

MÉTHODES D'ACQUISITION DES CONNAISSANCES¹

Par Gilbert Paquette

SOMMAIRE

Introduction	1
1. Méthodologie de développement d'un système à base de connaissances	1
1.2 Étapes de développement d'un système à base de connaissances	2
1.3 Intervenants et leur rôle	4
2. Acquisition des connaissances : objectifs et moyens.....	5
2.1 Étude d'opportunité	5
2.2 Planification d'une démarche d'acquisition des connaissances.....	6
2.3 Identification des connaissances	7
2.4 Conceptualisation des connaissances	8
2.5 Formalisation des connaissances	8
3. Induction des règles à partir d'exemples.....	8
3.1 Processus d'induction.....	9
3.2 Fiabilité d'une induction et ensembles caractéristiques.....	11
3.3. Choix d'un ensemble représentatif d'exemples	12
3.4 Arbre des contraintes entre attributs	15
3.5 Ensemble des règles correspondant à une contrainte	16
3.6 Sommaire d'une méthode d'induction de règles.....	17
4. Techniques d'entrevue	18
4.1 Les objectifs de l'entrevue	18
4.2 Préparation d'une entrevue	19
4.3 Déroulement d'une entrevue.....	21
4.4 Analyse d'une entrevue.....	23
5. Identification et structuration des attributs par la grille répertoire	25
5.1 Qu'est-ce que la grille objets-attributs?	25
5.2 Construction de la grille objets-attributs	26
5.3 Groupement des cas et identification de nouveaux attributs	28
5.4 Épuration de la grille objets-attributs	30
5.5 Sommaire de la méthode avec la grille objets-attributs.....	32
5.6 Avantages et limites de l'explicitation par grille	32
6. Outils et techniques d'induction de règles	33
6.1 Survol des outils d'acquisition des connaissances	33
6.2 Tableur comme outil d'induction des règles.....	35
6.3 Induction automatique à partir d'exemples.....	39
Conclusion.....	41
À retenir.....	43

¹ Ce texte est extrait en majeure partie du volume G.Paquette et L. Roy, « Systèmes à base de connaissances, Télé-université et Beauchemin, pp.175-244

Introduction

Le but de ce texte est d'élaborer une **méthodologie de développement** de systèmes à base de règles (SBR).

Dès les premiers essais de réalisation d'applications de plus ou moins grande envergure, on se bute au problème fondamental du développement des SBR : **l'acquisition des connaissances**.

Étant donné que les SBR ont, entre autres, pour vocation de reproduire le mieux possible le comportement d'un expert, il devient donc nécessaire d'analyser ce comportement afin de le traduire en données que peut traiter l'ordinateur. Mais comment cerner la connaissance de l'expert? Comment la qualifier et la quantifier? Comment la traduire?

On peut croire que les experts eux-mêmes, sans formation préalable, sont capables de développer leurs propres applications. La réalité est tout autre. Certains logiciels, tels les « coquilles » de systèmes experts sont faciles à utiliser, mais ils sont peu utiles dans la solution du problème principal : identifier, organiser et traduire en termes informatiques les connaissances qui alimenteront un système à base de règles. Ce problème est précisément celui de l'acquisition et de la formalisation des connaissances.

Ce texte a pour but de présenter des éléments de solution à ce problème. L'approche consiste à systématiser les opérations à effectuer, à appliquer des méthodes éprouvées et à identifier des outils appropriés à chaque tâche. Cela revient à proposer une démarche ou une méthodologie d'acquisition des connaissances. D'ailleurs, *méthodologie* est le *mot clé* des prochains chapitres. Nous allons en mettre une à l'épreuve en la confrontant d'abord à des cas pratiques simples, puis nous finirons par l'appliquer à des situations plus complexes afin de mieux évaluer l'utilité.

1. Méthodologie de développement d'un système à base de connaissances

Une simple observation nous apprend que la mise au point des systèmes informatiques se fait selon des objectifs clairement formulés, une démarche planifiée et un recours à des méthodes et à des outils de travail appropriés. Il devrait en être ainsi pour le développement des systèmes à base de connaissances. Pour ce faire, on peut *transposer dans le domaine des SBC des méthodologies déjà éprouvées* dans d'autres secteurs de l'informatique. Toutefois, les particularités inhérentes aux systèmes à base de règles nous obligent à des *adaptations nécessaires* des méthodes et à l'emploi d'outils spécifiques¹. **1.1 Ingénierie de la connaissance²**

La spécialisation des connaissances et des méthodes d'une discipline est un phénomène caractéristique du développement de la plupart des domaines du savoir. L'histoire de l'informatique en est un bon exemple. Ainsi, on reconnaît que la programmation et l'analyse de systèmes sont des spécialités qui ont acquis le statut de discipline. Quant à l'intelligence artificielle, elle est née de la rencontre de plusieurs disciplines apparemment étrangères. Il était prévisible que son développement conduise à l'émergence de nouvelles spécialités pour devenir des disciplines autonomes. Le développement des systèmes à base de connaissances confirme cette tendance puisqu'il constitue un champ d'activités spécialisées, l'ingénierie de la connaissance. Plusieurs faits témoignent de cette réalité : vocabulaire spécialisé, littérature abondante, revues spécialisées, prolifération de logiciels adaptés et constitution de cercles de spécialistes appelés cognitivistes ou ingénieurs de la connaissance. Au point de vue

² Les termes ingénierie du savoir, génie logiciel, génie cognitif et *knowledge engineering* tiennent parfois lieu de synonymes.

sociologique, on peut donc prétendre que l'ingénierie de la connaissance constitue bel et bien une discipline.

L'ingénierie de la connaissance étudie le processus de développement des systèmes à base de connaissances. La notion de représentation des connaissances y occupe une place centrale. Cela indique clairement que son objet d'étude est la connaissance prise au sens large par opposition aux données brutes ou factuelles d'un système informatique conventionnel. La tâche de bâtir un système à base de connaissances implique des opérations comme la collecte de connaissances, leur analyse, leur organisation et leur codification pour les implanter dans un système.

1.2 Étapes de développement d'un système à base de connaissances

La figure 1 montre les principales tâches prises en charge par l'ingénierie de la connaissance, comparativement à celles d'un système conventionnel. La démarche que nous proposons s'inspire de celle déjà éprouvée pour des systèmes informatiques conventionnels. Cette démarche fait place à l'analyse, à la conception, à la programmation, aux tests et à l'implantation. Ce sont des tâches couramment exécutées lors du développement de n'importe quel type de système informatique.

La différence dans la terminologie ne devrait pas cacher la *similitude* dans le développement des deux types de systèmes. Par contre, chaque démarche présente ses particularités. La principale originalité du développement des SBC réside dans l'importance accordée aux connaissances. Elles font l'objet de trois étapes : l'identification, l'acquisition et la représentation. D'autre part, les outils employés ne sont pas les mêmes.

Ainsi, on développera un système conventionnel avec un langage de programmation évolué et très souvent, avec un système de gestion de base de données qui supportent des générateurs de rapports des programmes de traitement et une partie de l'interface usager.

Dans le cas d'un système à base de règles, on aura recours à des outils d'acquisition des connaissances pouvant générer des règles qui seront traitées par un moteur d'inférence.

Voici un survol des grandes étapes de développement d'un système à base de connaissances.

Étude d'opportunité. On étudie le domaine de connaissances à modéliser et les tâches effectuées par les experts afin d'évaluer la possibilité, la pertinence et les rapports coûts-bénéfices du développement d'un système à base de règles. Ensuite, on définit les besoins et les objectifs de l'organisation qui envisage le développement d'un SBC. Cette étude doit répondre à la question : « Est-ce qu'un SBC est une solution satisfaisante aux besoins identifiés? » Si la réponse est positive, on peut passer aux étapes suivantes.

Identification des connaissances. On définit les types de données et les types de solutions normalement produites par l'expert. Ensuite, on identifie les tâches que devrait effectuer le SBC, pour reproduire le comportement de l'expert. On repère les objets types, leurs principaux attributs et leurs relations les plus évidentes. Ce travail fournit des indications sur la nature des connaissances à rechercher au cours des prochaines étapes.

Acquisition des connaissances. On recueille les connaissances nécessaires à l'exécution des tâches identifiées précédemment, par des entrevues auprès des experts, des études de cas, de la recherche documentaire, etc.

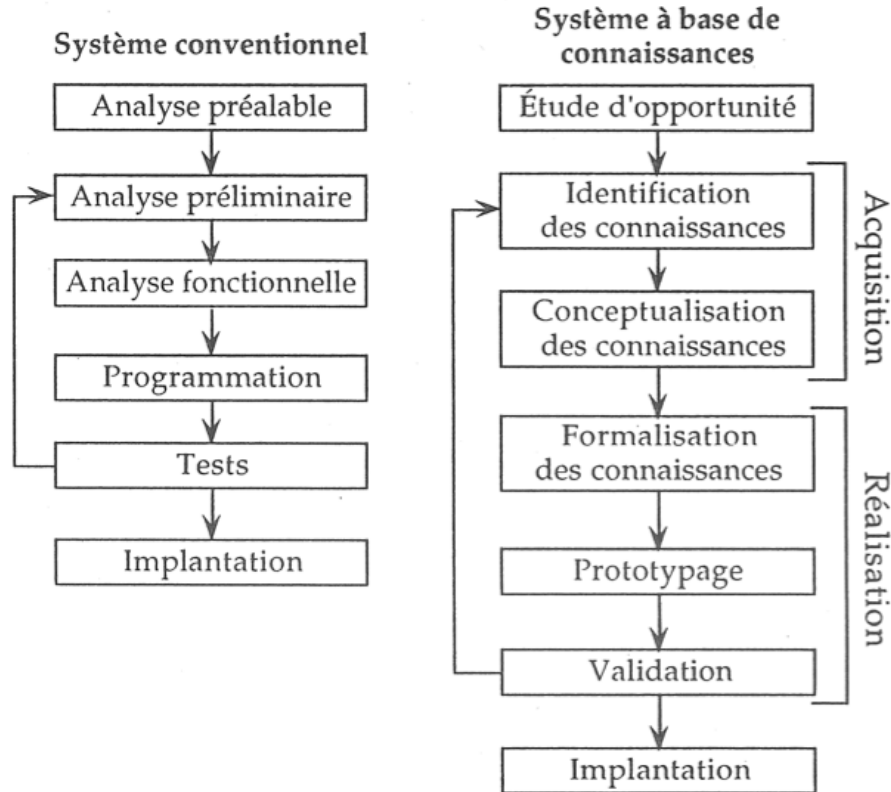


Figure 1 Étapes de développement de systèmes informatiques.

Représentation des connaissances. C'est l'étape de la formalisation des connaissances pour les traduire en règles ou en schémas selon la représentation choisie. On utilise des outils d'organisation et de structuration des connaissances pour construire progressivement une base à partir des connaissances acquises.

Prototypage. On implante progressivement sur ordinateur la base de connaissances et on conçoit l'interface usager. C'est un travail de raffinement progressif. On construit d'abord une version réduite du futur système. Puis, on y ajoute les connaissances acquises progressivement et on enrichit le dialogue avec l'utilisateur.

Validation. On vérifie l'exactitude des résultats fournis par le système auprès de l'expert, des utilisateurs et du responsable de projet.

Implantation. On met en marche le système dans son contexte normal d'opération et on initie les utilisateurs à son exploitation.

Bien que la démarche semble séquentielle, certaines étapes peuvent être exécutées en parallèle ou laissées en suspens pour être reprises à un stade ultérieur de développement du système. On procède généralement par *raffinement progressif* pour compléter certaines étapes. On pense surtout à l'acquisition et à la représentation des connaissances, au prototypage et à la validation que l'on effectue généralement par phases cycliques successives. On raffine notre modèle des connaissances, la base de règles et le prototype sur ordinateur, à chacune de ces étapes.

1.3 Intervenants et leur rôle

Intervenants est un terme générique qui désigne les personnes impliquées dans le développement d'un système informatique. Dans le cas d'un SBC, les intervenants sont les suivants :

- le responsable de projet,
- les usagers,
- les experts,
- les **cogniticiens** ou ingénieurs de la connaissance,
- les programmeurs.

Les *usagers* sont la raison d'être de tous les systèmes informatiques! Ils doivent donc participer à sa définition et à sa validation. Les *programmeurs* sont les personnes qui maîtrisent les techniques de l'informatique et la programmation est une de leurs spécialités. Dans le développement d'un système à base de règles, un programmeur est responsable de la traduction des connaissances en règles et de la programmation rendue nécessaire par les particularités du système, notamment les détails de l'interface usager, les liens avec d'autres applications, les calculs ou autres traitements spécialisés. Quant au *responsable de projet*, on peut le considérer comme la personne qui prend les décisions (ou qui est mandatée pour les faire connaître) concernant les orientations d'un projet, les ressources humaines et matérielles requises, et l'échéancier à respecter.

Les *experts* et les *cogniticiens* sont deux types d'intervenants spécifiques au développement d'un système à base de connaissances. Les experts possèdent les connaissances que les cogniticiens tentent d'acquérir et de représenter pour les traduire en faits et en règles dans un système à base de règles, ou en schémas s'il s'agit du mode de représentation choisi. D'autre part, les cogniticiens sollicitent les experts et colligent des données pour constituer une base de connaissances dont l'exploitation par un moteur d'inférence peut reproduire des tâches normalement exécutées par des experts humains. La figure 2 illustre de quelle manière interagissent les intervenants lors du processus de développement d'un SBC.

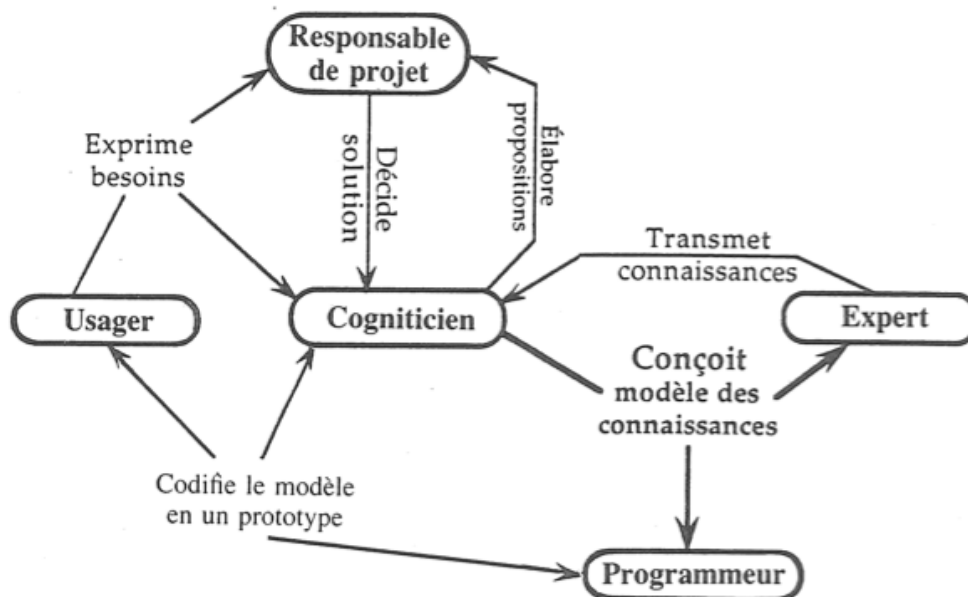


Figure 2 Les échanges entre les intervenants dans le développement d'un SBC.

Comme le montre la figure 2, le cogniticien joue un rôle fondamental dans le développement des SBC, soit l'identification, l'acquisition et la représentation des connaissances. Nous tenons à préciser ce que l'on attend de lui. Dans son livre *Acquisition du savoir pour les systèmes experts*, Anna Hart suggère une liste d'aptitudes souhaitées chez les cogniticiens. En voici quelques-unes :

- savoir bien communiquer;
- avoir du tact, de la diplomatie;
- être patient(e);
- savoir se mettre à la place de la personne experte;
- être persistant(e);
- être logique;
- être capable de s'adapter;
- avoir un esprit inventif;
- avoir confiance en soi;
- avoir une connaissance du domaine.

Le cogniticien ou ingénieur de la connaissance sert de lien entre l'expert et le programmeur du système à base de règles, ou parfois, il réalise le système si la programmation n'est pas nécessaire. Il doit recueillir les connaissances nécessaires au développement d'un système, les organiser et les structurer dans une forme telle qu'elles puissent être directement codées dans un formalisme informatique. D'autre part, le cogniticien est responsable de la conception du dialogue que le SBC entretient avec l'utilisateur ainsi que de la spécification des interfaces comme les menus, les fenêtres, les icônes, etc. Enfin, de concert avec l'expert, il voit à la validation des résultats produits par le système.

Selon l'envergure du système, plusieurs des rôles identifiés ici peuvent être tenus par une même personne. Ainsi, une personne familière avec le développement de systèmes peut adapter ses méthodes et ses techniques pour représenter les connaissances des experts et programmer le prototype du système. Par contre, à moins de prouesses d'empathie, il est plutôt irréaliste d'échanger les rôles d'expert et d'utilisateur.

2. Acquisition des connaissances : objectifs et moyens

L'acquisition des connaissances regroupe toutes les tâches préalables à la construction d'une représentation et à la codification des connaissances sous forme de règles ou autrement. Son but est donc la cueillette et la structuration des connaissances. Ces tâches sont normalement effectuées par le cogniticien en étroite collaboration avec l'expert. Les moyens employés pour y arriver varient selon la nature, la complexité et la quantité de connaissances à acquérir. On insiste sur le fait que l'acquisition des connaissances doit s'inscrire dans une démarche planifiée et toujours orientée vers l'objectif final : représenter les connaissances de l'expert pour les traduire en règles. Enfin, on retient quelques outils d'organisation et de structuration des connaissances pour nous aider dans cette tâche.

2.1 Étude d'opportunité

Le but de cette étape est de juger de la pertinence du développement d'un système à base de règles, suite à l'étude de la définition du problème à résoudre, du survol du domaine d'expertise, des ressources humaines et matérielles requises et des solutions de remplacement disponibles.

Pour les intervenants, cette étape est souvent l'occasion de commencer à se familiariser avec les concepts et les techniques liés au développement de SBC. Tous les intervenants devraient y participer. C'est le gestionnaire du projet ou ses mandants qui prennent la décision finale quant à la solution proposée. Cette solution doit expliciter le circuit complet de l'information traitée par le SBR projeté, c'est-à-dire de l'expert aux usagers, les ressources humaines et matérielles requises et les raisons qui motivent le développement du système proposé.

Il s'agit d'une approche identique à celle suivie pour l'étude d'opportunité des systèmes conventionnels. Sauf que dans le cas des SBR, cette étude doit démontrer les avantages de la technologie des bases de connaissances par rapport à d'autres technologies comme les bases de données. Certains auteurs³ suggèrent des conditions à respecter pour que l'on adopte une solution orientée système à base de connaissances. Les voici sous forme interrogative.

- Est-il possible de *délimiter le domaine d'expertise* à implanter dans un système?
L'expérience a montré que les meilleures applications de ce genre sont celles qui ont une fonction limitée et bien identifiée. Un SBC doit être spécialisé pour maximiser les chances de son implantation.
- La solution informatique actuelle est-elle déficiente? Il ne sert à rien de remplacer ou de doubler une technologie appropriée par une autre.
- L'expertise est-elle nécessaire? Il faut que des experts du domaine soient beaucoup plus performants que des novices pour résoudre les problèmes envisagés. Autrement, il n'y a pas de raison de développer un système expert.
- Les experts sont-ils disponibles et accessibles?
- Est-ce que le sujet est enseigné ou documenté? Lorsque le savoir sur un sujet est déjà documenté, cela constitue parfois une source de connaissances qui peut se substituer à l'expert.

Enfin, une étude d'opportunité sérieuse doit évaluer les rapports coûts-bénéfices qui peuvent résulter de l'implantation d'un SBC dans une organisation. Cela implique une estimation des coûts en machine et en main-d'œuvre, ainsi que des retombées prévisibles. Un échéancier de réalisation est absolument nécessaire. Dans le cas d'un SBR, il faut, en plus, étudier l'impact d'une nouvelle technologie pour les usagers et démontrer que cette approche produira de meilleurs résultats qu'un système conventionnel. Accepter la solution proposée signifie que l'on peut passer aux étapes suivantes. Dans le cas contraire, le développement du système s'arrête ici!

2.2 Planification d'une démarche d'acquisition des connaissances

L'acquisition des connaissances doit se faire conformément à un plan élaboré conjointement par les experts et les cognitivistes. C'est l'étape qui prépare à la construction d'un premier prototype, notamment à la rédaction des règles, dans le cas d'un SBR. Les principales tâches à effectuer lors de cette étape sont :

- faire l'inventaire des sources de connaissances;
- élaborer un échéancier et un scénario de cueillette des connaissances;
- familiariser les membres de l'équipe de développement avec le domaine d'expertise;
- initier les experts aux techniques de représentation et de codification des connaissances;

³ Notamment Parsaye et Chignell.

- sélectionner des problèmes types qui serviront de références au développement d'un prototype de SBR;
- préparer et réaliser les entretiens avec les experts et en analyser les résultats;
- organiser et structurer les connaissances recueillies pour produire un modèle cohérent à partir duquel on pourra induire les règles.

Toutes ces opérations sont planifiées et supervisées par le cognicien. En outre, c'est lui qui se charge de la dernière tâche. Par ailleurs, le nombre de personnes impliquées dans un projet dépend de son importance. Dans les prochaines sections, nous décrirons en détail chacune des tâches d'acquisition. Pour en faire une synthèse, nous proposons un schéma⁴ du cheminement d'acquisition des connaissances, à la figure 3.

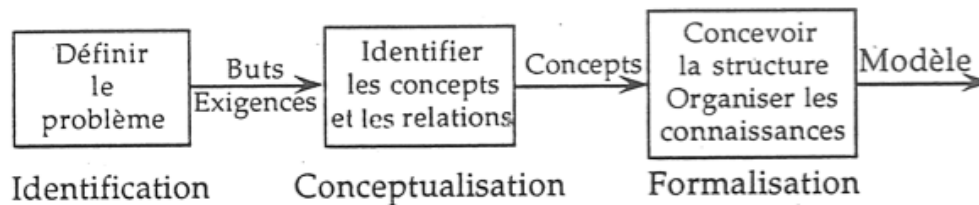


Figure 3 Phases d'acquisition des connaissances.

2.3 Identification des connaissances

Cette étape consiste essentiellement à définir le problème à résoudre. On commence généralement par cerner les objectifs du système projeté, c'est-à-dire les résultats qu'il doit produire et leur forme.

Ensuite, il faut identifier les connaissances qui sont nécessaires pour produire les résultats attendus. Dans un contexte de développement d'un SBR, cela veut dire interroger les experts pour faire un inventaire de leurs connaissances et découvrir leur façon de résoudre les problèmes du domaine étudié. Anna Hart suggère que l'étape d'identification se fasse en répondant à des questions. Les voici présentées sous forme d'une liste d'interrogations que le cognicien pourra consulter pour s'assurer qu'il a envisagé la plupart des facettes du problème avec l'expert.

- Quelles sont les données à fournir au système?
- Quels sont les résultats produits par le système? Comment sont-ils présentés?
- Quels sont les types de problèmes?
- Quelles sont les caractéristiques des solutions?
- Quels types de connaissances doit-on acquérir?
- Comment peut-on diviser le problème en sous-problèmes?
- Quelles données manqueront peut-être?
- Quelles sont les relations entre les données?
- Quel est le degré de précision des données et des résultats à fournir?
- Quels sont les postulats de base (parfois implicites) de la personne experte?
- Quelles sont les contraintes à respecter dans les données ou les solutions?

⁴ Cheminement proposé par Hayes-Roth, Waterman et Lenat.

- Quelles sont les inférences faites par la personne experte?
- De quelle manière les buts sont-ils choisis?
- Quels sont les problèmes faciles, fréquents, difficiles ou intéressants?

2.4 Conceptualisation des connaissances

À cette étape, le cogniticien, de concert avec l'expert, doit faire ressortir les *concepts importants et les relations significatives* du domaine étudié. Il est important de schématiser les concepts et les relations pour établir clairement la base conceptuelle du prototype à développer. Comme pour l'identification des connaissances, une liste de questions peut servir de guide lors de la conceptualisation. En voici quelques-unes :

- Quelles sont les données disponibles? Quelles sont celles obtenues directement et celles déduites?
- Est-il possible d'identifier des tâches et des stratégies de résolution de problèmes?
- Quelles sont les hypothèses de travail?
- Quelles sont les relations entre les objets?
- Y a-t-il une hiérarchie des relations et des objets?
- Quelles sont les contraintes à respecter?
- Quelles sont les connaissances nécessaires à la solution des problèmes et celles utilisées pour justifier une solution?
- Dès l'étape de la conceptualisation, le modèle ébauché par le cogniticien doit être confronté à des cas concrets auxquels l'expert fait régulièrement face.

2.5 Formalisation des connaissances

Il s'agit de l'étape de construction d'un modèle opérationnel des concepts et des relations identifiés à l'étape précédente. C'est ici que le cogniticien procède au choix de la représentation et des outils de développement appropriés au problème. Il doit d'abord opter pour une représentation : logique, base de règles ou base de schémas. Ensuite, il choisit l'environnement informatique qui supporte la représentation choisie. Le reste du travail consiste à traduire les connaissances conceptualisées dans le formalisme de l'outil informatique adopté.

Précisons que, comme nous envisageons de développer des systèmes à base de règles, nous retiendrons la règle comme étant l'unité fondamentale de nos représentations. Selon cette approche, la modélisation doit faire ressortir les attributs d'un domaine et leurs relations. Précisons enfin que la conception d'une représentation est un processus de découverte et de création. C'est une tâche *d'induction*, c'est-à-dire la construction d'un modèle à partir de connaissances brutes exprimées par les experts. Des techniques et des outils de modélisation sont décrits dans les sections suivantes.

3. Induction des règles à partir d'exemples

En philosophie des sciences, on distingue généralement deux grands processus largement utilisés à différentes phases de l'activité scientifique : la déduction et l'induction. On les retrouve également dans la construction des systèmes à base de connaissances. Nous avons déjà mis en évidence le processus de déduction ou d'inférence au chapitre précédent. Celui-ci sert à mettre à profit une base de connaissances sous forme de règles ou de schémas. Afin de pouvoir appliquer la déduction, il faut déjà connaître un ensemble de règles générales reliant

des faits entre eux. Ainsi, la connaissance de certains faits permet d'en déduire de nouveaux. Par exemple, à l'aide de la règle suivante applicable à tout animal :

Si Respiration (animal) = branchies

Alors Classe (animal) = poisson

Du fait Respiration (cette truite) = branchies, on déduit par cette règle un fait nouveau :

Classe (cette-truite) = poisson

La déduction est un processus davantage mécanique que créateur de nouvelles connaissances, puisqu'on peut dire que les faits déduits sont déjà là implicitement, dans la règle générale. La déduction ne permet que de les mettre en évidence. Tout est dans les règles. L'induction, au contraire, est un processus beaucoup plus créateur qui permet justement de construire ces règles générales à partir de l'examen attentif de cas particuliers ou d'exemples. Dans un tableau donnant les attributs d'une cinquantaine d'animaux, on constate que chaque fois qu'un animal respire par des branchies, il s'agit d'un poisson. On est alors tenté d'en faire une règle générale. C'est ce qu'on appelle procéder par induction.

L'induction joue un rôle fondamental dans l'acquisition des connaissances. À cette étape, nous ne connaissons pas encore les règles qui constitueront la base de connaissances. Notre but est justement de les découvrir. S'il arrive quelquefois que les experts peuvent énoncer directement des règles, il faudra, le plus souvent, examiner le fonctionnement de l'expert dans des cas particuliers, pour en induire des règles générales.

3.1 Processus d'induction

Pour comprendre le processus d'induction, prenons un exemple simple. Un physiothérapeute doit déterminer les divers paramètres d'un traitement aux ultrasons, notamment la *durée* du traitement. En discutant avec lui, on apprend que les facteurs importants dans un traitement aux ultrasons sont :

- le *but* du traitement qui peut être la guérison (gu), l'analgésie (an), la réduction des adhérences (ad) ou la réduction de l'œdème (œ);
- le *stade* de la maladie qui peut être aigu (ai), subaigu (sa) ou chronique (ch);
- la *profondeur* de l'endroit à traiter qui est un nombre décimal entre 0 et 7 cm;
- la *réflexion* osseuse, phénomène changeant qui est l'effet du faisceau qui est une valeur logique binaire oui ou non.

On lui demande alors de nous dire comment il fixerait l'attribut *durée du traitement* dans un certain nombre de cas de patients fictifs dont on lui donne les valeurs des quatre attributs relatifs à l'état du patient. Le tableau 1 montre ce que l'on peut obtenir de cette façon.

En examinant attentivement ces résultats, on constate tout de suite des régularités, grâce au bon ordonnancement de cet échantillon de cas. Les quatre premières lignes du tableau nous suggèrent que la durée est de 3 minutes, dès que le stade est aigu. Les trois autres paramètres ne semblent avoir aucune influence sur la durée. Cela nous incite à induire la règle :

Tableau 1 – Un premier échantillon de cas pour l'attribut *durée*

Cas-patient	But	Stade	Profondeur (cm)	Réflexion	Durée (min)
cas 1	gu	ai	1	oui	3
cas 2	an	ai	2	non	3
cas 3	ad	ai	3	oui	3
cas 4	ce	ai	4	non	3
cas 5	gu	sa	3	oui	5
cas 6	an	sa	2	non	5
cas 7	ad	sa	1	oui	5
cas 8	ce	sa	2	non	5
cas 9	gu	ch	1	oui	5
cas 10	an	ch	2	non	5

Règle 1 : Si stade (cas) est aigu
Alors durée (cas) est 3 min

De même, les quatre lignes suivantes nous suggèrent la règle :

Règle 2 : Si stade (cas) est subaigu
Alors durée (cas) est 5 min

Et finalement, les deux dernières lignes semblent poursuivre la tendance dans le cas chronique, ce qui nous suggère de généraliser la règle précédente et de la remplacer par :

Règle 2* : Si stade (cas) est subaigu ou stade (cas) est chronique
Alors durée (cas) est 5 min

Contrairement à la déduction où chaque pas déductif est fiable, l'induction comporte toujours une part d'incertitude. Supposons qu'en continuant d'interroger l'expert, on obtient le tableau 2 qui est plus complet :

Tableau 2 – Un deuxième échantillon de cas pour l'attribut *durée*

Cas-patient	But	Stade	Profondeur (cm)	Réflexion	Durée (min)
cas 1	gu	ai	1	oui	3
cas 2	an	ai	2	non	3
cas 3	ad	ai	3	oui	3
cas 4	œ	ai	4	non	3
cas 5	gu	sa	3	oui	5
cas 6	an	sa	2	non	5
cas 7	ad	sa	1	oui	5
cas 8	œ	sa	2	non	5
cas 9	gu	ch	1	oui	5
cas 10	an	ch	2	non	5
cas 11	ad	ch	3	oui	10
cas 12	œ	ch	2	non	10

Les deux nouvelles lignes du tableau compliquent les choses légèrement. Elles n'obéissent pas à la Règle 2*, puisque le stade est chronique alors que la durée n'est pas de 5 minutes, mais de 10 minutes. On dira de chacun de ces deux cas qu'il est un *contre-exemple* de la Règle 2* car il

la contredit, alors que les cas 5 à 8 sont des *exemples* de cette règle. Les cas 5 à 8 sont également des exemples de la Règle 2, laquelle n'a toujours pas de contre-exemple. Nous allons donc éliminer la Règle 2*, puisqu'elle a des contre-exemples et conserver la Règle 2 qui tient toujours. On dit d'une règle qui n'a pas de contre-exemple qu'il s'agit d'une *règle valide* (par rapport à un ensemble de cas). Il nous reste à trouver une façon de traiter correctement les cas où le stade est chronique. Dans ces cas, la durée est parfois de 5 minutes, parfois de 10 minutes. La différence semble causée par le but plutôt que par les deux autres paramètres. Pour nous rassurer, il faut ici examiner un plus grand nombre de cas où on fait varier ces deux derniers paramètres. Mais si la tendance se maintient, on peut induire les deux règles suivantes :

Règle 3 :	Si	stade (cas) est chronique et (but (cas) est guérison ou but (cas) est analgésie)
	Alors	durée (cas) est 3 min
Règle 4 :	Si	stade (cas) est chronique et (but (cas) est adhérences ou but (cas) est œdème)
	Alors	durée (cas) est 10 min

L'induction nous laisse toujours un certain degré d'incertitude : reste-t-il des contre-exemples à l'une ou l'autre des règles que nous avons définies? C'est là le prix de la création. En effet, l'induction est un processus mental « synthétique », plus créateur que la déduction qui est, par essence, analytique. Dans l'induction, on construit de nouvelles règles générales à partir de cas particuliers, d'exemples. Ces connaissances générées par induction, bien que plus risquées, sont d'un niveau plus général que celles qui ont servi à les construire.

3.2 Fiabilité d'une induction et ensembles caractéristiques

Pour réduire les risques inhérents à l'induction d'une règle, il faut prendre certaines précautions. Pour toute règle et un ensemble de cas qui ont servi à induire la règle, on peut définir trois ensembles caractéristiques :

- CE : l'ensemble des contre-exemples; ceux pour lesquels la condition s'applique, mais non la conclusion;
- EX : l'ensemble des exemples; ceux pour lesquels la condition et la conclusion s'appliquent;
- NC : l'ensemble des autres cas qui ne contredisent ni ne renforcent la règle; ce sont ceux où la condition ne s'applique pas.

Pour la Règle 1 :

Si	stade (cas) est aigu
Alors	durée (cas) est 3 min

et l'ensemble de cas du tableau précédent,

- CE est vide (il n'y a pas de contre-exemple)
- EX est formé des quatre premiers cas
- NC est formé des huit derniers cas

Pour la Règle 2* :

Si stade (cas) est subaigu ou
 stade (cas) est chronique
Alors durée (cas) est 5 min

On trouvera les ensembles caractéristiques sur le diagramme de la figure 4 :

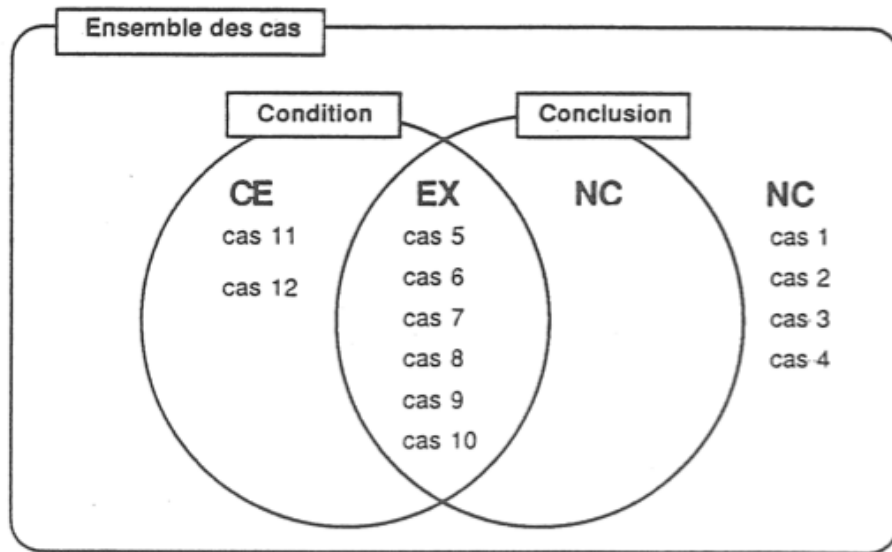


Figure 4 Ensembles caractéristiques d'une règle.

La fiabilité de l'induction d'une règle dépend beaucoup du nombre d'éléments dans chaque ensemble caractéristique. Par exemple, s'il n'y a qu'un seul contre-exemple sur une centaine de cas, on peut affirmer que la règle tient à cette exception près. À l'opposé, s'il n'y a pas de contre-exemple mais *EX* et *NC* contiennent très peu d'exemples, on accordera peu de fiabilité à la règle. Pour la Règle 2*, si on augmente le nombre de cas, le nombre de contre-exemples augmente en proportion, ce qui renforce notre décision de rejeter cette règle. Par contre, pour la Règle 1, on peut se demander ce qui arrive pour des valeurs supérieures à 4 cm. La durée est-elle toujours égale à 3? Autrement dit, bien que la Règle 1 semble valide, il y a encore trop peu d'exemples et ceux-ci ne sont pas bien répartis quant à certains attributs comme la profondeur.

3.3. Choix d'un ensemble représentatif d'exemples

Mais ce facteur quantitatif n'est pas le seul en cause. La fiabilité que l'on accorde à une induction dépend énormément du contexte. Par exemple, supposons que l'on débarque sur une île inconnue du Pacifique et que l'on nous montre un oiseau bleu appelé Wiwi, puis on nous montre un habitant obèse de la tribu des Onitara. On accordera plus de crédibilité à la règle :

Si nom(oiseau) = Wiwi
Alors couleur(oiseau) = bleu

qu'à la règle :

Si tribu(habitant) = Onitara
Alors forme(habitant) = obèse

La raison réside dans nos connaissances acquises antérieurement : les types d'oiseaux sont souvent tous de la même couleur, alors que dans tous les peuples, il y généralement des personnes minces et des personnes obèses. Évidemment, il n'y a pas de règle absolue puisque l'induction est un processus non algorithmique. Cependant, on peut se fier à la fois à la taille des ensembles caractéristiques et à notre connaissance du domaine à modéliser qui va en s'améliorant, au fur et à mesure que nous progressons dans la représentation des connaissances.

Reprenons l'exemple du traitement aux ultrasons en physiothérapie. Cet exemple va nous donner des indications sur le choix d'un ensemble représentatif de cas. Cette fois, nous désirons connaître comment l'expert fixe la fréquence du faisceau à partir des attributs du cas-patient : but, stade, profondeur et réflexion. Le tableau 3 nous montre un ensemble bien réparti, sauf en ce concerne le but. Les trois autres paramètres prennent toutes les combinaisons possibles. Évidemment, la profondeur étant un attribut numérique qui peut prendre un nombre infini de valeurs entre 0 et 7 cm, nous nous sommes limités pour le moment à trois valeurs où l'endroit à traiter est superficiel (1 cm), moyen (3 cm) ou profond (6 cm).

En examinant le tableau, nous retenons immédiatement que le stade et la réflexion n'influencent pas la valeur de la fréquence. De plus, dans tous les cas, la fréquence diminue lorsque la profondeur augmente. On peut tout de suite avancer les trois règles suivantes :

Si profondeur(cas) = 1 cm
 Alors fréquence(cas) = 3 MHz
 Si profondeur(cas) = 3 cm
 Alors fréquence(cas) = 1 MHz
 Si profondeur(cas) = 6 cm
 Alors fréquence(cas) = 0,75 MHz

Les 18 cas du tableau 3 sont-ils vraiment représentatifs? Non, car le but du traitement prend toujours la valeur guérison (gu). Pour valider les trois règles précédentes on peut choisir 9 cas où on donne au but les trois autres valeurs possibles (an, ad et ce), et à la profondeur, les trois valeurs 1, 3 et 6, sans faire varier le stade et la réflexion. Ce processus confirmerait la validité des trois règles. Comme elles n'ont aucun contre-exemple et beaucoup d'exemples, en fait 9 exemples chacune sur 27 cas choisis, on peut leur accorder une très grande fiabilité. Cependant, ces règles n'ont pas une très grande utilité. On dit que l'on a une *règle utile* si elle s'applique dans un nombre significatif de cas parmi tous ceux possibles. Or ici, on reconnaît que chaque règle ne couvre qu'une valeur de la profondeur sur une infinité de cas possibles. Qu'arrive-t-il lorsque la profondeur est différente de 1 cm, 3 cm ou 6 cm? Pour le savoir, nous prenons un ensemble de cas où seule la profondeur varie, puisque nous avons découvert que les autres attributs n'ont aucune influence sur la fréquence.

Tableau 3 – Un premier échantillon de cas pour l'attribut *fréquence*

Cas-patient	But	Stade	Profondeur (cm)	Réflexion	Fréquence MHz
cas 1	gu	ai	1	oui	3
cas 2	gu	ai	1	non	3
cas 3	gu	ai	3	oui	1
cas 4	gu	ai	3	non	1
cas 5	gu	ai	6	oui	0,75
cas 6	gu	ai	6	non	0,75

cas 7	gu	sa	1	oui	3
cas 8	gu	sa	1	non	3
cas 9	gu	sa	3	oui	1
cas 10	gu	sa	3	non	1
cas 11	gu	sa	6	oui	0,75
cas 12	gu	sa	6	non	0,75
cas 13	gu	ch	1	oui	3
cas 14	gu	ch	1	non	3
cas 15	gu	ch	3	oui	1
cas 16	gu	ch	3	non	1
cas 17	gu	ch	6	oui	0,75
cas 18	gu	ch	6	non	0,75

Du tableau 4 qui suit, nous pouvons induire avec une assez grande fiabilité les règles suivantes :

- Si profondeur(cas) \leq 2 cm
Alors fréquence(cas) = 3 MHz
Si profondeur(cas) > 2 cm ET profondeur(cas) < 4,5 cm
Alors fréquence(cas) = 1 MHz
Si profondeur(cas) \geq 4,5 cm
Alors fréquence(cas) = 0,75 MHz

Tableau 4 – Un deuxième échantillon de cas pour l'attribut *fréquence*

Cas-patient	But	Stade	Profondeur (cm)	Réflexion	Fréquence MHz
cas 1	gu	ai	0,5	oui	3
cas 2	gu	ai	1	oui	3
cas 3	gu	ai	1,5	oui	3
cas 4	gu	ai	2	oui	3
cas 5	gu	ai	2,5	oui	1
cas 6	gu	ai	3	oui	1
cas 7	gu	ai	3,5	oui	1
cas 8	gu	ai	4	oui	1
cas 9	gu	ai	4,5	oui	0,75
cas 10	gu	ai	5	oui	0,75
cas 11	gu	ai	6	oui	0,75
cas 12	gu	ai	6,5	oui	0,75
cas 13	gu	ai	7	oui	0,75

Cependant, une incertitude demeure. Est-ce bien à 2 cm et à 4,5 cm que la valeur de la fréquence change? Pourquoi pas à 2,1 cm ou à 4,4 cm? On peut résoudre ce dernier problème en ajoutant plusieurs cas où la profondeur est proche des deux valeurs critiques. Peut-être aussi un expert peut-il trancher cette question, car dans tous les cas, lorsque nous franchissons une étape significative d'induction, l'avis d'un expert est requis pour valider les règles et s'assurer qu'elles couvrent tous les cas. Il se peut alors que l'expert élimine ou ajoute des attributs ou leur

fixe de nouvelles valeurs possibles; ce qui oriente le processus d'induction dans de nouvelles directions.

3.4 Arbre des contraintes entre attributs

Nous allons maintenant regarder le processus d'induction plus globalement. Dans les exemples précédents, nous avons supposé que les attributs avaient déjà été identifiés et que leurs interrelations avaient été établies. Par exemple, nous avons supposé au départ que la durée, tout comme la fréquence, ne pouvaient dépendre que des attributs but, stade, profondeur et réflexion. Après une analyse plus poussée à l'aide d'ensembles représentatifs de cas, nous nous sommes aperçus que la durée ne dépendait que du stade et du but et que la fréquence ne dépendait que de la profondeur. Cette réalité est illustrée à la figure 5.

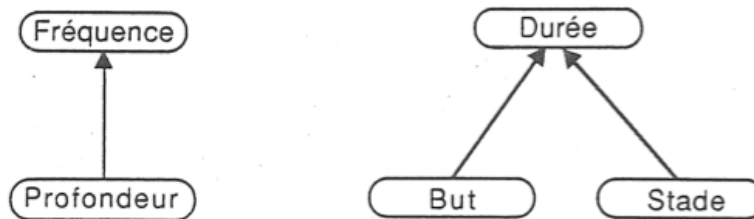


Figure 5 Les contraintes « fréquence » et « durée ».

De tels ensembles de liens entre attributs sont appelés *contraintes*. L'arbre des contraintes de la figure 5 nous montre donc deux contraintes : la contrainte « fréquence », qui ne dépend que de l'attribut *profondeur*, et la contrainte « durée » qui ne dépend que des attributs *but* et *stade*. Pour chaque contrainte, on veut pouvoir définir par induction un ensemble de règles valides qui couvrent tous les cas qui peuvent se présenter, comme nous venons de le faire.

Le travail sur le choix d'un ensemble représentatif d'exemples montre l'importance de comprendre d'abord les rapports des contraintes entre les attributs avant de commencer à définir des règles. Si nous avions su, au départ, que la fréquence ne dépendait que de la profondeur, nous aurions choisi nos cas de façon à éliminer les colonnes but, stade et réflexion, inutiles pour déterminer la fréquence, et à nous donner une plus grande variété de valeurs de la profondeur, attribut déterminant pour la fréquence.

Pour déterminer les contraintes entre attributs, on doit se laisser guider par le problème. Une analyse du problème doit faire ressortir les relations de dépendance entre les attributs. Pour ce faire, on peut procéder en « chaînage arrière », c'est-à-dire en partant de la finalité du système et en remontant vers les conditions qui doivent être vérifiées pour que le but final soit atteint. Que voulons-nous que le système donne comme réponse finale? De quels attributs dépend cette réponse? Puis, de quels attributs chacun dépend-il à son tour? On continue ainsi jusqu'à ce que tous les liens de dépendance entre les attributs soient identifiés. Par exemple, si le but du système est de déterminer un traitement aux ultrasons, on peut sans doute obtenir de l'expert un arbre des contraintes entre attributs comme celui du schéma de la figure 6.

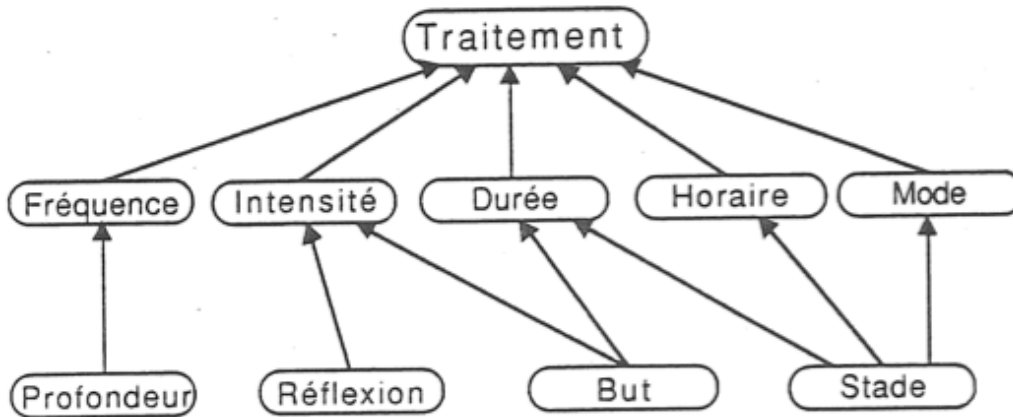


Figure 6 L'arbre des contraintes d'un traitement aux ultrasons.

Cet arbre contient six contraintes. Les deux contraintes « fréquence » et « durée » que nous avons vues plus haut; trois nouvelles contraintes correspondant à trois autres attributs importants pour le traitement : *l'intensité*, le *mode* du faisceau à ultrasons, et *l'horaire*, soit le nombre d'applications qu'il faudra faire; enfin, une dernière contrainte permet d'afficher le « traitement » requis à partir des cinq attributs « fréquence », « intensité », « durée », « horaire » et « mode ». Par contre, si le problème consiste à déterminer le coût du traitement, nous obtenons un arbre des contraintes tout à fait différent comme celui de la figure 7.

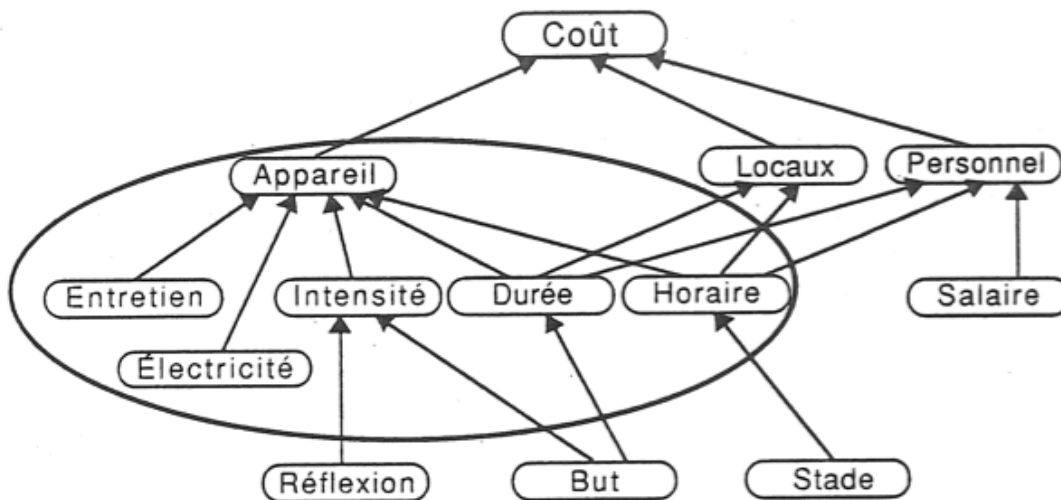


Figure 7 L'arbre des contraintes du coût d'un traitement aux ultrasons.

3.5 Ensemble des règles correspondant à une contrainte

Une fois l'arbre des contraintes établi, chaque contrainte donne lieu à une ou plusieurs règles. Pour chaque ensemble de règles correspondant à une contrainte, nous devons définir un ensemble de règles cohérentes qui couvrent tous les cas le plus efficacement possible. Un ensemble de règles sur une contrainte est un ensemble consistant s'il ne contient que des règles valides qui ne se contredisent pas. Par exemple, pour la contrainte « durée », l'ensemble des règles {Règle 2, Règle 3, Règle 4} est consistant. Par contre, l'ensemble {Règle 2*, Règle 3, Règle 4} ne l'est pas.

- Règle 2 : Si stade (cas) est subaigu
Alors durée (cas) est 5 min
- Règle 2* : Si stade (cas) est subaigu ou
stade (cas) est chronique
Alors durée (cas) est 5 min
- Règle 3 : Si stade(cas) est chronique et
(but(cas) est guérison ou but(cas) est analgésie)
Alors durée (cas) est 5 min
- Règle 4 : Si stade(cas) est chronique et
(but (cas) est adhérences ou but (cas) est œdème)
Alors durée (cas) est 10 min

La Règle 2* n'est pas contradictoire à la Règle 3, elle a tout simplement plus d'exemples. Cependant, elle est contradictoire à la Règle 4, car les cas où le stade est « chronique » et le but est « adhérences » ou « œdème » satisfont les conditions des deux règles. Cependant, on obtient des valeurs différentes pour la durée : 5 min avec la Règle 2* et 10 min avec la Règle 4.

Un ensemble de règles sur une contrainte est un ensemble complet si chaque cas possible est couvert par au moins une règle, c'est-à-dire si chaque cas satisfait la condition d'au moins une règle. Si l'ensemble de règles est complet, la valeur de l'attribut d'où provient la contrainte pourra être établie dans tous les cas. Par exemple, ni {Règle 2, Règle 3, Règle 4} ni {Règle 2*, Règle 3, Règle 4} ne sont des ensembles complets, puisque les cas où le stade est « aigu » ne sont pas couverts. Cependant, les ensembles {Règle 1, Règle 2, Règle 3, Règle 4} et {Règle 1, Règle 2*, Règle 3, Règle 4} sont complets, car leurs règles couvrent tous les cas possibles. Notons que le premier ensemble est consistant, alors que le deuxième ne l'est pas, puisque les Règle 2* et Règle 4 se contredisent.

Un ensemble de règles est un ensemble complet et efficace si chaque règle couvre des cas différents. On dit aussi qu'il n'y a pas de redondance ou de règles inutiles. Un tel ensemble réduit les risques d'explosion combinatoire si on met le moteur d'inférence en marche. Par exemple, l'ensemble {Règle 1, Règle 2, Règle 3, Règle 4}, en plus d'être consistant, est complet et efficace, puisque chaque règle couvre des cas différents. Par contre, si on y ajoutait la règle valide suivante :

- Règle 5 : Si stade (cas) = chronique et
but (cas) = œdème
Alors durée (cas) = 10 min

l'ensemble demeure consistant et complet. Par contre, il n'est pas efficace puisque tous les cas couverts par la Règle 5 le sont déjà par la Règle 4.

3.6 Sommaire d'une méthode d'induction de règles

L'induction des règles d'un système à base de connaissances consiste à créer des règles générales à partir de l'examen attentif de cas particuliers ou d'exemples. Il s'agit d'une démarche intellectuelle créatrice, puisqu'elle permet de découvrir des connaissances significatives et générales à partir de données factuelles et particulières. Bien que l'induction laisse toujours un certain degré d'incertitude quant à ses résultats, on peut la pratiquer

efficacement et obtenir des résultats très vraisemblables en suivant une méthode systématique dont voici les grandes lignes :

- Faire l'étude du problème et obtenir un premier échantillon d'objets caractéristiques du domaine étudié. Cela permet de faire un *inventaire des attributs* des objets identifiés.
- Trouver les relations de dépendance entre les attributs. On procède en chaînage arrière pour identifier d'abord le but final du système à développer. Ensuite, on détermine de quels attributs dépend la solution du problème, puis de quels autres attributs dépendent ceux-ci à leur tour. On procède ainsi jusqu'à l'obtention d'un *arbre des contraintes*. Une contrainte est formée d'un attribut de l'arbre et de ses descendants, c'est-à-dire les attributs dont il dépend.
- Déterminer les valeurs possibles de chaque attribut de l'arbre; ce qui permet de construire des exemples. Le nombre de cas nécessaires dépend des valeurs que peuvent prendre les attributs. Ce nombre varie aussi en fonction du contexte, c'est-à-dire des connaissances que nous avons déjà acquises sur le domaine. L'induction étant un processus plus créateur mais moins fiable que la déduction, il faut se donner un *ensemble représentatif* d'exemples ou de cas.
- Pour chaque contrainte, on cherche à *induire un ensemble consistant, complet et efficace* de règles. L'induction des règles est un processus de *raffinement graduel*. On peut procéder de la manière suivante :
 - dégager des règles approximatives d'un premier ensemble de cas;
 - confirmer ou infirmer les règles avec d'autres cas;
 - reformuler des règles pour les rendre plus précises et plus générales, en ajouter de nouvelles ou en retrancher au besoin;
 - vérifier l'ensemble des règles induites pour s'assurer qu'elles ne se contredisent pas et qu'elles couvrent tous les cas efficacement.

4. Techniques d'entrevue

Il existe plusieurs techniques qui permettent l'acquisition de connaissances en sollicitant des experts d'un domaine. L'observation directe, l'entrevue et le questionnaire sont les techniques les plus utilisées. C'est le contexte, la disponibilité des experts, la nature du sujet traité et le degré d'avancement d'un projet qui déterminent le choix de la technique la plus appropriée. Cependant, dans tous les cas, il incombe au cogniticien d'instaurer la collaboration avec les experts et de planifier soigneusement chaque rencontre avec ceux-ci. De plus, le cogniticien doit aussi trouver des moyens appropriés pour faire un suivi de ces rencontres. Ainsi, il voit à enregistrer les conversations, recueillir les questionnaires, en faire l'analyse et en structurer le contenu pour en induire une représentation des connaissances du domaine étudié qu'il pourra confronter à l'avis des experts. Cela veut dire maintenir une *collaboration soutenue basée sur la confiance réciproque entre experts et cogniticiens* durant toute la période de développement d'un système à base de connaissances.

4.1 Les objectifs de l'entrevue

L'entrevue est sans doute la technique la plus couramment utilisée pour solliciter la connaissance de l'expert. Il s'agit de la forme d'interaction la plus familière et un des moyens de transfert de connaissances les plus efficaces si on prend la précaution de structurer et d'enregistrer une conversation dont l'objectif est clairement défini et connu des interlocuteurs. Au cours d'un dialogue, le cogniticien pose des questions à l'expert sur son domaine d'expertise. L'expert s'exprime alors à sa manière et le cogniticien doit décoder son langage

pour en extraire les concepts et les méthodes que ce dernier utilise pour résoudre des problèmes. Dans son vocabulaire, l'expert décrira des objets et les relations entre ceux-ci. Ce sont ces matériaux qui serviront de base à la conception d'une représentation des connaissances de l'expert.

La conduite d'une séquence d'entrevues est une étape cruciale du processus d'acquisition des connaissances. Elle suppose une rétroaction entre l'expert et le cogniticien. Lors de l'entrevue, on doit utiliser une terminologie familière à l'expert pour faire ressortir clairement les objets du domaine étudié, leurs relations et leur organisation ainsi que les méthodes appliquées par l'expert pour résoudre les problèmes. Il faut absolument que l'entrevue soit centrée sur la manière dont l'expert raisonne sur le domaine des connaissances à étudier. Le cogniticien doit être le plus objectif possible, c'est-à-dire qu'il doit être attentif à ce que dit et fait l'expert; il doit aussi éviter de faire trop intervenir sa propre compréhension du problème, car ce n'est pas lui l'expert.

En général, en se documentant, on peut trouver les connaissances factuelles à intégrer dans une base de règles. Il n'est donc pas nécessaire d'accaparer indument l'expert pour obtenir ce genre de savoir. C'est pourquoi, l'entrevue doit faire ressortir les connaissances conceptuelles et surtout les heuristiques appliquées par l'expert pour résoudre les problèmes. Une manière d'y parvenir est de *procéder par cas*. Cette technique consiste à demander à l'expert de *fournir des exemples* pour décrire les connaissances de son sujet et la manière dont il procède pour résoudre les problèmes. On demande aussi à l'expert de décrire des exemples et des contre-exemples pour constituer un échantillon de cas représentatifs qui décrivent correctement le domaine étudié. Ensuite, le cogniticien peut procéder par induction pour construire une représentation des connaissances à partir des exemples recueillis auprès de l'expert. Enfin, on doit s'assurer que l'expert peut valider le modèle des connaissances que l'on s'appête à codifier dans une base de règles.

Exemple

Pour illustrer les techniques d'entrevue, nous prendrons le cas du développement d'un système capable d'effectuer un diagnostic de condition physique d'une personne et de lui prescrire un programme d'éducation physique personnalisé (PÉPP⁵) selon ses objectifs. Le système effectue ces tâches en puisant dans une banque d'exercices de conditionnement physique et en appliquant des règles de sélection de ces exercices basées sur un *modèle* de conditionnement appliqué par des experts du domaine. L'équipe de ce projet était constituée de « cogniticiens » (deux enseignants en informatique spécialisés en analyse de systèmes et en programmation) et d'experts (deux enseignants en conditionnement physique). L'objectif des entrevues était :

- d'identifier et de codifier les paramètres d'un modèle de conditionnement physique;
- de proposer un modèle des méthodes appliquées par les experts, lesquels s'inspirent de connaissances physiologiques, d'une compréhension des procédés métaboliques, des théories sur l'entraînement et du bon sens pour prescrire un programme de conditionnement personnalisé.

4.2 Préparation d'une entrevue

Le cogniticien doit élaborer une *stratégie d'entrevue* pour s'assurer qu'il peut obtenir d'un expert toutes les données nécessaires à la codification d'une base de règles sur le sujet étudié.

⁵ Programme d'Éducation Physique Personnalisé.

L'analyse préalable du problème étant faite, il est en mesure d'inventorier les sources d'informations sur le sujet pour colliger les connaissances disponibles sans le recours aux experts. Cet exercice lui permet d'identifier les connaissances manquantes qu'il ne peut obtenir sans l'apport de l'expertise. Ensuite, il établit un contact avec un ou des experts et les informe du projet en cours. Il faut que l'expert en connaisse les objectifs, le contexte de son application et les connaissances requises à son élaboration. De plus l'expert doit savoir ce que l'on attend de lui, c'est-à-dire connaître et partager les objectifs du projet. Ce sont des préalables à l'établissement d'une *relation de confiance entre les experts et l'équipe de développement* d'un système.

C'est le cogniticien qui établit le *scénario de chaque entrevue* basé sur l'atteinte d'objectifs connus de l'expert. Pour ce faire, il peut décomposer le problème à traiter en sous-problèmes, chacun couvrant un aspect bien identifié du système à concevoir. D'autre part, le cogniticien et l'expert doivent avoir une terminologie commune pour s'assurer que la communication se fasse sans ambiguïté. Cela veut dire que le cogniticien doit apprendre le vocabulaire du domaine et formuler ses questions dans des termes qui minimisent les chances de confusion avec l'expert. Le cogniticien doit présenter à l'expert une documentation pertinente pour appuyer ses questions. Il peut s'agir d'articles spécialisés, de références bibliographiques ou de comptes rendus d'expériences décrivant des cas précis à partir desquels l'expert peut s'exprimer et livrer un aspect de son expertise. Les connaissances et le vocabulaire acquis par le cogniticien contribuent à sa crédibilité, relèvent le niveau de discours et maintiennent une relation de confiance avec l'expert.

Une liste de questions peut servir de base à une entrevue. Les questions doivent appeler des réponses précises de la part de l'expert.

D'ailleurs, il est toujours intéressant de placer l'expert devant un problème concret et de lui demander de le résoudre en décrivant sa démarche de réflexion. On lui demande de « penser tout haut ». C'est un moyen privilégié d'avoir accès non seulement au savoir de l'expert, mais surtout à sa manière de l'utiliser. On doit donc préparer des cas à soumettre à l'expert. Dans les cas choisis, on retient des exemples typiques et des contre-exemples, de façon à obliger l'expert à préciser les limites de ses énoncés. Il est parfois souhaitable que l'on transmette à l'expert, avant l'entrevue, les questions et les cas qui lui seront soumis.

Il ne faut pas sous-estimer la préparation matérielle des entrevues. Cela concerne le lieu, le moment et le matériel requis. Le lieu de travail de l'expert est indiqué, puisque c'est là qu'il peut normalement avoir accès à la documentation, aux ordinateurs ou autres appareils requis selon le cas. Au besoin, on prend rendez-vous et on confirme les coordonnées et l'ordre du jour d'une réunion. Dans le cas où un projet nécessite plusieurs entrevues, on doit établir un *calendrier des réunions* pour permettre à chaque partie de s'y préparer correctement et de planifier son emploi du temps. Enfin, on doit s'assurer de disposer du matériel nécessaire : un équipement audiovisuel nécessaire à l'enregistrement de l'entrevue, des ordinateurs dans le cas où une démonstration serait requise ou tout autre matériel approprié.

Exemple

Les premiers préparatifs aux entrevues pour le PÉPP furent de chercher dans la documentation afin d'apprendre le vocabulaire : exercice de conditionnement, phase d'entraînement, type d'exercice, condition physique, objectifs d'entraînement, habitudes de vie, posologie, progression, diagnostic, prescription, etc. La terminologie est spécialisée, mais la documentation accessible ne manque pas. Ensuite, les cogniticiens ont élaboré un plan de travail, une

répartition des tâches et un échéancier des réunions avec les experts. Ces derniers étaient très enthousiastes face au projet, puisqu'ils allaient être les premiers à en profiter.

Après cette étape, les cognitiens ont proposé une ébauche de ce que devrait être le PÉPP, ses fonctions générales, les données qu'il faudrait lui fournir et son cycle d'utilisation par une personne qui voudrait suivre un de ses programmes de conditionnement. Cette ébauche qui comportait des diagrammes a servi de base au développement du système. Toutes les discussions ultérieures ont permis de préciser et d'améliorer ce modèle original. Puisque l'on a convenu tacitement de faire l'analyse et la conception par raffinement graduel, on a décomposé le problème en sous-problèmes : diagnostic, banque d'exercices de conditionnement, groupes musculaires, mécanisme de prescription d'un programme d'entraînement, posologie et progression des exercices. Cette approche a permis de centrer chaque entrevue sur un objectif précis et ainsi, de mettre au point assez rapidement des modules de PÉPP.

4.3 Déroulement d'une entrevue

C'est le cogniticien qui dirige l'entrevue conformément à un scénario préparé et connu de l'expert. Un scénario peut servir de cadre pour le déroulement d'une entrevue et être utilisé comme un menu des termes à clarifier, des questions à répondre et des problèmes à résoudre. On peut admettre des digressions pertinentes pourvu qu'elles se terminent par un retour au déroulement prévu. Les interlocuteurs doivent toujours avoir à l'esprit les objectifs de l'entrevue.

Une entrevue peut prendre diverses formes selon le sujet traité et le degré d'avancement d'un projet. L'auteur Waterman nous suggère, au tableau 5, des techniques pour solliciter un expert en entrevue.

Tableau 5 – Quelques techniques d'entrevue

Technique	Description
Observation	Examen de la manière dont l'expert résout des problèmes
Discussion	Exploration concernant les types de données, les concepts et les méthodes nécessaires à la solution d'un problème
Analyse	Présentation de problèmes concrets à l'expert et formulation du raisonnement pour les résoudre
Critique	Évaluation par l'expert d'un prototype de base de règles induites et du mécanisme d'inférence proposé
Validation	Vérification du système à l'aide des problèmes déjà analysés lors d'entrevues précédentes

Le déroulement d'une entrevue doit être centré sur les connaissances de l'expert et surtout sur la manière dont il les utilise. D'autre part, il faut absolument conserver ces données recueillies car, lors des entrevues, elles serviront à l'élaboration d'une base de règles. Le cogniticien doit donc être méthodique tout en admettant une certaine souplesse dans le déroulement d'une entrevue. Anna Hart propose une liste de conseils pour bien mener une entrevue. En voici une adaptation :

- être spécifique, plutôt que général;
- ne pas imposer d'outils ou de termes étrangers à l'expert;
- ne pas interrompre l'expert;

- enregistrer tout sur vidéo ou cassette;
- soumettre des cas concrets à l'expert;
- noter comment l'expert utilise son savoir;
- accorder beaucoup d'importance à la manière à laquelle l'expert se comporte, lorsque l'information est incomplète ou non vérifiée;
- faire réfléchir l'expert à voix haute;
- insister sur les cas intéressants, critiques;
- identifier les caractéristiques des décisions;
- distinguer les buts;
- faire usage de diagrammes ou de schémas;
- amener l'expert à formuler des règles;
- reclassifier au besoin (faire du « chainage arrière »);
- faire des mises au point, récapituler.

Lorsque l'entrevue est terminée, on peut immédiatement en faire un bilan sommaire pour dégager les consensus et identifier les ambiguïtés et les informations à obtenir en complément à celles dont on dispose déjà. Si d'autres entrevues s'avèrent nécessaires, il faut prendre rendez-vous avec l'expert et proposer une ébauche du contenu de la rencontre.

Exemple

Dans le cas de PÉPP, les entrevues furent nombreuses pour deux raisons : les facilités d'organisation des rencontres entre experts et « cognitiens », tous professeurs de la même institution, et la complexité du sujet traité qui nécessitait de fréquentes rencontres. Les rencontres furent fréquentes jusqu'aux premières simulations sur ordinateur. Une fois le modèle de conditionnement conçu, la programmation a débuté et les rencontres avec les experts furent plus espacées. L'équipe a appliqué la plupart des techniques d'entrevue identifiées dans cette section. Elle n'a pas enregistré les conversations, mais un membre de l'équipe était chargé de rédiger des comptes rendus exhaustifs afin d'assurer un suivi. De plus, tous les documents afférents à ce projet ont été conservés : comptes rendus, articles de revues, notes personnelles, simulations sur ordinateurs, ébauches de modèles, etc.

Chaque rencontre se déroulait selon un ordre du jour précis et commençait par un point statutaire : suites de la dernière rencontre et état du projet. Le plus souvent, chaque membre de l'équipe avait une série de questions à soumettre aux autres. On a accordé beaucoup d'importance à des questions qui étaient à la base d'un modèle d'entraînement.

Comment caractériser et quantifier la condition physique d'une personne? Peut-on définir une côte? Qu'est-ce qui définit un exercice de conditionnement? Quels sont ses attributs? Comment définir un objectif d'entraînement? En répondant à ces questions, les experts ont fourni les matériaux de base à un modèle de connaissances. Ensuite, il fallait identifier les relations entre les objets. Cela nous a conduits à d'autres questions. Comment tenir compte de la condition physique et de l'objectif d'entraînement dans la sélection d'exercices? Quel est le lien entre condition physique et objectif d'entraînement? Comment associer un exercice à une partie du corps? Les réponses à ces questions nous ont éclairés sur les règles de sélection des exercices à prescrire à une personne. Enfin, on a abordé la posologie et la progression des exercices, c'est-à-dire le rythme ou l'intensité d'exécution d'un exercice selon la phase d'entraînement.

Les cognitiens ont observé les experts en train d'élaborer des programmes de conditionnement physique. Nous leur avons demandé de s'observer eux-mêmes pour qu'ils soient en mesure de décrire leur manière de raisonner. De plus, nous avons procédé à partir de cas types comme celui-ci : une femme d'une trentaine d'années de bonne condition physique veut développer sa force musculaire. Quels sont les groupes musculaires à solliciter? Quels exercices sont appropriés? Quelle est la posologie initiale? Quelle progression appliquer? Quels sont les appareils requis? Une fois une telle mise en situation faite, on demandait aux experts de répondre explicitement à chacune des questions, en les incitant à décrire leur manière d'arriver à des conclusions. À chaque étape du raisonnement, on devait identifier les données requises et les conclusions intermédiaires possibles. Nous proposons des modèles écrits ou sous forme de diagrammes aux experts pour valider notre compréhension de leur manière de raisonner et faire ressortir les liens qu'ils pouvaient créer. Ce procédé a été très enrichissant et productif pour tous les membres du groupe.

4.4 Analyse d'une entrevue

Pendant l'analyse, l'information brute recueillie lors des entrevues est raffinée, éditée et réorganisée. C'est là que le cognitiens organise et structure les connaissances obtenues de l'entrevue. Il y a deux types de connaissances à formaliser : les éléments de savoir et les relations ainsi que la manière dont l'expert utilise ces éléments. En termes de représentations des connaissances, on parle d'objets et d'attributs ainsi que de procédures ou de méthodes. Or, toutes ces entités peuvent être traduites en règles, si c'est la représentation adoptée. L'analyse de l'entrevue constitue donc un premier pas vers la codification en règles des connaissances de l'expert.

La première étape d'analyse consiste à *transcrire les enregistrements* et les notes manuscrites de l'entrevue pour en faire un tout cohérent. On doit toujours *conserver les originaux*, c'est la source de nos connaissances. Le travail d'analyse se fait à partir des transcriptions. Ensuite, on peut procéder à un *inventaire des connaissances* invoquées implicitement ou explicitement dans les transcriptions. À cette étape, il est conseillé de construire progressivement un dictionnaire des connaissances. C'est un outil similaire au dictionnaire des données dans un système conventionnel. Le dictionnaire des connaissances permet un suivi lors du développement et même lors de l'implantation et de l'opération d'un système. Pour chaque article consigné dans ce dictionnaire, Hart suggère d'enregistrer les renseignements suivants :

- nom;
- type (objet, fait, but, procédé, etc.);
- description;
- attributs ou caractéristiques;
- valeurs possibles;
- préalables (conditions d'utilisation);
- buts (conséquences de l'utilisation);
- référence de la source;
- commentaires.

On reconnaît ici la manière de définir un objet par une liste d'attributs. Comme il s'agit d'une approche qui nous est familière, elle facilite la traduction du dictionnaire des connaissances en règles. Le dictionnaire est un document toujours en cours d'élaboration, tout au long du

développement du système. Pour rendre cette tâche plus facile, on utilise des outils informatiques, comme un éditeur ou une base de données.

Avant de passer à l'écriture des règles, on doit traduire les méthodes et les « trucs » utilisés par l'expert pour résoudre des problèmes. Ce sont généralement des marches à suivre que l'on peut codifier sous forme de règles. Enfin, on doit schématiser toutes ces informations pour en montrer la structure et les liens entre chaque élément. Il est intéressant, voire nécessaire, de *traduire en schémas et en diagrammes les connaissances* acquises parce que cela constitue :

- une manière de synthétiser un sujet, d'en dégager les éléments essentiels;
- un excellent moyen de communiquer sans ambiguïté;
- un outil de conception, de simulation et d'expérimentation;
- une représentation intermédiaire entre les données brutes et les règles.

Le cognicien doit toujours commencer une entrevue en présentant, à l'expert, les résultats de l'analyse des entrevues précédentes. De cette manière, l'expert voit comment le cognicien décode ses connaissances et ainsi, il peut mieux apprécier la pertinence des questions et des problèmes qui lui seront soumis. En résumé, lors de l'analyse des entrevues, le cognicien doit effectuer les tâches suivantes :

- transcrire les enregistrements et les notes d'entrevues;
- toujours conserver les originaux;
- constituer un dictionnaire des connaissances;
- traduire en diagrammes les connaissances acquises.

Exemple

L'analyse des entrevues de PÉPP a permis de concevoir un modèle des connaissances basé sur les objets suivants : exercices de conditionnement, objectifs d'entraînement, fiche d'entraînement et fiche de condition physique. Par ailleurs, c'est à cette étape que l'on a élaboré un scénario d'opération du système par ses usagers. Cela a obligés les cogniciens à identifier les fonctions du système. Les experts ont participé à cette analyse. Il en a résulté une description des objets, des attributs et des procédures aux formes variées : diagrammes hiérarchiques, diagrammes de fonctions, diagrammes d'interaction avec le système, tables de décisions, descriptions de fichiers et schémas des relations entre les fichiers. La figure 8 représente les principales fonctions, entités et relations du système PÉPP :

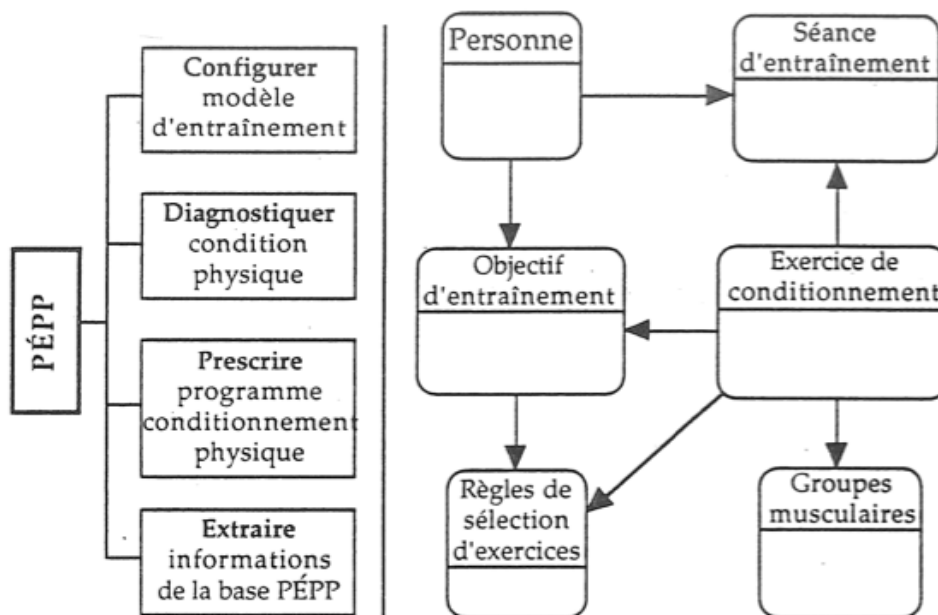


Figure 8 Quelques diagrammes de PÉPP.

On a également rédigé une description du fonctionnement du système projeté, de chacune de ses fonctions avec les données requises, les résultats fournis et le procédé par lequel la fonction produirait le résultat attendu. Nous avons estimé que toutes ces représentations (graphiques et textuelles) étaient complémentaires.

5. Identification et structuration des attributs par la grille répertoire⁶

L'analyse préalable d'un problème et les entrevues avec les experts fournissent des données qu'il faut organiser et structurer. Nous devons procéder par induction pour construire une base de règles qui traduisent correctement et complètement les connaissances d'un domaine. À la section 3, on a présenté une démarche systématique d'induction. La difficulté d'une telle démarche est qu'elle ne fournit aucune garantie quant à l'identification des attributs les plus significatifs et à leur structuration afin que la rédaction des règles soit facilitée. Nous proposons dans cette section une méthode qui permet d'automatiser en partie le processus d'induction en disposant de façon ordonnée les attributs et les exemples qui décrivent un objet. Il s'agit de la grille répertoire ou grille objets-attributs.

5.1 Qu'est-ce que la grille objets-attributs?

Supposons que l'on s'apprête à créer une base de connaissances sur un sujet précis. Le point de départ de notre démarche est la prise en note d'exemples représentatifs du contexte à analyser. Chaque exemple doit illustrer un cas précis d'objet et des attributs qui le caractérisent. On appelle ensemble expérimental ou *échantillon*, ces exemples à partir desquels on essaie d'induire un modèle de connaissances. En nous inspirant du paradigme objet-attributs-valeurs, les paramètres d'un échantillon de données sont :

- N : le nombre total d'exemples ou de cas
- M : le nombre total d'attributs

⁶ Cette section est enrichissement qui peut être omis lors d'une première lecture.

Cas_{*i*} : un exemple
 Attribut_{*j*} : un attribut de l'objet
 Valeur_{*i,j*} : une valeur de l'attribut *j* pour le cas *i*

On peut donc décrire un domaine par des listes de valeurs d'attributs correspondant à des *exemples*. Le sommaire de notre échantillon est consigné dans le tableau 6.

Tableau 6 – Grille objets attributs

	Attribut 1	Attribut 2	Attribut 3	...	Attribut _M
Cas ₁	valeur _{1,1}	valeur _{1,2}	valeur _{1,3}	...	valeur _{1,M}
Cas ₂	valeur _{2,1}	valeur _{2,2}	valeur _{2,3}	...	valeur _{2,M}
Cas ₃	valeur _{3,1}	valeur _{2,3}	valeur _{3,3}	...	valeur _{3,M}
...
Cas _N	valeur _{N,1}	valeur _{N,2}	valeur _{N,3}	...	valeur _{N,M}

Une ligne de cette grille correspond à un cas ou objet. À chaque colonne de ce tableau, on associe un attribut. Une telle approche de représentation d'un contexte ou d'une réalité nous amène à poser trois questions.

1. Selon quels critères détermine-t-on le nombre d'attributs et comment procède-t-on à leur choix?
2. Selon quels critères peut-on *grouper des cas* pour définir des classes dans un échantillon de données?
3. Selon quels critères peut-on *ordonner les attributs* pour reconnaître les plus importants?

La réponse à la première question exige un haut niveau de connaissance du contexte à analyser. Un choix judicieux des attributs conduit à un modèle qui rend bien compte de la réalité. Pour répondre aux autres questions, nous proposons la méthode de la grille objets-attributs ou grille répertoire. En effet, le but de cette approche est de faire des groupements d'attributs et de cas, afin de sélectionner des attributs signifiants et de former des classes d'objets.

5.2 Construction de la grille objets-attributs

Pour construire la grille objets-attributs, on considère les attributs comme des indicateurs bipolaires, c'est-à-dire dont la valeur se situe quelque part entre deux pôles déterminés sur une échelle calibrée arbitrairement. On appelle *max* et *min* les valeurs respectivement maximales et minimales de tous les attributs définis pour caractériser les cas d'un échantillon. Ainsi, on peut apprécier les caractéristiques d'objets sur une échelle de 1 à 7, 1 à 10 ou 1 à 100, selon la situation. Cela revient à dire *qu'une échelle commune sert à fixer les valeurs de tous les attributs*. Bien que l'exigence d'une échelle commune ne soit pas une nécessité, elle permet de systématiser la construction d'une grille objets-attributs et d'en faciliter le traitement ultérieur et son interprétation. On doit donc prendre la précaution de construire la grille objets-attributs en appliquant une échelle commune aux valeurs de tous les attributs.

Exemple

Nous voulons constituer une base de connaissances sur une classification de vertébrés afin de déterminer à quel groupe appartient un vertébré particulier.

Les cas de notre étude sont donc des vertébrés. Pour déterminer leurs attributs, nous nous sommes inspirés de la taxinomie de Linné. Ce naturaliste suédois du XVIII^e siècle a proposé une classification des êtres vivants basée sur les notions de genre et d'espèce. On peut identifier les attributs de ce contexte en consultant des ouvrages de botanique ou de zoologie ou en interrogeant des personnes compétentes sur le sujet. C'est ainsi que l'on peut construire un *échantillon* de cas représentatifs à partir desquels on peut formuler des règles quant à une classification des vertébrés.

Nous avons retenu neuf attributs différents pour caractériser des vertébrés. Quant aux valeurs des attributs, nous jugeons *qu'une échelle commune de 1 à 4* peut assez bien traduire la diversité des cas possibles. Le tableau 7 est un sommaire de notre grille d'analyse.

Tableau 7 – Table des valeurs d'attributs de vertébrés

Valeurs	1	2	3	4
Attributs				
1. <i>Enveloppe</i>	plumes	écailles	lisse	poils
2. <i>Fécondation</i>	interne	externe	–	–
3. <i>Locomotion</i>	ailes	membres	nageoires	anneaux
4. <i>Milieu</i>	terrestre	aquatique	aérien	mixte
5. <i>Régime</i>	carnivore	herbivore	–	–
6. <i>Reproduction</i>	ovipare	poche externe	placentaire	vivipare
7. <i>Respiration</i>	poumons	branchies	–	–
8. <i>Squelette</i>	cartilagineux	osseux	–	–
9. <i>Sang</i>	chaud	froid	–	–

Tous les attributs peuvent prendre l'une des quatre valeurs prévues. Le signe « – » veut dire qu'une valeur ne s'applique pas.

Nous sommes à l'étape où l'on doit colliger des *exemples représentatifs* en vue de construire la grille répertoire. Nous avons répertorié dix-sept exemples de vertébrés avec leurs caractéristiques. Nous les disposons dans la grille objets-attributs du tableau 8.

Tableau 8 – Grille objets-attributs de vertébrés

Attributs	1	2	3	4	5	6	7	8	9
Cas	Envel.	Fécon.	Locom.	Milieu	Régime	Repro.	Respi.	Squel.	Sang
1. Manchot	1	1	3	2	1	1	1	2	1
2. Autruche	1	1	2	1	2	1	1	2	1
3. Pinson	1	1	1	3	2	1	1	2	1
4. Faucon	1	1	1	3	1	1	1	2	1
5. Saumon	2	2	3	2	1	1	2	2	2
6. Requin bleu	2	1	3	2	1	4	2	1	2
7. Mako	2	1	3	2	1	1	2	1	2
8. Ornithorynque	4	1	1	4	2	1	1	2	1
9. Girafe	4	1	2	1	2	3	1	2	1
10. Loup	4	1	2	1	1	3	1	2	1
11. Dauphin	4	1	3	2	1	3	1	2	1
12. Chauve-souris	4	1	1	3	1	3	1	2	1
13. Koala	4	1	1	1	2	2	1	2	1
14. Kangourou	4	1	1	1	2	2	1	2	1
15. Boa	2	1	4	1	1	4	1	2	2
16. Python	2	1	4	1	1	1	1	2	2
17. Grenouille	3	2	2	4	1	1	1	2	2

5.3 Groupement des cas et identification de nouveaux attributs

Nous pouvons regrouper les cas du tableau 8 en calculant leur ressemblance. Par exemple, si on veut calculer l'écart entre le Boa et le Python, on compare les lignes 15 et 16 du tableau précédent et on compte le nombre de fois où les valeurs des attributs diffèrent. Pour 8 attributs sur 9, ce sont les mêmes valeurs; seul l'attribut Reproduction distingue ces deux animaux. On dira qu'ils se ressemblent à $8 \cdot 100/9 = 89\%$. Par contre, en comparant les lignes 3 et 6, on constate que la ressemblance du Pinson et du Requin n'est que de 11 %.

Pour faciliter une lecture globale de la situation, on peut disposer les résultats dans une matrice 17 x 17 comme dans le tableau 9.

Cette matrice comporte deux particularités. Tous les éléments de sa diagonale sont nuls, car l'écart entre un cas et lui-même est nul. De plus, l'écart entre un premier et un deuxième cas égale l'écart entre le deuxième et le premier. Cela veut dire qu'il est nécessaire de calculer seulement les éléments situés au-dessus de la diagonale de la matrice. En général, pour une matrice de M cas, on a $M(M-1)/2$ écarts à calculer; dans notre exemple, $17(17-1)/2 = 136$.

Tableau 9 – Matrice de ressemblance entre les animaux

	Manc.	Autr.	Pins.	Fauc.	Saum.	Requ.	Mako	Orni.	Gira.	Loup	Daup.	Chau.	Koala	Kang.	Boa	Pyth.	Gren.
Manchot	-	67	67	78	44	44	44	56	44	44	78	56	44	44	44	56	44
Autruche	-	-	67	67	22	11	22	67	67	67	44	44	44	56	44	56	44
Pinson	-	-	-	89	22	11	22	78	56	44	44	67	78	67	33	44	33
Faucon	-	-	-	-	33	22	33	67	44	56	56	78	56	56	44	56	44
Saumon	-	-	-	-	-	67	78	22	11	22	44	22	11	11	33	44	56
Requin	-	-	-	-	-	-	89	11	11	11	44	22	11	11	56	44	22
Mako	-	-	-	-	-	-	-	22	11	22	44	22	11	11	44	56	33
Ornithor.	-	-	-	-	-	-	-	-	67	56	56	67	78	78	33	44	44
Girafe	-	-	-	-	-	-	-	-	-	89	67	67	78	78	44	44	33
Loup	-	-	-	-	-	-	-	-	-	-	78	78	67	67	56	56	44
Dauphin	-	-	-	-	-	-	-	-	-	-	-	78	56	56	44	44	33
Chauve-s.	-	-	-	-	-	-	-	-	-	-	-	-	67	67	44	44	33
Koala	-	-	-	-	-	-	-	-	-	-	-	-	-	100	33	33	22
Kangourou	-	-	-	-	-	-	-	-	-	-	-	-	-	-	56	56	22
Boa	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	89	44
Python	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	56
Grenouille	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Il y a évidemment d'autres façons de calculer la ressemblance qu'en comptant le nombre de cas où deux attributs n'ont pas la même valeur. Lorsque les attributs à comparer sont des nombres, on peut faire la somme des différences en valeur absolue des attributs de chaque cas. Quelle que soit la méthode retenue, le calcul de tous les pourcentages de ressemblances entre les cas est une opération assez fastidieuse. Elle peut être grandement facilitée si on utilise un tableur ou un logiciel statistique, pour éditer la grille objets-attributs et utiliser des fonctions de calcul sur des colonnes du même type que les calculs d'écart que nous venons de décrire. On doit donc utiliser des *outils informatiques appropriés* pour raffiner la grille objets-attributs.

Une fois la matrice des ressemblances obtenue, on constate que certains cas se ressemblent à plus de 85 %, d'autres à plus de 65 %, d'autres à plus de 45 %, etc. En se basant sur ces pourcentages, on peut alors tracer le graphique de la figure 9.

Ce graphique permet de classer les cas selon leur ressemblance. On y retrouve des catégories familières :

1. Les oiseaux
2. Les mammifères
3. Les reptiles
4. Les poissons
5. Les animaux à sang chaud
6. Les animaux à sang froid
7. Les vertébrés

Par la suite, rien ne nous empêche de faire de ces catégories de nouveaux attributs qui serviront à classer un nouvel animal. Cet arbre peut devenir un arbre des contraintes où, de proche en proche, on identifie de plus en plus précisément tous les attributs d'un animal inconnu.

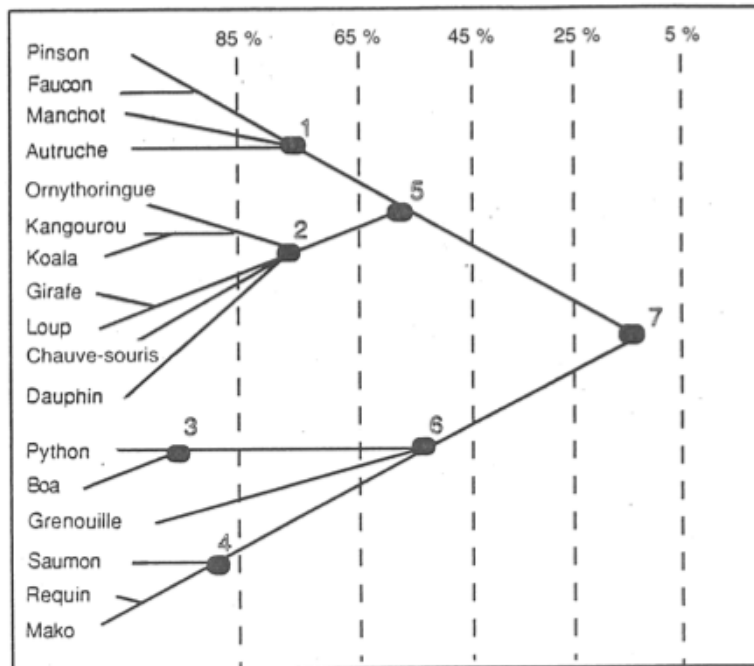


Figure 9 – Arbre des ressemblances entre des animaux.

5.4 Épuration de la grille objets-attributs

Nous venons de voir comment on peut découvrir de nouveaux attributs par regroupement de cas similaires. Le processus inverse, l'élimination d'attributs moins discriminants, est aussi intéressant. Il n'est pas certain que tous les attributs retenus dans la grille originale soient nécessaires. On pourrait en réduire le nombre et ainsi, simplifier notre modèle objets-attributs. En réduisant le nombre d'attributs, on diminue le nombre de règles que l'on doit construire.

Une façon de réduire le nombre d'attributs pour ne retenir que ceux nécessaires est de s'assurer qu'il n'y a pas d'attributs redondants. Des attributs « semblables » sont regroupés pour n'en former qu'un seul.

On peut définir la ressemblance entre attributs en leur associant une « distance » respective, c'est-à-dire d'une façon analogue à ce que nous avons fait pour regrouper les cas dans la section précédente. À partir de la grille du tableau 8, nous nous intéressons cette fois aux colonnes. *L'écart entre deux attributs est une mesure de la différence entre leurs valeurs respectives pour des cas différents.* Deux attributs sont d'autant plus ressemblants que leur écart mutuel est petit.

Cela nous amène à construire la *matrice des écarts* entre les M attributs d'un échantillon. Il s'agit d'une matrice de M lignes et de M colonnes. Nous avons construit la matrice des écarts entre les neuf attributs des animaux, au tableau 10.

Pour obtenir l'écart entre deux attributs, on compare les colonnes correspondantes du tableau 4.8 et on compte le nombre de fois où elles sont différentes. Par exemple, la différence entre l'attribut 1 (Enveloppe) et l'attribut 3 (Locomotion) est obtenue par la comparaison des colonnes 1 et 3. La valeur obtenue, 15, signifie que ces deux attributs diffèrent pour 15 des 17 animaux. Ils sont donc très peu semblables. Par contre, les attributs Respiration et Sang ont un écart de 3, ce qui signifie qu'ils ne diffèrent que pour 3 cas sur 17. Ce sont donc des attributs très semblables.

Une fois la matrice des écarts entre attributs construite, on peut grouper les attributs selon leurs distances mutuelles. Ainsi, on peut *regrouper des attributs dont les écarts mutuels se rapprochent*. C'est une manière de diminuer le nombre d'attributs qui décrivent des objets et de réduire les connaissances qui les décrivent. De plus c'est une excellente façon de faire la synthèse d'une grande quantité de données en identifiant les attributs réellement significatifs.

Appliquons cette technique à notre exemple. Commençons par regrouper les attributs selon leurs écarts mutuels pour éliminer les redondances et identifier les attributs les plus significatifs. Selon la matrice des écarts entre les attributs (tableau 10), les attributs les plus « rapprochés » sont, dans l'ordre : Fécondation et Respiration (3), Respiration et Sang (4), Fécondation et Sang (4), Fécondation et Régime (8), etc.

Nous retrouvons trois attributs qui forment un groupe de caractéristiques très voisines : Fécondation, Respiration et Sang. Tous ces attributs sont-ils nécessaires pour identifier des catégories générales de vertébrés? Si les écarts entre deux attributs sont nuls, on peut en éliminer un. Dans le cas présent, on peut retenir l'attribut Respiration et prévoir des règles pour déduire les autres attributs. Par exemple, quand la respiration est branchiale, le sang est froid. En éliminant pour le moment les deux autres attributs, on obtient une grille épurée, c'est-à-dire qui comporte moins de données que la grille originale.

Tableau 10 – Matrice des écarts entre les attributs de vertébrés

	1	2	3	4	5	6	7	8	9
	Envel.	Fécon.	Locom.	Milieu	Régime	Repro.	Respir.	Squel.	Sang
Envelope		12	15	12	15	13	10	14	8
Fécondation			10	9	8	10	3	13	4
Locomotion				15	13	12	11	13	10
Milieu					14	14	7	14	9
Régime						9	9	9	12
Reproduction							10	14	12
Respiration								16	3
Squelette									13
Sang									

La réduction du nombre d'attributs doit faciliter l'induction des règles pour déterminer le groupe auquel appartient un vertébré particulier. Le passage aux règles s'effectue de la manière décrite dans la section 4 : arbres des contraintes, induction d'ensembles de règles pour chaque contrainte et raffinement des règles.

5.5 Sommaire de la méthode avec la grille objets-attributs

La construction et le raffinement d'une grille objets-attributs sont des étapes qui s'inscrivent dans une démarche d'induction et dont nous nous rappelons les principales.

- Recueillir un échantillon de données représentatives d'un domaine. C'est *l'ensemble expérimental*.
- Identifier les attributs des objets à classifier.
- Définir les valeurs possibles de chaque attribut. Dans le cas où un attribut peut prendre un spectre continu de valeurs, on doit regrouper celles-ci pour former des catégories. Cela revient à convertir un attribut continu en un attribut discret. On prend également soin d'exprimer les valeurs de tous les attributs sur une *échelle commune* pour uniformiser les calculs et faciliter l'interprétation des résultats.
- Disposer l'échantillon des cas (attributs et valeurs) dans une grille objets-attributs.
- Calculer les écarts entre les cas, pour construire la *matrice des écarts entre les cas*, de façon à découvrir des attributs significatifs.
- Calculer les écarts entre les attributs, pour construire la *matrice des écarts entre les attributs*, afin de regrouper les attributs selon leurs écarts mutuels et faire *l'épuration de la grille* objets-attributs.

5.6 Avantages et limites de l'explicitation par grille

La grille est un bon outil de classement et d'organisation d'un ensemble de données attributs-cas décrivant un domaine de connaissances. Elle permet de synthétiser, donc de donner une vue d'ensemble des données brutes. Par la simplicité de son organisation, la grille objets-attributs est utilisable sans explications préalables par plusieurs personnes impliquées dans un projet. Elle est donc un bon outil de communication. Par ailleurs, la grille présente des données dans un format standard pour plusieurs logiciels, comme les tableurs et les progiciels statistiques. Ce sont des outils très répandus, d'usage assez convivial et qui peuvent faciliter grandement le traitement de données en tableau.

Le principal intérêt d'une grille objets-attributs est qu'elle amène les experts à réfléchir attentivement à un problème. Elle permet d'envisager des changements dans les attributs ou les cas, afin d'en arriver à une grille plus représentative de la réalité que l'on veut traduire en règles. De plus elle permet de voir facilement les relations entre les cas et les attributs. Elle permet donc un gain d'information du simple fait qu'elle suppose une structuration et une comparaison des données disposées dans un même tableau. Après avoir contribué à la construction d'une grille, les experts ont probablement davantage d'idées sur la nature des règles. Leur perception du problème est plus claire. La méthode peut aussi être employée pour choisir des attributs et des cas pour l'induction ou pour subdiviser un problème en sous-problèmes plus petits.

Quant aux limites de cet outil, elles tiennent à l'aspect figé et rigide des connaissances confinées dans un tableau. La grille donne, de l'expertise, une idée plus statique que dynamique. C'est un moyen bien adapté à des problèmes de classification et de taxinomie, mais moins adapté pour des problèmes de conception, de planification ou de prédiction. De plus, la grille n'est pas utile pour l'analyse de connaissances procédurales, stratégiques ou pour la description des processus. Enfin, comme la plupart des outils d'induction, la grille ne nous permet jamais d'affirmer que l'on a traité tous les cas ou toutes les relations possibles d'un sujet.

6. Outils et techniques d'induction de règles

L'acquisition des connaissances est souvent décrite comme le goulot d'étranglement du processus de construction d'un système expert. Selon la complexité du domaine, cette étape préliminaire à l'implantation informatique proprement dite peut prendre jusqu'à 6 à 24 mois et peut représenter plus de 75 % du temps consacré au projet.

Voilà pourquoi on s'est intéressé très tôt aux outils informatiques d'acquisition des connaissances. Beaucoup de recherches ont été réalisées sur le sujet et certaines ont conduit à des systèmes commerciaux d'acquisition des connaissances. On espère ainsi réduire le temps et les coûts d'acquisition et augmenter la fiabilité des connaissances, notamment en diversifiant les sources d'expertise.

On peut répartir les outils disponibles sur un continuum. À une extrémité, l'ingénieur de la connaissance interroge ou consulte les sources d'expertise, puis il structure les connaissances manuellement sans intervention directe de l'expert. À l'autre extrémité, des systèmes automatiques ou interactifs d'acquisition sont utilisés directement par les experts. Dans ce cas, l'ingénieur de la connaissance

- conseille les experts sur le processus d'acquisition;
- choisit et configure correctement les outils d'acquisition;
- édite les connaissances avec l'expert pour en faciliter le codage;
- valide les connaissances et prépare les interfaces-usagers en collaboration avec des experts et des usagers.

6.1 Survol des outils d'acquisition des connaissances

Même si la beaucoup de projets sont réalisés manuellement, la quantité d'informations en cause dans tout grand projet fait en sorte que l'on peut difficilement se passer d'outils informatiques qui aident à l'acquisition des connaissances. On peut classer ces outils en six grandes catégories :

- les progiciels d'application;
- les simulations et les prototypes;
- les outils automatiques d'induction;
- les outils interactifs d'entrevue;
- les outils basés sur la grille répertoire;
- les systèmes intégrés d'acquisition.

Les *progiciels d'application* ont cette souplesse et cette facilité d'accès qui les rendent utiles aux diverses phases du processus d'acquisition. Ils peuvent être utilisés par l'ingénieur de la connaissance pour synthétiser l'état des connaissances identifiées jusqu'à présent. Ils peuvent aussi servir à l'expert à décrire une partie de ses connaissances d'une manière facilement modifiable et même utilisable par le système. Ainsi, on utilise un éditeur graphique ou un processeur d'idées pour maintenir à jour l'arbre des contraintes ou les arbres de décision, à travers les multiples séances d'entrevue avec les experts. Le traitement de texte facilite l'écriture des règles sur une contrainte puisque celles-ci ont toutes la même structure générale. Un gestionnaire de base de données ou un tableur peut servir à colliger les exemples qui servent à la construction de connaissances plus générales. Un tableur permet d'ajuster les règles par essai et erreur.

Les *simulations et les prototypes* peuvent être utilisés pour aider l'expert à compléter ou à reformuler les connaissances. Par exemple, en observant un pilote expert qui est en train d'utiliser un simulateur de vol, l'ingénieur de la connaissance peut observer comment il résout concrètement les problèmes qui se posent à lui et en extraire des modèles qui serviront à construire le système à base de connaissances.

Une autre façon de procéder est de mettre à la disposition de l'expert un prototype, même très sommaire, construit à l'aide d'une coquille de systèmes experts. Par son interaction avec la coquille, l'expert peut mieux identifier les contradictions, les incomplétudes et les redondances de la base de connaissances. Beaucoup de coquilles utilisent des méta connaissances pour déceler elles-mêmes ce type de faille dans une base de connaissances et les soumettre à l'expert pour une décision. Certaines coquilles (*Persona! Consultant Plus*, KEE) permettent d'intégrer des images actives, car elles peuvent être modifiées au cours de sessions d'acquisition de connaissances. Ainsi, l'expert peut modifier l'image d'un cadran pour obtenir un résultat plus conforme à ses connaissances, ce qui a pour effet de modifier la règle qui supporte le graphique.

Les travaux en apprentissage des machines ont donné naissance aux *outils automatiques d'induction*. Ce sont des programmes qui découvrent de nouvelles connaissances à partir de cas particuliers. Par exemple, META-DENDRAL (Buchanan et Feigenbaum, 1978) apprend des règles permettant de prédire la décomposition de classes de composés chimiques dans un spectromètre de masse. AM (Lenat, 1977) et BACON (Langley, 1987) sont des programmes qui découvrent des lois à partir d'exemples, le premier en mathématiques, et le second, en chimie.

Plusieurs programmes construisent automatiquement des règles à partir d'exemples fournis par un expert. Par exemple, l'algorithme AQII (Michalski et Chilausky, 1980) a connu des succès encourageants. Un ensemble de règles permettant d'identifier des maladies chez les plants de soja a été induit automatiquement à partir d'exemples décrivant les attributs des plants. Cet ensemble de règles s'est révélé plus efficace en présence d'exemples nouveaux qu'un autre ensemble de règles obtenu en interrogeant l'expert. Plusieurs algorithmes semblables ont été créés et certains sont intégrés dans des coquilles de systèmes experts. Ainsi, l'algorithme ID3 (Quinlan, 1982) fait partie intégrante de la coquille 1st CLASS sur les ordinateurs IBM ou compatibles.

Les *outils interactifs d'entrevue* sont des programmes qui posent des questions à l'expert suivant une méthodologie précise, comme celle proposée dans ce chapitre. On demande à l'expert d'identifier clairement les attributs en cause, leurs valeurs possibles et leurs relations. Dans certains cas, le programme génère des cas possibles pour la condition et demande à l'expert de spécifier la valeur de l'attribut dans la conclusion, obtenant ainsi un ensemble d'exemples qui permettent d'induire des connaissances plus générales.

Les *outils basés sur la grille répertoire* permettent d'appliquer la technique, présentée à la section précédente, d'une façon interactive. Ils sont généralement intégrés dans des systèmes qui permettent également d'interroger d'une façon interactive un ou plusieurs experts, de l'aider à structurer les connaissances et à induire, automatiquement ou avec un mode conversationnel, les connaissances sous forme de règles ou de schémas.

Compte tenu de la diversité des outils d'acquisition et de la complexité de chacun, nous devons nous contenter de décrire, à l'aide d'exemples, deux techniques d'acquisition relativement simples à utiliser. Celles-ci illustrent deux approches différentes de l'acquisition des connaissances sous forme de règles : l'induction interactive et l'induction automatique.

6.2 Tableur comme outil d'induction des règles

Les tableurs comme EXCEL sont des outils fort utilisés. Ils sont faciles d'accès et on peut les utiliser à divers niveaux de complexité. Nous allons nous inspirer d'un exemple, pour illustrer une méthode d'induction interactive des règles utilisant le tableur EXCEL.

Le jury d'une commission d'examen doit, chaque année, se réunir pour décider du résultat des étudiants. Ceux-ci peuvent réussir, échouer ou reprendre les examens qu'ils ont manqués. La décision du jury se base sur les facteurs suivants :

- Nb-échecs (NE) : le nombre d'examens où l'étudiant a échoué.
- Nb-près (NP) : le nombre d'examens échoués où la note était près du seuil de passage.
- Assiduité (ASS) : l'étudiant a-t-il assisté régulièrement aux cours?
- Excuse (EXC) : l'étudiant a-t-il une excuse, par exemple la maladie?

Ces attributs de l'étudiant nous donnent une contrainte entre attributs où le résultat dépend de quatre facteurs. (Voir la figure 10.)

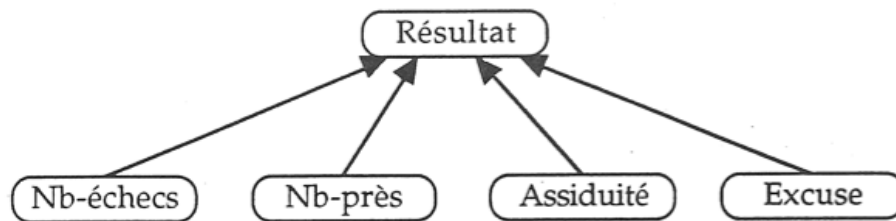


Figure 10 La contrainte Résultat.

Sur la base de cette contrainte, nous devons établir un ensemble de règles permettant de décider du résultat de l'étudiant, dans tous les cas pouvant se présenter. Le tableur au tableau 11 regroupe un certain nombre de cas types d'étudiants.

Une analyse du tableau 11 nous montre que les cas types sont bien répartis pour chacun des attributs. Donc, l'ensemble est assez représentatif et fournit une bonne base pour l'induction des règles. Notons que, si jamais on veut étendre l'ensemble, on peut facilement ajouter des lignes au tableau. Généralement, les cas types ont été insérés dans le tableau dans un ordre quelconque, au hasard des questions. La fonction de tri accessible dans le tableur permet de réordonner les lignes de différentes façons, pour mieux faire ressortir les régularités et faciliter l'induction des règles.

Tableau 11 – Situation des candidats à un examen

Cas	Nb Échecs	Nb Présences	Assiduité	Excuse	RÉSULTAT
1	0	0	bonne	non	succès
2	0	0	bonne	oui	succès
3	3	0	bonne	non	échec
4	3	1	mauvaise	non	échec
5	3	0	bonne	non	échec
6	3	2	bonne	oui	reprise
7	2	1	bonne	non	reprise
8	2	2	bonne	oui	reprise
9	1	0	mauvaise	oui	reprise
10	0	0	mauvaise	oui	succès
11	1	1	bonne	oui	reprise
12	1	1	mauvaise	non	reprise
13	1	0	mauvaise	non	échec
14	3	2	mauvaise	non	échec
15	2	2	bonne	non	reprise
16	2	1	bonne	oui	reprise
17	2	0	mauvaise	non	échec

Le tableau 12, obtenu après plusieurs essais, permet de mieux identifier les règles. En l'analysant attentivement, on peut facilement induire l'ensemble de règles suivant :

1. Si Nb-échecs(étudiant) = 0
Alors Résultat(étudiant) = succès
2. Si Nb-échecs(étudiant) = 1 et Nb-Près(étudiant) = 1
Alors Résultat(étudiant) = reprise
3. Si Nb-échecs(étudiant) = 1 et Nb-Près(étudiant) = 0 et
Excuse(étudiant) = oui
Alors Résultat(étudiant) = reprise
4. Si Nb-échecs(étudiant) = 1 et Nb-Près(étudiant) = 0 et
Excuse(étudiant) = non
Alors Résultat(étudiant) = échec

Tableau 12 – Candidats à un examen classés selon le nombre d'échecs

Cas	Nb Échecs	Nb Présences	Assiduité	Excuse	RÉSULTAT
1	0	0	bonne	non	succès
2	0	0	mauvaise	oui	succès
3	0	0	bonne	oui	succès
4	1	1	mauvaise	non	échec
5	1	0	mauvaise	oui	reprise
6	1	1	mauvaise	non	reprise
7	1	1	bonne	oui	reprise
8	2	2	mauvaise	non	échec
9	2	1	bonne	non	reprise
10	2	1	bonne	oui	reprise
11	2	2	bonne	non	reprise
12	2	2	bonne	oui	reprise
13	3	0	bonne	non	échec
14	3	0	bonne	non	échec
15	3	1	mauvaise	non	échec
16	3	2	mauvaise	non	échec
17	3	2	bonne	oui	reprise

5. Si Nb-échecs(étudiant) = 2 et Nb-Près(étudiant) = 0
Alors Résultat(étudiant) = échec
6. Si Nb-échecs(étudiant).= 2 et Nb-Près(étudiant) = 1 et
Excuse(étudiant) = oui
Alors Résultat = reprise
7. Si Nb-échecs(étudiant) = 2 et Nb-Près(étudiant) = 1 et
Excuse(étudiant) =non
Alors Résultat(étudiant) = échec
8. Si Nb-échecs(étudiant) = 2 et Nb-Près(étudiant) = 2
Alors Résultat(étudiant) = reprise
9. Si Nb-échecs(étudiant) = 3 et Excuse(étudiant) = oui
Alors Résultat(étudiant) = reprise
10. Si Nb-échecs(étudiant) = 3 et Excuse(étudiant) = non
Alors Résultat(étudiant) = échec

On peut maintenant se servir du tableur pour construire un prototype basé sur ces règles. Ce nouveau tableur nous permettra de mettre notre ensemble de règles à l'épreuve et de le corriger au besoin.

Tableau 13 – Test d'un ensemble de règles

CAS À L'ÉTUDE :		Nb-Échecs	Nb-Près	Assiduité	Excuse
		1	0	bonne	non
RÉSULTAT SELON LES DIFFÉRENTES RÈGLES :					RÉSULTAT
Règle 1	SI(NE = 0; "réussir"; "n/a")				n/a
Règle 2	SI(ET(NE = 1; NP = 1); "reprise"; "n/a")				n/a
Règle 3	SI(ET(NE = 1; NP = 0; Exc = "oui"); "reprise"; "n/a")				n/a
Règle 4	SI(ET(NE = 1; NP = 0; Exc = "non"); "reprise"; "n/a")				reprise
Règle 5	SI(ET(NE = 2; NP = 0); "échec"; "n/a")				n/a
Règle 6	SI(ET(NE = 2; NP = 1; Exc = "oui"); "reprise"; "n/a")				n/a
Règle 7	SI(ET(NE = 2; NP = 1; Exc = "non"); "échec"; "n/a")				n/a
Règle 8	SI(ET(NE = 2; NP = 2); "reprise"; "n/a")				n/a
Règle 9	SI(ET(NE = 3; Exc = "oui"); "reprise"; "n/a")				n/a
Règle 10	SI(ET(NE = 3; Exc = "non"); "échec"; "n/a")				n/a

Les valeurs des quatre attributs du cas à l'étude peuvent être modifiées de toutes les façons possibles et le résultat des règles s'affiche. Les règles du tableau 13 sont écrites dans le formalisme du tableur EXCEL :

SI (condition; valeur si oui; valeur si non)

Si la condition de la règle s'applique, le résultat de l'étudiant s'affiche dans la colonne de droite, sinon on obtient « n/a », ce qui nous indique que la règle ne s'applique pas. Voici diverses façons d'utiliser ce tableur :

On passe en revue les 17 cas types initiaux, de façon à nous assurer que nous avons induit les bonnes règles.

En faisant plusieurs essais, on constate qu'il n'y a toujours qu'une règle qui s'applique à la fois. Ce fait nous assure que deux règles ne peuvent être contradictoires.

On peut essayer d'autres cas pour voir si les règles prédisent un résultat correct pour des cas autres que ceux de l'ensemble de cas types, C'est ce que nous avons fait dans le tableau précédent, où la règle 3 s'applique dans un cas nouveau et nous donne un bon résultat.

Dans notre exemple, nous avons 48 cas possibles. De façon générale, le nombre de cas possibles est le produit du nombre de valeurs possibles de chaque attribut. Nous avons 4 attributs qui peuvent prendre respectivement 4, 3, 2 et 2 valeurs possibles. On arrive donc à $(4 \times 3 \times 2 \times 2) = 48$ cas possibles. En vérifiant tous ces cas, on a la certitude que l'ensemble de règles est complet, c'est-à-dire qu'il couvre tous les cas. Dans des problèmes plus complexes, on ne peut énumérer tous les cas possibles, particulièrement lorsqu'un attribut prend une infinité de valeurs réelles. Dans ces cas, il existe maintenant des progiciels qui étendent les possibilités du tableur. On peut obtenir, par exemple, un graphique qui nous indique quels cas sont couverts par l'ensemble des règles. Un tel progiciel guide également l'utilisateur dans la construction d'un ensemble consistant, complet et efficace de règles, en lui offrant des conseils méthodologiques et des outils conçus à cette fin.

6.3 Induction automatique à partir d'exemples

À l'opposé de cette approche interactive où l'utilisateur, l'ingénieur de la connaissance ou l'expert, construit lui-même son ensemble de règles à partir des exemples, plusieurs programmes ont été créés pour construire cet ensemble automatiquement. Certains de ces programmes sont autonomes de l'outil de développement. Par exemple, KNOWLEDGE MAKER ou AUTO-INTELLIGENCE permettent de construire des règles qui sont par la suite intégrées à un outil de développement. D'autres sont intégrés dans des coquilles de systèmes experts. C'est le cas de 1st CLASS qui utilise une variante de l'algorithme ID3 de Quinlan.

Cet algorithme part du principe qu'un ensemble représentatif de cas nous donne les règles implicitement. Lorsque l'ensemble des cas possibles est petit, le problème est simple. Dans l'exemple précédent, on peut donner les 48 cas possibles et écrire une règle par cas. Cependant, un tel système est totalement inefficace. Pour obtenir un *algorithme capable de construire un ensemble réduit mais complet de règles*, observons comment l'ingénieur de la connaissance a procédé dans le cas précédent où l'on a utilisé le tableur. Il a ordonné les cas et a découvert que l'attribut *Nb-échecs était le plus déterminant*, car il sépare bien les valeurs des résultats. Ensuite, dans les cas où plusieurs résultats demeurent possibles, l'attribut *Nb-près* est le second en importance. Enfin, dans les cas qui restent à séparer, l'attribut *Excuse* doit être utilisé, sans qu'on ait besoin d'un recours à l'attribut *Assiduité*.

Comment un programme peut-il lui aussi découvrir un ordre efficace dans lequel utiliser les attributs? La base de l'algorithme est simple. Il faut d'abord identifier le critère selon lequel on peut ranger les cas, c'est-à-dire *l'attribut de classement*. Dans notre exemple, il s'agit de l'attribut *Résultat*. Ensuite, pour chacun des attributs restants et pour chacune des valeurs prises, calculons le nombre relatif pour chaque valeur prise par l'attribut de classement. Cela correspond aux valeurs « succès », « reprise » et « échec » dans notre exemple.

Pour obtenir le premier tableau, on compte le nombre de cas types où on obtient les résultats « succès », « reprise » ou « échec » lorsque le nombre d'échecs est 0; on obtient 3/3, 0/3 ou 0/3. Pour $NE = 1$, on obtient 0/4, 3/4 et 1/4. Pour $NE = 2$, on a 0/5, 4/5 et 1/5. Pour $NE = 3$, on a 0/5, 1/5 et 4/5.

Si on compare ce tableau de l'attribut *NE* avec ceux obtenus de la même façon pour les attributs *NP*, *ASS* et *EXC*, on constate qu'il « sépare » bien les valeurs possibles du résultat. Entre autres, quand $NE = 0$, on est sûr du succès. Quand $NE = 1$, on a 3 chances sur 4 d'obtenir une reprise. Quand $NE = 2$, on a 4 chances sur 5 d'obtenir également une reprise. Et quand $NE = 3$, on a 4 chances sur 5 d'obtenir un échec.

Pour chacun des quatre tableaux du tableau 14, on peut combiner ces probabilités au moyen d'une fonction qui donne, pour chacun des quatre attributs, une évaluation de la façon dont il sépare les valeurs possibles du résultat. On constate que *NE* est le plus « séparateur » avec 0,929, puis ce sera *NP* avec 0,631, puis *ASS* avec 0,473 et enfin *EXC* avec 0,455.

La façon dont on calcule les valeurs pour chaque attribut est facile d'application, mais son explication dépasse le cadre de ce texte. Elle est fondée sur la mesure de Shannon qui permet de calculer la quantité d'information associée à une combinaison de valeurs. Qualitativement, on peut dire que la formule de Shannon fournit une mesure de l'étalement des valeurs possibles d'une variable. Cette mesure est nulle quand une variable ne prend qu'une seule valeur. Dans ce cas limite, la fonction d'information de Shannon nous suggère que la variable ne nous apprend rien sur l'ensemble de cas étudié. Un attribut qui ne prend qu'une seule valeur est inutile. L'autre cas limite, celui pour lequel la mesure de Shannon est maximale, correspond à la situation où un attribut d'un ensemble d'objets prend autant de valeurs différentes qu'il y a d'objets.

Tableau 14 – Ordonnement des attributs avec ID3

	NE = 0	NE = 1	NE = 2	NE = 3	I(NE) =
succès	1	0	0	0	0,929
reprise	0	0,75	0,8	0,2	
échec	0	0,25	0,2	0,8	
	NP = 0	NP = 1	NP = 2		I(NP) =
succès	0,375	0	0		0,631
reprise	0,125	0,8	0,75		
échec	0,5	0,2	0,25		
	ASS = b	ASS = m			I(ASS) =
succès	0,2	0,143			0,473
reprise	0,2	0,571			
échec	0,6	0,286			
	EXC = o	EXC = n			I(EXC) =
succès	0,286	0,1			0,455
reprise	0,714	0,3			
échec	0	0,6			

Pour chacun des quatre tableaux du tableau 14, on peut combiner ces probabilités au moyen d'une fonction qui donne, pour chacun des quatre attributs, une évaluation de la façon dont il sépare les valeurs possibles du résultat. On constate que NE est le plus « séparateur » avec 0,929, puis ce sera NP avec 0,631, puis ASS avec 0,473 et enfin EXC avec 0,455.

La façon dont on calcule les valeurs pour chaque attribut est facile d'application, mais son explication dépasse le cadre de ce texte. Elle est fondée sur la mesure de Shannon qui permet de calculer la quantité d'information associée à une combinaison de valeurs. Qualitativement, on peut dire que la formule de Shannon fournit une mesure de l'étalement des valeurs possibles d'une variable. Cette mesure est nulle quand une variable ne prend qu'une seule valeur. Dans ce cas limite, la fonction d'information de Shannon nous suggère que la variable ne nous apprend rien sur l'ensemble de cas étudié. Un attribut qui ne prend qu'une seule valeur est inutile. L'autre cas limite, celui pour lequel la mesure de Shannon est maximale, correspond à la situation où un attribut d'un ensemble d'objets prend autant de valeurs différentes qu'il y a d'objets.

Pour un tel ensemble, chaque valeur contribue à l'information associée à toutes les combinaisons possibles.

L'algorithme ID3 procède par itération en se concentrant chaque fois sur l'attribut non encore utilisé qui amène le plus d'information.

Choisir un ensemble de cas types. Répéter :

- en utilisant l'attribut le plus séparateur, construire un ensemble de règles basé sur l'attribut ayant la plus grande valeur d'information quant au résultat;
- générer un nouvel ensemble de cas types, ajoutant les exceptions aux règles.

Arrêter dès qu'il n'y a plus d'exception à l'ensemble de règles.

Appliqué à notre exemple, l'algorithme nous donne en trois itérations un ensemble de règles semblable à celui que nous avons généré à l'aide du tableur. (Un astérisque « * » indique une règle non encore valide.)

- Premier attribut : NE
 - SI NE = 0 alors RÉSULTAT = succès
 - * SI NE = 1 alors RÉSULTAT = reprise
 - * SI NE = 2 alors RÉSULTAT = reprise
 - * SI NE = 3 alors RÉSULTAT = échec

- Deuxième attribut : NP
 - SI NE = 0 alors RÉSULTAT = succès
 - SI NE = 1 et NP = 1 alors RÉSULTAT = reprise
 - * SI NE = 1 et NP = 0 alors RÉSULTAT = échec
 - SI NE = 2 et NP = 0 alors RÉSULTAT = échec
 - * SI NE = 2 et NP = 1 alors RÉSULTAT = reprise
 - SI NE = 2 et NP = 2 alors RÉSULTAT = reprise
 - * SI NE = 3 alors RÉSULTAT = échec

- Troisième attribut : EXC
 - SI NE = 0 alors RÉSULTAT = succès
 - SI NE = 1 et NP = 1 alors RÉSULTAT = reprise
 - * SI NE = 1 et NP = 0 alors EXC = oui alors
RÉSULTAT = reprise
 - SI NE = 1 et NP = 0 et EXC = non alors
RÉSULTAT = échec

 - SI NE = 2 et NP = 0 alors RÉSULTAT = échec
 - SI NE = 2 et NP = 1 EXC = oui alors
RÉSULTAT = reprise
 - SI NE = 2 et NP = 1 et EXC = non alors
RÉSULTAT = échec
 - SI NE = 2 et NP = 2 alors RÉSULTAT = reprise
 - SI NE = 3 et EXC = oui alors RÉSULTAT = reprise
 - SI NE = 3 et EXC = non alors RÉSULTAT = échec

Après ces trois itérations, il n'y a plus d'exception à l'ensemble de règles.

Quinlan a démontré, même pour des problèmes beaucoup plus complexes, que l'algorithme ID3 était capable d'induire correctement un ensemble de règles valides pour l'ensemble de cas types. Pour un problème de classification impliquant 14 attributs et près de 2000 cas types, il a obtenu un ensemble correct d'une cinquantaine de règles en seulement 4 itérations. C'est justement dans des cas aussi complexes qu'il est difficile d'induire manuellement et d'une façon correcte un ensemble de règles. On peut donc affirmer que plus la taille de la base de connaissances à construire est grande, plus l'induction automatique des règles est une solution qui donne de bons résultats.

Conclusion

L'importance de la planification, de l'analyse systématique et de l'emploi d'outils appropriés est de mise, quelle que soit la nature de l'application informatique. Dans le cas des SBR et plus généralement des systèmes à base de connaissances, l'accent est mis principalement sur les connaissances à identifier, à acquérir et à représenter. La méthodologie suggérée repose sur une séquence d'étapes à effectuer, de l'étude d'opportunité à l'implantation d'un SBC. Il s'agit d'un *processus par raffinement graduel* : on procède en spirale en reprenant les étapes déjà effectuées. Certains auteurs parlent d'un processus itératif. À chaque passage,

on améliore le travail fait précédemment. C'est ainsi que la construction de la base de règles et sa validation peuvent suggérer des améliorations aux attributs et à leurs contraintes qui produiront à leur tour des modifications aux règles.

L'acquisition des connaissances est certainement l'aspect le plus difficile dans la construction d'un système à base de connaissances. Plusieurs raisons expliquent ce fait :

- On connaît encore mal la nature exacte de l'expertise à modéliser.
- Les modes de représentation des connaissances les plus répandus, les règles et les schémas, ne conviennent pas toujours pour représenter certains aspects de l'expertise.
- Il n'y a pas encore de méthodologie bien établie et validée scientifiquement, sauf dans les grandes lignes.
- Il y a encore peu d'outils informatiques qui facilitent l'acquisition des connaissances; celle-ci se fait encore largement manuellement en utilisant des trucs du métier.
- Il est souvent difficile d'avoir accès aux experts et aux sources d'expertise; des problèmes de disponibilité, de confidentialité et en définitive de coût viennent compliquer la tâche.
- La communication avec les experts est un processus long et difficile, même lorsque l'expert est disponible.

L'identification, la représentation et la formalisation des connaissances constituent les tâches essentielles d'une nouvelle discipline, l'ingénierie de la connaissance pratiquée par les cognitiens. Avec les experts, ce sont les principaux intervenants dans le développement d'un SBR. Les cognitiens sont responsables de la planification et de la réalisation, conjointement avec les experts, des tâches préalables à la formulation sous forme de règles des connaissances d'un domaine.

À retenir

1. La méthodologie de développement d'un système à base de connaissances (SBC) s'apparente à celle des autres types de systèmes informatiques. On procède par raffinement graduel pour chacune des principales étapes de développement comme l'étude d'opportunité, l'identification et l'acquisition des connaissances, la représentation et la codification des connaissances, le prototypage, la validation et l'implantation.
2. On peut recommander le développement d'un SBC lorsqu'il existe un besoin réel d'expertise pour pallier l'absence d'experts humains, lorsque les experts sont accessibles lors du développement, lorsque les connaissances peuvent être d'une façon précise identifiées, circonscrites, représentées et implantées dans une base de connaissances et lorsque le rapport coûts-bénéfices le justifie.
3. L'acquisition des connaissances se fait par des études documentaires, des entrevues et l'application de techniques de représentation des connaissances et d'induction.
4. Les experts et les cognitivistes participent conjointement à l'acquisition des connaissances. Les cognitivistes recueillent les connaissances, les organisent et les représentent sous une forme exécutable pour l'ordinateur. En retour, les experts valident les modèles construits par les cognitivistes.
5. L'entrevue du cognitiviste avec l'expert est une séance de travail qui doit être minutieusement préparée. Elle devrait se dérouler selon un scénario et des objectifs connus de tous les participants. Les résultats d'une entrevue doivent être analysés pour en extraire le maximum de connaissances.
6. Lors de la phase d'acquisition des connaissances, la principale préoccupation du cognitiviste consiste à maintenir la confiance et l'intérêt des experts et à conserver leur collaboration active.
7. L'acquisition des connaissances par induction consiste à appliquer des techniques de découverte de règles à partir d'un ensemble de données brutes ou d'exemples. Les étapes d'induction sont : l'identification des attributs, la constitution d'un échantillon d'exemples représentatifs, le regroupement et l'ordonnement des attributs et la rédaction des règles.
8. La quantité et la diversité des données d'un domaine ainsi que la complexité des traitements à leur appliquer pour en induire des connaissances sont deux raisons majeures qui nous incitent à recourir à des outils informatiques. Les chiffriers, les progiciels statistiques et les générateurs de règles à partir d'exemples sont de bons outils informatiques d'aide à l'acquisition et à la codification des connaissances.