Interprétation de l'arrêt d'une consultation en chaînage arrière

Lorsqu'une consultation en chaînage arrière est arrêtée :

- soit un nœud contenant une réponse a été élu : il permet au système de fournir une réponse. Si l'utilisateur le désire, la recherche peut continuer pour compléter la réponse,
- soit tous les nœuds indéterminés, et contenant une réponse, ont été testés et aucun n'a été élu :

-s'il reste des nœuds contenant une réponse et qui sont encore indéterminés : alors il manque au système des faits pour pouvoir fournir une réponse,

- s'il n'en reste plus : le système ne sait pas répondre car la conjonction des faits est en dehors de son domaine de connaissance.

- soit le système a trouvé un nœud incohérent, et cela situe de façon précise une incohérence.

Donc, en chaînage arrière aussi, le système peut interpréter de façon précise l'arrêt de la consultation. Le dernier échange à examiner, entre le système et le consulteur, est la fourniture d'une preuve par le système.

4.4.1.4 Recherche d'une preuve

Une preuve est la décomposition d'une réponse du système en lemme, c'est à dire en enchaînements de conjonctions de faits, remontant jusqu'aux faits fournis par le consulteur.

Pour fabriquer la preuve de la description d'une solution, il suffit de parcourir le sous-graphe du RcDP dont la racine est le nœud contenant la réponse à prouver, à la recherche des faits fournis par le consulteur. Les procédures qui permettent de parcourir virtuellement un RcDP, à l'intérieur d'un RDP, sont décrites dans [Sainte Marie 91].

Pour améliorer la compréhension de la preuve, le système pourra s'arrêter sur les nœuds qui contiennent une formule du langage de décision (ou langage de réponse), quitte à prouver ensuite ces formules si le consulteur le désire.

Pour cela, une procédure récursive peut être suivie :

- dans le RDP, pour établir la preuve d'une formule d'un nœud qui contient un fait établi par le consulteur, il suffit de retourner ce fait,

- de même, dans le RDP, pour établir la preuve d'une formule d'un nœud, qui contient une formule du langage de décision, il suffit de retourner cette formule,

- sinon, dans le RcDP, pour établir la preuve d'une formule d'un nœud dont un père est éligible, il faut établir la preuve d'une formule de ce père, et la retourner,

Le système peut aussi prouver une incohérence lorsqu'elle est située, c'est à dire lorsqu'il a détecté un nœud incohérent :

- si le nœud était élu, il faut prouver une formule de ce nœud, et prouver la négation de la formule du nœud qui a permis de détecter l'incohérence,
- si le nœud était rejetable, il faut prouver la négation d'une formule de ce nœud, et prouver la formule du nœud qui a permis de détecter l'incohérence.

Pour prouver la négation d'une formule d'un nœud, il faut d'abord examiner si un fait fourni par le consulteur permet de nier une formule du nœud, sinon il faut examiner, dans le RcDP, si un fils est rejetable, auquel cas il suffit de prouver la négation d'une formule de ce fils.

4.4.2 Interface expert

Les échanges entre l'expert et le système sont plus complexes. En plus des échanges prévus en figure 3.5, il nous faut donner la possibilité à l'expert d'enrichir le langage de description. Ainsi, nous obtenons :

- de l'expert vers le système :

- échange n°5 = description d'un problème et de sa solution,
- échange n°6 = description d'une solution,
- échange n°7 = contre-exemple,
- échange n°8 = descripteur ou modalité,

- du système vers l'expert :

- échange n°9 = preuve,
- échange n°10 = description d'un problème.

4.4.2.1 Description d'un problème et de sa solution, par l'expert

Pour décrire un problème et sa solution (échange n°5), l'expert procède comme un consulteur en chaînage avant : il choisit, dans une liste, les descripteurs qu'il désire instancier. Cette liste est la même qu'en consultation, augmentée du langage de décision.

Le système doit ensuite prouver son accord ou son désaccord avec l'expert. Pour cela, le système utilise un mécanisme de chaînage avant avec propagation de l'élection ou du rejet des nœuds. Ainsi, lorsque ce mécanisme s'arrête, les différents états possibles du RDP sont :

- si tous les faits fournis par l'expert ont été traités (y compris les faits exprimés avec le langage de décision), alors tous les nœuds contenant une réponse de l'expert ont été élus (aucune incohérence n'a été détectée). Eventuellement, d'autres nœuds contenant une réponse ont aussi été élus.

- sinon :

- soit un nœud incohérent a été détecté : cela situe de façon précise une incohérence,
- soit le système ne trouve pas de nœud couvrant un fait en cours de traitement : ce fait est en dehors du domaine de connaissance du système.

Dans tous les cas le système sait interpréter l'arrêt du traitement des faits fournis par l'expert, et peut éventuellement fournir une preuve (échange n°9).

4.4.2.2 Recherche d'une preuve

Si tous les faits fournis par l'expert ont été traités, il n'a donc été détecté aucune incohérence : le système est d'accord avec l'expert, sauf, peut-être, pour certaines réponses que l'expert n'avait pas données et qui appartiennent à des nœuds élus. La recherche d'une preuve d'une réponse est identique à celle proposée pour le mode consultation. Si l'expert ne réfute aucun lemme, l'exemple peut être intégré au RDP, avec l'algorithme présenté en figure 4.5.

Sinon, si un nœud incohérent a été détecté, l'incohérence est située de façon précise, et la recherche d'une preuve est identique à celle proposée pour le mode consultation.

Dans le dernier cas, aucune preuve ne peut être fournie : le système ne peut qu'affirmer que ce fait sort de son domaine de connaissance.

4.4.2.3 Traitement d'un contre-exemple

L'expert doit alors exprimer son accord ou son désaccord avec la preuve ou l'affirmation du système (échange n°7).

Dans le cas d'une affirmation (un fait sort du domaine de connaissance du système), l'expert ne peut qu'accepter ou renoncer à l'extension du domaine de connaissance du système. Si l'expert accepte, le contre-exemple est tout trouvé : il s'agit de celui en cours de traitement, c'est à dire la conjonction de tous les faits fournis par l'expert. L'intégration de ce nouvel exemple peut être faite avec l'algorithme présenté en figure 4.5. Sinon, si l'expert refuse, le système considère que l'expert a voulu le "piéger", et arrête là (que pourrait-il faire de plus ?) le traitement des faits fournis par l'expert.

Dans le cas où l'expert réfute un lemme, un contre-exemple doit être fourni pour que le système modifie sa base de connaissance, et propose à nouveau son accord ou son désaccord à l'expert. L'expert est alors concentré sur le lemme qu'il veut réfuter, et peut donc fournir un contre-exemple simple : un lemme étant un raisonnement très court, l'effort de l'expert est donc moindre, et le risque qu'il se trompe fortement diminué.

La description du contre-exemple se fait de la même façon que la description du couple (problème, solution). Le système intègre ce contre-exemple à l'aide de l'algorithme proposé en figure 4.5. Ensuite le système reprend tout son raisonnement (en chaînage avant) pour exprimer son accord ou son désaccord avec l'expert. Et ainsi de suite jusqu'à ce que l'expert accepte toutes les preuves du système (i.e. qu'il ne les réfute pas). A ce moment, si le système est d'accord avec l'expert, l'exemple peut être intégré au RDP.

4.4.2.4 Evolution du domaine et facteurs cachés

L'évolution du domaine peut se traduire de deux façons : soit il s'agit d'une modification du langage de description, soit il s'agit d'une modification dans la résolution des problèmes.

La modification du langage de description correspond à l'échange n°8, et consiste, pour l'expert, à introduire de nouveaux termes dans le langage de description (aussi bien dans le langage d'observation que dans le langage de décision). Cette facilité a déjà été étudiée au § 4.3.2.7, et nous y avons vu quelles modifications du langage de description respectent l'incrémentalité de représentation.

La modification de la résolution d'un problème est plus difficile à traiter, car il peut s'agir de ne plus prendre en compte certains anciens exemples. Or l'algorithme que nous adoptons respecte un critère de consistance fort : il faut que les concepts couvrent tous les exemples et aucun

contre-exemple. Il faut donc prévoir un mécanisme capable d'identifier les exemples qu'il faut éliminer afin que la consistance soit toujours respectée, ainsi que la validité de la connaissance.

Considérons donc une réponse \mathcal{R} et la formule \mathcal{F} (éventuellement complexe) qui couvre tous les exemples contenant \mathcal{R} , et seulement ceux-ci. Le problème à résoudre est de savoir quoi faire dans le RDP, lorsqu'une partie de la formule \mathcal{F} ne doit plus mener à \mathcal{R} mais à une autre réponse \mathcal{R}' . Ce problème apparaîtra lorsqu'un expert fournira un exemple dont la description contient \mathcal{R}' mais pas \mathcal{R} , et dont \mathcal{F} est une description. Le système prouvera alors à l'expert pourquoi il faut conclure \mathcal{R} plutôt que \mathcal{R}' . Ceci mettra en lumière une conjonction de faits (un lemme), contenue dans \mathcal{F} , qui est en cause. L'expert fournira alors un contre-exemple réfutant ce lemme. Pour que l'apprentissage soit rapide, il faudra que l'expert fournisse un contre-exemple le plus général possible, de façon à éliminer d'un seul coup toute une partie de la formule \mathcal{F} qui conduit à \mathcal{R} . Ce contre-exemple sera intégré au RDP, mais s'il s'agit d'une évolution du domaine, alors il se peut que certains anciens exemples, couverts par ce contre-exemple, et conduisant à \mathcal{R} soient à supprimer du RDP pour que la consistance et la validité soient respectées.

Soient ae la partie problème d'un tel ancien exemple, et soit ce celle du contre-exemple fourni par l'expert. Si l'on intègre (ce, \mathcal{R}') au RDP selon l'algorithme proposé en figure 4.5, alors, puisque la consistance est respectée, et puisque ce recouvre, ou est égale à ae , \mathcal{R}' se trouve seul dans un nœud, et ce nœud contenant \mathcal{R}' est, soit le fils d'une feuille vide, soit une feuille, selon que \mathcal{R}' est déjà apparu dans un autre exemple, ou non. La validité n'est alors plus respectée puisqu'on ne peut plus conclure ni \mathcal{R}' , ni \mathcal{R} à partir de la description de ae ou celle de ce .

Habituellement, cette situation révèle la présence d'un facteur caché, c'est à dire d'une formule, que l'expert n'a pas fournie, et qui permet, dans la réalité, de discriminer ae de ce . Puisque, dans notre cas, il peut s'agir aussi d'une évolution du domaine, il convient donc d'interroger l'expert en le prévenant des conséquences de son choix.

Evolution du domaine :

Si l'expert confirme son contre-exemple et accepte la suppression d'anciens exemples — si l'expert le désire, le système peut lui présenter tous ces exemples — nous considérons que nous nous trouvons en présence d'une évolution du domaine. Pour déterminer quels sont les exemples à supprimer dans le RDP, il faut former la conjonction des formules constituant le

contre-exemple, et considérer l'ensemble des exemples du RDP, couverts par cette conjonction : tous ces exemples (sauf le contre-exemple que l'expert vient de fournir) sont à supprimer.

La suppression d'un exemple doit s'accompagner d'opérations de restructuration du graphe, de façon à obtenir un RDP correct (généralisations minimales, critère de consistance). Soit N le nœud représentant l'exemple à supprimer : parcourir les nœuds n , successeurs de N , en commençant par N et en allant vers la racine, par ordre topologique (n'examiner un nœud qu'après avoir examiné tous ses pères), et effectuer les opérations suivantes :

soit P l'ensemble des nœuds prédécesseurs de n :

- si $P = \emptyset$ alors

- retirer n de V_{RDP} , et
- retirer tous les arcs dont il est l'origine, de E_{RDP} , car n ne repose désormais sur aucun exemple,

- si P est un singleton, soit $P = \{p\}$: alors la couverture (i.e. l'ensemble des exemples couverts par ce nœud, [Royer 90]) de n et celle de p sont identiques. Il faut donc réunir p et n (ne garder que les formules les plus spécifiques de p et de n), après avoir mis à jour les liens autour de ce nouveau nœud, de façon à supprimer les éventuels liens redondants :

- supprimer (p, n) de E_{RDP} ,
- s'il existe $x \in V_{RDP}$ tel que $(n, x) \in E_{RDP}$ et $\text{chemin}(p, x) = \text{vrai}$ alors supprimer (n, x) de E_{RDP} ,
- réunir p et n ,

- si P n'est ni un singleton, ni vide, alors former l'ensemble D des nœuds représentant la disjonction de tous les éléments de P . Pour chaque $d \in D$, tel que $\text{chemin}(n, d) = \text{faux}$, faire :

- pour tout $p \in P$, tel que $(p, d) \in E_{RDP}$ faire supprimer (p, d) de E_{RDP} ,
- pour tout $x \in E_{RDP}$, tel que $(n, x) \in E_{RDP}$ et $\text{chemin}(d, x) = \text{vrai}$ faire supprimer (n, x) de E_{RDP} ,
- ajouter (n, d) à E_{RDP} .

Cet algorithme n'est que l'algorithme de base pour un RDP où ne sont pas représentées les formules complexes (cf. § 4.3.2.3). Pour plus de détails sur l'algorithme complet, le lecteur pourra consulter [Sainte Marie 91].

Présence d'un facteur caché :

Si l'expert refuse la suppression d'anciens exemples, nous considérons que la présence d'un facteur caché a été détectée. Pour aider à l'identification du facteur caché, le système simule l'existence de ce facteur caché en inventant un nouveau descripteur qu'il appelle *paramètre_inconnu*. Le système recherche alors les exemples couverts par la description du contre-exemple (cf. ci-dessus), les retire du RDP, leur adjoint ce nouveau descripteur avec une valeur inconnue baptisée *valeur1*, et les ré-introduit dans le RDP. Puis le système retire le contre-exemple, lui adjoint le nouveau descripteur avec une autre valeur inconnue *valeur2*, et ré-injecte ce contre-exemple dans le RDP. Le système est alors en mesure de fournir à l'expert toutes les Dépendances Perçues qui font intervenir le descripteur *paramètre_inconnu*. Si l'expert devine quel est ce descripteur inconnu, il le définit, et le RDP est mis à jour; sinon tous les exemples couverts par le contre-exemple, et le contre-exemple lui-même, sont retirés du RDP. L'expert doit alors choisir exclusivement entre la réintroduction des anciens exemples et l'oubli de son contre-exemple, ou la suppression des anciens exemples et l'introduction de son contre-exemple.

4.4.2.5 Recherche d'un problème par le système

Le dernier échange entre le système et l'expert, est celui qui consiste, pour le système, à identifier un problème pour lequel il n'a pas de solutions (une lacune dans sa base de connaissance), et à proposer ce problème à un expert (échange n°10). L'expert fournit alors une solution à ce problème (échange n°6).

Les problèmes pour lesquels le système n'a pas de solution sont d'abord ceux qui ont été soumis par un consultant, et auxquels le système n'a pas su répondre : il suffit de les enregistrer pour ensuite les proposer tels quels à un expert, en prenant soin de les ré-essayer avant de les proposer. L'expert pourra, s'il le désire, compléter ces exemples, puis les résoudre. Le système n'a alors plus qu'à traiter ce nouvel exemple selon le schéma ordinaire décrit au § 4.4.2.1.

Mais le système peut détecter de lui-même un autre type de problèmes pour lesquels il n'a pas de solution : il peut facilement détecter, pour chaque descripteur, les limites de sa connaissance.

Pour cela, le système forme l'ensemble E des nœuds du RDP qui contiennent une réponse, et, pour le descripteur étudié, forme l'ensemble E' des nœuds, prédécesseurs (dans le RcDP) des nœuds de E , qui contiennent une formule portant sur ce descripteur. Ensuite le système peut former la réunion des ensembles de valeurs de ce descripteur pour chaque nœud de E' . Soit C cette réunion. C représente le domaine de connaissance du système pour ce descripteur.

Si C est différent du domaine de définition du descripteur, le système peut fabriquer des exemples pour essayer d'agrandir son domaine de connaissance pour ce descripteur.

Prenons l'exemple d'un descripteur d à un seul attribut à valeur entière : si $C=[4,25]$ alors que le domaine de définition de d est $[0,25]$, le système peut refabriquer l'exemple pour lequel $d=4$, et proposer ce même exemple en ne changeant que la valeur de d (2 par exemple). De même si $C=[4,10] \cup [14,25]$, le système peut essayer pour $d=12$ avec l'exemple pour lequel $d=10$ ou celui pour lequel $d=14$, etc.

4.4.2.6 Validité, pertinence et cohérence du RDP

Nous avons prévu au § 3.3.1.2 que la validité des réponses de l'Apprenti était garantie par l'algorithme de construction incrémentale de la théorie. On constate, en effet, que les modes de construction de la connaissance du système que nous proposons, permettent d'affirmer que cette connaissance est toujours valide : lorsqu'un exemple est intégré au RDP, le critère de consistance nous assure que le système, pour ce même exemple, fournira la même réponse que celle que l'expert avait fournie avec l'exemple. Les situations où ça ne serait pas le cas sont identifiées et résolues selon la procédure décrite au § 4.4.2.4 qui nous garantit la validité de la connaissance représentée par le RDP.

En ce qui concerne la pertinence de la connaissance représentée par le RDP, le succès des procédures que nous proposons n'est pas garanti. En effet, pour que la pertinence soit respectée, il faudrait que toutes les Dépendances Perçues du RDP soit examinées et acceptées par un expert. Cela est réalisable mais demanderait un travail beaucoup trop fastidieux à l'expert. Nous nous contentons donc d'une semi-pertinence : seules les preuves réfutées par un expert ne sont plus fournies à un consultant. Les preuves qui n'ont pas été réfutées, car elles n'ont pas été examinées, sont supposées a priori pertinentes.

La cohérence d'une base de connaissance est habituellement scindée en deux ([Ayel – Rousset 90]) :

- la cohérence statique ne porte que sur des parties de la base de connaissance, sans exploiter la puissance déductive de cette connaissance,
- la cohérence dynamique porte sur tous les résultats que peut produire la base de connaissance.

Pour étudier la cohérence, il faut se définir un "modèle conceptuel du monde réel", appelé C-modèle dans [Ayel – Rousset 90], par rapport auquel on peut juger de la cohérence d'une base de connaissance. Par exemple, dans le système INDE ([Ayel & al. 88]), le C-modèle ne contient que la contrainte de monovaluation des descripteurs, et en conséquence une base de connaissance est incohérente ssi elle permet de déduire deux valeurs différentes pour un même descripteur.

Avec ce C-modèle, puisque les exemples fournis par les experts sont individuellement statiquement cohérents, alors les mécanismes proposés aux paragraphes précédents nous garantissent que le RDP construit est statiquement et dynamiquement cohérent ([Royer 90]). Mais nous pouvons aussi ajouter à notre C-modèle : l'absence de bouclage de relations de Dépendance Perçue (car le RDP est un graphe sans circuit) et l'absence de redondance de relations de Dépendance Perçue (grâce à la procédure mettre-à-jour- ϵ_{RDP} : cf. figure 4.6).

Nous disposons donc d'un système capable de se fabriquer une connaissance semi-empirique d'un domaine, et capable de redistribuer cette connaissance au fur et à mesure qu'elle se constitue. Le dernier problème à examiner est celui de la simultanéité des opérations effectuées sur cette connaissance, car nous désirons que notre système soit accessible, à tout moment, aussi bien aux experts, qu'aux consultants.

4.4.3 L'accès multi-voie au système

Le problème qui se pose est celui de *l'exclusion mutuelle* (cf. [Tanenbaum 87]) de certaines parties des processus de lecture ou d'écriture sur un RDP, afin que des accès conflictuels sur un même RDP soient évités.

Nous avons remarqué que les opérations effectuées sur un RDP sont locales, en ce sens qu'elles ne nécessitent pas la reconstruction complète du graphe. Néanmoins, lorsqu'un nouvel exemple est introduit dans un RDP, il n'est pas possible de prévoir où seront effectuées les modifications : il n'est donc pas possible d'isoler a priori un sous-graphe du RDP sur lequel seraient effectuées toutes les modifications, et qui serait déclaré en *section critique* (cf. [Tanenbaum 87]).

D'un autre côté, nous ne pouvons pas nous satisfaire de déclarer en *section critique* uniquement l'entourage des nœuds N, n et m de l'algorithme de la figure 4.5, car les modifications apportées localement autour de ces nœuds produisent un "mauvais" RDP, tant que l'algorithme n'est pas arrivé à son terme. Nous risquerions donc de proposer certaines déductions à un consultant, qui ne seraient pas compatibles avec de futures déductions, pendant la même consultation, au fur et à mesure que l'intégration du nouvel exemple avancerait.

De même, nous ne pouvons pas nous satisfaire de remarquer que lors de l'intégration d'un nouvel exemple, si l'algorithme traite le nœud n, alors tous les successeurs de n sont déjà traités et ne seront plus modifiés. En effet, l'algorithme d'intégration d'un nouvel exemple ne garantit que la complétude de la connaissance représentée par le RDP. Sa validité n'est traitée qu'après intégration complète du nouvel exemple et peut avoir des répercussions sur tout le RDP.

La seule solution qui nous reste est de déclarer en *section critique* tout le programme d'intégration d'un nouvel exemple (y compris le traitement de la validité de la connaissance) de façon à ce qu'un seul expert à la fois puisse modifier le RDP, et de façon à ce qu'aucune consultation du RDP ne soit en cours lorsqu'un expert intègre un nouvel exemple (*exclusion mutuelle*).

Il sera peut-être possible d'alléger cette exclusion mutuelle, en permettant aux consultants de travailler sur une copie du RDP effectuée avant toute modification. Cette possibilité sera dépendante de l'environnement informatique du système.

4.5 DESCRIPTION FONCTIONNELLE DU SYSTEME TELEMATIQUE

Dans ce paragraphe, nous ne décrivons que fonctionnellement le système Moïse. Sa description organique sera réalisée ultérieurement dans notre équipe.

Rappelons, en premier lieu, les principales fonctions que devra remplir notre système :

- être accessible par télématique, et plus particulièrement par Minitel®,
- acquérir la connaissance par apprentissage automatique à partir d'exemples, et par un dialogue entre le système et l'expert,
- repérer des lacunes dans sa base de connaissance pour interroger des experts, et enfin
- redistribuer sa connaissance en étant capable de l'expliquer.

Les principaux composants du système Moïse sont :

- une base de connaissance sous forme d'un Réseau de Dépendances Perçues (RDP),
- un module de communications Vidéotex[®] chargé de gérer l'interface entre le système et les utilisateurs, selon la norme CCITT V23, et chargé aussi de la gestion des éventuels codes d'accès au système, des boîtes aux lettres (b.a.l.), des pages d'informations générales sur le système, des statistiques d'accès, etc...,

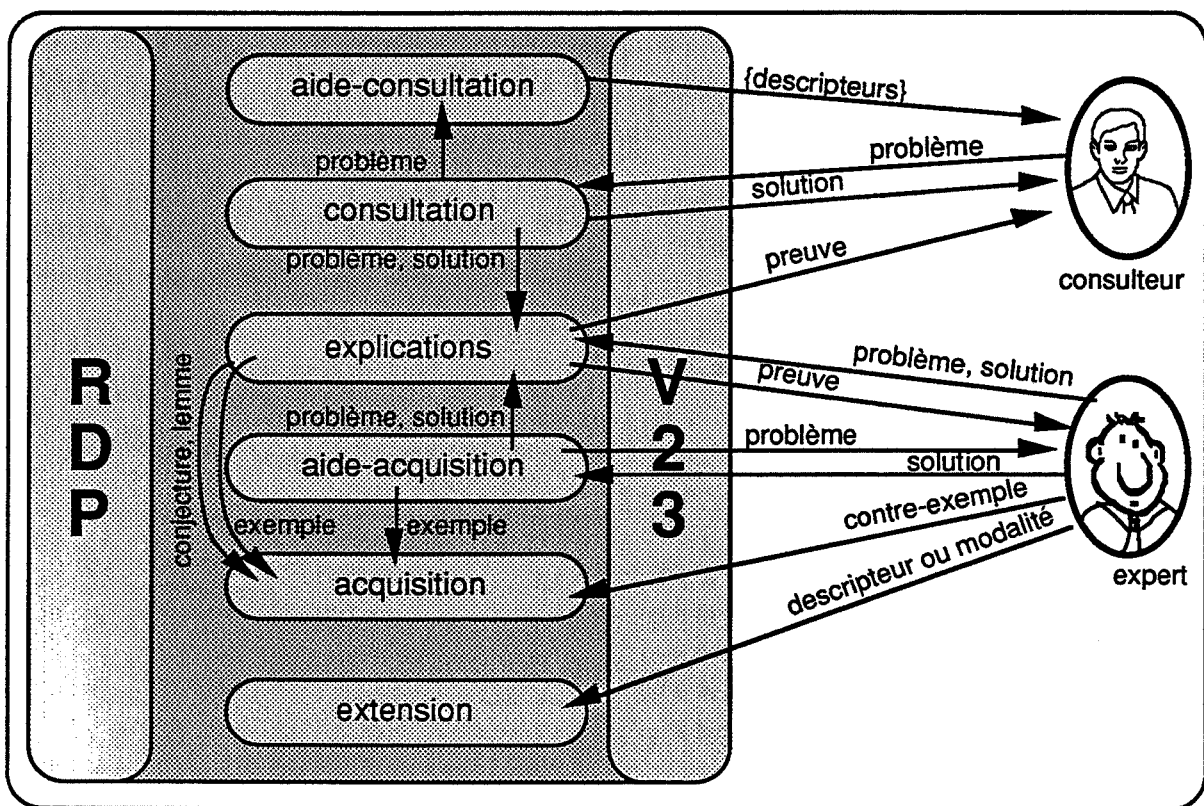


fig. 4.8 : description fonctionnelle de Moïse

- un module de consultation du RDP : en entrée, ce module reçoit un problème, en sortie il émet une solution,
- un module d'explications du raisonnement : en entrée, il reçoit un couple (problème, solution), et en sortie il émet une preuve, c'est à dire un enchaînement de lemmes,

- un module d'acquisition de la connaissance, dont le rôle est de modifier le RDP de façon à prendre en compte la réfutation d'un lemme par un expert. En entrée, ce module reçoit un triplet (conjecture, lemme, contre-exemple). Lorsque toutes les preuves sont acceptées par l'expert, si le couple (problème, solution) est un exemple, alors il est intégré au RDP.

- un module d'extension du langage de description : son rôle est de permettre, principalement, à un expert, d'entrer un nouveau descripteur ou une nouvelle modalité d'un attribut nominal ou structuré déjà existant,

- un module d'aide à la consultation qui permet à un consulteur de savoir quel(s) paramètre(s) il lui faut encore fournir pour espérer avoir une réponse de la part du système. En entrée, ce module reçoit le problème posé par le consulteur, et en sortie il émet un ensemble de descripteurs à instancier,

- un module d'aide à l'acquisition de la connaissance qui est capable de proposer à un expert un problème que le système ne sait pas résoudre. En sortie il émet un problème, et en entrée il reçoit une solution,

Comme nous pouvons le constater, nous n'avons pas respecté scrupuleusement la formalisation du dialogue que nous avons présentée au chapitre 3 (cf. figure 3.5) : nous y ajoutons deux fonctionnalités, via deux modules : l'aide à la consultation et l'extension. Le premier de ces modules n'offre qu'une facilité d'utilisation du système par un consulteur. En revanche, le deuxième module est d'une importance capitale pour que le système apprenne vraiment : c'est lui qui permet au système d'enrichir son vocabulaire. Nous pouvons penser, qu'en fait, dans l'environnement d'apprentissage de Reitz (cf. figure 3.3), cet enrichissement est implicite : l'apprenti enrichit son vocabulaire lorsque le Maître le critique, ou bien lorsque l'Oracle ou la Sonde lui fournissent des problèmes résolus, avec de nouveaux "mots". Nous avons préféré expliciter cette fonctionnalité, car elle a une influence sur l'incrémentalité de construction du RDP.

4.6 CONCLUSION

Nous avons donc montré la faisabilité d'un système d'acquisition de la connaissance à partir d'exemples qui permet de prendre en compte des données éventuellement bruitées, tout en étant capable d'être opérationnel dans des domaines évolutifs. Pour cela, nous n'avons pas suivi la

voie habituelle du traitement des données bruitées, qui consiste en un traitement statistique du bruit, mais avons opté pour un traitement de ces données par dialogue avec l'expert.

En effet, si le domaine est évolutif, et si des exemples peuvent être bruités, alors il n'est pas possible au système de décider, lorsqu'un exemple vient contredire une connaissance déjà acquise, s'il s'agit d'une évolution du domaine ou d'un exemple bruité : les symptômes sont identiques. Par un traitement statistique de ces cas, le système pourrait décider entre ces deux situations : si une connaissance est de plus en plus remise en cause par des nouveaux exemples, il s'agit très certainement d'une évolution du domaine. A l'inverse, lorsqu'un exemple vient confirmer une connaissance contredite par un exemple précédent, il y a de fortes chances que l'exemple qui avait contredit la connaissance, était bruité. Mais entre le moment où un exemple contredit une connaissance déjà acquise, et le moment où le système est capable de décider des mesures à prendre, la connaissance en suspens est inutilisable, ainsi que l'exemple qui a déclenché l'ambiguïté.

Dans notre système, d'un côté, les experts ne sont pas considérés comme étant des êtres infailibles, et leurs faiblesses sont prises en compte (possibilité de bruit dans les exemples fournis). Mais, d'un autre côté, leurs capacités d'expertise, et de réflexion sur cette expertise, sont pleinement utilisées lors des dialogues avec le système (réfutation de preuves, découverte de facteurs cachés). Ainsi, les situations ambiguës sont résolues par ces dialogues, et le choix entre évolution du domaine et exemple bruité est effectué immédiatement : la base de connaissance du système est donc à tout moment utilisable par des consultants.

La connaissance représentée par un RDP n'est donc plus une connaissance purement empirique, mais plutôt une connaissance semi-empirique : cette connaissance ne s'appuie pas uniquement sur l'analyse des exemples, mais aussi sur leur étude, effectuée par l'expert avec l'aide du système.

Notre système ne sera donc pas qualifié de "système d'apprentissage automatique", mais de "système d'acquisition interactive de la connaissance", formulation qui est plus évocatrice des véritables objectifs de notre système, car elle sous-entend l'intervention active d'éléments extérieurs qui permettent l'acquisition de la connaissance.

Chapitre 5

Moïse : application à l'assainissement autonome des eaux usées

Dans le chapitre précédent, nous avons présenté le système Moïse de façon générale : comment il peut acquérir la connaissance, par dialogue avec les experts, et comment il peut redistribuer cette connaissance acquise, en étant capable de l'expliquer. Dans ce chapitre, nous allons présenter son application au domaine qui a motivé cette recherche : l'assainissement autonome des eaux usées, qui sera le premier domaine d'application du système Moïse.

L'application du système Moïse à un domaine particulier consiste uniquement en une initialisation du RDP, telle que définie au § 4.3.3, c'est à dire dans le choix d'une relation fondatrice de la relation de Dépendance Perçue, et dans la définition de l'Univers du Discours.

Ce dernier point est à la fois relativement peu important car, comme nous l'avons vu, le langage de description peut évoluer en cours d'acquisition de la connaissance, et capital, car il conditionne de manière déterminante le début de la vie du système. En effet, un mauvais langage de description, ne pourra fournir, au début, que de mauvais exemples. Il faudra

attendre que les experts aient suffisamment amélioré ce langage pour que l'intégration des exemples soit performante. Il convient donc de bien définir ce langage, de façon à ce que l'acquisition de la connaissance soit la plus rapide possible, pour ne pas décourager les experts.

Après avoir présenté la relation fondatrice choisie, et le langage de description du domaine, nous examinerons la pertinence de la connaissance acquise, et proposerons quelques pistes pour mieux prendre en compte ce critère.

5.1 LA RELATION FONDATRICE

La relation fondatrice de la relation de Dépendance Perçue (notée \mathfrak{R} au chapitre 2), actuellement implantée dans le logiciel EDGE, est la relation de dérivation (cf. § 2.6.6.2). Avec ce choix de relation fondatrice, la relation de Dépendance Perçue prend la signification suivante :

$\mathbf{F}_a \rightarrow \mathbf{F}_b$ si et seulement si il existe un exemple qui est décrit par \mathbf{F}_a
et tous les exemples qui sont décrits par \mathbf{F}_a , sont aussi décrits par \mathbf{F}_b .

où \mathbf{F}_a et \mathbf{F}_b sont deux formules du langage de description du domaine.

Mais avec la définition de plusieurs sous-langages du langage de description, la relation de Dépendance Perçue peut prendre plusieurs sens : on peut déjà prévoir "toujours concomitant avec" à l'intérieur d'un sous-langage, "toujours suivi de" entre un sous-langage d'observations et un sous-langage de prédiction ([Sainte Marie 90a]), "permet de décider" entre un sous-langage d'observations et un sous-langage de décision, ou pour une relation d'Equivalence Perçue : "est synonyme de" pour deux sous-langages décrivant les mêmes objets, etc.

C'est l'étude du domaine d'application, et en particulier la définition des différents sous-langages, qui permet de déterminer ces différents sens (cf. § 5.2.5).

5.2 LE LANGAGE DE DESCRIPTION DU DOMAINE

Pour étudier le langage de description du domaine de l'assainissement autonome des eaux usées, nous disposons de la base de connaissance de la maquette de serveur expert (décrite au § 4.1), et d'un ensemble d'expertises réalisées récemment (certaines de ces expertises figurent en annexe de ce mémoire).

La base de connaissance de la maquette de serveur expert a été élaborée, par les hydrogéologues de notre équipe, à l'aide de divers ouvrages, dont des textes officiels souvent très flous, ainsi que des discussions avec des spécialistes (D.D.A.S.S., Agence Financière de Bassin, ...) : pour plus de précisions, le lecteur pourra consulter [Cres 89].

5.2.1 L'expertise en assainissement autonome des eaux usées

On a coutume de distinguer quatre fonctions en assainissement autonome des eaux usées : la collecte, le pré-traitement, le traitement et l'évacuation. La figure 5.1 explicite ces quatre fonctions pour le cas de l'assainissement individuel. Pour l'assainissement autonome, le principe est le même, mais concerne plusieurs bâtiments regroupés pour un seul dispositif de traitement.

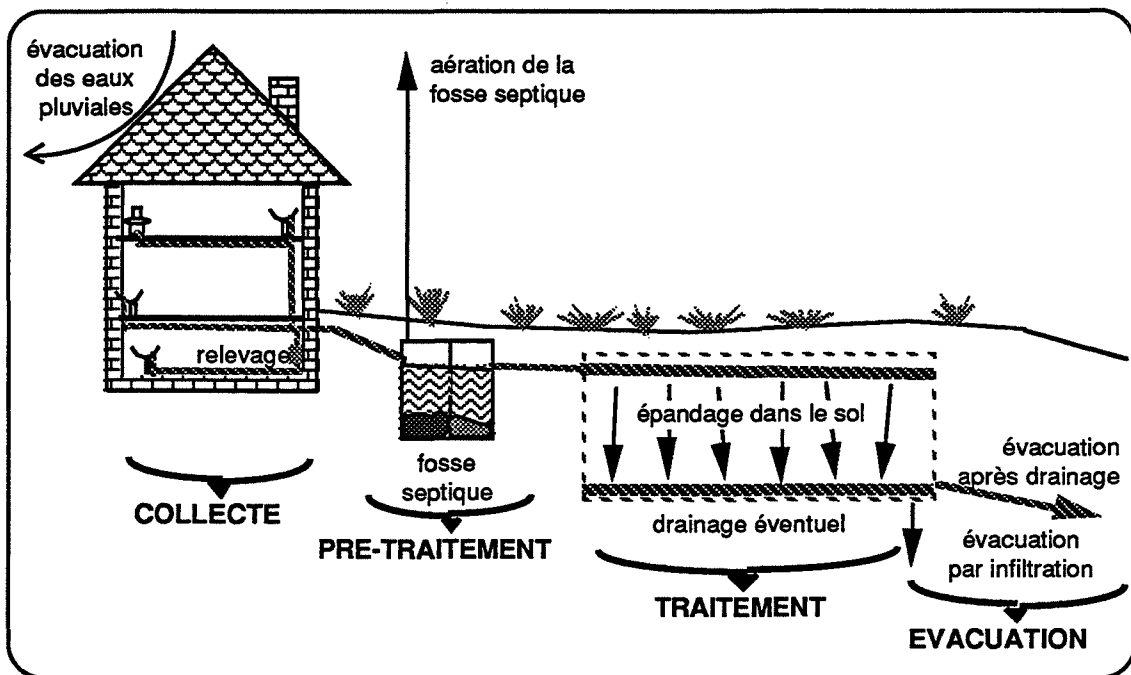


figure 5.1 : principe de l'assainissement individuel ([Cres 89])

Parmi ces quatre fonctions, seules nous intéressent, pour l'instant, celles qui posent le plus de problèmes de conception, à savoir les deux dernières : le traitement et l'évacuation. Dans le futur, lorsque nous nous intéresserons au diagnostic de dysfonctionnement des dispositifs d'assainissement, les quatre fonctions devront être considérées.

Une expertise, en assainissement autonome des eaux usées, consiste en un recueil :

- de données, relatives au terrain sur lequel on espère installer le dispositif d'assainissement, et
- de données permettant d'évaluer la quantité d'eau à assainir, de façon à prévoir un dimensionnement du dispositif d'assainissement et à le comparer à la place disponible du terrain étudié.

A partir de ces données, l'expert détermine quel dispositif est le plus apte à assainir les eaux usées, et quelles précautions il faut prendre pendant l'installation ou pour l'entretien.

Dans cette application, qui peut être caractérisée d'"aide à la décision", deux sous-langages apparaissent immédiatement : un sous-langage permettant de décrire les données nécessaires à la prise de décision (le langage des observations), et un langage de description de la décision (le langage des décisions). Le but du système d'aide à la décision est alors de mettre en correspondance ces deux sous-langages. La relation de Dépendance Perçue, lorsqu'elle reliera une formule du langage des observations, et une formule du langage des décisions, prendra le sens "permet de décider". Par exemple :

(pente > 25%) → (filière = aucune)

pourra se lire :

la pente du terrain est supérieure à 25% permet de décider qu'aucune filière ne convient.

Nous allons, dans les paragraphes suivants, étudier ces deux sous-langages.

5.2.2 Le langage d'observation

Dans un premier temps, nous ne considérons que les caractéristiques du terrain. Nous laissons de côté l'aspect dimensionnement du dispositif d'assainissement.

Le calcul de ce dimensionnement ne soulève aucune difficulté (c'est une fonction très simple de quelques caractéristiques du bâtiment, telles que vocation du bâtiment, nombre de chambres s'il s'agit d'une habitation, etc.). Mais, en revanche, la recherche de la place disponible sur la parcelle est beaucoup plus délicate à étudier, car elle fait intervenir des notions géométriques, des notions d'utilité relative d'occupation de la parcelle, etc., et pour l'instant, nous n'avons pas traité d'expertises à ce sujet, qui nous permettraient de définir une première mouture du langage de description de l'occupation d'une parcelle.

Nous nous limitons donc, dans un premier temps, à la description du terrain. Dans les expertises dont nous disposons, un descripteur, nommé **taille de la parcelle**, apparaît pour certaines filières : en lui-même, il ne signifie pas grand chose, si l'on n'étudie pas la quantité d'effluents à traiter et donc les dimensions du dispositif. Nous décidons donc de ne pas le prendre en compte, tant que nous n'étudions que les caractéristiques du terrain.

5.2.2.1 Hétérogénéité de la parcelle étudiée

Si une parcelle n'est pas homogène, l'expert peut être amené à fournir plusieurs valeurs pour un même descripteur. Lorsque cette hétérogénéité est fonction de la profondeur, il nous suffit de prévoir des descripteurs dont l'une des variables est la profondeur (cf. § 5.3.2).

En revanche, lorsque cette hétérogénéité est surfacique, l'étude du terrain se fait pour chaque sous-parcelle homogène, puis l'installation du dispositif est envisagée sur une ou plusieurs sous-parcelles, en tenant compte de leurs dimensions respectives, et du dimensionnement prévu pour le dispositif. Par exemple, si l'épaisseur du sol d'une parcelle varie de 0 à 3 mètres, mais qu'aucune sous-parcelle n'est d'une profondeur et d'une surface suffisantes pour installer un épandage en tranchées, il faut envisager une filière moins encombrante et/ou moins exigeante en épaisseur de sol (c'est le cas de l'expertise n° 21).

Etant donné que, dans un premier temps, nous ne tenons pas compte de la géométrie de la parcelle, nous ne pouvons donc pas prendre en considération les différentes sous-parcelles constitutives d'une parcelle hétérogène. Nous sommes donc obligés, dans un premier temps, de prendre en compte uniquement des parcelles homogènes (ou, ce qui revient au même, des sous-parcelles indépendantes les unes des autres). Dans les expertises dont nous disposons, certaines modifications ont été effectuées pour "homogénéiser" les parcelles.

La prise en compte de l'hétérogénéité d'une parcelle, ainsi que de la notion de place disponible d'une parcelle, pourra se faire si nous parvenons à faire apparaître un langage simple, permettant la description géométrique d'une parcelle. L'étude de ce langage sera faite ultérieurement.

5.2.2.2 Caractérisation symbolique et numérique d'un même paramètre

Les experts en assainissement des eaux usées évaluent les paramètres caractéristiques d'un terrain, à la fois qualitativement et quantitativement. Pour la maquette du serveur expert, nous

avons résolu ce problème en fixant, a priori, une correspondance entre fourchettes de valeurs numériques et qualificatifs, et en proposant ces fourchettes avec leurs correspondants numériques. Ainsi, pour la pente du terrain, nous demandions à l'utilisateur : "la pente est-elle très faible (inférieure à 2%) ?". L'avantage de cette approche est de mettre l'accent sur un ordre de grandeur d'un paramètre, plutôt que sur sa valeur précise.

Avec le module d'acquisition de la connaissance, ces correspondances peuvent être apprises : elles peuvent apparaître sous forme d'Equivalences Perçues entre formules portant sur un même paramètre auquel correspondent plusieurs descripteurs, selon les différentes natures du paramètre. Ainsi, on peut définir, pour un paramètre **pente**, au moins deux descripteurs : **pente_numérique** et **pente_symbolique** .

Mais si un expert donne un exemple où ne figure que **pente_symbolique**, alors on risque de perdre involontairement des correspondances entre valeurs symboliques et valeurs numériques. Aussi faut-il demander aux experts, lorsqu'ils donnent une valeur symbolique d'un paramètre pouvant aussi être décrit numériquement, d'en donner une valeur numérique, ou d'accepter la correspondance déjà apprise. Et inversement, il faut demander aux experts, lorsqu'ils donnent une valeur numérique d'un paramètre pouvant aussi être décrit symboliquement, d'en donner une valeur symbolique, ou d'accepter la correspondance déjà apprise.

Cette demande se fait par l'intermédiaire du dialogue par preuves et réfutations, entre le système et l'expert : si, par exemple, l'expert donne une valeur de **1%** pour la pente, alors le système en déduit que la pente est **très faible** : si l'expert accepte, tout va bien; sinon, il doit réfuter.

On définit ainsi, à l'intérieur du langage de description des observations, différents sous-langages, relatifs chacun à un paramètre. Chaque sous-langage permet de décrire un paramètre grâce à différents descripteurs, et permet d'établir des correspondances entre ces descripteurs.

Mais ces correspondances peuvent ne pas être uniques : "une pente est forte" a-t-il le même sens pour un expert exerçant dans les Alpes, que pour un expert exerçant en Normandie ? Se pose ici le problème que nous avons évoqué au chapitre 1, et qui est une constante du domaine de l'assainissement des eaux usées : celui de la régionalisation de la connaissance experte. Nous étudierons ce problème au § 5.2.2.4.

5.2.2.3 Caractérisation multiple d'un même paramètre

Au delà de l'aspect symbolique/numérique de la caractérisation d'un paramètre, il peut aussi arriver qu'un même paramètre soit décrit de plusieurs façons par les experts. Par exemple, dans

l'expertise n°18, la nature du sol est donnée en fonction de la profondeur, alors que dans d'autres expertises elle est donnée en une seule valeur. Il en va de même, pour certains autres paramètres (par exemple la perméabilité du sol ou la nature du substratum).

La solution réside, comme pour la dualité symbolique/numérique, dans la définition pour un paramètre donné, d'un sous-langage permettant de décrire ce paramètre, avec différents descripteurs.

5.2.2.4 La régionalisation de la connaissance experte

Le problème de la régionalisation de la connaissance ne pose pas de problèmes à notre système d'acquisition de la connaissance : il suffit de demander aux experts où a été faite l'expertise (c'est plus le lieu de l'expertise, que celui où exerce l'expert, qui importe, car un expert peut exercer dans divers lieux, et s'adapte toujours là où il exerce) et d'apprendre ce paramètre de l'expertise. L'entité géographique retenue, est le département, car l'administration qui valide les installations, et qui impose par ce biais ses choix, est départementale (ce sont les D.D.A.S.S.).

Ainsi, nous obtiendrons des Dépendances Perçues conditionnées par le lieu d'expertise. Par exemple, si dans le département **d1**, "pente faible" équivaut à "5% < pente < 10%", et dans le département **d2**, "pente faible" équivaut à "8% < pente < 14%", nous aurons les Dépendances Perçues suivantes :

- "département=**d1**" & "5% < pente < 10%" → "pente faible", et
- "département=**d2**" & "8% < pente < 14%" → "pente faible".

La régionalisation de la connaissance, peut aussi concerner la liste des paramètres pris en compte dans les expertises. Par exemple, si dans le département **d1**, ce sont la nature du sol et sa perméabilité qui sont prises en compte, alors que dans le département **d2**, ce sont des perméabilités à différentes profondeurs, nous pourrions avoir une Dépendance Perçue du genre :
"département=**d1**" & " α < perméabilité < β " & " & "nature du sol = σ " →
"département=**d2**" & " δ < perméabilité(ϕ < profondeur < γ) < ϵ ".

5.2.3 Le langage de décision

Le langage de décision, ou langage de réponse, est le langage avec lequel les décisions d'un expert sont exprimées, et avec lequel, par conséquent, le système fournit ses conseils.

Pour le domaine qui nous intéresse, il s'agit de caractériser un dispositif d'assainissement. Cette caractérisation comporte plusieurs champs : tout d'abord le principe du dispositif est nommé (les experts du domaine parlent de *filière*), ensuite des aménagements de cette filière sont prévus, tels que drainage, étanchéification, position et profondeur des tranchées pour un épandage en tranchées, etc. Mais comme ces aménagements peuvent être communs à plusieurs filières (le drainage par exemple), il convient de distinguer deux sous-langages de décision.

Un premier sous-langage de décision qui correspond à la filière choisie : il constitue ce que nous avons nommé "les réponses intéressantes" au § 4.4.1.1, et qui sont les réponses recherchées en chaînage arrière. En effet, à quoi servirait-il de se préoccuper du drainage du dispositif avant de connaître quel est le dispositif retenu ? Nous nommons ce langage, le langage de réponses.

Le deuxième sous-langage de décision correspond aux aménagements des filières. Ces réponses permettent de préciser le conseil, et doivent être examinées après avoir obtenu une "réponse intéressante". Nous nommons ce langage, le langage des précisions.

5.2.4 Le langage intermédiaire

En plus des observations de terrain, et des conclusions finales de l'expert, on peut déceler, dans les expertises, des conclusions intermédiaires qui permettent d'améliorer les explications grâce à des étapes intermédiaires.

C'est typiquement le cas pour le paramètre **aptitude à l'assainissement par le sol en place**, qui ne constitue pas à lui seul une réponse permettant de caractériser une filière ou un aménagement de filière, mais il permet à l'expert de rejeter certaines filières qui nécessitent que le sol en place soit capable d'assainir les eaux usées.

Mais c'est aussi le cas pour les paramètres auxquels sont attachés plusieurs descripteurs. Par exemple, affirmer que "la pente est faible" peut constituer une conclusion intermédiaire entre le choix d'une filière et le fait que "pente = 2%". Dans ce cas là, le paramètre peut à la fois être observable directement, et déductible d'autres descripteurs plus élémentaires. Néanmoins, "la pente est faible" ne peut être observable directement que par un expert : pour un novice, cette proposition ne veut rien dire, si elle n'est pas détaillée par une correspondance numérique. Nous choisissons donc de ne proposer aux consultants, que des descripteurs qui ne font pas partie du langage intermédiaire.

En résumé, nous nous définissons 4 sous-langages :

- un langage des réponses, qui permet de caractériser les filières d'assainissement,
- un langage des précisions, qui permet de compléter le résultat de l'expertise,
- un langage des conclusions intermédiaires, qui permet d'améliorer les preuves,
- un langage d'observation, pour exprimer les problèmes à résoudre.

5.2.5 Sémantiques de la Dépendance Perçue

Lorsqu'une Dépendance Perçue relie deux formules d'un même sous-langage, portant sur des paramètres différents, elle prend le sens "est toujours co-occurent avec". Par exemple : "la proximité d'un puits de captage d'eau est toujours co-occurent avec la présence d'une nappe aquifère".

Si elle relie deux formules portant sur le même paramètre, mais sur deux descripteurs différents, elle peut se lire "signifie que". Par exemple : "dans le département de la Loire, la pente vaut 2% signifie que la pente est faible".

Comme nous l'avons vu au § 5.2.1, une relation de Dépendance Perçue entre une formule quelconque et une formule du langage de réponse a le sens "permet de décider". C'est aussi le cas pour une Dépendance Perçue entre une formule quelconque et une formule du langage des précisions, ou entre une observation et une conclusion intermédiaire, ne portant pas sur le même paramètre.

5.2.6 Incomplétude des expertises

La dernière remarque que nous pouvons faire sur les expertises dont nous disposons, concerne l'incomplétude de certaines d'entre elles. Notamment, à partir de l'expertise n°4, de nombreuses données sont manquantes, mais ont, en fait, été prises en compte par l'expert, par rapport aux expertises précédentes qui avaient été faites dans le même secteur.

Dans notre application, nous demandons aux experts de nous fournir toutes les informations qui leur ont permis d'effectuer leur expertise, et uniquement celles-ci. Si une information n'est pas fournie par l'expert, nous estimons que l'expert ne l'a pas considérée lors de son expertise. Nous ne prenons donc pas en compte les données manquantes.

Parmi les expertises dont nous disposons, plusieurs ont été effectuées dans des secteurs identiques, et certaines observations de terrain n'ont été faites que globalement. En conséquence, ces observations ne sont pas toujours répétées dans chaque compte-rendu d'expertise, mais ont pourtant été considérées, à chaque fois, par l'expert. Nous avons donc complété artificiellement les données avec les expertises précédentes du même secteur, et avec l'aide des hydrogéologues de notre équipe.

5.3 DEFINITION DU LANGAGE INITIAL DU SYSTEME

Nous sommes donc en mesure de proposer un premier langage de description du domaine de l'assainissement autonome des eaux usées. Ce langage devra être amélioré après discussion avec quelques experts du domaine, de façon à en proposer une première version qui permette une bonne acquisition de la connaissance dès le début de la mise en service du système.

5.3.1 Les opérateurs

Les opérateurs à définir sont de deux sortes : ceux qui relieront des propositions (les opérateurs booléens), et ceux qui relieront des attributs à une ou plusieurs valeurs (les opérateurs d'affectation).

Ils interviennent dans les langages mis à la disposition des consultants ou des experts, pour qu'ils décrivent leurs problèmes, et aussi leurs solutions pour les experts.

5.3.1.1 Les opérateurs booléens

La conjonction est présente implicitement dans la description d'un problème et/ou d'une solution : les différents faits énumérés sont reliés implicitement par une conjonction.

Quant à la disjonction, elle est inutile, car d'une part elle peut être utilisée implicitement, et d'autre part, son explicitation risquerait d'alourdir considérablement l'expression des problèmes et /ou des solutions par les consultants et les experts. La disjonction peut intervenir pour attribuer "disjonctivement" plusieurs valeurs à un même descripteur : dans ce cas, les opérateurs d'affectation peuvent la remplacer. Par exemple, "la pente vaut 2 ou la pente vaut 3" et "la pente $\in \{2,3\}$ " sont des propositions équivalentes. Lorsqu'elle intervient pour relier deux propositions, dans un même exemple, on peut s'en passer en fournissant deux exemples qui contiennent chacun une proposition.

5.3.1.2 Les opérateurs d'affectation

Les opérateurs d'affectation que nous prévoyons sont l'égalité, l'inégalité, l'appartenance à un ensemble (qui peut être un intervalle pour un descripteur linéaire). Ces derniers opérateurs seront utiles, notamment, pour que les experts puissent fournir des contre-exemples pas trop spécifiques.

Ces opérateurs ne seront manipulés que par les experts. Les consultants, quant à eux, devront répondre à des questions précises et n'auront pas besoin de ces opérateurs. Par exemple, si le système a besoin de connaître la valeur de la proposition suivante :

$$"1 \% \leq \text{la pente} \leq 5 \%"$$

il demandera au consultant :

- soit de donner une valeur précise de la pente, s'il la connaît,
- soit de répondre par "oui", "non", ou "je ne sais pas" à la question suivante : "est-ce que la pente est comprise entre 1 et 5 % ?".

5.3.2 Le langage d'observation

Dans ce paragraphe, nous examinons, un par un, tous les paramètres qui permettent de décrire les caractéristiques d'un terrain, principalement à partir de l'ensemble d'expertises dont certaines figurent en annexe de ce mémoire. Certaines "améliorations" préalables de ces expertises ont été effectuées, en collaboration avec les hydrogéologues de notre équipe, de façon à pouvoir toutes les prendre en compte. Les plus importantes de ces modifications sont indiquées ci-dessous.

5.3.2.1 La pente du terrain

Comme nous l'avons vu ci-dessus, la pente du terrain peut être caractérisée quantitativement et qualitativement. Nous choisissons la caractérisation qualitative pour faire partie du langage des conclusions intermédiaires, et fixons la liste ordonnée des valeurs initiales possibles à (très faible, faible, moyenne, forte, très forte).

La caractérisation quantitative devra se faire à l'aide d'entiers positifs (une précision supérieure à celle apportée par les entiers est inutile).

D'autres caractérisations sont possibles. En effet, la pente peut ne pas être homogène sur tout le secteur étudié. L'expert considère alors plusieurs pentes. C'est le cas pour l'expertise n°23 : les pentes, à l'endroit prévu pour l'installation du dispositif, et à son aval, sont différentes; celle de

l'aval est plus élevée et risque d'entraîner des résurgences. Il faut donc prévoir un paramètre décrivant la **pen**te_aval (quantitativement et qualitativement). De même, l'expertise n° 31, nous suggère de prévoir un paramètre **pen**te_amont.

En résumé, nous retenons trois paramètres : **pen**te, **pen**te_aval, et **pen**te_amont comprenant chacun deux descripteurs (numérique et symbolique). Les descripteurs symboliques font partie du langage intermédiaire et ne seront pas proposés aux consultants. Nous prévoyons aussi un descripteur **remar**que, qui pourra recevoir du texte libre, relatif à la pente du terrain :

- **pen**te*n* : une variable linéaire entière,
- **pen**tes : une variable linéaire définie en extension, (fait partie du langage intermédiaire),
- **pen**te_aval*n* : une variable linéaire entière,
- **pen**te_avals : une variable linéaire définie en extension, (fait partie du langage intermédiaire),
- **pen**te_amont*n* : une variable linéaire entière,
- **pen**te_amonts : une variable linéaire définie en extension, (fait partie du langage intermédiaire),
- **remar**que : une variable de type texte, (fait partie du langage intermédiaire).

Les modifications apportées aux expertises sont les suivantes :

- modification de certains qualificatifs, comme, par exemple, *importante en forte, particulièrement faible en très faible*,
- transfert de la proposition *risquant d'entraîner des résurgences* dans les remarques,
- suppression d'indications de direction de la pente (cf. expertise n° 23).

5.3.2.2 Le sol

Le sol est caractérisé, selon les expertises, par sa qualité, sa nature, son épaisseur, sa perméabilité.

La nature du sol n'est pas toujours renseignée, car elle n'importe pas toujours. Lorsqu'elle est renseignée, il s'agit d'une caractérisation symbolique (terre végétale, arène granitique, zone d'altération sablo-argileuse, etc.) accompagnée d'une épaisseur, et éventuellement d'une profondeur si le sol n'est pas homogène. Toutefois, l'épaisseur d'une couche n'est pas toujours renseignée, notamment lorsqu'elle est très importante.

Sa perméabilité est une quantité mesurable. Parfois un expert peut la qualifier sans effectuer de mesures. Certains experts considèrent non pas une perméabilité du sol, mais des perméabilités en fonction de la profondeur.

L'épaisseur, si elle n'a pas été prise en compte dans l'évaluation de la nature ou de la perméabilité, est parfois évaluée par les experts.

La caractéristique qui semble pouvoir résumer toutes ces évaluations est la qualité du sol, qui n'apparaît pas explicitement dans les expertises, mais que l'on retrouve parfois sous la rubrique "nature du sol". Cf. par exemple l'expertise n°13 où nature du sol vaut "excellente". La qualité du sol fera donc partie du langage intermédiaire.

En résumé, les descripteurs sont les suivants :

- **nature/épaisseur_n** : une variable nominale et une variable linéaire réelle,
- **nature/épaisseurs** : une variable nominale et une variable linéaire définie en extension, (fait partie du langage intermédiaire),
- **nature/profondeur_épaisseur** : une variable nominale et deux variables linéaires réelles,
- **perméabilité_n** : une variable linéaire entière,
- **perméabilités** : une variable linéaire définie en extension, (fait partie du langage intermédiaire),
- **perméabilité_n/profondeur** : une variable linéaire entière et une variable linéaire réelle,
- **perméabilités/profondeur** : une variable linéaire définie en extension et une variable linéaire réelle, (fait partie du langage intermédiaire),
- **épaisseur_n** : une variable linéaire entière,
- **épaisseurs** : une variable linéaire définie en extension, (fait partie du langage intermédiaire),
- **qualité** : une variable linéaire définie en extension, (fait partie du langage intermédiaire).
- **remarque** : une variable de type texte, (fait partie du langage intermédiaire).

Les modifications apportées aux expertises sont :

- si la nature du sol est décrite par "imperméable", alors on transfère ce renseignement dans "perméabilité" qui prend la valeur "très faible",
- si la nature du sol est décrite par un qualificatif du genre : excellente, bonne qualité, etc., on transfère cette valeur dans "qualité du sol",
- transfert de certains renseignements (par exemple : "épaisseur du sol insuffisante pour un épandage par le sol en place", ou "pente risquant d'entraîner des résurgences") dans le descripteur "remarque".

5.3.2.3 Le substratum

Le substratum constitue ce que nous avons nommé "sous-sol" dans la maquette de serveur expert. Il est décrit par sa qualité, sa nature, sa profondeur, et sa perméabilité.

Sa caractérisation est, pour le langage initial du système, semblable à celle du sol :

- **nature** : une variable nominale,
- **nature/profondeur** : une variable nominale et une variable linéaire réelle,
- **nature/profondeur_épaisseur** : une variable nominale et deux variables linéaires réelles,
- **perméabilité_n** : une variable linéaire entière,
- **perméabilités** : une variable linéaire définie en extension, (fait partie du langage intermédiaire),
- **perméabilité_n/profondeur** : une variable linéaire entière et une variable linéaire réelle,
- **perméabilités/profondeur** : une variable linéaire définie en extension et une variable linéaire réelle, (fait partie du langage intermédiaire),
- **profondeur_n** : une variable linéaire entière,
- **profondeurs** : une variable linéaire définie en extension, (fait partie du langage intermédiaire),
- **qualité** : une variable linéaire définie en extension, (fait partie du langage intermédiaire).
- **remarque** : une variable de type texte, (fait partie du langage intermédiaire).

5.3.2.4 La nappe aquifère

La proximité d'une nappe (par rapport à la surface du sol) est une notion très importante, à cause de ses risques de pollution. Lorsqu'une nappe est présente sur le site, elle est caractérisée qualitativement (traces d'hydromorphies, par exemple), et éventuellement par sa profondeur. Si plusieurs nappes existent, elles sont caractérisées en fonction de leur profondeur.

Sa caractérisation se fera avec les descripteurs suivants :

- **présence d'une nappe** : une variable nominale,
- **présence d'une nappe/profondeur_n** : une variable nominale et une variable linéaire réelle,
- **présence d'une nappe/profondeurs** : une variable nominale et une variable linéaire définie en extension (fait partie du langage intermédiaire),
- **remarque** : une variable de type texte, (fait partie du langage intermédiaire).

Dans un deuxième temps, il faudra structurer la variable représentant la présence d'une nappe : une nappe peut-être captive ou libre, lorsqu'elle existe. De plus, il faudra prévoir un paramètre pour permettre d'indiquer si la nappe est drainée.

5.3.2.5 La présence d'un captage d'eau

La caractérisation de la proximité d'un captage d'eau est simple : s'il existe un captage d'eau dans les environs, il suffit de mesurer sa distance au futur dispositif d'assainissement. La réglementation interdit d'installer un dispositif d'assainissement à moins de 35 mètres d'un captage. Néanmoins la position relative de ce captage et du dispositif d'assainissement (amont, aval) peut avoir de l'importance, mais n'est pas toujours renseignée par les experts.

Ses descripteurs sont :

- **présence d'un captage** : une variable nominale,
- **présence d'un captage/distancen** : une variable nominale et une variable linéaire réelle,
- **présence d'un captage/distancen_position** : une variable nominale, une variable linéaire réelle, et une variable nominale,
- **remarque** : une variable de type texte, (fait partie du langage intermédiaire).

5.3.2.6 L'existence d'un point de rejet superficiel

Pour certaines filières, lorsque le sol ne peut pas recevoir les effluents en sortie du dispositif d'assainissement, il faut évacuer ces effluents vers un fossé, une rivière, un lac, etc. Si aucun point de rejet superficiel, n'est disponible (à proximité), certaines filières peuvent être inenvisageables. Si le seul rejet superficiel possible est situé en amont du dispositif d'assainissement, il peut être envisageable, dans certains cas, de l'utiliser, à condition d'installer une pompe conduisant les effluents vers ce point de rejet.

Sa caractérisation se fera avec les descripteurs suivants :

- **possibilité de rejet superficiel** : une variable nominale,
- **possibilité de rejet superficiel/position** : deux variables nominales,
- **remarque** : une variable de type texte, (fait partie du langage intermédiaire).

Dans un deuxième temps, il faudra structurer la variable représentant la possibilité de rejet superficiel : en effet, lorsque le rejet superficiel est possible, un grand nombre de possibilités est envisageable (lac, ruisseau, fosse, drain, ...)

Dans les expertises dont nous disposons, la position relative du rejet superficiel et du dispositif d'assainissement n'est jamais précisée. On considère donc que, dans tous les cas, le rejet superficiel se situait en aval.

5.3.2.7 La localisation géographique de l'expertise

Le problème de la régionalisation de la connaissance experte nous oblige à considérer le lieu où s'est déroulée l'expertise. Au § 5.2.2.4, nous avons vu que l'entité géographique retenue est le département. Nous définissons donc, dans le langage des observations, le descripteur suivant :

- **département** : une variable nominale.

Eventuellement, si cela peut avoir un intérêt, on pourra définir une structure pour regrouper les départements en régions administratives.

D'autres informations relatives à la localisation sont présentes dans les expertises : il s'agit du nom de l'expert et du lieu précis de la parcelle étudiée (commune, secteur, et même parfois n° cadastral de la parcelle). Ces informations n'ont aucune utilité pour les consultants, mais pourront servir à l'expert du domaine chargé de la maintenance du système, afin de pouvoir éventuellement contacter un expert, pour avoir des précisions sur une expertise particulière. Ces informations doivent donc être enregistrées, mais ne font pas partie du langage d'observation.

5.3.3 Le langage des conclusions intermédiaires

Le langage des conclusions intermédiaires contient déjà certains descripteurs définis au paragraphe précédent :

- pentes, pente avals, pente amonts, et remarque pour la pente,
- perméabilités, perméabilités/profondeur, épaisseurs, qualité, et remarque pour le sol,
- perméabilités, perméabilités/profondeur, profondeurs, qualité, et remarque pour le substratum,
- remarque pour la nappe aquifère,
- remarque pour la présence d'un captage d'eau, et

- remarque pour l'existence d'un point de rejet superficiel.

Nous pouvons y ajouter un descripteur pour les remarques générales :

-remarque : une variable de type texte,

ainsi que les descripteurs suivants :

- **aptitude à l'assainissement par le sol en place** : une variable linéaire définie en extension,

- **dispersion des effluents** : une variable linéaire définie en extension,

- **traitement des effluents par le dispositif** : une variable linéaire définie en extension,

- **protection de la nappe** : une variable linéaire définie en extension,

Ces quatre derniers descripteurs n'ont qu'une variable qui prendra ses valeurs dans une extension du domaine booléen. Nous proposons (oui, limite, non).

En ce qui concerne les descripteurs "remarque" (qui peuvent recevoir du texte libre) nous prévoyons que leurs valeurs soient examinées régulièrement par un expert du domaine, chargé de la maintenance du système, afin de voir si elles ne permettent pas la création de nouveaux descripteurs.

Les principales modifications apportées aux expertises sont uniquement d'exprimer ce qui se trouve sous la rubrique "utilisation du dispositif", dans un descripteur "remarque".

5.3.4 Le langage des réponses

Le langage des réponses ne comporte qu'un seul descripteur, permettant d'exprimer les filières d'assainissement retenues par les experts, à savoir : épandage en tranchées, filtre à sable, terre filtrant, filtre bactérien percolateur (aucune expertise ne le prévoit), lagunage, ou bien aucune (dans les cas où la parcelle étudiée ne peut pas recevoir de dispositif d'assainissement) :

- **dispositif** : une variable nominale.

Les modifications apportées aux expertises sont de ne retenir, dans la rubrique "filière", que le principe du système d'assainissement, et de reporter toutes les autres informations dans le langage des précisions. Les filières "lit filtrant" et "lit d'infiltration" ont été traduites par "tertre filtrant".

5.3.5 Le langage des précisions

Les options pouvant être retenues pour l'adaptation d'une filière donnée, sont très diverses :

- drainage du dispositif d'assainissement,
- orientation de l'écoulement des eaux dans un filtre à sable,
- orientation des tranchées par rapport à la pente, pour un épandage en tranchées,
- étanchéification du dispositif d'assainissement,
- profondeur des tranchées, pour un épandage en tranchées,
- aération du dispositif, dans le cas d'un lagunage.

Nous en déduisons les descripteurs suivants :

- **drainage** : une variable booléenne,
- **orientation de l'écoulement des eaux** : une variable nominale,
- **orientation des tranchées** : une variable nominale,
- **étanchéification** : une variable booléenne,
- **profondeur des tranchées** : une variable nominale,
- **profondeur des tranchées_n** : une variable linéaire réelle,
- **aération** : une variable booléenne.

Le langage de description, ainsi défini, permet de décrire les exemples d'expertise dont nous disposons. Le lecteur trouvera en annexe de ce mémoire quelques résultats de l'apprentissage sur cette base d'exemples. Dans le paragraphe suivant, nous examinons un point particulier soulevé par ces tests : celui de la pertinence des conseils que peut fournir la base de connaissance ainsi formée.

5.4 LA PERTINENCE DES CONSEILS FOURNIS

Au § 3.3.1.2, nous avons défini une théorie valide comme étant une théorie qui permet de retrouver toutes les solutions aux exemples fournis par les experts et qui n'ont pas été remis en cause. Et nous avons aussi défini une théorie pertinente, comme étant une théorie qui ne permet d'engendrer que des preuves acceptées, et ensuite jamais réfutées, par un expert. Au § 4.4.2.6, nous avons affaibli cette condition de pertinence, et avons choisi une semi-pertinence : une théorie est semi-pertinente si elle permet d'engendrer uniquement des preuves qui n'ont jamais été réfutées par un expert.

Nous admettons donc qu'un conseil fourni par le système est pertinent s'il est issu d'une théorie semi-pertinente. Nous montrons, ci-dessous, qu'il est possible d'améliorer ce critère en considérant le "substrat" d'une formule présente dans le RDP. Nous appelons substrat d'une formule d'un RDP, l'ensemble des exemples qui sont en Dépendance Perçue avec cette formule, et nous considérons le nombre d'éléments de cet ensemble.

Lorsqu'une preuve est fournie par le système, elle l'est sous la forme $\mathcal{F}_a \Rightarrow \mathcal{F}_b$: plus le nombre d'éléments du substrat de la formule \mathcal{F}_a est grand, plus cette formule est crédible, et par conséquent plus la relation $\mathcal{F}_a \Rightarrow \mathcal{F}_b$ est à son tour crédible. En effet, plus le substrat de la formule \mathcal{F}_a contient d'éléments, plus cette formule a été observée de fois, et cela conforte d'autant plus notre opinion que $\mathcal{F}_a \Rightarrow \mathcal{F}_b$. Nous pouvons donc moduler le critère de pertinence d'un conseil fourni par le système, par une considération statistique sur l'ensemble d'apprentissage.

Le système peut aussi indiquer si cette Dépendance Perçue a été acceptée par un expert, combien de fois elle l'a été, et même le nombre d'experts différents qui l'ont acceptée.

Si le système indique au consulteur (ou à l'expert) le nombre de fois qu'il a perçu une dépendance (en le comparant au nombre total d'exemples formant l'ensemble d'apprentissage), et le nombre de fois qu'un expert a accepté cette dépendance, le consulteur sera plus à même d'apprécier la confiance qu'il doit accorder à la preuve fournie.

Mais nous n'envisageons pas d'utiliser automatiquement cette modulation de la pertinence : une Dépendance Perçue peut n'être fondée que sur un seul exemple, n'avoir jamais été acceptée par un expert (car jamais proposée), et pourtant être très pertinente ! Il faut laisser le soin à l'interlocuteur du système de juger lui-même une preuve fournie par le système, et pour l'aider dans cette tâche, lui fournir le plus d'éléments possible. La seule garantie qu'a le consulteur, est que chaque preuve fournie par le système n'a encore jamais été refusée par un expert.

5.5 CONCLUSION

La définition d'un premier langage de description du domaine de l'assainissement autonome des eaux usées, qui ne tient pas compte de la géométrie des parcelles étudiées, n'a pas posé de problèmes. Au fur et à mesure de la saisie des exemples, le langage a été enrichi, et des améliorations possibles sont apparues, notamment :

- la définition de types structurés (arborescences) pour regrouper les départements en régions administratives ou géographiques, et pour préciser la présence d'une nappe ou celle d'un point de rejet superficiel, et

- la nécessité d'inclure des opérateurs d'affectation autres que l'égalité (seul opérateur prévu par la préversion du logiciel EDGE).

Ces limitations du langage de description (non prise en compte des hétérogénéités surfaciques, un seul opérateur d'affectation, absence de types structurés) nous ont obligé à "appauvrir" les expertises dont nous disposions. Mais ce langage nous a quand même permis (cf. annexe) de retrouver des correspondances entre descriptions symboliques et descriptions numériques, ainsi que quelques règles élémentaires d'applicabilité de certaines filières d'assainissement autonome. Il faut toutefois noter que ces essais d'apprentissage ont été effectués sur une base très restreinte (environ 30 exemples seulement) pour pouvoir en tirer des conclusions intéressantes du point de vue du domaine d'application.

La réalisation du système Moïse, et notamment son application à l'assainissement autonome des eaux usées, se poursuit avec le soutien de la Région Rhône-Alpes, dans le cadre d'un Programme Pluriannuel de Recherche (Intelligence Artificielle, Système Experts et Applications), et en collaboration avec la société 3ASI.

D'autres applications de système d'acquisition interactive de la connaissance peuvent être envisagées. A l'Ecole des Mines de Saint-Etienne, nous étudions d'ores et déjà l'entretien incrémental d'une base de compétences d'experts en pollution des eaux marines et continentales.

A priori, la consultation d'une telle base se rapproche de celle d'un système de Recherche Documentaire, mais dans lequel la base d'informations évolue dans le temps, en fonction de l'évolution des concepts qui peuvent décrire les compétences. La description d'un (ou plusieurs) domaine scientifique, telle que peut la faire un spécialiste, ne suffit pas pour décrire et gérer des compétences d'experts. Certains liens interdisciplinaires, nécessaires à une expertise efficace, ne peuvent être découverts qu'en examinant des interventions réelles d'experts (des interventions réussies ou ayant conduit à un échec, peu importe). Cet entretien de la base de compétences doit donc pouvoir s'effectuer à partir de cas réels : les concepts, et les dépendances, à découvrir sont davantage liés à la notion d'interventions, qui sont concrétisées sous forme de cas réels, qu'à la notion de disciplines, ou sous-disciplines, scientifiques.

Conclusion

L'objectif initial de notre travail était de montrer la faisabilité d'un transfert de connaissance depuis des experts vers un système informatique, dans un domaine où la connaissance experte est dispersée et évolutive. Le traitement simultané de l'évolution du domaine et de l'éventualité de données bruitées nous a conduit à rejeter l'approche statistique, couramment employée en apprentissage automatique. A la place, nous adoptons un mécanisme de dialogue entre le système et l'expert, lorsque l'apprentissage automatique se révèle incapable de décider, de façon sûre, de la marche à suivre.

Dans ce travail nous proposons donc un moyen d'acquisition de la connaissance, pour les cas où la connaissance experte est dispersée et/ou évolutive. Pour ces cas-là, les méthodes analytiques, qui font intervenir un intermédiaire humain, se montrent inefficaces. De même, les méthodes d'apprentissage automatique, tant qu'elles ne font que de l'analyse (symbolique ou numérique) de données, sont inefficaces car elles ne parviennent pas à une extraction complète de la connaissance des experts : elles se limitent aux informations que les experts pensent à fournir.

Ainsi, dans une première phase, notre système apprend automatiquement, à partir de cas réels résolus par les experts, jusqu'à ce qu'un conflit surgisse entre une information fournie par un expert et la connaissance apprise auparavant. A ce stade-là, le système ne peut décider de lui-même, s'il faut douter de certaines règles apprises, ou s'il faut douter de l'information qui a fait surgir le conflit. Il ne peut décider s'il s'agit d'une évolution du domaine, ou bien si un expert a oublié de lui fournir certaines informations.

Dans la deuxième phase d'apprentissage, le système dialogue alors avec l'expert pour lui demander de résoudre le conflit : pour cela le système fournit à l'expert certaines informations, et le guide dans son choix. Notamment, si l'expert confirme l'information qui a fait surgir le conflit, le système peut montrer à l'expert tous les exemples précédemment appris qui vont être supprimés. Dans le cas où l'expert accepte l'idée d'un facteur caché, le système peut simuler l'existence de ce facteur caché pour aider l'expert à l'identifier.

Le système télématique Moïse, qui sera doté de ces mécanismes d'acquisition interactive de la connaissance, nous permettra donc de former une base de connaissance en assainissement autonome des eaux usées, représentative de l'expertise utilisée dans tous les départements français. Dans un premier temps, Moïse sera aussi capable de redistribuer cette connaissance, au fur et à mesure qu'elle se constituera, afin de simplifier la tâche des Directions Départementales des Affaires Sanitaires et Sociales, chargées de contrôler les systèmes d'assainissement des eaux usées.

Dans un deuxième temps, cette base de connaissance pourra aussi être utile au Ministère de l'Environnement et au Ministère des Affaires Sanitaires et Sociales, dans lesquels des efforts sont actuellement entrepris pour réaliser une unification des concepts mis en œuvre dans les expertises en assainissement des eaux usées, et qui devraient déboucher à terme sur une nouvelle réglementation plus précise.

Mais cette base de connaissance est aussi destinée à alimenter la Station de Travail Intelligente de l'aménageur en eau (d'autres systèmes similaires devront être réalisés pour recueillir les expertises utilisées dans d'autres sous-domaines de l'aménagement en eau). Pour cela nous avons défini une architecture informatique de la Station de Travail qui distingue plusieurs niveaux de connaissances (connaissance de surface, connaissance statique, et connaissance de stratégie). La connaissance de surface, connaissance mise en œuvre dans la réalisation d'une tâche particulière, est acquise au moyen du système télématique.

En ce qui concerne l'acquisition de la connaissance de stratégie, qui spécifie la résolution globale d'un problème général d'aménagement en eau, nous proposons la réalisation d'un simulateur de projet, qui pourra aussi être utilisé comme système d'EIAO (Enseignement Intelligemment Assisté par Ordinateur), et dont l'architecture informatique sera très proche de celle de la Station de Travail elle-même. La méthode d'acquisition de la connaissance à adopter pour cet outil fera l'objet de travaux futurs dans notre laboratoire, mais on peut d'ores et déjà avancer qu'elle sera à base d'apprentissage à partir d'exemples, et qu'un dialogue avec les experts y jouera un rôle primordial.

Enfin, nous avons scindé la connaissance statique (c'est à dire la connaissance qui décrit le domaine d'expertise) en deux parties : d'un côté la connaissance statique générale, c'est à dire celle relative à la résolution d'un problème général d'aménagement en eau, et d'un autre côté, la connaissance statique spécifique, c'est à dire celle relative à sous-domaine particulier de l'aménagement en eau. Son acquisition est donc réalisée en même temps que l'acquisition des deux autres types de connaissances. Pour la connaissance statique générale, elle sera acquise par le simulateur, au travers du dialogue avec l'expert, lorsqu'il décrira de nouveaux objets. De même, la connaissance statique spécifique sera acquise par le système télématique lorsque l'expert décrira de nouveaux descripteurs qui lui permettent de décrire la situation pour laquelle il prend une décision.

L'originalité de notre approche de l'acquisition de la connaissance tient dans la place que le système réserve aux experts : d'un côté, il ne leur fait pas confiance et prévoit que certaines erreurs puissent être commises par les experts, d'un autre côté il utilise au maximum leur capacité de réflexion sur leur propre expertise pour corriger ces éventuelles erreurs, ainsi que pour prendre en compte les évolutions du domaine.

Toutefois, les aides du système à l'extraction de la connaissance pourront être améliorées, notamment dans la recherche des problèmes que le système ne sait pas résoudre. Pour l'instant, nous n'avons prévu que les problèmes soumis par des consultants, et que le système n'a pas su résoudre, ainsi que les problèmes comprenant un descripteur dont le domaine de connaissance, par le système, est inférieur au domaine de définition.

Cette recherche de lacunes dans une base de connaissance, et la constitution d'exemples à faire résoudre par un expert, pourraient être appliquées à "l'apprentissage à partir de simulations",

afin de pouvoir fabriquer un macro-modèle d'un système complexe pour lequel on ne possède qu'un modèle descriptif fin, qui sert de base à la simulation.

En effet, dans certains domaines, des décisions sont prises après consultation d'un simulateur du système réel mais, malheureusement, ces simulations peuvent être longues et ne sont alors d'aucune utilité lorsqu'il s'agit de prendre des décisions urgentes. Il est donc intéressant, dans ces cas-là, d'avoir recours à un modèle "plus rapide" du système réel. Le problème qui se pose alors est de réaliser ce macro-modèle : une solution consiste à coupler un système d'apprentissage au simulateur :

- le système d'apprentissage est chargé de fabriquer des cas qu'il ne sait pas résoudre, puis de les proposer au simulateur,
- en retour, le simulateur fournit au système d'apprentissage les résultats de sa simulation en termes d'indicateurs, à partir desquels le système d'apprentissage peut enrichir sa base de connaissance.

Un tel système peut avoir des applications très importantes dans les domaines où la simulation joue un grand rôle, car elle est la seule source de connaissance du système réel (autre que le système réel lui-même) : c'est le cas, par exemple, en Gestion de la Production.

Bibliographie

- [Albert-Vogel 90] **ALBERT P. et VOGEL C.** - KOD-STATION : un environnement intégré pour le génie cognitif - *Génie logiciel et systèmes experts* - 1990 - n° 19 - pp. 28-30.
- [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 & al. 88] **AYEL M., CHEIN M., PIPARD E. et ROUSSET M.-C.** - De la cohérence dans les bases de connaissances - *Journées Nationales Intelligence Artificielle* - Toulouse : Teknea - 1988 - pp. 357-384.
- [Ayel-Rousset 90] **AYEL M. et ROUSSET M.-C.** - *La cohérence dans les bases de connaissances* - Toulouse : Cepadues-Editions - 1990 - 106 p.
- [Beaune-Graillet 90] **BEAUNE P. et GRAILLOT D.** - Un "Serveur Expert" utilisant l'apprentissage automatique - *ERGO-IA'90* - Biarritz : AFCET, SELF et IDLS - 1990 - pp. 294-303.
- [Boucheron & al. 86] **BOUCHERON S., QUINQUETON J., SALLANTIN J. et SOLDANO H.** - CALM : Contester pour Apprendre en Logique Modale - *Système d'apprentissage CALM* - Montpellier : CRIM - 1986 - 89 p. - Rapport sur le Projet "Apprentissage" du G.S. DIALR.
- [Boy-Faller-Sallantin-88] **BOY G., FALLER B. et SALLANTIN J.** - Acquisition et ratification de connaissances - *Journées Nationales Intelligence Artificielle* - Toulouse : Teknea - 1988 - pp. 321-356.
- [Brodhag 90] **BRODHAG, C.** - *Objectif terre, Les Verts, de l'écologie à la politique* - Paris : éditions du félin - 1990 - 325 p.

- [Brunet-Dorbes 90] **BRUNET E. et DORBES G.** - KADS & MERISE vers une unification du génie cognitif et du génie logiciel - *Génie logiciel et systèmes experts* - 1990 - n° 19 - pp. 10-27.
- [Carbonell & al. 84] **CARBONELL J.G., MICHALSKI R.S. et MITCHELL T.M.** - An overview of Machine Learning - *Machine Learning : An Artificial Intelligence Approach* - Michalski, Carbonell et Mitchell - Berlin : Springer-Verlag - 1984 - pp. 3-23.
- [Carbonell 89] **CARBONELL** - Paradigms for Machine Learning - *Artificial Intelligence* - 1989 - vol. 40 - pp. 1-9.
- [Centre Royaumont 79] **CENTRE ROYAUMONT POUR UNE SCIENCE DE L'HOMME** - *Théories du langage, Théories de l'apprentissage* - Paris : Seuil - 1979 - 533 p.
- [Cohen-Feigenbaum 82] **COHEN P.R. et FEIGENBAUM E.A.** - *The Handbook of Artificial Intelligence* - Los Altos : W. Kauffmann - 1982 - vol.3 - 639 p.
- [Cres 89] **CRES F.N.** - *Contribution des systèmes à bases de connaissance en sciences de l'eau; Promise : un simulateur de projet; Moïse : un système de diagnostic en assainissement autonome* - Thèse de Doctorat : Ecole Nationale Supérieure des Mines de Saint-Etienne - 1989 - 279 p.
- [Decaestecker 89a] **DECAESTECKER C.** - *ADECLU : an Incremental Conceptual Clustering System* - Bruxelles : CADEPS, Université Libre de Bruxelles - 1989 - 25 p. - Rapport Technique I.A.-06-89
- [Decaestecker 89b] **DECAESTECKER C.** - Incremental Concept Formation via a Suitability Criterion - *Data Analysis, Learning Symbolic and Numeric Knowledge* - New York : Nova Science Publishers, Inc. - 1989 - pp. 435-442.
- [Delmas-Vernet 88] **DELMAS D. et VERNET P.** - *Intelligence artificielle et Vidéotex : état de l'art et prospective* - Paris : EC2 - 1988 - 238 p.
- [Deslisle-Lefevre 91] **DELISLE F. et LEFEUVRE J.C.** - Le gavage des eaux - *Sciences et Avenir* - 1991 - Hors série - n° 83 - pp. 7-10.

- [Dieng 90] **DIENG R.** - Méthodes et outils d'acquisition des connaissances - *ERGO-IA'90* - Biarritz : AFCET, SELF et IDLS - 1990 - pp. 245-271.
- [Dietterich-Michalski 84] **DIETTERICH T.G.** et **MICHALSKI R.S.** - A Comparative Review of Selected Methods for Learning from examples - *Machine Learning : An Artificial Intelligence Approach* - Michalski, Carbonell et Mitchell - Berlin : Springer-Verlag - 1984 - pp. 41-81.
- [Dubreuil 90] **DUBREUIL P.** - *Rapport d'évaluation du potentiel français en hydrologie* - Paris : secrétariat d'Etat auprès du Premier Ministre chargé de l'Environnement et de la Prévention des Risques technologiques et naturels majeurs, Ministère de la Recherche et de la Technologie - 1990 - 72 p.
- [Ermine 89] **ERMINE J.L.** - *Systèmes experts : théorie et pratique* - Cachan : Lavoisier - 1989 - 221 p.
- [Farreny 85] **Farreny H.** - *Les Systèmes Experts : principes et exemples* - Toulouse : Cepadues-Editions - 1985 - 254 p.
- [Farreny-Ghallab 87] **FARRENY H.** et **GHALLAB M.** - *Eléments d'intelligence artificielle* - Paris : Hermès - 1987 - 363 p.
- [Fisher 87] **FISHER D.H.** - Knowledge Acquisition Via Incremental Conceptual Clustering - *Machine Learning* - 1987 - vol. 2 - pp. 139-172.
- [Ganascia 88] **GANASCIA J.G.** - CHARADE : une sémantique cognitive pour les heuristiques d'apprentissage - *Huitièmes Journées Internationales, Les Systèmes Experts et leurs Applications, Conférence Générale* - Avignon : EC2 - 1988 - pp. 567-585.
- [Gascuel-Caroux 91] **GASCUEL O.** et **CARAUX G.** - Sur l'interprétation statistique des résultats en apprentissage - *6ièmes Journées Françaises de l'Apprentissage* - Sète : CJN-IA, PRC-GDR IA et LIRMM - 1991 - pp. 186-200.

- [Gennari & al. 89] **GENNARI J.H., LANGLEY P. et FISHER D.** - Models of Incremental Concept Formation - *Artificial Intelligence* - 1989 - vol. 40 - pp. 11-61.
- [Graillet 86] **GRAILLOT D.** - *Faisabilité d'un système d'ingénierie pour la réalisation de projets d'aménagement en eau à partir du modèle de simulation : MISE* - Thèse de Doctorat d'Etat : Université des Sciences et Techniques du Languedoc - 1986 - 507 p.
- [Habrias 88] **HABRIAS H.** - *Le modèle relationnel binaire, Méthode I.A. (NIAM)* - Paris : Eyrolles - 1988 - 302 p.
- [Hart 88] **HART A.** - *Acquisition du savoir pour les systèmes experts* - Paris : Masson - 1988 - 142 p.
- [Hayes-Roth-McDermott 78] **HAYES-ROTH F. et MAC DERMOTT J.** - An interference matching technique for inducing abstractions - *CACM 26* - 1978 - pp. 401-410.
- [Hofstadter 85] **HOFSTADTER D.** - *Gödel Escher Bach : les Brins d'une Guirlande Eternelle* - Paris : InterEditions - 1985 - 884 p.
- [Hyckman & al. 89] **HYCKMAN F.R., KILLIN J.L., LAND L., MULHALL T., PORTER D. et TAYLOR R.M.** - *Analysis for Knowledge-Based Systems, A Practical Guide to the KADS Methodology* - Chichester : Ellis Horwood Limited - 1989 - 189 p.
- [Kassel 88] **KASSEL G.** - Deux directions de recherche pour l'explication du raisonnement - *Huitièmes Journées Internationales, Les Systèmes Experts et leurs Applications* - Avignon : EC2 - 1988 - vol. 3 - pp. 305-318.
- [Kassel 89] **KASSEL G.** - Systèmes Experts de Seconde Génération et explication du raisonnement - *Neuvièmes Journées Internationales, Les Systèmes Experts et leurs Applications, Conférence Spécialisée : Systèmes Experts de Seconde Génération* - Avignon : EC2 - 1989 - pp. 223-234.
- [Kodratoff 86] **KODRATOFF Y.** - *Leçons d'apprentissage symbolique automatique* - Paris : Cepadues-Editions - 1986 - 191 p.

- [Lakatos 84] **LAKATOS I.** - *Preuves et réfutations, essais sur la logique de la découverte mathématique* - Paris : Hermann - 1984 - 218 p.
- [Lalonde 91] **LALONDE B.** - Les espoirs de la science - *Sciences et Avenir* - 1991 - Hors série - n° 83 - pp. 4-5.
- [Lebowitz 86] **LEBOWITZ M.** - Concept Learning in a Rich Input Domain : Generalization-Based Memory - *Machine Learning : An Artificial Intelligence Approach, volume II* - Michalski, Carbonell et Mitchell - Los Altos : W. Kauffmann - 1986 - pp. 193-214.
- [Lebowitz 87] **LEBOWITZ M.** - Experiments with Incremental Concept Formation : UNIMEM - *Machine Learning* - 1987 - vol. 2 - pp. 103-138.
- [Lesaffre & al. 89] **LESAFFRE F.M., CORSI P., LAPICQUE Y., MANAGO M., CONRUYT N., BLYTHE J. et BLANCARD D.**- Acquérir des connaissances à l'aide de l'apprentissage - *Neuvièmes Journées Internationales, Les Systèmes Experts et leurs Applications, Conférence Générale* - Avignon : EC2 - 1989 - pp. 165-182.
- [Liquière-Mephu Nguifo 90] **LIQUIERE M. et MEPHU NGUIFO E.** - LEGAL (LEarning with GALois Lattice). Un système d'apprentissage de concepts à partir d'exemples - *5ème Journées Françaises de l'Apprentissage, Lannion* - Lannion : CNET, PRC-GDR IA et France Télécom - 1990 - pp. 93-113.
- [Manago-Kodratoff 87] **MANAGO M.V. et KODRATOFF Y.** - Noise and knowledge Acquisition - *Tenth International Joint Conference on Artificial Intelligence, Milan* - Los Altos : W. Kaufmann - 1987 - pp. 348-354.
- [Michalski 84] **MICHALSKI R.S.** - A Theory and Methodology of Inductive Learning - *Machine Learning : An Artificial Intelligence Approach* - Michalski, Carbonell et Mitchell - Berlin : Springer-Verlag - 1984 - pp. 83-134.

- [Michalski 86] **MICHALSKI R.S.** - Understanding the Nature of Machine Learning : Issues and Research Directions - *Machine Learning : An Artificial Intelligence Approach, volume II* - Michalski, Carbonell et Mitchell - Los Altos : W. Kauffmann - 1986 - pp. 3-25.
- [Mitchell 78] **MITCHELL T.M.** - *Version Spaces : An approach to concept learning* - PhD thesis : Stanford University - 1978.
- [Mitchell 82] **MITCHELL T.M.** - Generalization as Search - *Artificial Intelligence* - 1982 - vol. 18 - pp. 203-226.
- [Nicolas 87] **NICOLAS J.** - *ALLY : un système logique pour la généralisation en apprentissage automatique* - Thèse de Doctorat : Université de Rennes I - 1987 - 95 p.
- [Pottier 86] **POTTIER L.** - Généralisations d'exemples et de contre-exemples en calcul propositionnel - *Colloque International d'Intelligence Artificielle, Marseille* - Paris : Hermès - 1986 - pp. 461-471.
- [Quinlan 86] **QUINLAN J.R.** - Induction of Decision Trees - *Machine Learning* - 1986 - vol. 1 - pp. 81-106.
- [Reitz 92] **REITZ P.** - Thèse de Doctorat : Université des Sciences et Techniques du Languedoc - 1992 - à paraître.
- [Reynaud 89] **REYNAUD C.** - Une aide à l'acquisition des connaissances de surface à partir des connaissances profondes - *Neuvièmes Journées Internationales, Les Systèmes Experts et leurs Applications, Conférence Spécialisée : Systèmes Experts de Seconde Génération* - Avignon : EC2 - 1989 - pp. 207-222.
- [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 - 155 p.
- [Russel-Grosof 87] **RUSSEL S.J. et GROSOFF B.N.** - A Declarative Approach to Bias in Concept Learning - *6th AAAI* - 1987 - pp. 505-510.

- [Sainte Marie 89] **DE SAINTE MARIE C.** - Dépendance Perçue : une introduction - *Reconnaissance des Formes et Intelligence Artificielle, Paris* - Paris : AFCET et INRIA - 1989 - pp. 1445-1459.
- [Sainte Marie 90a] **DE SAINTE MARIE C.** - Apprentissage et acquisition de la connaissance - *Convention IA, Paris* - Paris : Hermès - 1990 - pp. 521-535.
- [Sainte Marie 90b] **DE SAINTE MARIE C.** - De l'observation à la connaissance empirique (en passant par la Dépendance Perçue) - *5ème Journées Françaises de l'Apprentissage, Lannion* - Lannion : CNET, PRC-GDR IA et France Télécom - 1990 - pp. 251-272.
- [Sainte Marie 91] **DE SAINTE MARIE C.** - Thèse de Doctorat : Institut National Polytechnique de Grenoble - 1991 - à paraître.
- [Sallantin et al. 91] **SALLANTIN J., SZCZECINIARZ J.J., BARBOUX C., LAGRANGE M.L. et RENAUD M.** - Théories semi-empiriques : conceptualisation et illustrations - *Revue d'Intelligence Artificielle* - 1991 - vol. 5 - n° 1 - pp. 9-67.
- [SESG 89] **DAVID J.M., KRIVINE J.P. et SIMMONS R.** - Préface des actes - *Neuvièmes Journées Internationales, Les Systèmes Experts et leurs Applications, Conférence Spécialisée : Systèmes Experts de Seconde Génération* - Avignon : EC2 - 1989 - pp. 4-6.
- [Steels 88] **STEELS L.** - The Deepening of Experts Systems - *IMACS'88, PARIS* - 1988 - pp. 323-326.
- [Tanenbaum 87] **TANENBAUM A.S.** - *Operating Systems : Design and Implementation* - Englewood Cliffs : PrenticeHall - 1987 - 719 p.
- [Utgoff 86] **UTGOFF P.E.** - Shift of Bias for Inductive Concept Learning - *Machine Learning : An Artificial Intelligence Approach, volume II* - Michalski, Carbonell et Mitchell - Los Altos : W. Kauffmann - 1986 - pp. 107-148.

- [Vere 75] **VERE S.A.** - Induction of concepts in the predicate calculus - *IJCAI* - 1975 - pp. 281-287.
- [Vogel 88] **VOGEL C.** - *Génie cognitif* - Paris : Masson - 1988 - 196 p.

Annexe A

Intégration d'un exemple à un RDP

Nous reprenons, dans cette annexe, le RDP décrit en figure 4.5 : il est issu des quatre exemples du § 4.3.1.1, et son contenu est décrit au § 4.3.1.2. Nous allons détailler les opérations qui permettent l'intégration d'un nouvel exemple, selon l'algorithme décrit en figures 4.6 et 4.7. L'exemple intégré au RDP est celui présenté au § 4.3.2.2 :

*e = (filière = épandage en tranchées et
pente = faible et
épaisseur = moyenne et
perméabilité = faible).*

Nous obtiendrons donc, en fin de traitement, le RDP de la figure 4.8.

Les fonctions de généralisations retenues sont, pour le descripteur nominal, la variabilisation, et, pour les descripteurs linéaires, la fermeture d'intervalle (cf. § 4.3.2.1).

A.1 Création du nouveau nœud :

L'algorithme commence par la création du nouveau nœud N. Ici ce nœud sera nommé n12, pour rester en cohérence avec les notations des figures 4.5 et 4.8.

$$f_{n12} \leftarrow \{ \text{filière}=\text{épan dage en tranchées}, \text{pente}=\text{faible}, \text{épaisseur}=\text{moyenne}, \text{perméabilité}=\text{faible} \}$$

A.2 Parcours topologique du RDP :

On va donc maintenant parcourir tout le RDP, en largeur d'abord, en commençant par la racine, et en n'examinant un nœud qu'après avoir examiné tous ses fils.

A.2.1 Examen du nœud n1 :

$G \leftarrow G(f_{n12}, f_{n1}) = \{ \text{filière}=\text{quelconque}, \text{faible} \leq \text{pente} \leq \text{moyenne}, \text{très faible} \leq \text{épaisseur} \leq \text{moyenne}, \text{faible} \leq \text{perméabilité} \leq \text{très forte} \}$. Seule la dernière formule a été généralisée.

$F = \emptyset$ car n12 n'étant relié à aucun nœud, il n'a aucun successeur.

$G - F = G \neq \emptyset$ et $G \neq f_{n1}$ car la perméabilité de n1 a été généralisée.

On crée donc un nœud m que l'on nommera n13.

$$f_{n13} \leftarrow f_{n1} - G = \{ \text{moyenne} \leq \text{perméabilité} \leq \text{très forte} \}$$

$$f_{n1} \leftarrow G = \{ \text{filière}=\text{quelconque}, \\ \text{faible} \leq \text{pente} \leq \text{moyenne}, \\ \text{très faible} \leq \text{épaisseur} \leq \text{forte}, \\ \text{faible} \leq \text{perméabilité} \leq \text{très forte} \}$$

n13 est relié aux pères de n1, c'est à dire à n3, n5 et n8, et les liens entre n1 et ses pères sont détruits; n1 et n13 sont reliés.

$f_{n12} \leftarrow f_{n12} - G = f_{n12}$,
car f_{n12} et G n'ont aucune formule en commun.

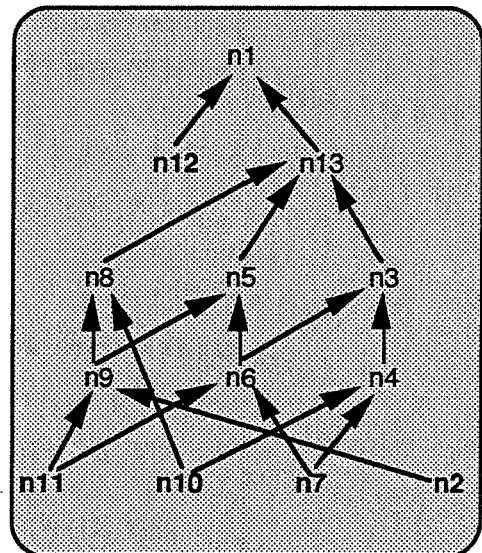


fig. A.1 : RDP après examen de n1

La procédure mettre-à-jour- $\epsilon_{RDP}(f_{n12}, f_{n1})$ ne fait qu'ajouter le lien (n12,n1) à ϵ_{RDP} car f_{n12} et f_{n1} n'ont, ni l'un ni l'autre, de fils.

A.2.2 Examen du nœud n13 :

$\mathcal{G} \leftarrow \mathcal{G}(f_{n12}, f_{n13}) = \{\text{faible} \leq \text{perméabilité} \leq \text{très forte}\}$ et $\mathcal{F} \leftarrow f_{n1}$ donc $\mathcal{G} - \mathcal{F} = \emptyset$:

on ne passe donc pas dans la première boucle.

$\mathcal{G} \neq f_{n13}$:

on ne passe donc pas non plus dans la deuxième boucle : rien n'est modifié dans le RDP.

On passe donc à l'examen de n3 (n12 n'est jamais examiné).

A.2.3 Examen du nœud n3 :

$\mathcal{G} \leftarrow \mathcal{G}(f_{n12}, f_{n3})$ donc :

$\mathcal{G} = \{\text{très faible} \leq \text{épaisseur} \leq \text{moyenne}\}$, et

$\mathcal{F} \leftarrow f_{n1}$ car n1 est le seul successeur de n12,

donc $\mathcal{G} - \mathcal{F} \neq \emptyset$ mais $\mathcal{G} = f_{n3}$: on ne passe donc pas dans la première boucle.

$\mathcal{G} = f_{n3}$: on passe dans la deuxième.

$f_{n12} \leftarrow f_{n12} - \mathcal{G} = f_{n12}$, car f_{n12} et \mathcal{G} n'ont aucune formule en commun.

$f_{n12} \leftarrow f_{n12} - \mathcal{G} = f_{n12}$,

car f_{n12} et \mathcal{G} n'ont aucune formule en commun.

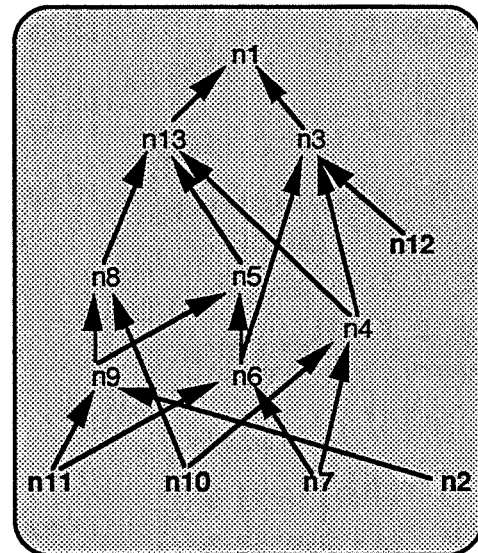


fig. A.2 : RDP après examen de n3

La procédure mettre-à-jour- $\mathcal{E}_{\text{RDP}}(f_{n12}, f_{n3})$ supprime le lien (n3, n13) puis ajoute (n4, n13), (n3, n1), et supprime (n12, n1) puis ajoute (n12, n3).

A.2.4 Examen du nœud n4 :

$\mathcal{G} \leftarrow \mathcal{G}(f_{n12}, f_{n4})$ donc : $\mathcal{G} = \{\text{filier} = \text{quelconque}, \text{faible} \leq \text{pente} \leq \text{moyenne}\}$.

$\mathcal{F} \leftarrow f_{n1} \cup f_{n3}$, donc $\mathcal{G} - \mathcal{F} = \emptyset$: on ne passe donc pas dans la première boucle.

$\mathcal{G} \neq f_{n4}$: on ne passe pas non plus dans la deuxième.

A.2.5 Examen du nœud n5 :

$\mathcal{G} \leftarrow \mathcal{G}(f_{n12}, f_{n5})$ donc :

$\mathcal{G} = \{ \text{moyenne} \leq \text{épaisseur} \leq \text{forte} \}$.

$\mathcal{F} \leftarrow f_{n1} \cup f_{n3}$, donc $\mathcal{G} - \mathcal{F} \neq \emptyset$, et

$\mathcal{G} = f_{n5}$: on ne passe pas dans la première boucle.

$\mathcal{G} = f_{n5}$: on passe dans la deuxième :

$f_{n12} \leftarrow f_{n12} - \mathcal{G} = f_{n12}$, car f_{n12} et \mathcal{G} n'ont aucune formule en commun.

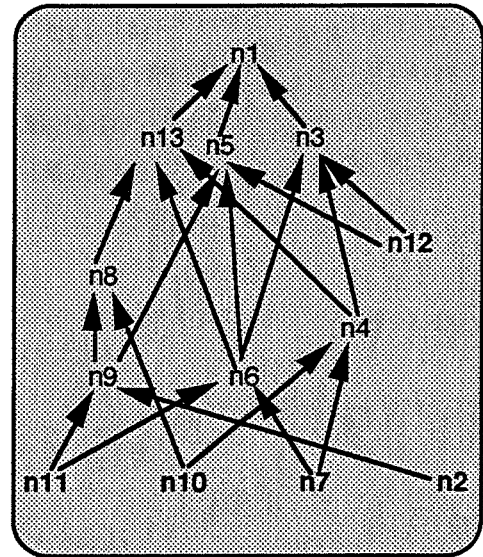


fig. A.3 : RDP après examen de n5

La procédure mettre-à-jour- $\epsilon_{RDP}(f_{n12}, f_{n5})$ supprime le lien (n5, n13), puis ajoute (n6, n13), (n5, n1), et (n12, n5).

A.2.6 Examen du nœud n8 :

$\mathcal{G} \leftarrow \mathcal{G}(f_{n12}, f_{n8})$ donc :

$\mathcal{G} = \{ \text{faible} \leq \text{perméabilité} \leq \text{moyenne} \}$.

$\mathcal{F} \leftarrow f_{n1} \cup f_{n3} \cup f_{n5}$, donc $\mathcal{G} - \mathcal{F} \neq \emptyset$, et

$\mathcal{G} \neq f_{n8}$: on passe dans la première boucle :

On crée le nœud n14 :

$f_{n14} \leftarrow f_{n8} - \mathcal{G} = \{ \text{perméabilité} = \text{moyenne} \}$

$f_{n8} \leftarrow \mathcal{G} = \{ \text{faible} \leq \text{perméabilité} \leq \text{moyenne} \}$

n9 et n10 sont reliés à n14, et les liens entre n8 et ses pères sont cassés; n8 et n14 sont reliés.

$f_{n12} \leftarrow f_{n12} - \mathcal{G} = f_{n12}$, car f_{n12} et \mathcal{G} n'ont aucune formule en commun.

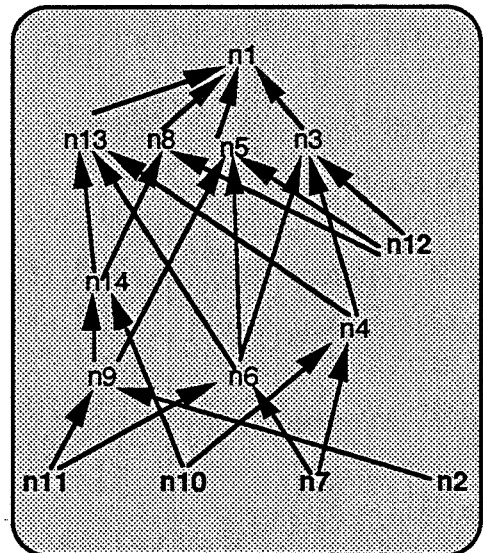


fig. A.4 : RDP après examen de n8

La procédure mettre-à-jour- $\epsilon_{RDP}(f_{n12}, f_{n8})$ supprime le lien (n8, n13), puis ajoute (n14, n13), (n8, n1), et (n12, n8).

A.2.7 Examen du nœud n6 :

$\mathcal{G} \leftarrow \mathcal{G}(f_{n12}, f_{n6})$ donc :

$\mathcal{G} = \{ \text{épaisseur=moyenne} \}$.

$\mathcal{G} = f_{n6}$: on ne passe pas dans la première boucle, mais on passe dans la seconde :

$f_{n12} \leftarrow f_{n12} - \mathcal{G}$ donc

$f_{n12} = \{ \text{filière=épanchage en tranchées, pente=faible, perméabilité=faible} \}$.

La procédure mettre-à-jour- $\epsilon_{RDP}(f_{n12}, f_{n6})$ supprime les liens (n6, n13), (n12, n3), (n12, n5), puis ajoute (n12, n6).

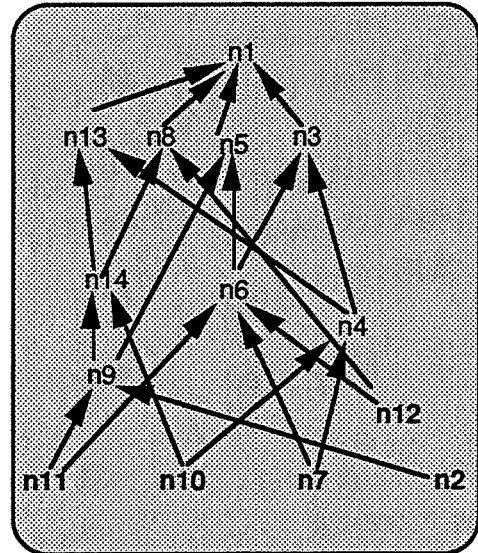


fig. A.5 : RDP après examen de n6

A.2.8 Examen du nœud n14 :

$\mathcal{G} \leftarrow \mathcal{G}(f_{n12}, f_{n14})$ donc : $\mathcal{G} = \{ \text{faible} \leq \text{perméabilité} \leq \text{moyenne} \}$.

$\mathcal{F} \leftarrow f_{n1} \cup f_{n3} \cup f_{n5} \cup f_{n6} \cup f_{n8}$, donc $\mathcal{G} - \mathcal{F} = \emptyset$: on ne passe donc pas dans la première boucle. $\mathcal{G} \neq f_{n14}$: on ne passe pas non plus dans la deuxième.

A.2.9 Examen du nœud n9 :

$\mathcal{G} \leftarrow \mathcal{G}(f_{n12}, f_{n9})$ donc :

$\mathcal{G} = \{ \text{filière=épanchage en tranchées, pente=faible} \}$.

$\mathcal{G} = f_{n9}$: on ne passe pas dans la première boucle, mais on passe dans la seconde :

$f_{n12} \leftarrow f_{n12} - \mathcal{G}$ donc :

$f_{n12} = \{ \text{perméabilité=faible} \}$.

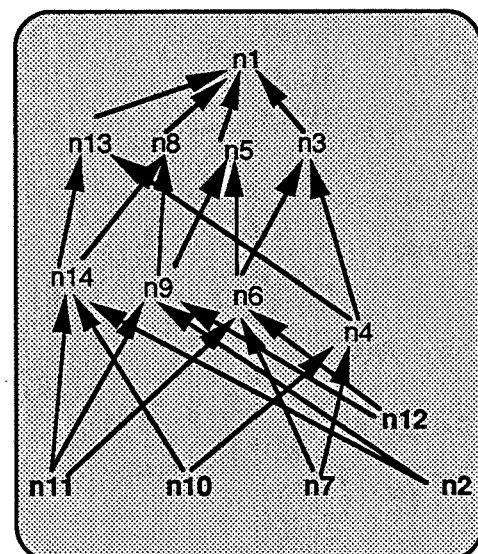


fig. A.6 : RDP après examen de n9

La procédure mettre-à-jour- $\mathcal{E}_{\text{RDP}}(f_{n12}, f_{n9})$ supprime le lien (n9, n14), puis ajoute (n11, n14), (n2, n14), (n9, n8), et supprime (n12, n8) et ajoute (n12, n9).

A.2.10 Examen des nœuds n2, n7, n10 et n11 :

Pour chacun de ces nœuds, nous obtenons : $\mathcal{G} - \mathcal{F} = \emptyset$, ou même $\mathcal{G} = \emptyset$, et $\mathcal{G} \neq \mathcal{f}$ ou $\mathcal{f} = \emptyset$, car ils n'ont rien en commun avec n12. Aucune autre modification n'est donc apportée au RDP de la figure A.6.

A.3 Vérification de l'existence d'un exemple identique :

En fin d'algorithme, il faut vérifier si un exemple identique a déjà été intégré au RDP. Si c'est le cas, alors, à la fin, $\mathcal{f}_{\text{N}} = \emptyset$.

Ici ce n'est pas le cas puisque : $\mathcal{f}_{n12} = \{ \text{perméabilité=faible} \}$.

Annexe B

La base d'expertises

Nous regroupons, dans cette annexe, quelques expertises qui nous ont servi à constituer notre base d'apprentissage. Elle sont issues d'expertises :

- de l'**ARASEE** (Association Rhône Alpes des Scientifiques pour l'Etude de l'Environnement),
- du Cabinet d'Etudes **GEOHYDROLOIRE**,
- de la **DDAF42** (Direction Départementale de l'Agriculture et des Forêts de la Loire), et
- d'**hydrogéologues agréés**.

Ces expertises sont exposées, dans cette annexe, telles qu'elles nous ont été fournies : seule une mise en page a été effectuée pour homogénéiser la présentation.

N° de l'expertise en assainissement autonome : 1

date de l'expertise : 1987

nom de l'expert : Cabinet Géohydroloire - ARASEE

Département : 42

Lieu : St Marcel de Félines - secteur Créonet - Croix de Bar

Nom du propriétaire : Jacquet A.

Filière préconisée : terre filtrant drainé

Paramètres de l'assainissement :

*** paramètres classiques :**

- pente du terrain : importante
- nature du sol :
- épaisseur du sol :
- perméabilité du sol :
- nature du substratum : roche mère
- profondeur du substratum : 0,2 m
- perméabilité du substratum :
- présence d'une nappe : sol hydromorphe
- profondeur de la nappe :
- présence d'un captage d'eau : non
- distance du dispositif au captage :
- existence d'un rejet superficiel : fossé d'une ancienne voie ferrée

*** paramètres spécifiques :**

- aptitude à l'assainissement par le sol en place : non
- taille de la parcelle :
- dispersion des effluents : oui
- traitement des effluents : oui
- utilisation du dispositif :

Remarques sur l'expertise :

existence d'un terrain inutilisable pour d'autres usages

circulation des véhicules ou autres impossible sur le terrain

N° de l'expertise en assainissement autonome : 6

date de l'expertise : 1987

nom de l'expert : Cabinet Géohydroloire - ARASEE

Département : 42

Lieu : St Marcel de Félines - secteur Créonet - Croix de Bar

Nom du propriétaire : Magat H.

Filière préconisée : terre filtrant drainé et étanchéifié ou lit filtrant

Paramètres de l'assainissement :

*** paramètres classiques :**

- pente du terrain : moyenne à forte
- nature du sol : terre végétale + arêne
- épaisseur du sol : faible à moyenne
- perméabilité du sol :
- nature du substratum :
- profondeur du substratum :
- perméabilité du substratum : moyenne
- présence d'une nappe : parfois traces d'hydromorphie
- profondeur de la nappe :
- présence d'un captage d'eau : non
- distance du dispositif au captage :
- existence d'un rejet superficiel : fossé de route

*** paramètres spécifiques :**

- aptitude à l'assainissement par le sol en place : non
- taille de la parcelle :
- dispersion des effluents : oui
- traitement des effluents : oui
- utilisation du dispositif : résurgences d'eaux usées actuellement observées avant installation

Remarques sur l'expertise :

N° de l'expertise en assainissement autonome : 9

date de l'expertise : 1987

nom de l'expert : Cabinet Géohydroloire - ARASEE

Département : 42

Lieu : St Marcel de Félines - secteur La Philipponière

Nom du propriétaire : Mathelin Y.

Filière préconisée : filtre à sable horizontal

Paramètres de l'assainissement :

*** paramètres classiques :**

- pente du terrain : pas très forte

- nature du sol :

- épaisseur du sol :

- perméabilité du sol : environ 0 mm/h

- nature du substratum :

- profondeur du substratum : 0,2 m

- perméabilité du substratum :

- présence d'une nappe : traces d'hydromorphie à 0,6m

- profondeur de la nappe :

- présence d'un captage d'eau : non

- distance du dispositif au captage :

- existence d'un rejet superficiel : pas de point de rejet en eau superficielle

*** paramètres spécifiques :**

- aptitude à l'assainissement par le sol en place : non

- taille de la parcelle :

- dispersion des effluents : oui

- traitement des effluents : oui

- utilisation du dispositif : réutilisation des eaux en sortie du filtre pour l'arrosage de pépinières

Remarques sur l'expertise :

N° de l'expertise en assainissement autonome : 13

date de l'expertise : 1987

nom de l'expert : Cabinet Géohydroloire - ARASEE

Département : 42

Lieu : St Marcel de Félines - secteur La Philipponière

Nom du propriétaire : système collectif.

Filière préconisée : filtre à sable drainé

Paramètres de l'assainissement :

*** paramètres classiques :**

- pente du terrain : importante risquant d'entraîner des résurgences
- nature du sol :
- épaisseur du sol :
- perméabilité du sol : environ 25 mm/h
- nature du substratum :
- profondeur du substratum : 1,4 m
- perméabilité du substratum :
- présence d'une nappe : hydromorphie à 0,6 m
- profondeur de la nappe :
- présence d'un captage d'eau : non
- distance du dispositif au captage :
- existence d'un rejet superficiel : ruisseau

*** paramètres spécifiques :**

- aptitude à l'assainissement par le sol en place : non
- taille de la parcelle : site en forme de cuvette
- dispersion des effluents : oui
- traitement des effluents : oui
- utilisation du dispositif :

Remarques sur l'expertise :

N° de l'expertise en assainissement autonome : 18

date de l'expertise : 1987

nom de l'expert : Cabinet Géohydroloire - ARASEE

Département : 42

Lieu : St Marcel de Félines - secteur Félines

Nom du propriétaire : Giroud J.

Filière préconisée : épandage par tranchées parallèles à la pente

Paramètres de l'assainissement :

*** paramètres classiques :**

- pente du terrain : faible à moyenne
- nature du sol :
- épaisseur du sol : > 1,50 m
- perméabilité du sol : environ 20 mm/h
- nature du substratum :
- profondeur du substratum :
- perméabilité du substratum :
- présence d'une nappe : non
- profondeur de la nappe :
- présence d'un captage d'eau : non
- distance du dispositif au captage :
- existence d'un rejet superficiel :

*** paramètres spécifiques :**

- aptitude à l'assainissement par le sol en place : oui
- taille de la parcelle : suffisante
- dispersion des effluents : oui
- traitement des effluents : oui
- utilisation du dispositif :

Remarques sur l'expertise :

N° de l'expertise en assainissement autonome : 21

date de l'expertise : 1985

nom de l'expert : Cabinet Géohydroloire

Département : 42

Lieu : St Romain le Puy (Plaine du Forez) - proximité source Parot

Nom du propriétaire : Cornet

Filière préconisée : filtre à sable horizontal drainé

Paramètres de l'assainissement :

*** paramètres classiques :**

- pente du terrain : particulièrement faible
- nature du sol : terre végétale sur 5 cm, sable feldspathique argileux jusqu'à 30cm, argile compacte jusqu'à 60 cm, sable argileux jusqu'à 75 cm
- épaisseur du sol : 75 cm
- perméabilité du sol : extrêmement faible (1,5 à 2,6 mm/h) jusqu'à 1m
- nature du substratum : argileux (argiles vertes), imperméable
- profondeur du substratum : >75 cm
- perméabilité du substratum : extrêmement faible
- présence d'une nappe : nappe superficielle captive et drainée en permanence et nappe profonde
- profondeur de la nappe : nappe superficielle : 1,5m (peu profonde),
nappe profonde : >2m (5m)
- présence d'un captage d'eau : oui, plusieurs
- distance du dispositif au captage : 200 à 300 m.
- existence d'un rejet superficiel : ruisseau bordant le CD8

*** paramètres spécifiques :**

- aptitude à l'assainissement par le sol en place : non
- taille de la parcelle :
- dispersion des effluents : oui
- traitement des effluents : oui
- utilisation du dispositif :

Remarques sur l'expertise :

influence aval d'une crue du ruisseau

N° de l'expertise en assainissement autonome : 23

date de l'expertise : 1986

nom de l'expert : Cabinet Géohydroloire - ARASEE

Département : 42

Lieu : Le Devey (Saint-Etienne) parcelles n° 3 et 4

Nom du propriétaire : Grandjon

Filière préconisée : épandage en tranchées

Paramètres de l'assainissement :

*** paramètres classiques :**

- pente du terrain : faible (d'Est en Ouest)
- nature du sol : remblai homogène constitué de sable et de machefer
faible épaisseur d'argile non constante
- épaisseur du sol : > 1,50 m
- perméabilité du sol : 30 mm/h à 70 mm/h
- nature du substratum : grès et schistes
- profondeur du substratum : 0 à 2 m
- perméabilité du substratum : faible
- présence d'une nappe : non
- profondeur de la nappe :
- présence d'un captage d'eau : non
- distance du dispositif au captage :
- existence d'un rejet superficiel :

*** paramètres spécifiques :**

- aptitude à l'assainissement par le sol en place : oui
- taille de la parcelle :
- dispersion des effluents : oui
- traitement des effluents : oui
- utilisation du dispositif :

Remarques sur l'expertise :

N° de l'expertise en assainissement autonome : 31

Informations générales :

date de l'expertise : 1986

nom et adresse de l'expert : DDAF 42 - Geohydroloire

Département : 42

Lieu : St. Romain les Atheux fond de vallon de la Valcherie

Nom du propriétaire : collectif

Filière préconisée : lagunage avec étanchéification par apport d'argile

Paramètres de l'assainissement :

*** paramètres classiques :**

- pente du terrain : élevée à l'amont, plus faible à l'aval, risques de résurgences
- nature du sol : terre végétale, sol brun sablo-limoneux perméable + blocs,
niveau d'argile gris-bleuté (1, 15 m),
niveau de sable micacé perméable (40mm/h) mais sec
- épaisseur du sol : au moins 1,5 m.
- perméabilité du sol : 20 mm/h entre 0,45 et 0,6 m., 10 à 35 mm/h entre 0,85 et 1,0 m,
nulle entre 1 et 1,15 m (niveau d'argile gris bleuté).
- nature du substratum : arène granitique argileuse au dessus du granite fissuré
- profondeur du substratum : au moins 1,5 m.
- perméabilité du substratum : moyenne
- présence d'une nappe : sol gorgé d'eau, végétation aquatique en surface, traces
d'hydromorphie importantes dans le niveau d'argile gris-bleuté, rares venues
d'eau dans le niveau de sable micacé
- profondeur de la nappe : 1,2 m par endroits
- présence d'un captage d'eau : non
- distance du dispositif au captage :
- existence d'un rejet superficiel : ruisseau

***paramètres spécifiques :**

- aptitude à l'assainissement par le sol en place : non
- taille de la parcelle : suffisante
- dispersion des effluents : oui
- traitement des effluents : oui
- protection de la nappe contre l'infiltration des eaux usées : oui

- utilisation du dispositif : permanente

Remarques sur l'expertise

cas limite et complexe faisant ressortir :

- l'homogénéité des terrains dans le plan et en vertical
- le volume d'argile disponible
- l'épaisseur de l'écran imperméable par rapport à la pente et donc l'interception de ce niveau par les terrassements
- l'existence d'écoulements superficiels

Ici, l'imperméabilisation sera réalisée par apport d'argile pour la placer là où il en manque

N° de l'expertise en assainissement autonome : 32

Informations générales :

date de l'expertise : 1987

nom et adresse de l'expert : DDAF 42 - Geohydroloire

Département : 42

Lieu : Gencenas (commune de Bessay), vallon du Vignard

Nom du propriétaire : collectif

Filière préconisée : lagunage avec étanchéification par apport d'argile

Paramètres de l'assainissement :

*** paramètres classiques :**

- pente du terrain : relativement forte (15%)
- nature du sol : homogénéité verticale mais pas horizontale
 - à l'ouest : sols sableux et très légers
 - au centre : sols sableux et limoneux et argileux en profondeur, hydromorphes
 - à l'est : sols sableux
- épaisseur du sol : à l'ouest et à l'est : peu épais et variable
 - au centre : plus épais
 - plus épais à l'amont qu'à l'aval
- perméabilité du sol : forte
 - à l'ouest : très perméables (180 mm/h)
 - au centre : perméabilité plus faible (10 à 70 mm/h)
 - à l'est : perméable
- nature du substratum : roche-mère anatexites à Cordiérite idem granite du Pilat
- profondeur du substratum : faible
 - à l'ouest : substratum très proche (entre 0,5m et 1,4m)
 - au centre : plus profond (entre 0,5m et 1,9 m.)
 - à l'est : plus proche (entre 0,7m et 1,4 m)
- perméabilité du substratum :
- présence d'une nappe : au centre : hydromorphie importante
- profondeur de la nappe :
- présence d'un captage d'eau : non
- distance du dispositif au captage :
- existence d'un rejet superficiel : ruisseau du Vignard

***paramètres spécifiques :**

- aptitude à l'assainissement par le sol en place : non
- taille de la parcelle : suffisante et acquise par la mairie
- dispersion des effluents : oui
- traitement des effluents : oui
- protection de la nappe contre l'infiltration des eaux usées : oui
- utilisation du dispositif : permanente

Remarques sur l'expertise

Un filtre à sable drainé serait possible mais risquerait de provoquer des écoulements sous-jacents nuisibles pour le mur de soutènement se situant à l'aval de la zone

Les sols en place doivent être éliminés étant donné leur qualité géotechniques médiocres

Annexe C

Quelques résultats d'apprentissage

Cette annexe présente quelques résultats de l'apprentissage à partir de la base d'expertises dont nous disposons (un peu plus d'une trentaine d'exemples). Cette base est, bien entendu, trop restreinte pour en tirer des conclusions intéressantes du point du domaine d'application : l'assainissement autonome des eaux usées. Mais nous avons retrouvé, néanmoins, quelques résultats concernant les correspondances symboliques/numériques, ainsi que quelques règles d'applicabilité de certaines filières d'assainissement.

C.1 Le RDP issu de la base d'expertises :

La figure C.1 montre le RDP construit à partir de la trentaine d'exemples que nous avons traitée. Ce RDP contient 363 nœuds, 480 formules atomiques et quelques formules complexes désignant des relations entre attributs (cf. § 4.3.2.3).

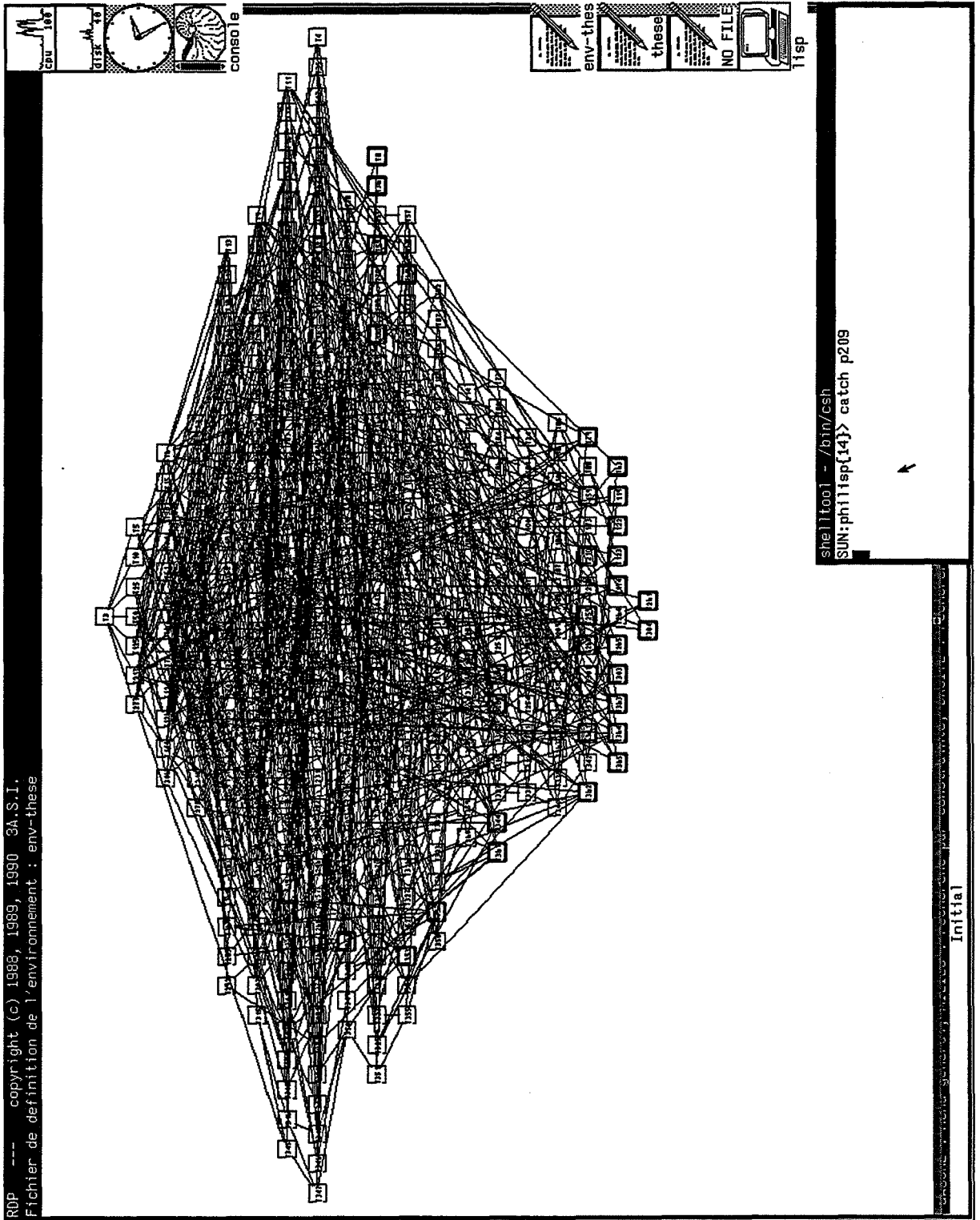


figure C.1 : le RDP issu de la base d'expertises

C.2 Correspondance symbolique/numérique :

Pour faire apparaître clairement les correspondances entre valeurs numériques et valeurs symboliques, nous choisissons un paramètre qui est souvent renseigné dans les expertises, tant numériquement que symboliquement : la perméabilité du sol, et nous extrayons ce paramètre des expertises pour construire un RDP ne contenant que ce paramètre (cf. figure C.2).

Dans ce RDP, nous remarquons l'existence d'exemples "haut placés" dans le RDP (représentés par les nœuds T5, T14, T69, T92 et T96) : ce sont les exemples pour lesquels n'est fournie qu'une perméabilité symbolique, et aucune perméabilité numérique.

Le nœud T5 contient **perméabilité = très faible**, et son père, le nœud T107 contient **perméabilité = 0**. On a donc la correspondance suivante :

$$\text{perméabilité} = 0 \Rightarrow \text{perméabilité} = \text{très faible} .$$

De même, avec les nœuds T14 et T27 on a :

$$\text{perméabilité} \in [5, 20] \Rightarrow \text{perméabilité} = \text{faible} .$$

Avec les nœuds T69 et T 110 :

$$\text{perméabilité} \in [25, 82] \Rightarrow \text{perméabilité} = \text{moyenne} .$$

Avec le nœud T92 : dans le seul exemple où **perméabilité = très forte.**, aucune perméabilité numérique n'est donnée. Nous n'obtenons donc aucune correspondance pour **perméabilité = très forte** .

Et enfin, avec les nœuds T96 et T91 :

$$\text{perméabilité} = 100 \Rightarrow \text{perméabilité} = \text{forte} .$$

Un grand nombre d'autres correspondances sont présentes dans ce RDP : par exemple l'arc reliant T90 à T95 nous indique :

$$\text{perméabilité} \in [25, 100] \Rightarrow \text{perméabilité} \in [\text{moyenne}, \text{forte}] .$$

RDP --- copyright (c) 1988, 1989, 1990 3A.S.I.
 Fichier de definition de l'environnement : env-these

```

shelltool - /bin/csh
-rwxr--r-- 1 philisp      62 Aug 13 16:41 llsp
SUN:philisp[16]> catch permea
  
```

Initial Read

cpu 100
 disk 60
 console
 llsp
 env-these
 these
 permea
 Llsp
 cd player
 /bin/csh

figure C.2 : le RDP representant les valeurs de permeabilite

C.3 Règle d'applicabilité d'une filière :

Parmi toutes les règles d'applicabilité des différentes filières, des règles simples sont facilement retrouvables.

Par exemple (nœuds T46 et T44) :

assainissement par le sol en place \neq non \Rightarrow filière = *épandage en tranchées*.

Ou bien (nœuds T282 et T11) :

profondeur de la nappe = *très faible* \Rightarrow filière = *tertre filtrant*

Ou encore (nœuds T48 et T11) :

**substratum = *roche mère* et profondeur du substratum \in *[0.2, 0.3]*
 \Rightarrow filière = *tertre filtrant***

Résumé

La réalisation d'un outil d'aide à la conduite de projets, dans le domaine de l'aménagement en eau, nécessite l'étude du transfert de la connaissance, depuis les experts du domaine, vers l'outil informatique. L'architecture informatique de cette Station de Travail Intelligente est présentée, avec comme paradigme, celui des Systèmes Experts de Seconde Génération : plusieurs niveaux de connaissance apparaissent qui nécessitent plusieurs formes d'acquisition de la connaissance.

Pour l'acquisition de la connaissance profonde, un simulateur de projets est utilisé; et un système télématique, à base d'apprentissage automatique à partir d'exemples, permet de traiter l'acquisition de la connaissance de surface. Cet apprentissage est effectué au moyen de la construction incrémentale d'un Réseau de Dépendances Perçues.

Néanmoins, nous n'exigeons pas du système télématique qu'il apprenne seulement à partir des exemples de problèmes résolus et fournis par les experts. Le système doit aussi être capable de dialoguer avec les experts de façon à faire valider la connaissance apprise. Ceci permet d'éviter l'introduction de certaines données erronées, tout en autorisant les changements de la base de connaissance dus à une éventuelle évolution du domaine (innovation technologique, évolution réglementaire, ...).

Mots clés

apprentissage symbolique automatique, acquisition de la connaissance, gestion de projets, simulation de projets, aménagement en eau, assainissement des eaux usées.