



# THÈSE

En vue de l'obtention du

## DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par Institut National Polytechnique de Toulouse  
Discipline ou spécialité : Systèmes Industriels

---

Présentée et soutenue par Cédric BELER  
Le 14 novembre 2008

### Titre

Modélisation générique d'un retour d'expérience cognitif  
Application à la prévention des risques

---

### JURY

Jean-Pierre Bourey, Professeur des Universités, EC de Lille - Rapporteur  
Thierry Denœux, Professeur des Universités, UT de Compiègne - Rapporteur  
Stéphane Ducasse, Directeur de Recherche, INRIA de Lille - Examineur (Président du Jury)  
Bernard Comet, MEDES Toulouse, Responsable du projet SUP - Examineur  
Laurent Geneste, Professeur des Universités, ENIT - Examineur  
Xavier Desforges, Maître de Conférences, ENIT - Examineur

---

**Ecole doctorale** : Ecole Doctorale Systèmes  
**Unité de recherche** : Laboratoire Génie de Production, ENI de Tarbes  
**Directeur(s) de Thèse** : Laurent Geneste  
**Rapporteurs** : Jean-Pierre Bourey, Professeur des Universités, EC de Lille  
Thierry Denœux, Professeur des Universités, UT de Compiègne

# Remerciements

*Quand vous verrez la lumière au bout du tunnel, priez pour que ce ne soit pas le train (Daniel Lemire).*

Je tiens tout d'abord à remercier mes encadrants de thèse sans qui rien n'aurait été possible : M. Laurent Geneste, directeur de thèse et M. Xavier Desforges, encadrant de thèse. J'ai particulièrement apprécié travailler avec eux, que ce soit pour leurs qualités humaines et scientifiques, leurs disponibilités mais aussi et surtout pour bonne humeur permanente et ce, du début à la fin de la thèse. J'ai vraiment énormément apprécié travailler et évoluer dans ces conditions. La thèse est avant tout un travail personnel mais il est clair que sans eux, je n'aurais pu mener à terme ce travail de longue haleine. J'ai aujourd'hui acquis confiance en mes capacités de chercheur et je leur en suis également très reconnaissant.

J'associe à ces remerciements, Daniel Noyes, directeur du Laboratoire Génie de Production (LGP) de l'ENI de Tarbes (ENIT) qui m'a accueilli depuis le DEA en septembre 2003, ainsi que tous les membres du laboratoire et de l'ENIT. La cordialité des échanges m'a permis de toujours me sentir à l'aise dans ce cadre, ce qui indéniablement a joué dans mon intégration dans le milieu de la recherche. Je remercie également Germain Lacoste, directeur de l'ENIT, qui m'a accepté dans son établissement. Merci aux doctorants, compagnons d'infortune, avec qui nous partageons nos doutes, désillusions, frustrations classiques de thésard. Merci plus particulièrement à Alex, Benoît, Eric, François, Paul, Raymond, Roberta, Simon, Toufik, Valentin...

J'adresse également mes remerciements aux personnes qui ont accepté de participer à mon jury de thèse :

- M. Thierry Denœux, professeur à l'Université Technologique de Compiègne et M. Jean-Pierre Bourey, professeur à l'Ecole Centrale de Lille, qui m'ont fait l'honneur d'être rapporteurs de ma thèse,
- M. Stéphane Ducasse, directeur de recherche à l'INRIA de Lille, pour avoir accepté de présider ce jury,
- M. Bernard Comet, médecin et directeur du MEDES à Toulouse et chef du projet du projet européen SUP, pour avoir accepté de participer à ce jury en tant qu'examinateur.

Je remercie aussi énormément la communauté des Smalltalkers, entre autre l'ESUG (Européan Smalltalk User Group) et en particulier M. Noury Bouraqadi, M. Stéphane. Ducasse et M. Serge Stinckwitch. Le LGP n'étant pas un laboratoire d'informatique, leur soutien m'a permis de ne jamais me sentir isolé. Grace à eux, j'ai pu participer aux conférences ESUG et autres Smalltalk party française. Cette communauté, bien que petite en nombre, regroupe des gens talentueux dont la disponibilité est remarquable.

De façon plus personnelle, je tiens à remercier M. Raymond Houé. Bien que supporter de l'OM, j'ai beaucoup apprécié partager mon bureau avec Raymond. Nos discussions et débats, parfois houleux, portaient aussi bien sur des problématiques scientifiques (en particulier sur la modélisation au sens large) mais aussi sur des sujets de la vie de tous les jours. Je regrette juste qu'il ne m'ait pas accompagné dans la découverte de Smalltalk ! Quoi qu'il en soit, je pense que nous constituons un centre de ressources et de conseils apprécié dans le laboratoire, à savoir le CRCGCL (Centre de Ressources et de Compétences Gadget et Conseils Logiciels) qui est en rapport avec toutes choses à la fois inutiles et donc totalement indispensables de la vie de tous les jours ! A quand un tel axe dans le laboratoire...

A ces remerciements personnels, j'associe Agnès Letouzey, voisine de bureau qui a toujours été présente pour m'accompagner dans les pauses café. Elle est d'ailleurs devenue la fournisseuse officielle des pastilles dont je tairai la marque. Elle fut d'un soutien considérable pour juguler la mauvaise foi inévitable du supporter marseillais précédemment cité ! Aussi, comment ne pas remercier Roberta Affonso, doctorante et voisine du bureau dont le charme et la gentillesse m'ont accompagné depuis le DEA jusqu'à la fin de la thèse. Nos commentaires sur NS sont des moments que je retiens parmi tant d'autres. De même, je remercie M. Bernard Grabot, membre d'honneur du CRCGCL, à la fois pour sa bonne humeur permanente et pour tous les échanges hors-cadre universitaire que nous avons pu avoir.

Je ne saurais terminer ces remerciements sans citer mes proches. Merci à tous de m'avoir soutenu durant ces longues années et surtout de ne jamais m'avoir reproché mon indisponibilité grandissante. Un grand merci à ma famille, mes amis de Tarasteix et à toi Danielle pour m'avoir supporté (dans tous les sens du terme)...

# Table des matières

Remerciements .....	1
Table des matières .....	3
Introduction Générale .....	5
Chapitre I Le retour d'expérience, une gestion des connaissances centrée sur l'expérience	9
I.1. Gestion des Connaissances .....	10
I.1.1. Connaissance et transmission du savoir .....	10
I.1.2. Outils et technologies de la connaissance .....	13
I.1.3. Gestion incrémentale des connaissances .....	16
I.2. Retour d'expérience .....	18
I.2.1. Origines et influences .....	18
I.2.2. Terminologie et définitions .....	19
I.2.3. Mise en œuvre du retour d'expérience .....	22
I.2.4. Les grandes classes du retour d'expérience .....	24
I.2.5. Architecture générique .....	26
I.3. Synthèse .....	31
Chapitre II Modélisation des expériences .....	33
II.1. Représentation des connaissances .....	34
II.1.1. Approches basées sur la logique .....	34
II.1.2. Approche basée sur les <i>frames</i> .....	36
II.1.3. Approches hybrides de représentation des connaissances .....	37
II.1.4. Représentation « orientée objet » des connaissances .....	40
II.2. Modélisation de l'incertitude .....	43
II.2.1. Contexte de la représentation incertaine .....	43
II.2.2. Le modèle des croyances transférables .....	48
II.3. Modèle hybride objet croyance de l'expérience .....	59
II.3.1. Un formalisme hybride objet incertain .....	60
II.3.2. Formalisation du contexte .....	63
II.3.3. Formalisation de l'analyse .....	66
II.4. Synthèse .....	70
Chapitre III Exploitation, recherche et comparaison d'expériences .....	71
III.1. Du RàPC au retour d'expérience .....	71
III.1.2. Le RàPC objet et flou .....	74
III.1.3. Similarité, origine et calcul .....	75
III.2. Similarité attributs-valeurs et objet .....	77
III.2.1. Similarité attribut-valeurs uniforme .....	77
III.2.2. Décomposition similarité locale/globale .....	78
III.2.3. Fonction d'agrégation et modèle de poids .....	78
III.2.4. Mesure de similarité locale pour les attributs numériques .....	80
III.2.5. Mesure de similarité locale pour les listes de valeurs .....	81
III.2.6. Mesure de similarité locale pour les taxonomies .....	82
III.2.7. Mesure de la similarité entre objets .....	83
III.3. Similarité des ensembles .....	85
III.3.1. Similarité ensembliste sans connaissance sur les éléments .....	85
III.3.2. Similarité ensembliste avec connaissance sur les éléments .....	86
III.3.3. Similarité d'attribut multi-valués .....	88
III.4. Similarité sur la base de représentations incertaines .....	90
III.4.1. Similarité et théorie des possibilités .....	90

III.4.2.	Similarité de fonctions de masses .....	92
III.4.3.	Similarité de distribution de croyance avec connaissance de similarité entre éléments.....	95
III.5.	Evaluation de la similarité d'expériences .....	98
III.5.1.	Agrégation de descripteurs simples .....	99
III.5.2.	Généralisation par récursivité pour les descripteurs objet .....	103
III.5.3.	Prise en compte des analyses.....	105
III.6.	Synthèse.....	110
Chapitre IV	Extraction de connaissance Indication du risque .....	111
IV.1.	Définition générale du risque .....	111
IV.1.1.	Formulation générale du risque .....	111
IV.1.2.	Vision causes conséquences.....	112
IV.1.3.	Une modélisation imparfaite et subjective .....	112
IV.1.4.	Les différentes classes de risques .....	113
IV.2.	Hiérarchisation du risque.....	113
IV.2.1.	Notion de gravité et pyramide des risques .....	114
IV.2.2.	Les référentiels de risque.....	116
IV.2.3.	Synthèse .....	118
IV.3.	Proposition d'une mesure du risque par retour d'expérience cognitif .....	119
IV.3.1.	Association entre expérience cognitive et risque .....	119
IV.3.2.	Construction de la mesure.....	121
IV.3.3.	Algorithme d'extraction de la mesure du risque.....	121
IV.4.	Synthèse.....	124
Chapitre V	Application générique de retour d'expérience .....	125
V.1.	Une implémentation dirigée par le modèle de l'expérience .....	125
V.2.	Vue d'ensemble des applications de retour d'expérience .....	128
V.2.1.	Architecture client serveur web.....	128
V.2.2.	Persistance et traitement de l'information.....	130
V.2.3.	Principe des applications et cas d'utilisation .....	130
V.3.	Modélisation générique.....	132
V.3.1.	Patterns .....	132
V.3.2.	Framework.....	135
V.3.3.	Les approches <i>Meta</i> .....	137
V.4.	Modélisation et implémentation.....	139
V.4.1.	Choix d'implémentation .....	139
V.4.2.	Modèle applicatif du retour d'expérience .....	145
V.5.	Application à la sécurité en montagne .....	146
V.5.1.	Le projet SUP.....	146
V.5.2.	Application réalisée.....	148
V.5.3.	Synthèse sur le projet SUP.....	151
Conclusion Générale	.....	153
Table des figures	.....	157
Bibliographie	.....	159

# Introduction Générale

Depuis plusieurs années les entreprises sont incitées à mettre en place des démarches qualité afin de mieux garantir la satisfaction de leurs clients. Dans ces démarches, l'objectif d'amélioration continue est primordial et constitue d'ailleurs un aspect central de la famille de normes ISO 9000 maintenue par l'organisation ISO (International Organization for Standardization). Il se concentre sur l'amélioration de la satisfaction du client au moyen d'améliorations incrémentales des produits, des services et des processus. Une des sources d'amélioration réside dans la capacité, pour l'entreprise, à résoudre les problèmes qui la touchent de manière structurée et systématique en organisant un **processus de résolution de problème**. Ce processus est initié lorsqu'un événement négatif (c'est-à-dire ayant un impact négatif sur le client ou sur l'entreprise) se produit. Il vise à analyser et résoudre le problème courant (aspect curatif) et à en éviter une nouvelle occurrence (aspect préventif).

Différents processus de résolution de problèmes standardisés ont été élaborés et sont implantés dans les entreprises. L'un des plus connus est sans doute le processus Plan-Do-Check-Act (PDCA), également connue sous le nom de « Roue de Deming ». D'autres méthodes sont également largement utilisées comme : 8 Disciplines ou 8 Do (8D) également appelée TOPS (Team-Oriented Problem Solving), Six Sigma ou DMAICS (Define, Measure, Analyze, Improve, Control and Standardize), 7-step... Les activités principales dans ces processus sont :

- la formation d'une équipe de résolution de problème,
- la description et l'évaluation de la criticité de l'événement,
- l'analyse de l'événement par des experts afin d'en rechercher les causes racines,
- la validation de cette analyse par des actions appropriées,
- la proposition d'une solution au problème et la mise en œuvre de cette solution curative),
- la suggestion d'actions pour éviter une nouvelle occurrence du problème (solution préventive, leçons apprises, etc.).

Les processus de résolution de problème sont, en général, assez lourds à mettre en œuvre et ne sont souvent déclenchés que pour la résolution de problèmes complexes (nécessitant un niveau d'expertise élevé) et critiques (ayant un impact très négatif sur le client ou l'entreprise). Pourtant, l'un des défauts majeurs de ces processus réside dans l'impossibilité, a priori, de réutiliser la connaissance engagée pour la résolution d'un problème passé en particulier celle des experts en phase d'analyse. La capacité à capitaliser et à réutiliser cette connaissance « en action » constitue ainsi un moyen puissant d'optimisation du processus de résolution de problème. Nous qualifions d'expérience le fragment de connaissances capitalisées lors du travail des experts pour la résolution d'un problème et de retour d'expérience le processus qui permet d'organiser la capitalisation et l'exploitation de ces expériences.

Nous avons analysé la nature des informations produites lors de l'application des démarches classiques de résolution de problème afin d'identifier les invariants et les aspects divergents. La synthèse de cette analyse est représentée sur le schéma de la Figure Introduction. Nous pouvons constater que toutes les méthodes font intervenir quatre grandes catégories d'informations : le contexte, l'analyse, les solutions et les leçons apprises. Le **contexte** correspond à la description du problème et consigne les informations sur le lieu et la date d'occurrence, sur le produit concerné, sur le client, sur les observations de terrain... L'**analyse** intègre tous les éléments étudiés lors de la recherche des causes racines : cette analyse peut être structurée, sous forme d'arbre des causes par exemple, et peut impliquer des actions de validation des causes racines proposées par les experts. Les **solutions** sont proposées au cours du processus de résolution de problème. La démarche 8D prévoit, par exemple, la mise en place d'actions qualifiées de conservatoires permettant de limiter les conséquences du problème et d'actions correctives permettant de corriger le problème. La rubrique **leçons apprises**

correspond aux enseignements généraux tirés de l'expérience (et, le cas échéant, d'autres expériences antérieures) sous la forme, par exemple, d'actions préventives permettant d'éviter une nouvelle occurrence du problème.

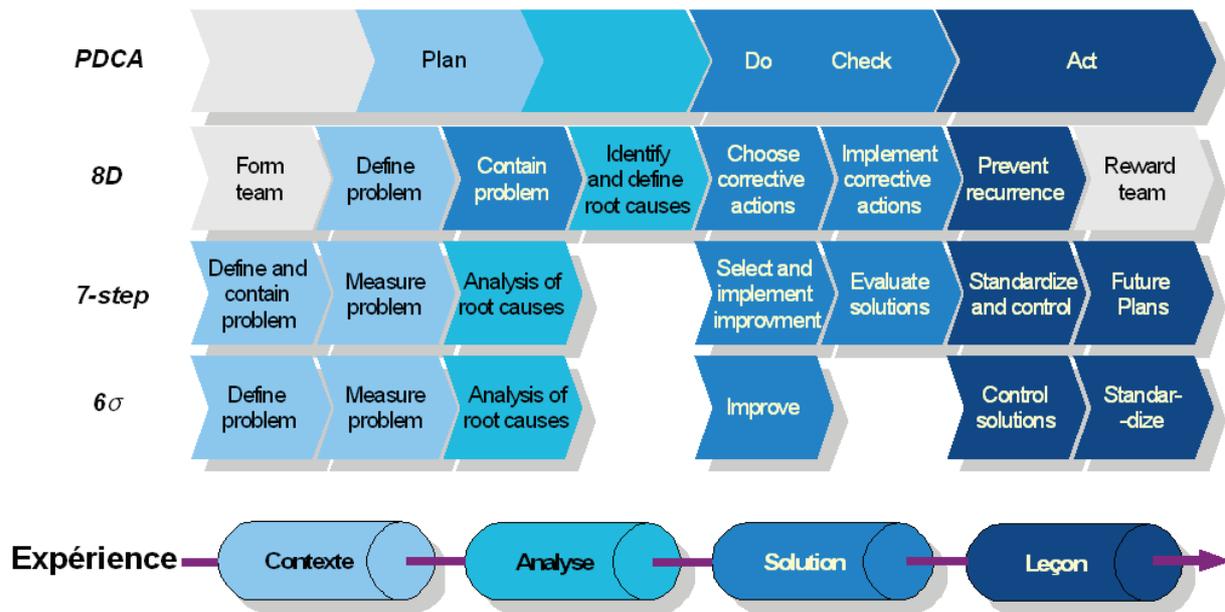


Figure Introduction - Correspondance entre méthodes de résolution de problèmes et retour d'expérience

Le retour d'expérience peut également être perçu comme un support de capitalisation et d'exploitation de connaissances expertes et constitue ainsi une alternative pragmatique aux approches basées sur une modélisation conceptuelle des connaissances (outils de type CommonKADS par exemple (Schreiber *et al.*, 1994) qui nécessitent de longues phases de modélisation, de mise en œuvre et de suivi des modèles de connaissances.

En effet, la gestion des connaissances est aujourd'hui devenue une priorité pour les entreprises et les organisations. Elle vise à éliciter le capital immatériel détenu par les différents acteurs en vue, d'une part, de permettre une meilleure efficacité dans la réalisation des processus et, d'autre part, de faciliter la transmission du savoir et de l'expertise des acteurs. Ces préoccupations sont aujourd'hui très importantes dans un contexte continuellement changeant où le turn-over des acteurs est une réalité omniprésente. Les approches conceptuelles et les outils classiques de gestion des connaissances sont souvent assez lourds à mettre en œuvre et, en particulier, ils demandent un investissement important. Ce type d'approche est dit « descendant » car la connaissance est modélisée de manière générale puis appliquée en contexte. Il existe aussi des approches plus pragmatiques, dites opérationnelles, dont l'objectif est avant tout la collecte de l'information et non son traitement de manière automatisée par un outil informatique (par exemple les approches de type gestion électronique de documents). L'approche par retour d'expérience est située entre ces deux pratiques de gestion des connaissances. Il s'agit selon nous d'une approche pragmatique et opérationnelle de gestion des connaissances qui vise à collecter des les expériences, c'est-à-dire des fragments de connaissances, qui sont des représentations structurées d'un événement ou d'un fait passé auxquelles sont associés des éléments d'analyse ou des descriptions des solutions apportées. Le retour d'expérience peut être perçu comme une approche de gestion des connaissances ascendante, partant d'une information factuelle structurée et analysée pour aboutir à des connaissances plus générales qui pourront être réutilisées par la suite.

Toutefois, la mise en œuvre d'un processus de retour d'expérience performant dans une entreprise nécessite une réflexion approfondie sur la manière de capitaliser les expériences et sur les moyens de les réutiliser. Le premier point (capitalisation des expériences) pose le problème de la caractérisation de la nature des informations contenues dans l'expérience (contexte, analyse des experts, solutions...) et de la structuration de cette expérience. Le deuxième point pose le problème de la réinjection des expériences et des connaissances capitalisées ainsi que de la définition d'outils adaptés.

Naturellement, cette problématique associée aux processus de retour d'expérience n'est pas limitée au domaine de l'entreprise. En effet, nous devons en permanence résoudre des problèmes dans le cadre de nos activités quotidiennes et la capacité à les résoudre de manière structurée, à partager la solution et à anticiper le même type de problème facilite et sécurise la pratique de ces activités. Nous nous sommes ainsi intéressés, dans le cadre du projet européen INTERREG IIIa SUP (Sécurité Urgence Pyrénées), à la prévention des risques pour les activités professionnelles et de loisirs en montagne. Notre rôle dans ce projet a été d'étudier et de caractériser un processus de retour d'expérience applicable à la prévention des risques liés à la pratique d'activités en montagne puis de développer un outil support de ce processus. Nous avons à cet effet réalisé un démonstrateur intégré dans un système d'information global développé pour le projet et basé sur un système d'information géographique (SIG). Si ce domaine d'application peut sembler assez éloigné de la problématique industrielle, nous montrons que les développements proposés peuvent être appliqués dans un contexte d'entreprise. En particulier, nous nous sommes préoccupés de la réalisation d'applications de retour d'expérience génériques, indépendantes du domaine d'application, pouvant être aisément transposées au domaine industriel de résolution de problème. Nous avons souhaité automatiser le plus possible et rendre transparentes les tâches logicielles « support » telles que les interfaces graphiques, la persistance, les algorithmes de recherche et de comparaison, la gestion des utilisateurs et de leurs droits...

Pour fournir des solutions à ce problème, nous avons étudié plusieurs formalismes de représentation des connaissances et sélectionné un formalisme hybridant une approche orientée objet et une représentation de l'incertitude. L'exploitation des expériences est focalisée sur la réutilisation des expériences capitalisées. Cette réutilisation est prévue, d'une part, dans le cadre d'une recherche d'expériences en tenant compte de l'incertitude et de l'analyse des experts et, d'autre part, en construisant un indicateur de risque (une connaissance plus générale) à partir d'un ensemble d'expériences.

Le mémoire de thèse est structuré en cinq chapitres.

Dans le **premier chapitre**, après avoir présenté les notions de connaissance et d'expérience, puis les différentes technologies de la connaissance et les outils de gestion des connaissances, nous introduisons le retour d'expérience en en donnant plusieurs définitions et exemples d'applications. Ensuite, nous détaillons une architecture générique de retour d'expérience qui sert de base à notre application. De manière assez succincte, le processus de retour d'expérience est composé de deux processus majeurs articulés autour d'une base d'expériences. Ces processus sont le processus de capitalisation et d'exploitation de l'expérience. Les différentes dimensions du retour d'expérience sont détaillées : la modélisation de l'information, l'implication des acteurs selon des rôles et des compétences précis, le développement et l'utilisation d'outils particuliers ou encore la définition de sous-processus spécifiques en capitalisation ou en exploitation.

Le **deuxième chapitre** est consacré à la représentation des expériences. Nous avons choisi une structure « objet » (attribut-valeur étendue) de l'expérience. Ce formalisme ne permet pas de réaliser des inférences aussi poussées qu'avec certains langages de représentation des connaissances plus formels mais, en contrepartie, il est plus facile pour les acteurs d'exprimer des connaissances. Nous proposons également d'intégrer la possibilité de représenter des informations incertaines, imprécises et incomplètes pour représenter le plus fidèlement possible des faits, des événements et d'autres informations en relation avec un environnement réel. Après avoir décrit le modèle de représentation de l'expérience sélectionné, nous présentons différents cadres et techniques de représentation des incertitudes. Nous avons choisi le Modèle des Croyances Transférables (Smets et Kennes, 1994) (variante de la théorie de Dempster-Shaffer) que nous avons couplé avec l'approche attributs-valeurs définie pour la structure de l'expérience. Le langage de représentation des expériences est ainsi un formalisme hybride de représentation des connaissances. Concernant l'expression des analyses expertes, nous avons intégré dans le modèle la possibilité pour l'expert de désigner les descripteurs les plus significatifs indispensables à l'explication de l'occurrence de l'événement. Cette analyse correspond à une conjonction de descripteurs à laquelle l'expert peut associer un degré de confiance relatif à son analyse.

Le **troisième chapitre** est consacré à l'aspect exploitation du retour d'expérience. Partant du formalisme de représentation proposé au deuxième chapitre, nous présentons plusieurs algorithmes d'exploitation de la base. Les techniques d'exploitation s'inspirent de mécanismes de recherche employés dans le domaine du Raisonnement à Partir de Cas et, en particulier, sur celles qui reposent sur le calcul de similarité. Nous détaillons dans ce chapitre les méthodes et techniques de calcul de similarité employées pour des formalismes objet ou attributs-valeurs. De plus, étant donné que les valeurs d'attributs sont des distributions de masses de croyance, nous présentons les mesures de similarité ensemblistes et des mesures permettant de comparer deux distributions de masses. De cet état de l'art apparaît qu'il n'existe pas de mesure de similarité couplant fonctions de masses et connaissances de similarité entre les éléments du cadre de discernement (domaine de valeur). Nous avons alors étudié et proposé une mesure de similarité particulière prenant en considération une connaissance de proximité sur les éléments (sous forme d'une matrice de similarité par exemple) et la nature ensembliste (disjonctive) des fonctions de masses. Cette mesure de similarité a ensuite été intégrée dans un algorithme récursif qui constitue le fondement de l'étape de recherche dans la base d'expériences. Nous avons proposé deux variantes de cet algorithme. La première est une recherche relative à certaines informations choisies et pondérées. La seconde est une recherche qui tient compte des analyses expertes réalisées durant la phase de capitalisation.

Le **quatrième chapitre** est consacré à l'extraction d'un indicateur du risque basé sur les expériences antérieures. Après quelques définitions générales du risque, nous proposons d'étendre la description de l'événement de l'expérience avec le niveau de gravité qui le caractérise. Nous présentons également le référentiel du risque qui est une généralisation des référentiels traditionnels « occurrence-gravité » utilisant des courbes iso-risques. Nous proposons un algorithme qui permet de calculer les similarités globales pour chaque gravité correspondante d'événement puis en positionnant ces résultats dans le référentiel adapté du risque d'extraire une mesure globale de risque.

Le **dernier chapitre** est consacré à la mise en œuvre des propositions précédentes. D'une part, nous avons réalisé un prototype pour le projet européen et, d'autre part, nous avons souhaité étendre la proposition à d'autres domaines, en particulier au domaine industriel. C'est pourquoi, nous avons choisi de développer un cadre générique permettant d'opérationnaliser des applications de retour d'expérience par la définition du modèle de l'expérience dans le formalisme attribut-valeurs étendu retenu. Ce choix permet notamment l'automatisation de nombreuses tâches répétitives comme la réalisation des interfaces, de la persistance des expériences, ou encore l'implémentation des algorithmes de recherche. La métamodélisation, métaprogrammation ou encore les *patterns* et les *frameworks* sont des moyens d'atteindre un certain degré de généralité. Nous présentons, après avoir mieux définis les caractéristiques des applications de retour d'expérience, chacun de ces éléments. Le système développé est un *framework* d'application basé sur un métamodèle simple de l'attribut ou plutôt du descripteur (le composant descriptif de base de l'expérience). Le cœur du système peut-être décrit en terme d'architecture dynamique (DOM – Dynamic Object Model). De plus, chaque descripteur, selon son type, contient des méta-informations (ou métadescriptions) qui, en étant interprétées, permettent d'automatiser certains traitements. Les développements ont été réalisés avec le langage Smalltalk. Nous avons également utilisé Magritte qui est un *framework* de métadescriptions et Seaside, un *framework* de développement d'applications Internet. Les technologies applicatives Internet sont aujourd'hui suffisamment développées pour permettre de réaliser à moindre coût des applications distribuées performantes. Par ce moyen, il est même aujourd'hui possible d'envisager des utilisations itinérantes en particulier avec les téléphones mobiles. Enfin, pour terminer ce chapitre, et avant de conclure la thèse, nous présentons l'opérationnalisation de ce cadre générique dans le cadre du projet européen SUP.

# Chapitre I

## Le retour d'expérience, une gestion des connaissances centrée sur l'expérience

La gestion des connaissances est aujourd'hui devenue une priorité pour les entreprises et les organisations. Elle vise à expliciter le capital immatériel détenu par les différents acteurs en vue, d'une part, de permettre une meilleure efficacité dans la réalisation des processus et, d'autre part, de faciliter la transmission du savoir et de l'expertise des personnels. Ces préoccupations sont aujourd'hui très importantes dans un contexte changeant et où la mobilité professionnelle des personnes a un impact très fort (changements de postes, changements de projets, départs massifs en retraite,...). En termes de solutions, les approches conceptuelles et outils classiques de gestion des connaissances sont assez lourds à mettre en œuvre et demandent un investissement important. Il existe également des approches plus pragmatiques, qualifiées d'opérationnelles, dont l'objectif est avant tout la collecte de l'information et non son traitement de manière automatisée par un outil informatique. Le retour d'expérience que nous présentons se situe entre ces deux pratiques de gestion des connaissances. Il s'agit, selon nous, d'une approche pragmatique, opérationnelle de gestion des connaissances qui vise à collecter des fragments de connaissances, les expériences, qui sont des représentations structurées d'un événement ou d'un fait passé auxquelles sont associés des éléments d'analyse ou des descriptions des solutions apportées.

Dans ce chapitre, après avoir présenté plus en détails le domaine de la gestion des connaissances, nous introduisons la notion de retour d'expérience en donnant plusieurs définitions et exemples d'applications. Ensuite, nous détaillons une architecture générique de retour d'expérience qui sert de cadre général aux développements proposés. Globalement, cette architecture décrit le processus de retour d'expérience comme l'association de deux processus majeurs articulés autour d'une base d'expériences. Ces processus sont le processus de **capitalisation** et le processus d'**exploitation** de l'expérience. Leur mise en œuvre repose sur des dimensions support particulières que sont la modélisation de l'information, l'implication des acteurs selon des rôles et des compétences spécifiques, le développement et l'utilisation d'outils dédiés ou encore la définition de sous-processus particuliers que ce soit en capitalisation ou en exploitation. La fin du chapitre est consacrée à l'extension de cette architecture générique par la définition du sous-processus de prévention en complément de celui, plus classique, de résolution de problème. Ce cas d'utilisation particulier est le point de départ des travaux sur l'extraction d'une mesure du risque présentés dans le chapitre IV.

## I.1. Gestion des Connaissances

La notion de Gestion des Connaissances (GC), ou Knowledge Management (KM), couvre un spectre très large qui fait référence à la façon dont les organisations rassemblent, gèrent et utilisent la connaissance (Zorn et Taylor, 2004). Ces termes font aussi référence aux approches visant à améliorer les performances des organisations en introduisant un ensemble spécifique de processus et de techniques pour identifier et capitaliser les connaissances, les savoir-faire, les expertises et autres richesses intellectuelles, pour les rendre disponibles au transfert et à la réutilisation dans l'organisation. (Easterby-Smith et Lyles, 2003). Généralement, les deux éléments fondamentaux d'un processus de gestion des connaissances dans une organisation sont la capitalisation et l'exploitation (Lebowitz, 1999). La capitalisation est le processus qui permet de capturer et stocker la connaissance pertinente du domaine, en vue de la réutilisation dans de nouvelles tâches (Dalkir, 2005). L'exploitation des connaissances est en rapport avec la diffusion des connaissances. Soit elles sont réinjectées dans les processus opérationnels pour en améliorer le déroulement, soit elles servent à accélérer la formation de futurs acteurs. Une difficulté majeure de toute approche de gestion des connaissances est son acceptation puis son adoption au sein des organisations. Pour faciliter cette intégration, deux pistes complémentaires peuvent être envisagées :

- la motivation des acteurs qui, par leur implication, vont permettre au système de se développer. Cette piste entre dans le champ des sciences sociales et nous n'aborderons pas cet aspect de la problématique car la motivation et l'implication du personnel ne peut être assurée que par des démarches de management qui sortent de notre champ de réflexion.
- le développement de processus standardisés et d'outils support adaptés aux réalités dans lesquelles évolue l'organisation afin de faciliter leur acceptation. Les acteurs manquent en effet souvent de temps pour être contributeurs et ne trouvent pas d'informations pertinentes disponibles (notamment en raison du manque de contributeurs...). La mise en place de processus et d'outils peu intrusifs pourra faciliter la capitalisation et justifier la pertinence de l'exploitation des connaissances.

C'est dans ce dernier cadre que se situe notre étude. Il concerne donc en priorité **l'instrumentation des processus de retour d'expérience** et s'inscrit ainsi dans le cadre de l'ingénierie des connaissances. C'est pourquoi, après avoir présenté la notion de connaissance d'un point de vue informationnel et précisé son lien avec la notion d'expérience, nous présentons succinctement les principales techniques d'ingénierie des connaissances et nous faisons en particulier une distinction entre les méthodes visant à une conceptualisation des connaissances et les méthodes basées sur la capitalisation de l'expérience.

### I.1.1. Connaissance et transmission du savoir

La connaissance est une notion à spectre large, difficile à cerner dans sa globalité. Nous nous limitons ici à la définition des connaissances modélisées dans les Systèmes d'Information (SI). Il s'agit une forme d'information riche et structurée (le lien entre données, information et connaissance d'un point de vue opérationnel est précisé dans la partie I.1.1.2).

#### I.1.1.1. Typologies des connaissances

Sous l'angle de la gestion des connaissances, et d'un point de vue pratique (Prax, 1997), la connaissance est une combinaison d'informations ou d'observations, de leur interprétation par des hommes qui puisent dans leur expérience personnelle et/ou collective, de modèles, de théories ou de croyances qui donnent leur sens à ces informations. La connaissance peut ainsi être vue comme une grille d'analyse des informations qui va permettre, en leur attribuant une interprétation, d'inférer, de

prévoir une situation qui en découle. Nous précisons ici brièvement trois typologies usuelles des connaissances en entreprise.

### **Connaissance tacite et connaissance explicite**

En psychologie cognitive, deux grands types de connaissances sont distingués : les connaissances tacites et les connaissances explicites. Cette distinction est reprise en gestion des connaissances. Elle a été proposée par Polanyi en 1962 (Polanyi, 1962) puis reprise par Hedlund et Nonaka (Hedlund et Nonaka, 1993) et par Nonaka et Takeuchi (Nonaka et Takeuchi, 1995).

Les connaissances tacites sont les connaissances qui appartiennent au monde des objets mentaux, des représentations mentales. Elles regroupent les compétences innées ou acquises, le savoir-faire et l'expérience. Elles sont généralement difficiles à formaliser et à transmettre par opposition aux connaissances explicites. Les connaissances explicites sont formalisées et peuvent être transmises au moyen d'un écrit ou d'un discours. Elles sont donc faciles à articuler et à transmettre : un vocabulaire analytique et rationnel, en mots précis et en valeurs significatives est utilisé. Ce type de connaissances peut être transmis au moyen de supports informatisés et entreposé dans des bases de données ou des bases de connaissances. Dans l'entreprise, ce sont, par exemple, les procédures et les règles rédigées pour la conduite des processus.

### **Connaissance individuelle et connaissance collective**

Chaque individu possède un ensemble de connaissances qu'il gère de manière intuitive et subjective et que l'on nomme connaissances individuelles. D'autres connaissances sont distribuées ou partagées entre différents individus, ce sont les connaissances collectives (Duizabo et Guillaume, 1996).

La connaissance individuelle permet à l'individu d'extraire de l'information pour lui donner du sens. En retour, ce processus met à jour la connaissance de l'individu. Au niveau de la connaissance collective, l'information rapportée par l'individu est collectivisée (avec la formalisation par exemple) pour produire un sens collectif désigné par l'expression « connaissances des entreprises » dans (Prax, 1997) ou parfois « référentiel d'entreprise ».

### **Savoir et savoir-faire – connaissance déclarative et procédurale**

Dans l'entreprise, la distinction est généralement faite entre connaissances déclaratives (statiques) et procédurales (dynamiques). Elle reprend la distinction plus ancienne entre savoirs et savoir-faire (Ermine *et al.*, 1996). Les connaissances déclaratives « donnent des informations sur les objets (réels ou hypothétiques) du monde » et les connaissances procédurales « donnent des indications sur les procédures et les conditions d'utilisation de ces procédures » (Weil- Barais, 1994).

#### **I.1.1.2. Information, Expérience et Connaissance**

Le retour d'expérience manipule des expériences mais aussi des données, des informations, et des connaissances comme tous les outils de gestion des connaissances. Afin de clarifier les développements, il est nécessaire de préciser les liens et le positionnement respectif des différents concepts utilisés. En Ingénierie des Connaissances, la distinction est traditionnellement faite, selon le niveau de structuration et les possibilités d'inférences associées, entre les notions de données, d'informations et de connaissances comme cela est représenté sur la Figure I-1.

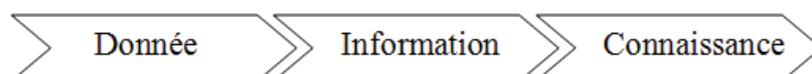


Figure I-1 – Distinction donnée/Information/Connaissance

- Le niveau « Donnée » représente des données brutes, des données atomiques, c'est-à-dire des faits ou des éléments non reliés entre eux, sans valeur informative en tant que tels car hors contexte.

- Le niveau « Information » représente des ensembles de données agrégées, mises en relation. Il y a une notion de contexte. Les données sont associées par des relations qui ajoutent de la valeur informative sous la forme d'une structuration. Ces données peuvent être regroupées de façon cohérente dans des bases de données. Un exemple d'information simple est une température de 10°C mesurée à Tarbes.
- Le niveau « Connaissance » correspond à des informations à forte valeur ajoutée qui permettent de générer, d'inférer de nouvelles informations. Une forme relativement simple de connaissance informatique est la règle de production utilisée dans les systèmes experts (par exemple, la règle suivante : «SI la température est inférieure à 10°C ALORS il fait froid »).

On trouve parfois, au delà du niveau « Connaissance », un niveau relatif au « savoir-faire », c'est-à-dire relatif à la manière de mobiliser les connaissances de façon pertinente pour résoudre un problème ou atteindre un objectif. Ce niveau est nommé « Sagesse » (Wisdom en anglais). Il est en relation avec l'expérience individuelle accumulée et avec la notion de compétence.

Afin de positionner l'expérience dans ce cadre, nous considérons le lien implicite qui existe entre expérience et connaissance. Bergmann mentionne dans (Bergmann, 2002) que la gestion de l'expérience est une forme de gestion des connaissances restreintes aux connaissances issues de l'expérience. L'expérience est construite à partir d'informations sélectionnées et éventuellement analysées, interprétées par des experts. L'expérience contient donc de la connaissance, plus ou moins explicite, acquise dans l'action. Il s'agit d'une connaissance opérationnelle. La connaissance peut être considérée comme un ensemble de savoirs plus généraux englobant les connaissances issues de l'expérience. Nous situons donc l'expérience comme une entité intermédiaire entre information et connaissance (Figure I-2). Nous considérons l'expérience comme un fragment de connaissance, un peu comme une pièce du puzzle de la connaissance, et, par conséquent, comme un vecteur de production de connaissances. Le niveau « Connaissance » peut ainsi être enrichi par la généralisation de fragments de connaissances capitalisés dans des ensembles d'expériences.



Figure I-2 - Positionnement de l'expérience dans le triptyque Donnée/Information/Connaissance

### I.1.1.3. Dynamique des Connaissances

La transmission du savoir et du savoir-faire implique une transformation de la connaissance. Le terme « connaissance » fait ici référence à l'ensemble des connaissances implicites, explicites, individuelles et collectives. Le passage d'un état à l'autre est connu sous le nom de dynamique des connaissances (Baumard, 1996). Sur la Figure I-3 les transitions possibles entre les différents types de connaissances sont recensées et nommées.

Les connaissances tacites et explicites peuvent être détenues de façon individuelle ou collective. Nonaka définit quatre états de connaissance (Nonaka, 1969) :

- la connaissance tacite individuelle, qui peut se trouver dans les schémas mentaux, les savoir-faire, les habitudes et la connaissance abstraite des individus (intuition),
- la connaissance explicite individuelle, qui concerne les connaissances et les compétences pouvant être facilement enseignées ou écrites, liées à l'expertise,
- la connaissance explicite collective, qui concerne les procédures opérationnelles, la documentation, les systèmes d'information et les règles,
- la connaissance tacite collective, qui réside typiquement dans les schémas de management, dans les consensus organisationnels sur les expériences passées ou dans la culture de l'entreprise (pratiques sociales).

Le retour d'expérience constitue un domaine particulier de la gestion des connaissances et permet, par conséquent, de transmettre un savoir. L'expérience apparaît comme la transcription de

connaissances contextualisées. La connaissance représente une généralisation de ces éléments particuliers. En d'autres termes, l'expérience est un épisode précis qui retranscrit la connaissance des acteurs impliqués. Les connaissances, quant à elles, sont l'ensemble des règles, des invariants qui peuvent être extraits des expériences ou construites indépendamment.

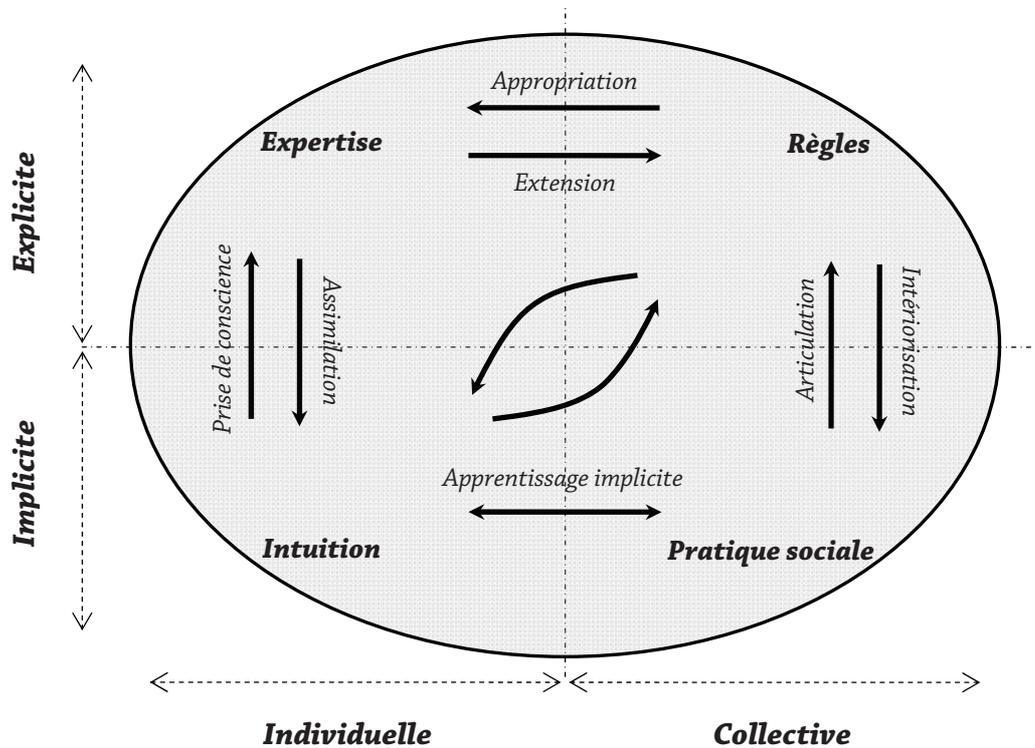


Figure I-3 - Dynamique des connaissances, d'après (Baumard, 1996)

Le retour d'expérience intègre savoir et savoir-faire et les connecte. Le savoir et le savoir-faire sont, en quelque sorte, instanciés dans chaque expérience. Aussi, l'un des enjeux du retour d'expérience est de favoriser le passage de la connaissance tacite à la connaissance explicite (Rakoto, 2004). Enfin, le passage entre connaissance individuelle et collective est typiquement produit par le retour d'expérience. Pour traiter un événement, les acteurs disposent de connaissances individuelles et collectives à partir desquelles ils génèrent des connaissances individuelles. Puis, cette connaissance est formalisée dans un langage commun de communication et devient ainsi une connaissance collective. A l'instar du passage des connaissances tacites aux connaissances explicites, le retour d'expérience favorise la conversion des connaissances individuelles en connaissances collectives dans les phases de capitalisation et d'exploitation.

En résumé, le retour d'expérience permet de faciliter la dynamique des connaissances sur plusieurs registres (cf. Figure I-3) :

- l'assimilation : passage de la connaissance explicite individuelle à la connaissance implicite individuelle,
- l'extension : passage de la connaissance explicite individuelle à la connaissance explicite collective,
- l'intériorisation : passage de la connaissance explicite collective à l'expérience implicite collective.

### I.1.2. Outils et technologies de la connaissance

Afin de situer notre approche, nous faisons la distinction entre les outils de gestion conceptuelle qui raisonnent sur la connaissance et les outils de gestion opérationnelle qui organisent la connaissance. Nous présentons ci-dessous les différentes technologies et outils de gestion des connaissances

indépendamment de l'aspect opérationnel/conceptuel. Nous montrons ensuite que le retour d'expérience est une approche conceptuelle. Les technologies de la connaissance se déclinent sur différents axes de développement :

- L'ingénierie des connaissances, issue de travaux en Intelligence Artificielle, concerne la construction de systèmes informatiques destinés à résoudre des problèmes de manière similaire à l'être humain,
- L'ingénierie basée sur la connaissance qui provient du domaine de la Conception Assistée par Ordinateur (CAO) et est en rapport avec la construction de systèmes qui assistent les ingénieurs (généralement les concepteurs) pour leur permettre plus d'efficacité dans leur travail,
- L'ingénierie ontologique qui a émergé de l'ingénierie des connaissances, de la philosophie et de l'informatique et qui est en rapport avec la construction des structures de connaissances (support d'expression) permettant l'interopérabilité entre différents systèmes informatiques et entrepôts.

Les champs d'application des technologies de la connaissance étant très nombreux, afin de bien positionner notre contribution dans l'univers de ces technologies, nous reprenons la classification présentée dans (Milton, 2008) et invitons le lecteur à se reporter à cet ouvrage pour une présentation complète des différentes techniques et pratiques. Une synthèse des usages et objectifs de ces technologies des connaissances est donnée dans le Tableau I :

<b>Usage</b>	<b>Exemples de technologies des connaissances utilisées</b>	<b>Objectifs</b>
Capitaliser et partager l'expertise d'une organisation	Portail Internet de Connaissances	Améliorer l'efficacité en apportant aux non-experts l'expertise nécessaire pour mieux réaliser une tâche
<b>Automatiser des tâches normalement réalisées par des experts</b>	<b>Systemes à Bases de Connaissances</b>	<b>Réduire le coût et le temps des processus métier et permettre aux experts d'être plus efficace</b>
Assurer l'interopérabilité de Systemes d'Information hétérogènes	Ontologies, Technologies sémantiques	Réduire les inefficacités engendrées par plusieurs bases de données et systemes hétérogènes
Capitaliser et stocker de la connaissance pour des besoins futurs	Portail Internet de Connaissances	Stopper la perte des connaissances quand les experts quittent l'organisation, et mieux prendre en compte les leçons apprises
Donner des moyens d'accès intelligent à l'information stockée dans les bases de données et les sites web	Technologies sémantiques, Ontologies	Apporter une utilisation efficace des technologies de l'information et des ressources web en filtrant et en présentant l'information de manière spécifique pour chaque utilisateur

Tableau I-1 - Usage des technologies de la connaissance d'après (Milton, 2008)

### **I.1.2.1. Approche conceptuelle et approche opérationnelle**

Nous avons mis en évidence dans le tableau de synthèse les Systemes à Base de Connaissances (SBC) (cf. Tableau I-1) car ce sont des outils servant à automatiser au moins en partie les tâches habituellement réalisées par des experts ce qui correspond à un des objectifs du retour d'expérience. Ces outils sont réalisés selon des approches de gestion conceptuelle des connaissances qui implique une

modélisation structurée de la connaissance en vue de l'application de traitements automatiques (raisonnements). Au contraire, les approches dites « opérationnelles » visent principalement l'organisation de l'information (Portail de connaissance, Gestion documentaire) en vue de faciliter sa restitution (regroupement par thèmes, recherche par mot-clés...). L'information peut aussi être davantage structurée en vue de faciliter l'interopérabilité et le partage entre applications (en utilisant par exemple des technologies sémantiques ou des ontologies).

Dans les approches conceptuelles, la connaissance est structurée de manière à éviter toute ambiguïté pour l'utilisateur et surtout pour le système. La structure de la connaissance, c'est-à-dire le « langage » qui sert de support à son expression, doit alors être précisément définie. On parle alors, pour cette caractéristique, de « transparence sémantique ». C'est la condition qui permet d'automatiser les raisonnements informatiques permettant l'inférence de connaissances. Les ontologies sont actuellement des supports privilégiés de la transparence sémantique. Nous précisons, dans la partie suivante, les contours de cette notion d'ontologie avant d'établir les liens possibles entre ontologies et retour d'expérience.

### I.1.2.2. Ontologie

L'ontologie, au sens large, est un terme employé pour désigner la définition du vocabulaire utilisé par des applications. Son sens varie suivant le domaine d'application et correspond à des définitions plus ou moins précises du point de vue formel. Etymologiquement, il s'agit de la « théorie de l'existence », c'est à dire la théorie qui tente d'expliquer les concepts qui existent dans le monde et comment ces concepts s'imbriquent et s'organisent. Dans les années 80, le terme a été repris par les chercheurs en intelligence artificielle et en ingénierie des connaissances pour décrire la connaissance commune à plusieurs domaines. En ce sens, une simple taxonomie était considérée comme une ontologie, même si celle-ci est a priori incomplète et peu précise. Dans les années 90, la définition a été précisée pour devenir la spécification formelle des conceptualisations (Gruber, 1993) ou encore la spécification formelle et explicite d'une conceptualisation partagée (Studer *et al.*, 1998).

Les ontologies sont utilisées par les chercheurs en science de l'information comme une structure hiérarchique sur les objets de la connaissance. A cette structuration est associé un ensemble de relations, de restrictions et/ou d'exceptions souvent exprimées par des formules logiques (prédicats). Depuis les travaux de Nebel (Nebel, 1990), il est d'usage de distinguer les T-Box pour ce qui est relatif à la taxonomie des A-Box pour ce qui est en rapport avec les assertions. Depuis la fin des années 90 et le développement du web sémantique et des technologies associées, la notion d'ontologie est fortement liée aux langages de représentation des ontologies dédiées au web sémantique (par exemple RDFS ou OWL).

Une définition générique et assez complète est donnée par le groupement W3C<sup>1</sup> (The World Wide Web Consortium) : « Une ontologie définit les termes utilisés pour décrire un espace de connaissance. Les ontologies sont utilisées par des gens, des bases de données et par des applications qui nécessitent de partager des informations concernant un domaine particulier (un domaine est uniquement un aspect spécifique ou une zone de connaissance, comme la médecine, la fabrication d'outils, l'immobilier, la réparation automobile, la gestion financière...). Les ontologies incluent des définitions utilisables par des ordinateurs de concepts de base d'un domaine ainsi que les relations associées<sup>2</sup>.

Nous employons le terme ontologie simplement comme la définition d'un vocabulaire partagé entre applications et utilisateur. Une ontologie est focalisée sur un domaine particulier et possède en principe une structure formelle qui la différencie des bases de connaissance. En pratique les ontologies prennent une des formes suivantes :

- une taxonomie (un arbre de classes) de concepts qui comporte éventuellement des associations d'attributs, ainsi que des relations restrictives entre objets des classes,
- un vocabulaire qui possède une grammaire et une structure,

---

<sup>1</sup> <http://www.w3.org/>

<sup>2</sup> <http://www.w3.org/TR/webont-req>

- un schéma de base de données qui définit une vue extrêmement commune d'un domaine.

Pour Fensel (Fensel, 2001), une ontologie est un schéma de base de données du 21<sup>ème</sup> siècle qui possède :

- un langage plus riche que les schémas classiques de base de données,
- une structure qui n'est pas sous forme de tables car cette contrainte est trop restrictive,
- une terminologie et une conceptualisation partagées par plusieurs experts de domaine.

Le retour d'expérience, tel que nous le définissons, est une démarche de gestion conceptuelle des connaissances. A l'instar d'un Système à Base de Connaissances, ce type de démarche peut s'appuyer sur des ontologies particulières, à savoir :

- l'ontologie du retour d'expérience, relative à la structuration de l'entité expérience et aux processus et traitements qui la transforment,
- l'ontologie du domaine, relative à la définition des termes du monde extérieur, qui est employée pour définir des termes du domaine d'application du retour d'expérience.

### I.1.3. Gestion incrémentale des connaissances

Parmi les approches de gestion des connaissances nous pouvons établir une distinction entre les méthodes ascendantes, qui généralisent des fragments de connaissance, et les méthodes descendantes, qui appliquent un modèle générique de connaissance à un domaine particulier.

#### I.1.3.1. Méthode descendante de gestion des connaissances

La plupart des méthodologies de conception des Systèmes à Base de Connaissances repose sur des processus séquentiels dont le but est de produire des systèmes complexes et fonctionnels de manière systématique. Il s'agit d'approches descendantes de gestion des connaissances. Des guides variés indiquent la démarche à suivre, étape par étape, permettant de passer progressivement des exigences informelles au modèle « exécutable ». La méthodologie de CommonKADS (Schreiber *et al.*, 1994) est une des plus connues basée sur ce principe. Elle est le résultat de plusieurs projets de recherche européens ESPRIT (P1098, P3178 et P2548) qui ont donné lieu, en particulier, au projet KADS (Schreiber *et al.*, 1993) puis CommonKADS. A l'origine dédiées à la création de Systèmes à Base de Connaissances, et donc tournées vers des problématiques de résolution de problèmes ou d'aide à la décision, les méthodologies de conception des Systèmes à Base de Connaissances sont aujourd'hui davantage orientées vers des problèmes de Gestion des Connaissances dans le cadre des entreprises. Les méthodologies les plus répandues sont KADS (Schreiber *et al.*, 1993), CommonKADS (Schreiber *et al.*, 1994), MKSM (Ermine *et al.*, 1996) et KOD (Vogel, 1988). Toutes ces méthodologies proposent différents modèles de représentation des connaissances et des supports méthodologiques pour la mise en place de systèmes de gestion des connaissances.

Ces méthodologies ont pour point commun d'être basées sur des modélisations conceptuelles de l'activité d'acteurs particuliers : les experts (Rakoto, 2004). Ce sont eux qui, par hypothèse, détiennent les connaissances clé des organisations. Un nouvel acteur, le cogniticien ou ingénieur des connaissances, intervient pour aider l'expert à modéliser et retranscrire certaines de ses connaissances dans un langage le plus souvent formel. Une fois la connaissance modélisée, elle peut être réinjectée dans les processus de l'entreprise. Ces démarches considèrent la construction d'un Système à Base de Connaissances comme un processus de modélisation. Le processus consiste à capturer et mémoriser les connaissances que les acteurs utilisent pour effectuer une tâche particulière (conception, résolution de problème...). Une fois qu'un type d'application est choisi, une série de modèles est développée, transformant progressivement les exigences du monde réel et l'expertise en un système opérationnel. Nous qualifions dans ce cas l'approche de « dynamique » car de nouvelles connaissances sont générées à l'aide d'outils qui proposent des mécanismes d'inférence qui s'appuient sur une connaissance structurée. Au moyen de traitements appropriés (faisant appel à des techniques d'intelligence artificielle le plus souvent), des

connaissances sont réutilisées et/ou inférées. Le système assiste partiellement l'acteur dans sa tâche d'acquisition et de réutilisation des connaissances.

Les méthodologies, orientées vers la conceptualisation des connaissances, demandent des efforts conséquents dont les bénéfices ne sont pas immédiats. Plusieurs difficultés apparaissent lors de leur mise en place. Citons par exemple le manque de disponibilité des experts, assez souvent accompagné d'un manque de motivation pour livrer leurs savoirs et savoir-faire qui fondent une partie de leur reconnaissance. En outre, la mise en place de tels systèmes est un travail lourd et de longue haleine qui engage fortement l'organisation. L'apprentissage des outils est long et souvent complexe. Enfin, l'évolution des connaissances nécessite une mise à jour régulière des modèles. Les inconvénients des méthodes de type CommonKADS sont qu'elles nécessitent:

- l'utilisation de techniques de modélisation conceptuelle, qui sont difficiles à s'approprier et qui sont fortement consommatrices de temps et de ressources humaines (Duribreux *et al.*, 2000) (Knublauch, 2002),
- l'introduction d'un cognicien, nouvel acteur de l'entreprise, dont le rôle est d'aider les experts à décrire leurs connaissances et qui implique des investissements spécifiques (Dupuis-Hepner 99),
- l'extraction hors contexte de la connaissance des acteurs qui demande des efforts d'abstraction,
- une actualisation des connaissances par de nouveaux audits d'experts,
- une traduction des modèles pour les rendre exploitables par les acteurs, experts ou non.

### **1.1.3.2. Méthode ascendante de gestion des connaissances**

Le retour d'expérience constitue une alternative aux approches de gestion des connaissances basées sur une conceptualisation globale des connaissances. L'objectif du retour d'expérience est de transmettre des connaissances concrètes ou des leçons apprises applicables à des niveaux opérationnels, tactiques ou stratégiques de telle sorte que, quand l'expérience est réutilisée, elle impacte positivement sur les résultats de l'organisation (Bergmann, 2002).

Les entreprises se tournent vers ces outils principalement pour éviter de reproduire les erreurs passées et bénéficier de tous les savoir-faire et connaissances disponibles impliqués, générés et utilisés lors de chaque résolution de problème. Les modèles de retour d'expérience sont en principe liés à des objectifs spécifiques de l'organisation et sont destinés à atteindre un but précis. C'est une forme particulière de gestion des connaissances dont la portée est opérationnelle bien que réalisée selon une approche conceptuelle. Contrairement aux Systèmes à Base de Connaissances, le retour d'expérience repose sur la modélisation d'un ensemble de connaissances partielles et spécifiques à une activité ou un processus. Nous considérons qu'il s'agit d'une méthode ascendante car l'objectif est de construire une base de connaissances à partir de connaissances fragmentaires.

Le retour d'expérience est aussi une approche permettant de mettre en œuvre une gestion des connaissances directement intégrée dans les processus opérationnels des organisations. La capitalisation est effectuée de manière peu intrusive, parallèlement au processus opérationnel. L'expérience peut-être analysée a posteriori et participer, éventuellement, à l'extraction de connaissances génériques par la suite. Ainsi, nous pouvons considérer l'expérience comme un relais utile à la construction de la connaissance comme cela est représenté sur la Figure I-4 (Ruet, 2002).

Les approches de type « retour d'expérience », par rapport aux méthodes conceptuelles descendantes de Gestion des Connaissances, sont avantageuses notamment pour les raisons suivantes :

- il est plus facile de formaliser l'expertise des acteurs en s'appuyant sur des expériences vécues que d'explicitier des connaissances génériques non contextualisées (l'expert participe toujours à la modélisation des expériences et des traitements associés mais de manière plus naturelle). La phase de capitalisation est rendue plus transparente, ce qui facilite la mise en œuvre et l'acceptation.

- l'aspect incrémental du système fait que la base de retour d'expérience s'étoffe au cours du temps, en fonction des possibilités et du besoin des acteurs. La base est opérationnelle dès que quelques expériences ont été capitalisées. Les mécanismes de recherche d'expériences similaires sont immédiatement disponibles. Concernant, l'extraction de connaissances, il faut attendre de disposer de suffisamment d'expériences dans la base. Le cycle de production des connaissances est donc plus long.

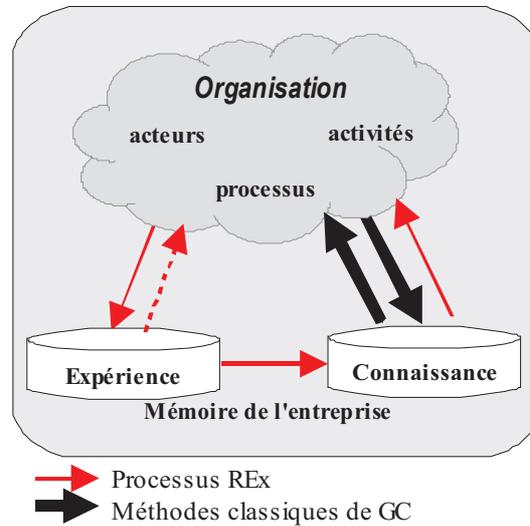


Figure I-4 - Vecteur de production de connaissance

L'inconvénient majeur, par rapport aux méthodes conceptuelles, est que la reconstitution des connaissances n'est pas une tâche triviale du fait de la fragmentation de la connaissance dans chacune des expériences. Nous ne pensons pas, toutefois, que cet aspect constitue un frein significatif à l'implantation de processus de retour d'expérience. En effet, le passage des expériences aux connaissances est souvent réalisé par des experts auxquels nous souhaitons apporter des outils adaptés pour interroger la base d'expériences. Dans la partie suivante, nous présentons plusieurs approches et variantes que l'on peut qualifier de retour d'expérience afin de positionner l'architecture que nous avons choisie.

## I.2. Retour d'expérience

Comme nous venons de le voir, le retour d'expérience peut être interprété comme un mécanisme de gestion des connaissances particulier, conceptuel et ascendant. Néanmoins, bien qu'il soit intéressant de le placer dans ce contexte, le principe exploité dans les applications de retour d'expérience correspond à une pratique empirique plus ancienne qui existe indépendamment de la notion de Système à Base de Connaissances. Nous précisons dans un premier temps les origines et influences de ce champ de recherche puis nous donnons des définitions générales du retour d'expérience. Nous présentons ensuite différentes applications significatives qui utilisent explicitement des approches basées sur l'expérience. Enfin, nous présentons et détaillons une architecture générique dans laquelle nous situons nos travaux.

### I.2.1. Origines et influences

Le retour d'expérience est une pratique très courante utilisée dans des domaines et sous des formes très variées. Dès que l'homme s'est organisé en société, il a pratiqué une forme de retour d'expérience afin de transmettre son savoir à la collectivité et en particulier à ces descendants. Ce retour d'expérience existait sous des formes très diverses, discours, enseignement mais aussi peinture, légendes. Il pouvait aussi prendre une forme plus structurée et ceci en même temps que la société s'organisait politiquement. Par exemple, des écrits du premier siècle montrent que le retour d'expérience fait l'objet

d'une préoccupation dans le domaine agricole avec des considérations sur la prise en compte des pratiques mises en œuvre : « *le chef de famille qui tient à avoir une méthode sûre pour cultiver ses terres se donnera la peine de consulter les fermiers les plus expérimentés de son époque ; et il devrait étudier avec soin les manuels des Anciens, réfléchir sur les opinions, sur les enseignements de chacun d'entre eux, pour voir si la leçon reçue peut s'appliquer telle quelle à son cas ou si elle doit être revue* », citation de l'ouvrage de Columella « *De re rustica* » datant du premier siècle et traduit dans (Cuneo, 2003). Ainsi, au fil des années, chaque personne applique les résultats d'une expérience passée pour améliorer son activité quotidienne. Cette citation pourtant très ancienne, reflète très bien les processus de retour d'expérience actuels présents dans les organisations.

L'expérience est essentielle dans l'apprentissage humain et ceci principalement en raison de la capacité d'apprentissage du cerveau qui est fondamentalement inductive. Cela signifie que le cerveau construit progressivement ses connaissances en se basant sur une confrontation avec la réalité, c'est-à-dire par l'expérience. De manière générale, l'induction, contrairement à la déduction, est un raisonnement logiquement inexact, qui est appuyé par sa vérification répétée, mais qui peut être démenti par un contre-exemple. On oppose souvent l'induction à la déduction qui est un type de raisonnement logique et exact qui fonctionne par application de règles préétablies. Dans une certaine mesure, le retour d'expérience construit en grande partie le savoir et les connaissances de l'homme. Une théorie cognitive, qui reflète fidèlement le processus de retour d'expérience, est celle du constructivisme défendue par Piaget (Piaget, 1950). Dans ce cadre, la compréhension, constamment renouvelée, est élaborée à partir des représentations plus anciennes d'événements passés, que le sujet a d'ores et déjà « emmagasinées » dans son vécu. En fait, le sujet restructure (« reconceptualise ») les informations reçues en fonction de ses propres connaissances : c'est le phénomène de restructuration conceptuelle à travers ses expériences. Cette reconceptualisation correspond à une mise à jour des connaissances de l'individu.

Plutôt que le terme d'expérience, le terme schéma introduit par Schank est aussi utilisé en Intelligence Artificielle (Schank, 1982). En psychologie cognitive, les schémas sont des représentations mentales abstraites qui résument et organisent de façon structurée des événements, des objets, des situations ou des expériences semblables. Les schémas, stockés en mémoire à long terme, permettent d'analyser, de sélectionner, de structurer et d'interpréter des informations nouvelles. Ils servent donc, en quelque sorte, de modèle, de cadre (pour reprendre l'expression équivalente utilisée en intelligence artificielle) pour traiter l'information et diriger les comportements. Ce modèle a été le point de départ de plusieurs travaux de recherche en Intelligence Artificielle tant en formalisation de la connaissance avec la théorie des *frames* de Minsky (Minsky, 1975) qu'en traitement ou en dynamique des connaissances par exemple en Raisonnement à Partir de Cas (Kolodner, 1993).

## I.2.2. Terminologie et définitions

Le retour d'expérience au sens large concerne l'analyse d'un fait ou d'un événement passé en vue de réutiliser la connaissance qui en découle. Suivant la nature positive ou négative des effets de la situation analysée, l'objectif sera de reproduire, ou bien d'éviter une situation de même type. Nous appelons « base de retour d'expérience » l'ensemble des expériences recueillies en vue d'une réutilisation. Deux types d'activités du retour d'expérience peuvent être d'emblée distingués :

- les activités de « capitalisation » pour tout ce qui concerne l'alimentation en expérience et en connaissances de la base,
- les activités d'« exploitation » en rapport avec l'utilisation, la réutilisation d'expériences de la base. La base est utilisée soit pour faire face à une nouvelle situation, soit pour faire de la prévention, de la formation... Les expériences peuvent être exploitées directement (un problème a déjà été résolu et sa solution est adaptée ou appliquée directement) ou par généralisation (plusieurs problèmes ont été résolus et il est possible de définir des règles générales qui permettront d'éviter qu'ils se reproduisent).

La base de retour d'expérience constitue alors un pivot entre ces deux types activités. Il existe un grand nombre d'expressions relatives à la prise en compte des expériences dans les organisations. En France, c'est le terme de retour d'expérience qui est le plus utilisé et c'est celui que nous utilisons dans

ce mémoire. En Europe, le terme d'Experience Management (EM) (management de l'expérience), en référence au Management des Connaissances (KM) (Knowledge Management), est utilisé et introduit la notion de gestion des connaissances liées à l'expérience (Bergmann, 2002). Aux Etats-Unis, c'est plutôt la notion de Lesson Learned (LL) (leçons acquises) qui est employée, mais nous trouvons également la terminologie « Experience Feedback » qui correspond à la traduction littérale du retour d'expérience. Dans tous les cas, c'est bien l'expérience et sa gestion qui sont au cœur de tous les travaux considérés.

De nombreuses définitions du retour d'expérience ont été proposées. Ces définitions varient essentiellement selon le domaine considéré.

### **Vision sûreté de fonctionnement et gestion des risques**

De nombreux systèmes de retour d'expérience sont appliqués au domaine des activités à risques (centrales électriques, transport aérien, etc.). En effet, les répercussions d'événements dans ce domaine peuvent être catastrophiques. Il convient alors de prendre les dispositions nécessaires pour contrôler et/ou éviter l'apparition de tels événements. Les outils de retour d'expérience dans ces domaines sont très nombreux. Début 2005, un état des pratiques de retour d'expérience en milieu industriel a été réalisé auprès de neuf entreprises, membres de l'Institut pour une Culture de Sécurité Industrielle<sup>1</sup> (ICSI), mandataire de ce bilan (Gaillard, 2005). Ces entreprises participent, depuis l'année 2003, à des travaux d'échanges dans divers thèmes ayant trait à la sécurité industrielle, dont celui relatif au retour d'expérience dans le cadre du GEC-REX (Groupe d'EChange sur le Retour d'Expérience). Les entreprises rencontrées à l'occasion de cet état des lieux sont issues de secteurs d'activité différents : Airbus, Arcelor, Arkema, EDF pôle nucléaire, RATP, Rhodia, SNCF, Solvay, et Total. Le rapport indique que la notion de retour d'expérience recouvre des réalités très variées et qu'une définition générale est assez difficile à donner.

Le groupement REXAO (Retour d'Expérience et Apprentissage Organisationnel), groupement d'études et de recherche animé par le pôle cyndinique de l'école des Mines de Paris, propose la définition suivante (REXAO, 2008) : « *d'une manière générale, le retour d'expérience est un outil de management, utilisé par l'encadrement pour identifier les causes de dysfonctionnement par l'analyse des faits (séances de débriefing, interviews, etc.) et pour en tirer des leçons (nouvelles consignes, modifications organisationnelles, etc.)* ». Dans cette logique, Y. Vérot propose une définition qui rejoint la précédente, mais qui fait apparaître de manière explicite la notion de démarche avec des activités bien définies (Verot, 2001) : « *la démarche de retour d'expérience peut être définie comme une démarche organisée et systématique pour :*

- *analyser toutes les anomalies, tous les incidents et accidents constatés,*
- *en rechercher les causes et les enchaînements,*
- *en retirer les divers enseignements,*
- *définir les mesures de correction et d'amélioration,*
- *assurer l'information pertinente aux parties intéressées ».*

Dans ces deux premières définitions, le retour d'expérience est considéré exclusivement pour le traitement d'événements négatifs liés à du matériel. Le retour d'expérience est ici placé au niveau d'un champ technique quantitatif. De plus, bien que les préoccupations soient globalement proches des nôtres, ce type de retour d'expérience insiste davantage sur les aspects organisationnels, c'est-à-dire sur l'intégration dans les organisations, la participation des acteurs, les processus de capitalisation de l'information. Notre objectif est de structurer davantage l'information en vue de traitements plus automatisés. L'intégration dans l'organisation devrait ainsi être facilitée et donc nécessiter moins d'effort au niveau du management.

### **Vision Experience Management (Experience factory)**

Bergmann a, au travers de l'Experience Factory (Basili *et al.* 1994), définit le champ du retour d'expérience comme traitant de la connaissance liée à tout type d'événement négatif et exprimée par l'expérience (Bergmann, 2002). La définition est donc restreinte au contexte de la résolution de

<sup>1</sup> <http://www.icsi-eu.org/>

problèmes et n'intègre que les événements négatifs : « *le management de l'expérience est un type de gestion des connaissances qui se restreint aux connaissances issues de l'expérience, c'est-à-dire aux connaissances spécifiques relatives à un contexte particulier de résolution d'un problème. Le management de l'expérience consiste à collecter, modéliser, stocker, réutiliser, évaluer et mettre à jour l'expérience* ». Nous retrouvons dans cette définition les activités à réaliser, ce qui présente le retour d'expérience comme un véritable processus.

C'est une vision aussi très proche de l'approche que nous avons retenue définissant le retour d'expérience en tant que macro-processus. Les outils réalisés sont néanmoins restreints à la capitalisation statique et très assistée des expériences/connaissances. Nous souhaitons pousser plus loin les traitements automatiques de l'information.

### **Vision leçons acquises (Lesson Learned)**

Aux Etats-Unis, de grandes organisations se sont intéressées à la mise en place du retour d'expérience ou « Lesson Learned » (leçons acquises) (Aha *et al.*, 2001a) (Aha *et al.*, 2001b). Les définitions proposées rejoignent les précédentes, mais considèrent en plus des événements négatifs, les événements positifs. Ainsi, le Département de l'Energie (DoE) aux Etats-Unis propose la définition suivante (Bicford, 2000) : « *une leçon acquise est une bonne pratique ou une approche innovante qui est capturée et partagée afin d'encourager sa réutilisation. Une leçon acquise peut également représenter un mauvais fonctionnement ou une expérience qui est capturée pour en éviter la récurrence* ». Il est à noter que dans cette définition, la notion d'expérience est associée à des événements dits positifs. Le traitement consiste à extraire des bonnes pratiques en vue de leur réinjection dans les processus opérationnels de l'organisation. La leçon acquise reste, en pratique, un inventaire d'expériences utiles. Ces variantes de l'expérience sont très utilisées dans le domaine militaire américain et de manière plus générale dans le domaine de la défense. Cela a donné lieu par exemple au AAR (US Army « After Action Review ») et au CALL (Center for Army Lesson Learned).

### **Définition retenue**

A partir de l'ensemble des définitions existantes et des caractéristiques du retour d'expérience, nous avons proposé et retenu la définition suivante (Rakoto *et al.*, 2002) (Clermont *et al.*, 2007) :

*« Le retour d'expérience est une démarche structurée de capitalisation et d'exploitation des connaissances issues de l'analyse d'événements positifs et/ou négatifs. Elle met en œuvre un ensemble de ressources humaines et technologiques qui doivent être organisées pour contribuer à réduire les répétitions d'erreurs et à favoriser certaines pratiques performantes ».*

Le retour d'expérience constitue ainsi une véritable démarche qui permet de capitaliser la richesse immatérielle de l'entreprise en vue de fiabiliser ses développements. Bien plus qu'un procédé technique, le retour d'expérience est un processus organisationnel qui nécessite un encadrement et un management des acteurs dans le même temps que la gestion des informations. Dans le travail présenté l'objectif est la réalisation d'applications qui devront éventuellement être intégrées dans un processus organisationnel complexe. Nous fournirons un cadre structuré, de formalisation de l'expérience qui prend en compte l'analyse experte. Cette formalisation est essentielle pour permettre l'exploitation future et en particulier l'extraction de connaissances.

A la lecture de l'ensemble de ces définitions, le retour d'expérience fait apparaître un ensemble d'aspects auxquels nous allons nous intéresser. Tout d'abord, la notion d'expérience apparaît clairement comme une source de connaissance. Celle-ci est exploitée au moyen d'une démarche qui génère un ensemble de flux d'informations. Pour conduire cette démarche, il est nécessaire de mettre en place un dispositif répondant à une finalité qui est le « gain à tirer de la connaissance a priori, du résultat d'une action déjà vécue, moyennant la connaissance du contexte » (Leroy, 2001). Enfin, la formalisation est un mécanisme indispensable pour permettre l'exploitation des connaissances.

La diversité des définitions témoigne d'une grande diversité d'applications. Nous analysons, plus en détails, différentes mises en œuvre du retour d'expérience dans la section suivante.

### I.2.3. Mise en œuvre du retour d'expérience

Le point central des processus de retour d'expérience et de gestion des connaissances est l'articulation de la mémoire autour de deux processus centraux, la capitalisation et l'exploitation (Lebowitz, 1999). Le schéma de la Figure I-6 reprend cette vision et présente les différentes entrées et sorties du processus de retour d'expérience dans son ensemble.

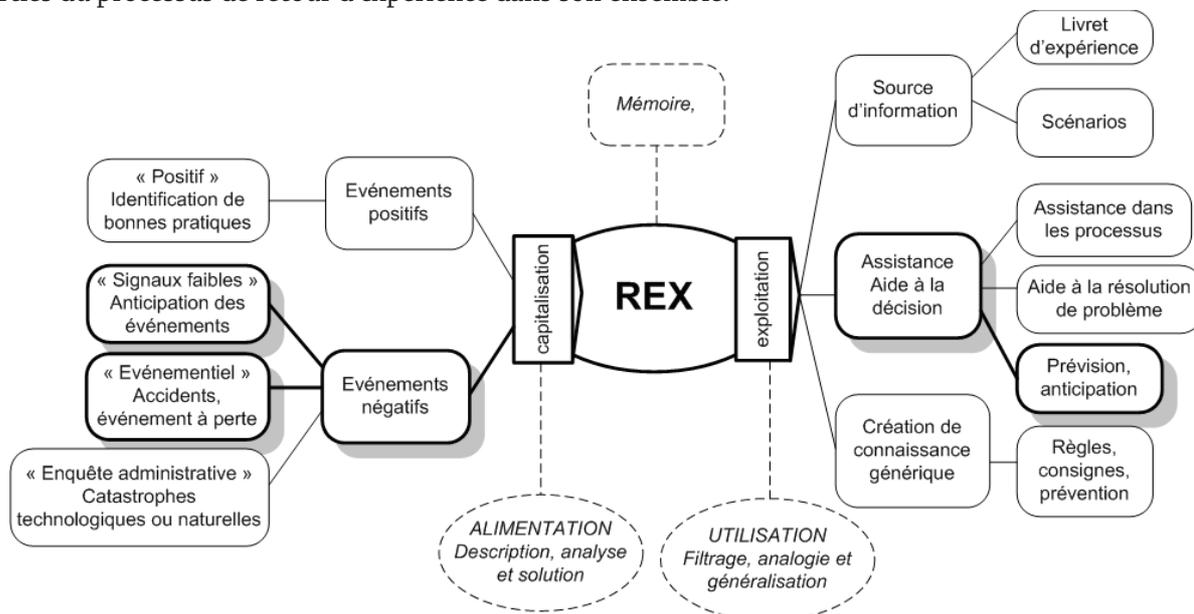


Figure I-5 - Vue d'ensemble des applications de retour d'expérience

Au centre, nous retrouvons la mémoire de l'organisation et les deux processus indissociables, capitalisation en entrée et exploitation en sortie. L'élément d'entrée d'un processus de retour d'expérience est l'événement, qu'il soit positif ou négatif. En sortie du processus de retour d'expérience nous pouvons distinguer différents modes d'exploitation :

- Source d'information : vise à la constitution de catalogues, de mémoires d'expériences, d'une mémoire de résolution de problèmes... directement à partir de la transcription des expériences,
- Assistance et aide à la décision : réutilisation des solutions, ou des analyses pour résoudre de nouveaux problèmes. Il est possible d'adapter une ancienne solution pour résoudre un nouveau problème ou bien de reproduire le processus de résolution,
- Création de connaissances: consiste en l'étude d'ensembles d'expériences pour inférer des connaissances plus globales par généralisation (apprentissage inductif).

Les différents types d'entrées pour le retour d'expérience mis en œuvre dans les entreprises sont représentés à la gauche du schéma de la Figure I-6. Il s'agit des enquêtes administratives, des retours d'expérience dits « positifs » concernant la capitalisation des bonnes pratiques, des retours d'expérience « signaux faibles », et enfin, les plus courants, les retours d'expérience « événementiel ».

#### Enquêtes administratives

Il s'agit d'études et d'enquêtes en comité de spécialistes ou comités de crise, liées à des catastrophes technologiques ou naturelles. L'objectif est en général de détecter les défaillances « systémiques » en répondant aux questions suivantes (Grelot et Guillaume, 2004) :

- *Qu'est ce qui s'est passé ?*
- *Comment cela s'est-il passé ?*
- *Pourquoi cela s'est-il passé ainsi ?*
- *Que faut-il faire pour que "la prochaine fois" les conséquences soient moins dramatiques ?*

Il s'agit ici d'une forme de retour d'expérience très commune mais très peu assistée d'outils de traitement de l'information. Les aspects méthodologiques sont en revanche souvent étudiés en détail et des outils de gestion documentaires peuvent aider dans le processus.

### **Retour d'expérience « positifs »**

Cette forme de retour d'expérience commence à être déployée dans les entreprises. Le « retour d'expérience positif » est une forme récente et encore peu systématisée du retour d'expérience qui vise non plus à apprendre des accidents ou des dysfonctionnements, mais à détecter les bonnes pratiques et à les renforcer.

Au-delà des échanges informels entre professionnels, le retour d'expérience sur les bonnes pratiques devient de plus en plus un thème d'échange structuré sous forme de plateformes de partage d'expériences (Gaillard, 2005). Cependant, une des difficultés majeures du retour d'expérience positif est l'effort supplémentaire qui est demandé pour décrire et étudier les bonnes pratiques dans les processus dont le déroulement est correct. Cela demande du temps et du recul ce qui n'est pas toujours possible dans le contexte des entreprises. Au contraire, lors d'un dysfonctionnement, des procédures d'étude et de résolution sont en général déclenchées et peuvent être combinées avec un processus de retour d'expérience.

### **Retour d'expérience signaux faibles**

Cette forme de retour d'expérience vise à anticiper la survenue d'un événement marquant sans que celui-ci ne soit réellement survenu. Le « retour d'expérience « Signaux faibles » est basé sur un des principes admis par les professionnels en maîtrise des risques, principe selon lequel tout ce qui est envisageable peut survenir, et qu'il convient donc d'anticiper autant que possible les événements redoutés, et de prendre les mesures qui permettront d'en contrôler la situation. C'est ce même principe qui prévaut dans les études de risques. La plupart des événements accidentels graves ont mis en évidence la présence de signaux faibles, repérés lors de l'analyse mais sans que les organisations en place ne les aient détectés comme source potentielle d'accident très grave. Cette forme de retour d'expérience est bien plus compliquée que le « retour d'expérience événementiel » et sa mise en œuvre passe par des équipes de spécialistes de la détection des signaux faibles et de leur traitement (Gaillard, 2005). C'est un type de retour d'expérience utilisé principalement pour les entreprises connaissant peu d'accidents mais dont les conséquences sont potentiellement catastrophiques (ex : l'industrie aéronautique, nucléaires...).

### **Retour d'expérience événementiel**

Ce type de retour d'expérience est le plus courant. Il repose sur la capitalisation de la réaction face à l'occurrence d'un événement. L'objectif est généralement de comprendre l'origine de l'événement de façon à éviter ou favoriser sa répétition. Le retour d'expérience événementiel est associé non seulement à des événements exceptionnels, mais aussi à ceux qui représentent un impact potentiel élevé. Pour cela, le retour d'expérience « événementiel » encourage la remontée du plus grand nombre d'événements depuis le terrain pour pouvoir repérer les cas les plus intéressants. Les mesures seront décidées à la vue des conclusions de l'analyse qui sera menée.

Le retour d'expérience événementiel est largement employé dans le monde industriel. Un pré-requis consiste à disposer d'une formalisation permettant de décrire les circonstances de l'événement et ses conséquences. Les outils de traitements de l'information associés à ce type de retour d'expérience sont principalement des gestions documentaires et éventuellement d'hyper-documentation (documents utilisant la structuration avec liens hypertextes et des contenus multimédia). Il n'existe pas, à notre connaissance, d'applications automatisant le processus de retour d'expérience à partir d'une représentation structurée de l'expérience.

## I.2.4. Les grandes classes du retour d'expérience

Afin de pouvoir classer les différents outils et architectures de retour d'expérience et positionner notre contribution, nous avons défini trois axes de classification des approches retour d'expérience. Nous considérons uniquement ici le retour d'expérience événementiel. L'événement constitue alors un marqueur fort de l'expérience et est obligatoirement présent. Les approches de retour d'expérience « événementiel » peuvent être classifiées selon les caractéristiques suivantes :

- Événement positif ou événement négatif,
- Connaissance statique ou dynamique,
- Raisonnement statistique ou cognitif.

### I.2.4.1. Retour d'expérience à événement « positif » ou « négatif »

Suivant la nature des événements traités le retour d'expérience sera qualifié de « négatif » ou de « positif ». L'événement objet du retour d'expérience est souvent un écart à la norme et au fonctionnement normal d'un système. Ainsi, les quasi-incidents ou toute situation considérée comme « anormale » peut en faire l'objet. Toutefois, une autre forme de retour d'expérience a été développée notamment dans plusieurs grandes entreprises : il s'agit d'une démarche généralement appelée « retour d'expérience positif ». Dans ce cadre, les « bonnes pratiques » sont identifiées et capitalisées en vue de favoriser une dynamique positive de communication du savoir. Sur le plan de son adoption par les acteurs, le retour d'expérience positif est plus facile à mettre en œuvre car la crainte d'un jugement ou de la désignation d'un coupable ne freine pas les échanges d'expériences.

Au contraire, le retour d'expérience basé sur des événements négatifs est plus facile à mettre en œuvre techniquement et à intégrer dans les processus opérationnels. En effet, comme nous l'avons souligné dans l'introduction générale de ce mémoire, il existe de nombreuses méthodes de résolution de problèmes, notamment dans le domaine de la qualité, qui peuvent servir de support à la capitalisation des expériences.

### I.2.4.2. Retour d'expérience statique et dynamique

La distinction entre retour d'expérience « statique » et retour d'expérience « dynamique » est utilisée pour différencier la manière dont les outils exploitent la connaissance avant de la restituer. Le retour d'expérience de type « statique » est le plus répandu dans les organisations.

Dans les approches statiques, l'objectif principal est de collecter et de diffuser la connaissance sans traitement « intelligent ». La connaissance est principalement consultée. Le système n'infère pas de nouvelles connaissances. Il peut être instrumenté par des outils de gestion documentaire qui assistent les utilisateurs dans leur découverte des expériences et par conséquent dans l'appropriation des connaissances. Dans le retour d'expérience de type « dynamique » au contraire, la création de nouvelles connaissances est un but. L'accent est mis sur la transparence sémantique (*cf.* II.1.4) entre l'utilisateur et le système. La connaissance doit être formalisée pour être accessible, compréhensible et manipulable par les deux parties. Le système va favoriser l'explicitation des connaissances implicites contenues dans la base d'expériences.

Généralement, dans les approches statiques, des formalismes semi-structurés, textuels, « hyperdocumentaires » sont utilisés pour représenter l'expérience. Ces outils reposent très souvent aujourd'hui sur une représentation XML qui est un langage semi-structuré à balises. Pour les approches dynamiques, il est possible d'utiliser des formalismes conceptuels que l'on retrouve en Ingénierie des Connaissances et en particulier dans la réalisation des Systèmes à Bases de Connaissances. Un exemple de retour d'expérience « statique » est celui de la méthode REX (Malvache *et al.*, 1994) développée par le CEA à la direction des Réacteurs Nucléaires. REX est une méthode et un logiciel support qui visent à capitaliser le retour d'expérience sur le démarrage de Superphénix. Il s'agit d'un ensemble de fiches, descriptions semi-structurées textuelles, schématiques sous forme « d'expériences ». Ces fiches sont

rédigées à partir des interviews des personnes qui ont vécu ces expériences. Elles sont ensuite mises à disposition dans un outil informatique avec un système de recherche par mots-clés, agrémenté d'un graphe de concepts, qui permet de naviguer dans les concepts du domaine pour mieux formuler sa requête (Ermine *et al.*, 1996).

Il n'y a pas, à notre connaissance, d'outils dynamiques dédiés au retour d'expérience. Les outils qui entrent dans cette catégorie sont en principe des outils de gestion descendante des connaissances, tels que CommonKADS. Notons, tout de même, qu'il existe des travaux visant à extraire de la connaissance à partir d'une approche statique. Par exemple, toujours pour la méthode REX, P. Seguela, dans ses travaux de thèses, cherche à établir des modèles conceptuels à partir d'une description textuelle. Une technique nommée Caméleon est développée et appliquée à REX. Elle repose sur l'analyse lexicale des textes et la découverte de concepts (Séguela, 2001).

### I.2.4.3. Retour d'expérience statistique et cognitif

La distinction entre retour d'expérience statistique et retour d'expérience cognitif repose sur la technique employée lors du traitement de la connaissance par le système. Dès lors que l'information stockée est en quantité suffisante, il devient possible d'utiliser des techniques statistiques, ces techniques pouvant être appliquées à la fois dans des approches statiques ou dynamiques. On peut utiliser des recherches de mots-clés, des recherches d'associations sémantiques ou encore des Analyses en Composantes Principales dès lors que l'information est suffisamment structurée.

Un exemple intéressant de système statistique est le BARPI<sup>1</sup> et l'ARIA (Analyse, Recherche et Information sur les Accidents) utilisés dans le domaine des crises sanitaires. La banque de données ARIA89 (Analyse, Recherche et Information sur les Accidents) est exploitée depuis février 1993 par le Bureau d'Analyse des Risques et Pollutions Industrielles (BARPI) créé en 1992. Le principal objectif de cette banque de données est de rassembler des informations précises et détaillées sur les incidents et accidents industriels et technologiques survenus en France pour la plupart mais aussi dans le reste du monde, afin d'en faciliter le retour d'expérience et de réaliser des analyses de risques. Cette banque de données recense actuellement 32.000 accidents (80% survenus en France) relatifs aux installations industrielles, au transport de matières dangereuses, aux pollutions agricoles etc.

Toutefois, lorsqu'on dispose de relativement peu d'informations, il devient difficile d'utiliser des techniques statistiques. Il est alors nécessaire de conférer aux expériences une formalisation adaptée qui permettra de consigner des analyses expertes qui seront exploitées par les mécanismes d'inférence. L'engagement des experts vient alors combler le manque d'informations statistiques. Nous qualifierons ces systèmes de « cognitifs » par opposition à « statistiques ».

### I.2.4.4. Bilan

Ces typologies des systèmes de retours d'expérience nous permettent de mieux situer le type de retour d'expérience auquel nous nous adressons. Nos travaux concernent en effet la réalisation de systèmes qui présentent les caractéristiques suivantes :

- **Événements négatifs** : ayant choisi de nous intéresser à l'aspect prévention des risques aidée par le retour d'expérience, la prise en compte des événements négatifs dommageables est naturelle. En outre, les événements négatifs sont les plus fréquents, les plus faciles à mettre en évidence et à formaliser.
- **Connaissances dynamiques** : nous souhaitons utiliser l'expérience comme un vecteur de production de connaissances. L'expérience constitue ainsi davantage un fragment de connaissance qu'un agrégat de données dont on dispose en nombre. La mise en relation de ces éléments de connaissance constitue un dispositif d'inférence d'une connaissance plus générale.
- **Retour d'expérience cognitif** : nous choisissons de faire des outils pour une capitalisation progressive des expériences, dans des circonstances réelles. Ceci implique un nombre de

<sup>1</sup><http://www.aria.developpement-durable.gouv.fr/>

données souvent insuffisantes pour des traitements d'ordre statistique. Pour palier ce problème, nous intégrons les éléments clé de l'analyse experte dans l'expérience.

Sur la base de ces caractéristiques, nous allons présenter une architecture générale pour les outils de retour d'expérience que nous développons.

## I.2.5. Architecture générique

L'architecture de retour d'expérience qui sert de support à nos travaux est une généralisation des approches existantes associées au domaine industriel. Cette architecture a été établie par Rakoto au cours de ses travaux de thèse à partir d'un état de l'art détaillé sur les systèmes industriels et les modèles formalisés de retour d'expérience (Rakoto, 2004). Nous reprenons en grande partie l'ensemble de l'architecture développée. La principale évolution concerne le positionnement de l'étape d'analyse du contexte qui intervient au moment de la définition de l'expérience. Nous revenons sur ce point plus en détail dans la partie I.2.5.1.

L'architecture de retour d'expérience proposée est basée sur une approche processus telle que préconisée dans le cadre des normes d'assurance qualité pour répondre à un besoin de standardisation et de normalisation. Ainsi, nous considérons le retour d'expérience comme un processus global de transmission du savoir dans lequel nous dissocions deux processus opérationnels majeurs : les processus de capitalisation et d'exploitation. Une base de retour d'expérience sert alors de pivot entre ces deux processus comme cela est représenté sur la Figure I-6. Elle contient à la fois les expériences et des connaissances généralisées.

- Le processus de capitalisation regroupe l'ensemble des activités permettant d'ajouter de nouvelles informations à la base de retour d'expérience y compris les opérations d'extraction de connaissances et de maintenance,
- Le processus d'exploitation est constitué des activités permettant de diffuser et d'employer ces informations, expériences et connaissances dans les processus industriels. Certaines activités de type exploitation sont éventuellement utilisées pour assister le processus de capitalisation.

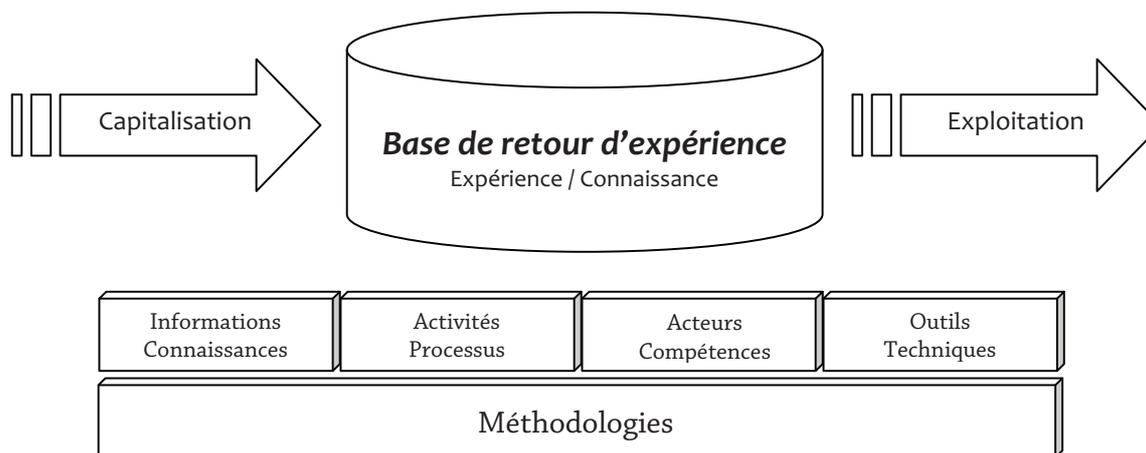


Figure I-6 - Méthodologie, processus et composants support du retour d'expérience

Sur le schéma de la Figure I-6 une vue globale du processus de retour d'expérience est représentée. La mise en œuvre opérationnelle nécessite de prendre en compte plusieurs dimensions. Quatre dimensions ont été distinguées :

- une dimension « informations et connaissances » qui concerne la modélisation de l'entité expérience, l'organisation de cette entité et en particulier le lien entre donnée, information et connaissance,

- une dimension « activités et processus » qui précise les différents processus du retour d'expérience et les tâches qui les composent. Les processus de capitalisation et d'exploitation sont déclinés en plusieurs sous-processus construit à partir de plusieurs activités,
- une dimension « acteurs et compétences » qui définit les rôles et les responsabilités des différents acteurs par rapport à leur utilisation du système,
- une dimension « techniques et outils » en rapport avec les moyens logiciels et managériaux mis en œuvre pour pouvoir opérationnaliser le retour d'expérience.

La mise en œuvre d'un processus de retour d'expérience nécessite de considérer ces quatre composantes et de les intégrer de manière cohérente dans le Système d'Information (SI) de l'organisation concernée. Aucune des dimensions n'est systématiquement prépondérante. Elles sont toujours présentes dans le processus et, idéalement, elles devraient être déployées en parallèle. Ce sont les méthodologies qui déterminent l'organisation et les enchaînements entre les éléments issus de chaque dimension. Néanmoins, certains aspects prennent le pas sur d'autres et sont développées plus en profondeur selon le type d'application réalisée. Les différentes dimensions support sont reprises et détaillées dans les parties suivantes.

### **I.2.5.1. Dimension « Informations et Connaissances »**

La relation entre informations, expériences et connaissances a déjà été présentée. L'aspect informations et connaissances traite de la modélisation de l'entité « expérience ». Plusieurs points sont à considérer pour trouver le bon compromis de modélisation.

- Il y a une nécessité de transparence sémantique, c'est-à-dire que les expériences doivent être formalisées pour être manipulables à la fois informatiquement et par des utilisateurs,
- Le degré d'abstraction nécessaire impose le choix de techniques de modélisation appropriées. Si la gestion des connaissances est de type gestion documentaire, alors la formalisation sera moindre. Si des systèmes offrant des capacités de raisonnement sont souhaités, il est utile d'adopter une formalisation de plus haut niveau permettant de décrire les concepts et les objets du monde réel, de les organiser et de les classer...
- Enfin, il est important selon nous de considérer en amont le caractère incertain, incomplet et éventuellement imprécis de l'information véhiculée dans le système de retour d'expérience. En effet, cette information provient de la description du monde et recense aussi des « avis » d'acteurs experts. Pour rendre le système souple dans son utilisation, il est nécessaire d'intégrer cet aspect qui est souvent éludé dans les systèmes de gestion des connaissances classiques.

Dans la littérature, nous trouvons plusieurs approches que l'on peut rapprocher du retour d'expérience comme le modèle d'apprentissage par l'expérience (*experiential learning*) (Kolb, 1984), le modèle des leçons apprises (*lessons learned*) (Weber *et al.*, 2001), la boucle de retour d'expérience (Faure et Bisson, 1999) (Faure et Bisson, 2000). Ces modèles ont trois limitations principales :

- une description imprécise du vocabulaire en liaison avec les modèles de connaissance,
- un manque d'outils formels permettant l'analyse rigoureuse des modèles,
- un manque d'outils concrets d'aide à la capitalisation et à l'exploitation.

C'est pourquoi l'approche que nous proposons prévoit un langage structuré de modélisation de l'expérience afin de pouvoir rendre automatisables certaines phases et tâches du processus de retour d'expérience et pouvoir extraire une connaissance générale.

### **I.2.5.2. Dimension « Activités et Processus »**

Le processus de capitalisation est basé sur les activités de saisie de l'ensemble des informations constituant l'expérience. Trois niveaux d'abstraction de l'information capitalisée peuvent être mis en évidence :

- niveau 0 : les événements sont décrits en vue d'une analyse de nature statistique (description simple du contexte de l'événement),

- niveau 1 : des analyses, des interprétations sont associées aux événements (recenser les causes, donner des solutions curatives, des points de vue...) ainsi que les solutions au problème posé,
- niveau 2 : l'information est analysée en vue de l'extraction de connaissances (procédures, règles, recherche d'invariants, de *patterns*...).

Le passage d'un niveau à l'autre est effectué au moyen de processus spécifiques qui, eux-mêmes, sont décomposés en plusieurs activités standard. Le schéma de la Figure I-7 est une synthèse de tous ces processus où nous précisons les enchaînements. Les processus A, B et C sont dédiés respectivement à la capitalisation des contextes, des analyses et des solutions. Le processus D est celui d'extraction des connaissances. Enfin le processus E de révision assure la maintenance à la fois des expériences et des connaissances.

Le processus d'exploitation est basé sur la diffusion de l'information indépendamment de chaque niveau de capitalisation selon deux modes possibles :

- « pull » : l'acteur sollicite lui-même la base de connaissances dans laquelle il va chercher l'information (logiciels spécifiques, fiches dans des classeurs...),
- « push » : l'information est ré-injectée dans les processus opérationnels sans que l'acteur n'ait à s'en soucier (intégration dans les processus opérationnels...).

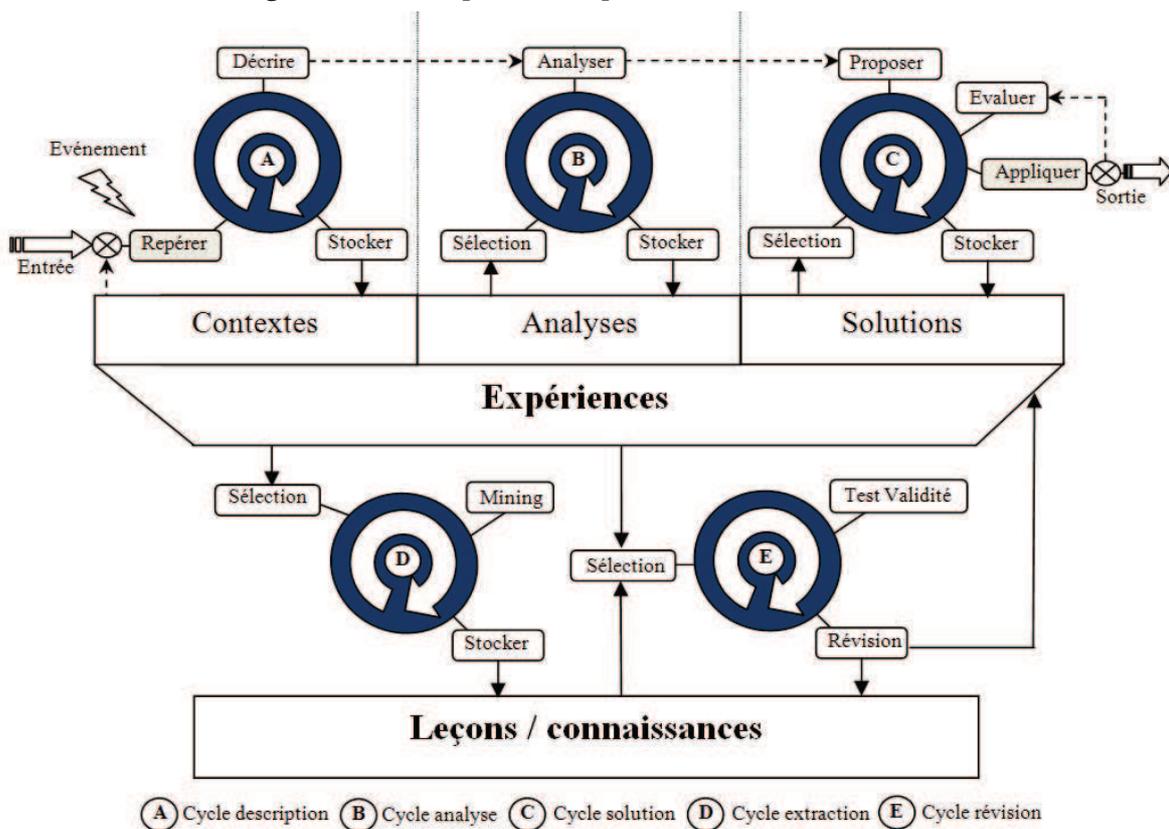


Figure I-7 - les différents cycles/processus et activités de capitalisation

De la même manière que pour le processus capitalisation, le macro-processus exploitation est décomposé en deux sous-processus :

- l'un correspondant au cycle de résolution de problème qui est déclenché lorsqu'un problème apparaît dans le déroulement d'un des processus de l'organisation,
- l'autre correspondant au cycle de prévention qui est activé en surveillance du bon déroulement d'un processus.

La différence fondamentale entre les deux processus est que dans le premier cas, un nouveau problème apparaît et déclenche un processus de résolution basé sur la réutilisation d'une ancienne solution éprouvée. Dans le deuxième cas, nous essayons de prévoir ou d'anticiper un problème en essayant de détecter des situations analogues à celles qui ont déjà posé problème. Ce cas d'utilisation implique la surveillance du contexte courant.

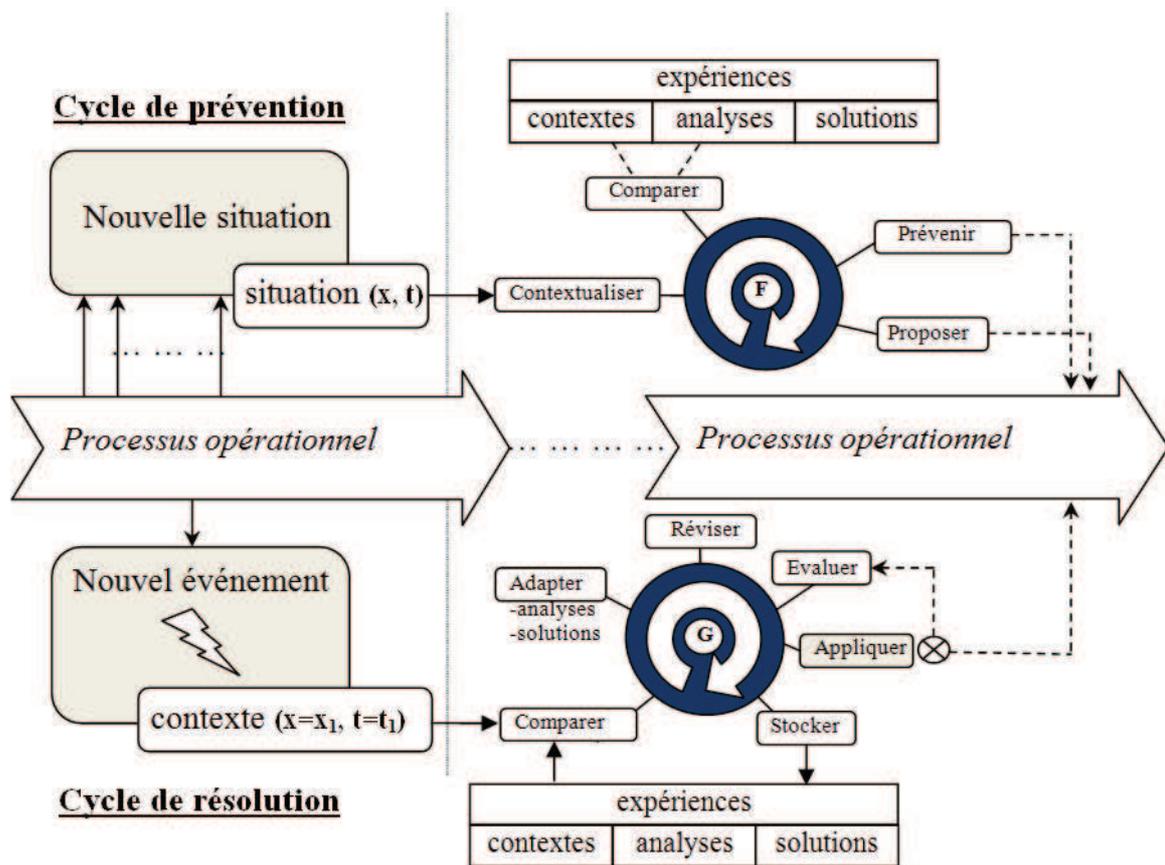


Figure I-8 - les différents cycles/processus et activités d'exploitation

La principale difficulté du cycle de prévention est l'étape de contextualisation. Dans le cycle de résolution, la correspondance des contextes est directe puisque l'événement s'est effectivement produit et donc possède une caractérisation compatible. Par contre, dans le cycle de prévention, il faut anticiper certains éléments qui ne sont pas forcément connus et effectuer un contrôle continu de la situation courante. Sur le schéma de la Figure I-8, ces deux processus (notés F et G) ainsi que les enchaînements d'activités typiques sont représentés. Il y est en particulier montré que l'utilisation en prévention (processus F) nécessite un contrôle continu du processus opérationnel supervisé.

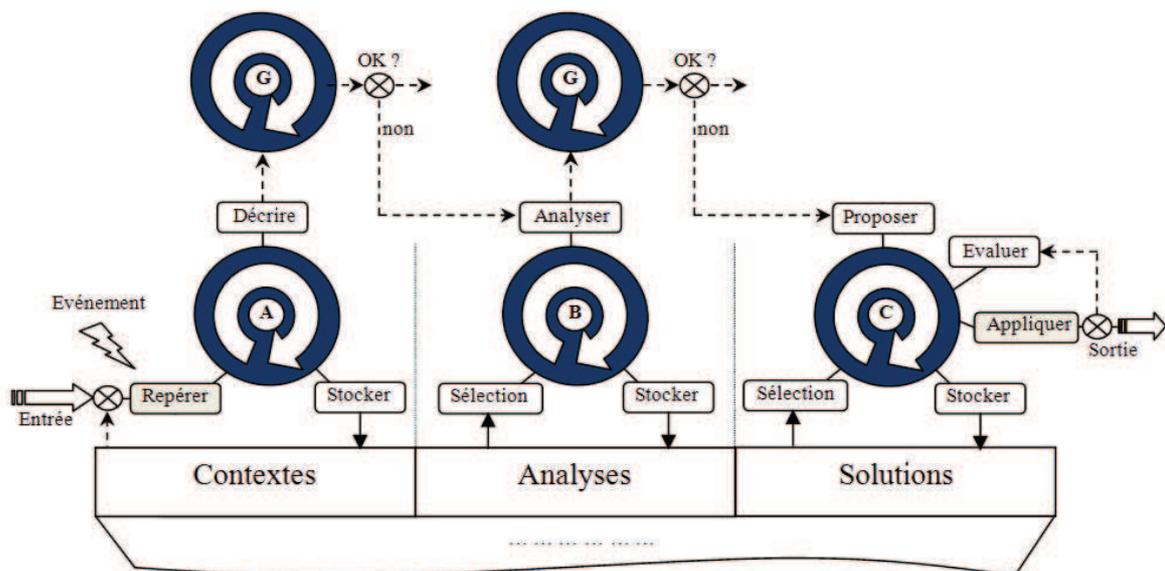


Figure I-9 - Intégration Processus Capitalisation et Exploitation

Il convient de remarquer que le processus de capitalisation, notamment dans ses tâches de description de la solution mais aussi d'analyse, est généralement assisté par le processus d'exploitation

de résolution de problème. Les processus ne sont donc pas indépendants et présentent des interactions. Pour illustrer cela, sur la Figure I-9, un aperçu de processus capitalisation/résolution est représenté en gardant les notations des deux schémas des Figure I-7 et Figure I-8.

### 1.2.5.3. Dimension « Acteurs et Compétences »

La réussite d'une démarche de retour d'expérience nécessite une implication importante des acteurs de l'organisation concernée. Plusieurs types d'acteurs interviennent dans le système à la fois dans les phases de conception du système de retour d'expérience, de capitalisation et d'exploitation et en maintenance. Ces acteurs ne disposent pas tous des mêmes compétences lorsqu'ils capitalisent des expériences. Ainsi, la démarche de retour d'expérience met en jeu des compétences différentes suivant qu'il s'agit de la description d'un cas, de son analyse ou de son interprétation ou qu'il s'agit d'extraction de connaissances.

Nous pouvons distinguer par exemple les types d'acteurs suivants :

- Utilisateur en exploitation : utilisateur qui sollicite le système en effectuant des requêtes dans la base (éventuellement de manière transparente).
- Utilisateur en capitalisation : utilisateur qui participe à la saisie de nouvelles expériences. En principe, cet utilisateur se contente de décrire et de capitaliser un événement et son contexte, et, éventuellement, la solution apportée.
- Utilisateur expert : utilisateur à qui sont confiées les tâches d'analyse et de jugement des contextes ou des solutions. Il s'agit a priori d'un expert du domaine qui en possède une bonne connaissance.
- Utilisateur cognicien : utilisateur qui intervient pour aider à la formalisation des connaissances. Il connaît les implications du formalisme de représentation sous-jacent et aide les différents acteurs à se conformer à ce langage. C'est lui, par exemple, qui participe aux tâches de modélisation antérieures à la mise en œuvre d'outils de retour d'expérience. Il assiste l'expert dans la formalisation de la connaissance du domaine nécessaire à l'expression des différentes entités (contexte, analyses, solution). C'est aussi lui qui assiste l'utilisateur expert dans la découverte de connaissances générales et qui propose une forme adaptée toujours selon le formalisme de représentation adopté.
- Utilisateur administrateur : utilisateur qui gère les droits d'accès des autres utilisateurs. C'est aussi lui qui valide d'éventuelles modifications du modèle. Idéalement, un système de retour d'expérience doit être adaptable puisqu'il fonctionne selon un mode incrémental. Il est possible d'imaginer que le modèle de l'expérience puisse évoluer après avoir été confronté à la pratique (cas de modèles qui ne se révéleraient pas vraiment adaptés à l'utilisation courante). Le système doit alors posséder des fonctionnalités permettant de modifier le modèle et d'adapter des expériences existantes.
- Utilisateur maintenance : utilisateur qui s'occupe des tâches de gestion du système, comme les sauvegardes, la mise en œuvre des différents serveurs ou encore la gestion des performances (création d'index) et des indicateurs...

Un autre point intéressant concernant les profils des utilisateurs est le lien que l'on peut établir entre l'utilité d'une expérience et le profil des acteurs. Il est en effet possible, pour assurer une meilleure diffusion des expériences, de modéliser et prendre en compte le profil et les compétences des acteurs impliqués dans les processus couverts par le retour d'expérience. Ainsi, les expériences les plus pertinentes peuvent être proposées aux utilisateurs en fonction de leur profil.

La complexité de l'intégration de ces considérations dépend de l'organisation des utilisateurs impliqués dans le processus de retour d'expérience ainsi que de la complexité même de ce processus (nombre d'étapes, de sous-processus...). Cette intégration seule n'est pas suffisante, elle doit également comporter une phase de modélisation des acteurs et des compétences selon un schéma général puisque ce besoin est commun à de nombreuses organisations (Hermosillo *et al.*, 2005).

En ce qui nous concerne, la modélisation de la dimension « acteurs et compétences » sera minimale car le domaine des compétences et de la modélisation est périphérique à notre problématique. C'est pourquoi, nous adopterons une modélisation simple des profils et compétences aussi bien pour traduire le rôle des acteurs dans le processus de retour d'expérience (et donc les droits d'accès aux modules de l'application support) que pour la représentation des compétences des acteurs dans leur métier.

#### **I.2.5.4. Dimension « Techniques et outils »**

Enfin, pour pouvoir mettre en œuvre et opérationnaliser les processus de retour d'expérience, il convient de considérer les dimensions, outils et techniques d'implémentation. Bien que le choix des techniques et outils soit relativement libre, nous proposons un découpage des fonctions nécessaires en deux catégories principales :

- les traitements usuels et classiques des systèmes d'information (interfaçage, persistance, ...),
- les traitements propres au retour d'expérience (modélisation, raisonnement, réutilisation, adaptation...).

Les traitements usuels sont les mêmes que ceux présents dans les Systèmes d'Informations classiques, c'est-à-dire « orientés données ». Ils permettent principalement l'intégration, la manipulation, la présentation et la sauvegarde des données. Aujourd'hui, ces traitements sont en général distribués. Ils doivent être flexibles pour pouvoir supporter les besoins permanents d'adaptation des organisations à leur environnement socio-économique. Nous trouvons par exemple des outils collaboratifs intégrés dans les entrepôts de données qui assurent l'interaction avec les données (interfaces utilisateurs) et la persistance sur support physique (sauvegarde de la base). Les interfaces utilisateurs (GUI) ou les Interfaces Homme-Machine (IHM) sont les parties visibles de l'application qui assurent l'interaction entre l'utilisateur et la machine. Cet aspect est très important et participe grandement à l'ergonomie de l'ensemble. Nous portons une attention particulière à cet aspect. En effet, les interfaces doivent respecter les exigences du processus de retour d'expérience mais aussi être suffisamment ergonomiques pour ne pas rebuter les acteurs et ainsi maintenir la dynamique du système.

Les traitements propres aux processus de retour d'expérience sont liés à l'exploitation particulière des informations que sont les expériences. Ces traitements sont appliqués aux éléments constitutifs de l'expérience et permettent de réaliser des tâches ou des activités qui appuient les processus du retour d'expérience. Tous ces traitements sont en liaison directe avec la formalisation de l'expérience choisie. Nous pouvons citer par exemple :

- la recherche de cas ou d'expériences selon différents critères grâce, par exemple, à des techniques utilisées en Raisonnement à Partir de Cas (RàPC),
- l'adaptation d'expériences à de nouvelles situations, par exemple à l'aide d'outils d'interpolation ou d'extrapolation,
- l'extraction de connaissances, par exemple au moyen des traitements de « rule mining » ou de « data mining »,
- ...

Nous définirons de manière la plus indépendante possible ces deux familles de traitements. Une formalisation suffisamment générale de l'expérience doit par exemple permettre de dissocier les traitements usuels des traitements spécifiques du retour d'expérience. Cette séparation délibérée permettra d'étudier le comportement des algorithmes de recherche indépendamment des choix de présentation et d'architecture des applications.

## **I.3. Synthèse**

Les applications et pratiques dites de « retour d'expérience » sont nombreuses, cependant, il existe finalement peu d'outils informatiques traitant l'expérience comme une entité cognitive. En effet, la

majorité des applications traitent l'expérience comme des documents et utilisent des outils de type GED (Gestion Electronique des Documents). Nous proposons une forme de retour d'expérience dit cognitif où l'expérience est une entité modélisée en vue de l'application de traitements et de raisonnements informatiques. Nous faisons d'ailleurs le rapprochement avec les systèmes à base de connaissances. Par rapport à ces derniers, le retour d'expérience cognitif a pour particularité notable que la connaissance est explicitée de manière incrémentale.

L'architecture présentée, proposée initialement dans (Rakoto, 2004) nous a semblé être suffisamment générale et pragmatique pour l'opérationnalisation d'un système de retour d'expérience cognitif générique. Nous avons étendu cette architecture en précisant le processus d'exploitation de prévention que nous avons étudié dans nos travaux et qui n'était pas détaillé auparavant. Dans la suite de ce mémoire, nous nous attacherons à l'étude des dimensions « informations et connaissances » et « techniques et outils » car ce sont les deux clés de l'opérationnalisation des systèmes de retour d'expérience. Ainsi, le chapitre II est consacré à la représentation. Cette représentation possède la particularité de tenir compte de l'imperfection des informations disponibles. Les chapitre III et IV sont consacrés aux techniques d'exploitation associées aux expériences, avec en particulier le développement d'algorithmes de similarité permettant de faire des recherches dans la base d'expérience.

# Chapitre II

## Modélisation des expériences

Ce chapitre est consacré à la formalisation et à la structuration de l'expérience. Le choix du formalisme servant à représenter l'expérience est justifié en considérant que le modèle est destiné aux traitements de recherche et d'extraction d'un indicateur du risque qui sont présentés dans les chapitres III et IV. Dans cet objectif, deux impératifs majeurs doivent être considérés :

- le langage du système doit être suffisamment intuitif et naturel pour ne pas compliquer son utilisation car les experts du domaine concerné ne sont pas, a priori, informaticiens ou logiciens,
- les incertitudes doivent être prises en compte afin de refléter, d'une part, l'imperfection de la représentation et des données capturées et, d'autre part, l'incertitude relative à l'analyse des experts.

Il faut donc définir un langage adapté à la représentation de l'expérience dont nous avons déjà proposé, dans le chapitre I, une structuration globale. Nous avons structuré l'expérience en plusieurs sections : un événement, son contexte, des analyses expertes réalisées à partir du contexte, des solutions et des leçons apprises.

Nous avons aussi choisi de nous focaliser sur le cycle de prévention essentiellement en raison du domaine d'application de nos travaux mais aussi parce que cet aspect du retour d'expérience est assez peu abordé dans la littérature. Dans ce cycle de prévention, les solutions qui ont été mises en œuvre auparavant ne sont pas forcément très intéressantes (dans le contexte de notre domaine d'application, une solution serait par exemple la planification d'une opération de secours en montagne). En revanche, un point important à prendre en compte est la manière dont les experts analysent le contexte pour en caractériser les éléments qui ont favorisé l'occurrence de l'événement négatif. C'est donc sur ce dernier aspect que nous focalisons nos travaux.

Nous avons également choisi d'orienter l'analyse en vue de l'extraction d'une connaissance de prévention des risques. Cependant, nous n'utilisons pas les techniques probabilistes classiques d'estimation des risques car le type d'expérience traité, de part sa nature incomplète, imparfaite et disponible en petit nombre, est mal adapté à une analyse de ce genre. Par contre, une relation intuitive existe entre l'évaluation d'un risque et l'expérience passée. Nous proposons une première approche d'explicitation de ce lien et nous la détaillons dans la partie IV.3. En effet, nous pensons que, de manière générale, une situation vécue qui conduit à un accident peut être reconnue par des signaux particuliers. Une analyse classique des risques consisterait à rechercher l'ensemble des causes racines par exemple en utilisant un arbre des causes. Bien que cela ne soit pas incompatible avec le retour d'expérience que nous proposons, nous avons décidé d'adopter une première approche plus simple et plus directe qui consiste à souligner, dans le contexte, la présence d'attributs, désignés comme des « signaux », qui sont des caractéristiques significatives sans lesquelles l'événement n'aurait pas eu lieu selon l'avis des experts du domaine.

Comme cela a été évoqué dans le chapitre I, les systèmes de retour d'expérience que nous visons peuvent être considérés comme des systèmes à bases de connaissances particuliers. C'est pourquoi, avant de préciser nos choix de modélisation de l'expérience, nous présentons les principaux formalismes de représentation des connaissances utilisés dans ces systèmes en distinguant les approches logiques, celles basées sur les *frames* et les approches hybrides. Ensuite, nous faisons une synthèse des différentes techniques de représentation et de prise en compte des incertitudes pour finalement détailler le choix que nous avons effectué pour la représentation de l'expérience.

## II.1. Représentation des connaissances

Les modèles généraux de représentation des connaissances sont nombreux. Il s'agit de langages qui permettent de représenter différents types d'entités : objets physiques, idées, actions, relations... par opposition à des modèles qui se concentrent sur un domaine particulier (relations temporelles, tâches, taxonomie...).

Nous trouvons dans la littérature deux grandes approches distinctes : les représentations basées sur une logique et les représentations basées sur les *frames* (approche structurale). Ces approches sont souvent combinées en approches qualifiées d'« hybrides ». Avant de détailler les principales approches de représentation des connaissances, nous précisons la distinction entre l'approche logique et l'approche structurale.

### II.1.1. Approches basées sur la logique

Les premiers langages utilisés pour représenter les connaissances étaient d'origine mathématique, basés sur un formalisme logique et en particulier sur la logique des propositions ou la logique des prédicats. Par exemple, de nombreuses ontologies ont été codées en utilisant des expressions logiques dans des formats développés par des logiciens. La logique, de manière générale, est un domaine des mathématiques et de la philosophie en rapport avec la manipulation d'expressions dont la valeur de vérité peut être établie. On trouve différents types de logiques comme, par exemple, la logique des propositions (aussi appelée calcul des propositions), ou la logique des prédicats qui définit des opérateurs comme « quelque soit », « il existe ». Dans ce deux cas, l'évaluation des expressions est binaire c'est-à-dire à valeur dans l'ensemble {vrai, faux}. Il existe beaucoup d'autres types de logique avec en particulier celles qui autorisent des valeurs graduelles comme les logiques modales, proches de la logique floue bien que cette dernière soit davantage basée sur la théorie des ensembles, ou encore des logiques temporelles qui prennent en compte les relations temporelles dans les inférences.

Un langage logique est caractérisé par un système de symboles et de règles qui définissent un langage en termes de syntaxe et par une interprétation qui fournit la sémantique de ce langage. Il est alors possible de combiner les symboles sous formes de formules logiques. La sémantique associée dépend naturellement de la logique considérée. A partir d'un ensemble d'expressions valides (la base de connaissances), un système de déduction (le moteur d'inférence) permet de raisonner en construisant des démonstrations et des propositions nouvelles. Ces dernières sont inférées par le système et viennent compléter la base pour créer le corps de connaissances. Les mécanismes d'inférence peuvent aussi permettre de répondre à une question posée.

Les approches logiques de représentation des connaissances reposent sur le même principe général : la construction de nouvelles expressions par inférence. Il existe plusieurs types d'inférences : la déduction, l'induction, ou encore l'abduction. Récemment, une distinction a été proposée dans le domaine de l'AGI (Artificial General Intelligence) sur le niveau d'axiomatisation des langages (Wang, 2006). Les systèmes de raisonnement basés sur une approche logique appartiennent à l'une des catégories suivantes :

- Axiomatique
- Semi-axiomatique
- Non-axiomatique

Les systèmes axiomatiques correspondent aux systèmes logiques mathématiques. L'hypothèse fondamentale est ici que le système possède suffisamment de connaissances et de ressources par rapport au problème à résoudre. Les systèmes semi-axiomatiques correspondent aux systèmes de raisonnement classiques que l'on retrouve en Intelligence Artificielle. La connaissance et les ressources sont suffisantes pour certains aspects de la résolution de problèmes mais pas tous. Cette approche repose sur les logiques mathématiques avec quelques adaptations en fonction de l'utilisation prévue de

la représentation (logique multi-valuée, temporelle...). Néanmoins, pour ces systèmes les outils mathématiques sont les mêmes que ceux employés pour les systèmes axiomatiques. Ils reposent donc sur les mêmes bases mathématiques. Les systèmes non-axiomatiques sont basés sur l'hypothèse qu'à tout moment, la connaissance et les ressources sont insuffisantes pour la résolution d'un problème. Le système doit pouvoir fonctionner en tenant compte de cette contrainte. P. Wang défend l'idée que pour l'Intelligence Artificielle, il est nécessaire d'adopter une approche non-axiomatique afin de ne pas faire d'hypothèses excessives sur la complétude de la connaissance disponible (Wang, 1995) (Wang, 2006). Par conséquent, certaines caractéristiques essentielles de la logique axiomatique, telles que la cohérence, ne sont plus assurées. En réponse, le système a un comportement intelligent, suivant la définition de Wang, car il s'adapte à la nature de la connaissance contenue dans la base. Il peut d'ailleurs avoir des comportements imprévisibles et surtout non renouvelables puisqu'il adapte ses réponses, ses inférences à la connaissance qui évolue en cours d'utilisation. Nous pouvons citer comme exemple de tels systèmes OpenNARS<sup>1</sup> et OpenCOG<sup>2</sup>.

Signalons qu'il existe aussi des langages d'échange des formalismes de représentation des connaissances basés sur la logique des prédicats. KIF (Knowledge Interchange Format) (Genesereth et Fikes, 1992) est une généralisation de ce type de langage. C'est un moyen d'échange de ce type de connaissances qui possède les caractéristiques suivantes :

- Une sémantique déclarative permettant d'exprimer des expressions formelles précises sémantiquement par l'utilisation de la logique des prédicats,
- Un méta-langage KIF Meta permettant de préciser la sémantique d'expressions particulières,
- Un ensemble d'entités et de notions comme l'objet, les fonctions et les relations qui permettent des règles de raisonnement non-monotones.

KIF n'a cependant pas abouti à un standard et depuis peu, un nouveau formalisme d'échange appelé Common Logic (CL<sup>3</sup>) approuvé par l'ISO en octobre 2007 est développé.

### ***Synthèse critique sur l'utilisation d'un formalisme logique en retour d'expérience***

D'un point de vue épistémologique, l'approche logique est bien adaptée à une large gamme de tâches de représentation (Hayes, 1977) (Moore, 1982). Néanmoins, les langages associés présentent de sévères limitations calculatoires. La logique des prédicats est seulement semi-décidable, et même pour des cas relativement simples. De plus elles sont souvent délicates à intégrer dans des systèmes logiciels actuels qui sont le plus fréquemment développés selon le paradigme orienté objet, dans des architectures de type MVC (Model - View - Controller).

Un inconvénient majeur de cette approche est que la logique du premier ordre est difficile à lire, à communiquer et à comprendre. Comme le font remarquer de nombreux auteurs, les modèles formels sont difficiles à construire (Ducournau *et al.*, 1998). Pour pouvoir utiliser des systèmes basés sur la logique, il faut disposer d'une compétence de logicien. Une autre possibilité consiste à utiliser des dispositifs de capture basés sur le traitement du langage naturel (NLP), mais il s'agit là d'un domaine de recherche à part entière. Ces observations nous ont poussé à ne pas nous orienter vers une approche de type logique.

Nous remarquons simplement que l'approche de logique non-axiomatique aurait, malgré tout, pu constituer un choix approprié puisque les bases et hypothèses sont similaires à nos considérations initiales. En effet, la construction de la connaissance est incrémentale et passe par la prise en compte progressive des nouvelles informations, à l'instar du retour d'expérience. Toutefois, des problèmes de lisibilité subsistent. Pour illustrer ce problème examinons deux expressions dans le formalisme NAL (Non-Axiomatic Logic) de NARS peuvent être écrites :

- Expression simple : < avalanche --> événement > %1.0 ;0.9% exprime avec certitude que l'avalanche est un (type d') événement.

<sup>1</sup> <http://code.google.com/p/open-nars/>

<sup>2</sup> [http://www.opencog.org/wiki/Main\\_Page](http://www.opencog.org/wiki/Main_Page)

<sup>3</sup> <http://common-logic.org/>

- Expression composée :  $\langle \langle \#x \rightarrow \text{asthmatique} \rangle \Rightarrow \langle (*, \#x, \text{ventoline}) \rightarrow \text{utilise} \rangle . \%0.9 ; 0.9\%$  exprime qu'une personne asthmatique utilise de la ventoline avec une certitude élevée.

Il existe des expressions bien plus complexes, en particulier quand les aspects temporels et procéduraux (actions qui modifient l'état du monde) sont pris en compte. Ces deux exemples montrent bien que la lecture n'est pas directe. Il faut connaître la signification des symboles et, pour construire de nouvelles expressions, il faut connaître et respecter la syntaxe du langage. Ceci nécessite donc un apprentissage très difficile à envisager pour les utilisateurs d'un système de retour d'expérience.

Il existe des formalismes plus structurés, plus contraints et donc limités dans leur expressivité. Ces langages permettent aisément de décomposer l'information sans entrer dans la définition sémantique de tous les liens existants. Cet aspect de décomposition et de structuration de l'information est important et ne nécessite pas une forte cohérence de l'information. La perte d'expressivité est compensée par le gain en termes de lisibilité et de prise en main. Les formalismes basés sur les *frames* et orientés objet permettent cette approche plus pragmatique et sont présentés dans les parties suivantes.

## II.1.2. Approche basée sur les *frames*

Le modèle des *frames* (schémas ou cadres) a été introduit par Minsky en 1975 et est apparu dans le cadre de la représentation des connaissances acquises par la vision (Minsky, 1975). Il propose d'organiser la connaissance au moyen d'un ensemble de *frames*. Ces derniers sont censés capturer l'essence de situations stéréotypées comme « être dans un lieu » ou « faire une activité » en regroupant toute l'information utile en rapport avec ces situations en vue de son exploitation future. Les collections de *frames* sont organisées et interconnectées dans des systèmes à base de *frames*. L'approche à base de *frames* trouve aussi des ramifications dans les premiers travaux sur les langages de programmation orientés objet tels que SIMULA et Smalltalk qui ont été développés dès la fin des années 60 et dans les années 70.

Un *frame* est donc un modèle qui organise la connaissance dans des structures qui contiennent un nom auquel est associé un ensemble d'attributs. Les attributs des *frames* sont représentés par un ensemble de facettes permettant de les caractériser selon des points de vue différents. Le terme de « facette » est inspiré de la métaphore des différentes faces sous lesquelles il est possible de voir un objet donné. Il existe, bien sûr, une facette qui permet de retourner la valeur de l'attribut mais il en existe beaucoup d'autres (contrainte de cardinalité, de validité, facette de calcul de valeurs par défaut, facette d'inférence de valeurs...). Cette notion de facette est centrale dans le modèle de représentation des connaissances à base de *frames* et peut être vue comme un médiateur entre l'attribut et sa valeur.

La communication de Minsky est assez vague concernant l'implémentation de ce modèle (Ducournau *et al.*, 1998) (Minsky, 1975). Il s'agit d'une première approche sur ce que les *frames* pourraient être en exposant les principes généraux. Contrairement aux représentations logiques, les *frames* sont flexibles et non-formelles. L'objectif est de pallier les faiblesses des représentations logiques, en particulier la difficulté d'appropriation et de lisibilité par des non logiciens. Parmi les autres raisons invoquées par Minsky, on trouve la critique de l'organisation de la connaissance qui est distribuée sur des axiomes indépendants, la monotonie de nombreuses logiques et surtout l'obligation de cohérence de la base de connaissances. Il estime que propriété de cohérence n'est même pas désirable dans une base de connaissances étant donné que la connaissance humaine ne la vérifie pas. Ce dernier argument est d'ailleurs repris comme un argument fondamental dans les logiques non-axiomatiques (Wang, 2004).

Le moyen d'inférence privilégié pour les approches à base de *frames* est l'« appariement » qui correspond à une mise en correspondance d'un *frame* volontairement incomplet (équivalent à une requête) avec les *frames* de la mémoire de connaissances. Ce processus consiste donc à comparer deux objets en fonction de leurs facettes et valeurs pour vérifier si elles correspondent ou non, si l'un est plus complet que l'autre ou si l'un peut être complété par l'autre de manière compatible. C'est un moyen très performant pour comparer les objets entre eux.

### **Synthèse critique sur l'utilisation d'un formalisme à base de frames en retour d'expérience**

La principale critique formulée à l'égard de l'approche de représentation à base de *frames* est le manque de formalisation de ce cadre qui, selon les logiciens, ne permet pas d'inférer des connaissances valides. Hayes a exposé la théorie des *frames* du point de vue de la logique (Hayes, 1980). Il a montré comment il était possible de reconstruire la théorie des *frames* sur les bases de la logique. Il n'existe pas aujourd'hui de modèle de *frames* exploitable car ce dernier n'a pas abouti à des standardisations suffisantes. Il a néanmoins été à l'origine des langages de programmation à prototypes mais aussi, dans une certaine mesure, de la définition des modèles objet de représentation des connaissances. Ces langages dit « post-frame » ou encore « centrés objet » sont les descendants directs de cette approche.

## **II.1.3. Approches hybrides de représentation des connaissances**

Les recherches en représentation des connaissances semblent avoir établi un consensus selon lequel il est intéressant de combiner le formalisme logique et celui basé sur les *frames*, en particulier pour établir un compromis intéressant entre adéquation épistémologique et adéquation heuristique. Nebel a introduit le terme de langage hybride pour y faire référence (Nebel, 1990). Nous présentons successivement les trois familles de langages hybrides les plus communes :

- les réseaux sémantiques et les graphes conceptuels,
- les langages post-frames et centrés objets,
- les logiques de description.

### **II.1.3.1. Réseaux Sémantiques et Graphes Conceptuels**

L'utilisation en représentation des connaissances de la logique seule n'est pas aisée car il faut avant tout maîtriser les aspects formels. Certains langages ont été développés de manière pragmatique tout en étant très liés à une interprétation logique. Les réseaux sémantiques et les graphes conceptuels en sont de très bons exemples, les seconds étant une évolution des premiers, plus précise sémantiquement.

#### **Réseaux sémantiques**

Les réseaux sémantiques trouvent leur origine dans des travaux du domaine de la psychologie associés aux traitements de la parole et de l'écriture (Richens, 1958), (Masterman, 1962). Quillian les a formalisés au cours de ces travaux de thèse (Quillian, 1966) avec l'objectif de constituer une « mémoire sémantique dont le but était de constituer une théorie sur l'organisation et l'utilisation de la mémoire humaine du point de vue sémantique et conceptuel. Les réseaux sémantiques reposent sur un modèle de la mémoire qui met en évidence la proximité entre des concepts mémorisés. Cette approche préconise l'utilisation de nœuds servant à représenter les entités conceptuelles (telles que des concepts, des mots...). Ces nœuds sont interconnectés par des liens (des arcs) représentant des relations conceptuelles. L'idée d'organiser ces concepts en taxonomie provient aussi de ces travaux. Par contre, la notion d'héritage, de subsumption de propriétés a été introduite plus tard (Collins et Quillian, 1970). Cette technique consiste à considérer des concepts avec des sur-concepts plus généraux ou des sous-concepts plus spécifiques. Concernant les moyens d'inférence, la technique générique utilisée est la recherche par propagation d'activation (*spreading activation search*). Elle est basée sur l'hypothèse que chaque concept est déterminé par tous les concepts qui peuvent être atteints directement ou indirectement par l'intermédiaire des arcs. Deux concepts sont comparés en déterminant l'ensemble des concepts qui peuvent être respectivement atteints.

De nombreux systèmes, basés sur les réseaux sémantiques, ont été développés. La raison de cette popularité est probablement leur aspect intuitif. En effet, il semble assez naturel d'organiser des concepts dans une hiérarchie interconnectant les concepts selon différents liens par exemple, linguistique, physique, ou encore conceptuel... Les réseaux sémantiques sont une représentation pratiquement directe de l'« espace des concepts ». L'implémentation est également assez simple. Le

caractère intuitif de ce formalisme est cependant aussi une de ses limites. Woods, par exemple, critique une tendance répandue dans la plupart des réseaux sémantiques : la représentation d'aspects fondamentalement différents avec le même élément, c'est-à-dire les mêmes arcs entre deux concepts (Woods, 1975). Il en résulte une confusion sémantique gênante pour toute interprétation automatisée. Ces confusions reposent, par exemple, sur la définition de nœuds concepts ou instances, valeurs possibles ou véritables... Directement inspiré des réseaux sémantiques, le modèle des graphes conceptuels (GC) défini par Sowa (Sowa, 1984) est lui-même doté d'une sémantique formelle (Chein et Mugnier, 1992).

## Graphes Conceptuels

D'après Sowa (Sowa, 1984), les graphes conceptuels sont des graphes finis, connectés et bipartis. Les deux sortes de nœuds du graphe biparti sont les concepts et les relations conceptuelles. Chaque relation conceptuelle possède un arc ou plus, et chaque arc doit être lié à un concept. Une relation conceptuelle relie plusieurs concepts. Ces deux types de nœuds définissent une structure ontologique à partir de laquelle sont créés des ensembles de concepts et de relations conceptuelles. Ces éléments permettent de construire les différents graphes conceptuels nécessaires à la représentation de la connaissance. Ainsi, les concepts et les relations sont définis précisément dans une ontologie formelle où peuvent être exprimées des contraintes, des pré-requis, des définitions et des conditions nécessaires et/ou suffisantes.

Dans le formalisme des graphes conceptuels, la connaissance est encodée à partir d'un support formé des treillis de concepts et des relations. La sémantique est ainsi définie précisément pour chaque relation et consiste principalement en des relations de subsomption ensembliste, proches de la notion d'héritage. Il est possible d'imbriquer les graphes conceptuels permettant ainsi d'associer tout nœud concept avec une description interne (Baget et Mugnier, 2002). Une représentation proche de l'esprit des *frames* est alors obtenue bien que les graphes conceptuels restent plus proches des approches logiques. En effet, l'une des caractéristiques fondamentales des graphes conceptuels est qu'il est possible de les formuler selon la logique des prédicats du premier ordre.

Concernant l'aspect inférence, l'opération privilégiée des graphes conceptuels est la projection (Sowa, 1984). Les graphes conceptuels peuvent être représentés en logique des prédicats et l'opération de projection permet d'effectuer des déductions du point de vue logique autorisant l'extraction des sous-graphes d'un graphe général en appliquant une succession de règles de spécialisation. Cette opération est « *sound* » et « *complete* » (Chein et Mugnier, 1992).

### ***Synthèse critique sur l'utilisation d'un formalisme à base de réseaux sémantiques ou de graphes conceptuels en retour d'expérience***

Les réseaux sémantiques semblent constituer une représentation intéressante pour la représentation des expériences. Toutefois, ils posent différents problèmes bien identifiés dans la littérature et que nous considérons comme rédhibitoires pour les approches que nous souhaitons mettre en œuvre (en particulier, ils ne possèdent pas une sémantique suffisamment précise).

Les graphes conceptuels répondent en grande partie aux problèmes posés. Ce sont également des outils transposables en logique sans avoir à travailler directement dans un formalisme logique. En effet, ils sont plus faciles d'accès, du fait de la notation linéaire et de la proximité de l'expression en langage naturel. Leur inconvénient majeur reste que l'opération de projection, mécanisme d'inférence privilégié, peut induire des temps de calcul importants dans certains cas. Malgré tout, les graphes conceptuels sont constitués d'un formalisme qui semble prometteur et nous avons d'ailleurs débuté des travaux (non consignés dans ce mémoire de thèse) sur leur utilisation pour la formalisation du retour d'expérience (Kamsu-Foguem *et al.*, 2008).

### II.1.3.2. Formalisme « Post-frames » et représentation des connaissances par objets

Les systèmes « post-frames » sont caractérisés, d'une part, par une rationalisation de l'utilisation des facettes introduites avec les *frames* qui sont ici conservées en nombre restreint mais parfois extensible à des fins de typage ou d'inférence des valeurs et, d'autre part, par l'adoption de la notion de classe (avec une distinction entre classe et instance) et d'une relation de type « généralisation-spécialisation » clairement établie. Dans les langages de *frames*, on ne trouve pas la distinction entre classe et instance. Par contre, en comparaison des Langages de Programmation Orientés Objet (LPOO) où cette distinction existe, une instance n'est pas contrainte à l'appartenance à une seule classe. Cette différence est fondamentale pour la représentation des connaissances. Une instance peut donc appartenir à plusieurs classes un peu à la manière de catégories.

Les systèmes dits de Représentation des Connaissances par Objet (RCO) sont apparus durant les années 90. Le plus connu dans la communauté française est certainement TROEPS (Euzenat, 1999). Ces systèmes sont très proches des langages post-frames mais la base de représentation est plus proche du paradigme orienté objet des langages de programmation. Ils possèdent cependant des spécificités développées au cours des années de recherche comme les notions de point de vue sur des concepts et de passerelle entre ces points de vue (Carré, 1989) (Dekker, 1994).

Les éléments communs à toutes ces approches ont été repris dans un protocole générique permettant d'accéder et d'échanger des connaissances exprimées nommé Open Knowledge Base Connectivity (OKBC) (Chaudhri *et al.*, 1998). En fait, OKBC est plus qu'un simple formalisme d'échange puisque ce protocole assez général présente les caractéristiques majeures des systèmes de représentation de connaissance à base de *frames*. Il est donc devenu un standard qui présente une articulation entre la structure permettant la représentation des connaissances (*frames*, axiomes, règles de production,...) et le support regroupant les traitements utilisés lors des raisonnements (classification, vérification de contraintes, chaînage avant et arrière...). Citons plus récemment, l'implémentation de Protégé<sup>1</sup> (Fridman-Noy *et al.*, 2000) (Grosso *et al.*, 1999) basée, à l'origine, sur un modèle de type OKBC (Protégé-frame).

Actuellement Protégé existe aussi sous une forme basée sur OWL (Protégé-OWL), un type de logiques de description définies pour être opérables sur Internet. Les logiques de description sont présentées dans la partie suivante. Protégé est certainement aujourd'hui l'outil de capitalisation des connaissances le plus couramment utilisés, en particulier dans des contextes industriels.

### II.1.3.3. Logiques de description (Description Logics, DL)

Dans la lignée des travaux de Brachman et Levesque (Brachman et Levesque, 1985), renforcés par un courant formel (Nebel, 1990), les représentations de connaissance hybrides ont évolué vers une formalisation croissante qui a conduit à définir la sémantique dénotationnelle de ces langages (formalisation mathématique du langage), à évaluer la décidabilité et la complexité de certains mécanismes de déduction et à concevoir des algorithmes corrects pour ces mécanismes. Les logiques de description forment une famille de langages de représentation de connaissance qui peuvent être utilisés pour représenter la connaissance terminologique d'un domaine d'application d'une façon structurée et formelle. On peut alors proposer des outils formels permettant de réaliser automatiquement des tâches comme la vérification de cohérence ou encore la classification de nouveaux concepts.

Les logiques de description constituent une formalisation du support de modélisation conceptuelle apporté en premier par le système KL-One (Brachman et Schmoltze, 1985). Les logiques de description utilisent les notions de concept, de rôle et d'individu. Les concepts correspondent à des « classes d'éléments » et sont interprétés comme un ensemble dans un univers donné. Les rôles correspondent aux « liens entre les éléments » et sont interprétés comme des relations et enfin les individus

---

<sup>1</sup> <http://protege.stanford.edu/>

correspondent aux éléments. Dans la plupart des langages de type « logiques de description », nous trouvons la dissociation entre la connaissance terminologique qui définit les concepts généraux, basiques et dérivés, et la connaissance assertionnelle qui apporte des connaissances en relation avec les individus. L'extension d'un concept représente des individus. L'intention d'un individu représente la ou les classes (concepts) dont il dérive.

Dans les logiques de description, toutes les informations connues sont alors modélisées comme un couple  $\langle T, A \rangle$ , où T est un ensemble de formule exprimant des connaissances générales (terminologiques) regroupé dans une T-Box et A est un ensemble de formules exprimant des connaissances spécifiques (assertionnelles) à des individus regroupé dans une A-Box. Une série d'outils a été développée comme F-Logic (Kifer *et al.*, 1995) qui a servi de référence à plus de cinquante langages de spécification des ontologies tel que Ontolingua (Gruber, 1993), LOOM (Mac Gregor et Burstein, 1991), le langage de modélisation de CommonKADS (Schreiber *et al.*, 1994), OIL (Fensel *et al.*, 2000) et plus récemment OWL<sup>1</sup> pour le web sémantique.

### **Synthèse critique sur l'utilisation d'un formalisme à Logiques de Description en retour d'expérience**

Les logiques de description sont une version formelle et précise de représentation des connaissances qui aspirent à assurer et à contrôler les inférences réalisées, les tâches de classification ou encore les requêtes. L'inconvénient majeur est que les systèmes existants, Protégé par exemple, sont utilisables à la manière d'une base de données à vocation sémantique et sont particulièrement adaptées pour élaborer des systèmes interopérables. D'ailleurs, récemment avec le Web sémantique, elles ont trouvé un champ d'application idéal. Dans le cadre de nos travaux, nous devons définir des modèles d'expérience dans un langage adapté et élaborer des mécanismes d'inférences très différents des mécanismes d'inférence standard des logiques de description. Nous n'avons pas d'objectif d'interopérabilité et c'est pourquoi les logiques de description et tous les outils formels associés ne sont pas, a priori, très adaptés.

## **II.1.4. Représentation « orientée objet » des connaissances**

Les méthodes d'Ingénierie des Connaissances et de Génie Logiciel avec en particulier les Langages de Programmation Orientés Objet (LPOO) sont relativement proches dans leurs principes. Bien que la littérature du génie logiciel et de l'ingénierie des connaissances fasse apparaître une séparation assez stricte entre les deux, un noyau commun existe comme cela est illustré sur la Figure II-1 extraite de (Knublauch, 2002). Dans cette figure, les primitives communes entre représentation d'ontologie et représentation orientée objet sont mises en relief.

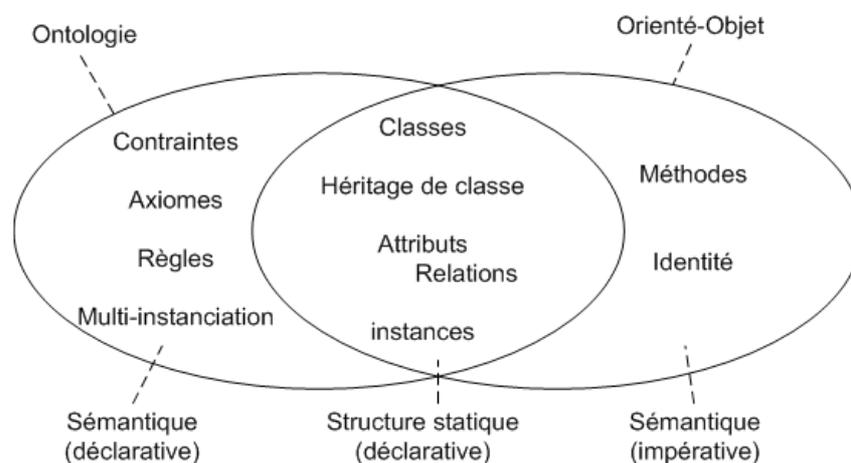


Figure II-1 - Primitives de modélisation ontologie et orientée objet (Knublauch, 2002)

<sup>1</sup> <http://www.w3.org/TR/owl-features/>

Comme nous pouvons le voir, il y a une communauté des primitives de représentation des ontologies préconisées en ingénierie des connaissances et de celles de la programmation orientée objet. Les deux approches sont des abstractions de domaines qui permettent de modéliser des entités et leurs relations. Certains principes de conception des modèles à base de classes et des modèles ontologiques sont identiques : modularité, cohérence interne élevée, extensibilité, utilisation de catégories et de noms naturels. Les primitives communes sont celles relatives à la structuration de l'information avec notamment la notion de classes, d'héritage, d'instances, d'attributs et de relations. Les différences majeures proviennent de la sémantique impérative des LPOO introduite, en particulier, par les méthodes. Au contraire, les langages de représentation des connaissances qui servent à définir et représenter des ontologies utilisent davantage la sémantique déclarative en apportant des cadres d'expression de contraintes, d'axiomes, de règles.

## Transparence sémantique

La particularité des langages de représentation des connaissances est leur non-ambiguïté qui est la conséquence de la transparence sémantique. Cette caractéristique fondamentale fait que chaque élément du modèle peut être interprété et compris par le système. Ainsi, les machines (et les utilisateurs) utilisent des modèles qui ont une signification précise, autrement dit, qui ne sont pas ambigus. L'interprétation par le système de ce que modélise l'utilisateur a donc une signification précise et correcte. La principale limite pour la représentation des connaissances du paradigme orienté objet est justement le manque de transparence sémantique. Les formalismes orientés objet ne définissent pas formellement toutes les primitives de modélisation associées et c'est pourquoi les modèles de domaine orientés objet ne peuvent pas permettre d'accomplir actuellement d'autres tâches que celles codées explicitement par les développeurs. Si nous voulons exprimer la sémantique de manière transparente, et ainsi rendre tout artefact de modélisation accessible informatiquement, il faut augmenter la sémantique du langage en précisant son modèle (cadre de la métamodélisation) ou en associant des méta-informations (informations qui qualifient les informations). Nous traitons cette problématique dans la partie V.3.

## UML comme support de représentation des connaissances

UML<sup>1</sup> est le langage de représentation universel orientée objet qui reprend les principes génériques de l'approche objet déclinée selon plusieurs diagrammes types. Il est standardisé par l'OMG<sup>2</sup>. Les diagrammes, associés à leur modèle standardisé, sont des outils graphiques en partie interprétables par un système informatique. UML est incontournable du fait de son universalité et de son acceptation généralisée dans les domaines logiciels mais aussi industriels. Cependant, une de ses limites pour la représentation des connaissances est relative à son déficit de sémantique, ce qui explique qu'il ne soit pas totalement exécutable. Pour combler cette lacune, une initiative du groupe pUML<sup>3</sup> (precise UML) (Evans et Kent, 1999) a pour objectif de définir la sémantique du modèle. Depuis 2000, les membres de ce groupe ont lancé la revue SOSYM<sup>4</sup> (Software and Systems Modeling). Bien que limité, UML peut servir de formalisme de représentation d'ontologies (Cranefield et Purvis, 1999). Les notes UML qui, bien que n'ayant pas d'implémentation logicielle directe, permettent de bien représenter la connaissance logique à appliquer au modèle du domaine. Pour illustrer l'utilisation d'UML en tant que support de représentation des connaissances, une modélisation simple mais typique de domaine à savoir celle d'une famille est représentée la Figure II-2. Cette modélisation de domaine est réalisée par l'intermédiaire d'un diagramme de classes. Ainsi, UML peut servir de support à la représentation de connaissances de domaine, et donc, dans une certaine mesure, UML permet de décrire des ontologies.

Dans le modèle simple d'une famille exprimé en UML (*cf.* Figure II-2), une grande partie de la sémantique du domaine est exprimée sous formes de notes. Ces assertions sont importantes car elles contraignent le système à des scénarios valides. Ce type d'informations n'est toutefois pas exploitable automatiquement par le système car la sémantique des notes n'est pas précise (du point de vue du système). Ces notes servent seulement à la définition du modèle conceptuel et, bien que transformées

---

<sup>1</sup> <http://www.uml.org/>

<sup>2</sup> <http://www.omg.org/>

<sup>3</sup> <http://www.cs.york.ac.uk/puml/>

<sup>4</sup> <http://www.sosym.org/>

en code lors de l'implémentation, il n'existe pas de règles et de méthodologies d'intégration de ces considérations. Afin de pallier les inconvénients des notes textuelles, une approche consiste à utiliser un langage de contraintes associé à UML comme OCL (Object Constraint Language) (Warmer et Keklepe, 1998). OCL est un langage objet qui sert à exprimer des contraintes logiques sur des instances (des objets). C'est une notation très pratique pour la représentation des relations logiques entre objets qui est proche des possibilités des LPOO (en particulier les langages dynamiques tels que Smalltalk, Ruby) (Renggli *et al.*, 2007). Cependant, OCL sert principalement à la documentation des modèles UML. Les contraintes exprimées ne peuvent pas être directement transformées en code tandis que les diagrammes de classes peuvent générer des parties de code. Il n'y a pas de règles qui définissent dans quelles méthodes ces contraintes pourraient être implémentées.

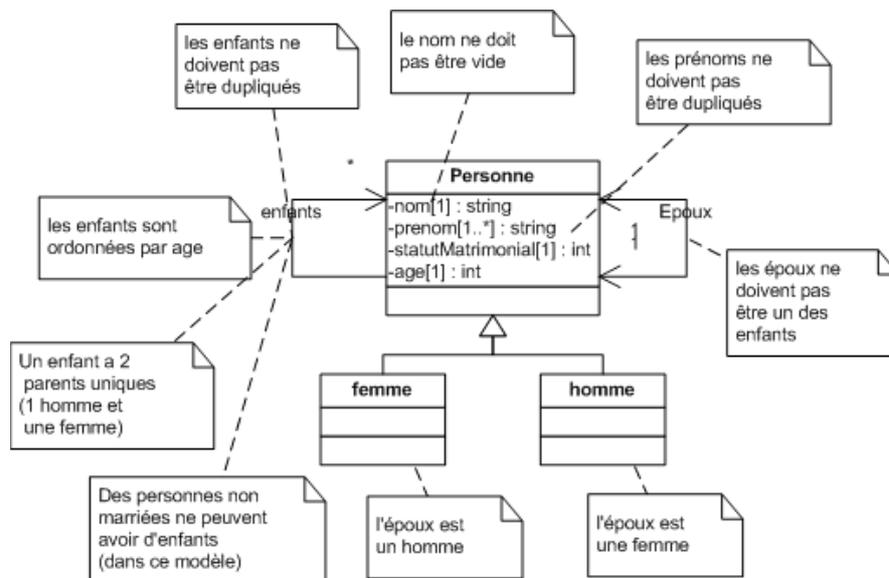


Figure II-2 Modélisation UML de connaissance (exemple : famille)

### **Synthèse critique sur l'utilisation d'un formalisme de type Langage de Programmation Orientée Objet en retour d'expérience**

Au contraire des différentes approches d'ingénierie des connaissances, l'approche orientée objet est considérablement plus mature et accessible sur le plan de l'opérationnalisation des systèmes qui constitue un des objectifs de nos travaux. Bien sûr, elle manque de support explicite de modélisation des connaissances mais elle est utilisée depuis de nombreuses années dans le développement d'applications modulaires et extensibles. La modularisation favorise une meilleure maintenance et facilite l'extension des programmes.

En outre, il existe plusieurs techniques de programmation dans les LPOO qui permettent d'exprimer et de prendre en compte des contraintes, des axiomes, et des règles explicitement. La difficulté est de combiner ces éléments sans trop alourdir le système. Les annotations de méthodes (*cf.* V.3.3.2), peuvent constituer une technique intéressante. Il s'agit d'associer des métadonnées en entête de la définition des méthodes. Les métadonnées sur les objets sont distinguées des objets eux-mêmes à la fois à l'exécution et lors de la conception. A l'exécution, les métadonnées sémantiques sont par exemple utilisées pour valider la base de connaissances, pour effectuer des raisonnements et pour implémenter des outils génériques d'acquisition des connaissances. Lors de la conception, les métadonnées peuvent être éditées par l'intermédiaire d'un outil qui serait équivalent à un outil simple de modélisation d'ontologie.

Une autre technique de programmation intéressante que nous aurions pu retenir, toujours dans l'objectif de dissociation des connaissances et des traitements, est la programmation par aspects qui permet de créer des règles, des contraintes de manière déclarative et de les tisser dans le corps procédural du programme lors de son exécution (Kiczales *et al.*, 1997). Le tissage est effectué à partir de points d'entrée présents dans chaque méthode. Ces points d'entrée peuvent être avant, après ou même dans le corps d'une méthode.

En outre, contrairement aux langages de représentation des connaissances basés sur des formalismes logiques, assez complexes et dont l'apprentissage n'est pas immédiat, la modélisation orientée objet est plus intuitive et correspond mieux à la représentation du monde des experts. Elle ne souffre pas d'être trop éloignée du code exécutable comme cela est le cas de nombreuses représentations proposées en ingénierie des connaissances. Enfin, la généralisation d'UML est aussi un facteur d'acceptation de ce type de langage comme moyen de représentation des connaissances.

Sur la base des différents points évoqués, nous considérons le paradigme orienté objet comme le point de départ de notre démarche de modélisation. Nous intégrons au fur et à mesure idées et *patterns* propres à la modélisation des connaissances que nous souhaitons utiliser et ceci dans un objectif de cohérence. C'est ainsi que nous avons réalisé un outil spécifique et homogène pour lequel nous avons souhaité obtenir un modèle directement exécutable. Ce modèle est alors pris en charge par des outils standard. Nous présentons la façon de rendre ce modèle « exécutable » dans le chapitre V.

Un des points essentiels de notre approche est lié à la possibilité de représenter de manière explicite l'incertitude liée aux informations contenues dans la base d'expériences et de construire des mécanismes d'inférence intégrant cette incertitude. Concernant cet aspect, nous décrivons dans la partie I.2 différents formalismes de représentation de l'incertitude. Dans la partie I.3 nous détaillons l'intégration de cette incertitude dans le formalisme orienté objet retenu.

## II.2. Modélisation de l'incertitude

Les expériences peuvent être considérées comme des collections d'informations qui traduisent un contexte dont les informations sont rarement connues avec précision. Elles peuvent aussi être totalement inconnues dans certains cas. Le formalisme retenu pour modéliser les expériences doit permettre de représenter ces imperfections de l'information et de les intégrer dans les différents afin de disposer de résultats convenables. L'incertitude doit alors être propagée des données initiales jusqu'au résultat final.

Nous commençons par la présentation du cadre de représentation que nous souhaitons utiliser. Dans un premier temps, nous introduisons les termes relatifs à l'imperfection de l'information comme l'imprécision, l'incomplétude... Dans un deuxième temps, nous introduisons de manière synthétique, la notation employée, la différence entre approche objective et subjective. Nous présentons ensuite globalement, en les positionnant les uns par rapport aux autres, différents cadres majeurs de représentation et de traitement des incertitudes comme la théorie des, la théorie des possibilités et la théorie des fonctions de croyance. Dans le cadre de ce dernier formalisme, nous introduisons la théorie de Dempster-Shafer (Shafer, 1976) plus spécifique car elle est l'approche que nous avons retenue. Après cette partie introductive de positionnement, nous décrivons plus en détails la variante des fonctions de croyance qu'est le Modèle des Croyances Transférables (MCT) (Smets et Kennes, 1994) que nous avons choisi d'utiliser. Enfin, nous présentons comment le couplage est réalisé avec la représentation structurelle « orientée objet » choisie.

### II.2.1. Contexte de la représentation incertaine

Les Systèmes à Base de Connaissances manipulent de l'information en provenance du monde réel, qui représente des activités, des processus... Les différents cadres de modélisation de l'incertitude permettent de représenter explicitement l'information telle qu'elle est obtenue. Ainsi, les résultats inférés sont souvent imprécis mais, en contrepartie, ils reproduisent plus fidèlement la réalité ou plus exactement ce que l'on sait de cette réalité. Dans la partie suivante, nous précisons la notion d'information imparfaite puis nous positionnons différents cadres classiques de modélisation de l'incertitude.

### II.2.1.1. Imperfection de l'information

L'information n'est que très rarement une donnée fiable et parfaite. Plusieurs défauts, comme l'incertitude, l'imprécision et l'incomplétude lui sont souvent associés. Nous résumons ces trois composantes sous le terme générique d'imperfection de l'information. Pour présenter les différentes facettes de cette imperfection, plusieurs termes sont employés. Nous les introduisons en considérant une information mono-source puis multi-sources.

#### *Information en provenance d'une source unique*

L'information qui décrit une caractéristique du monde est imparfaite par nature et ce pour plusieurs raisons (incomplétude, méconnaissance, imprécision...). Nous pouvons distinguer (Dubois et Prade, 2006) :

- L'**incertitude** qui est relative à la vérité d'une information, et caractérise son degré de conformité à la réalité. Elle fait référence à la nature de l'objet ou du fait concerné, à sa qualité, à son essence ou à son occurrence,
- L'**imprécision** qui concerne le contenu de l'information et indique donc son défaut quantitatif de connaissance,
- L'**incomplétude** qui caractérise l'absence d'information apportée par la source sur certains aspects du problème,
- L'**ambiguïté** qui exprime la capacité d'une information à permettre des interprétations différentes.

#### *Information en provenance de sources multiples*

Un des apports communs aux différentes approches de traitement de l'incertain est la possibilité de synthétiser plusieurs informations en une seule. Ceci n'a bien sûr de sens que si les sources sont différentes et expriment une information relative à un même phénomène. Nous trouvons différents termes qui permettent de caractériser une information en provenance de plusieurs sources :

- Le **conflit** caractérise plusieurs informations conduisant à des interprétations contradictoires et donc incompatibles. Les situations conflictuelles sont fréquentes dans les problèmes de fusion de données et aussi de prise de décision (cas des analyses multi-expertes),
- La **redondance** est l'opposé du conflit. Selon certains points de vue c'est un aspect positif, puisqu'il peut venir confirmer une information. La redondance entre les sources est souvent observée, dans la mesure où les sources donnent des informations sur le même phénomène. Dans ce cas elle n'est pas gênante et peut être agrégée.

Ce descriptif succinct nous permet de mieux appréhender les différentes facettes que peut revêtir l'imperfection de l'information en provenance du monde réel. Le lecteur intéressé trouvera plus de définitions de l'imperfection de l'information dans (Smets, 1997). Nous allons maintenant introduire les différentes familles de représentation de ces imperfections en commençant par distinguer les approches objectives des approches subjectives.

### II.2.1.2. Approche objective et subjective

Développée dès les premiers travaux sur la notion de chance (Dubois et Prade, 2006), l'approche subjective a progressivement été écartée au profit de l'approche objective. Or, aujourd'hui, il est reconnu que les approches objectives, sont inutilisables dans la plupart des problèmes réels de prise de décision (Baudrit et Dubois, 2005) C'est pourquoi, plusieurs cadres adaptés à la représentation des multiples facettes de l'information imparfaite ont été développés récemment.

- L'approche objective suppose que la probabilité est une propriété objective du monde modélisé. Etant relative à une épreuve du monde réel, cette approche correspond à l'observation d'un phénomène particulier reproductible. C'est aujourd'hui l'approche la plus connue et utilisée en

particulier par dans le cadre des probabilités et des statistiques. Dans cette catégorie, deux interprétations prédominent (Baudrit, 2006) : l'interprétation fréquentiste où la probabilité d'un événement est définie comme la fréquence limite d'occurrence de cet événement dans un nombre infini d'épreuves et l'interprétation propensive où la probabilité d'un événement est déterminée par les propriétés physiques et objectives de l'objet ou par le processus produisant de l'événement.

L'approche subjective considère la probabilité comme la mesure personnelle de la croyance en l'occurrence de l'événement (Dubois *et al.*, 2003). Dans cette approche, la probabilité n'est pas interprétée comme le passage à la limite d'une fréquence, mais comme la traduction numérique d'un état de connaissance (la connaissance d'un expert par exemple). Citons comme cadres de développements théoriques :

- La méthode bayésienne constitue un cadre applicable à des probabilités a priori subjective. L'inférence bayésienne permet de calculer (ou de réviser) la probabilité d'une hypothèse. Le choix est fait de modéliser une attente, d'estimer un jugement quitte à le réviser en fonction des nouvelles informations dont on dispose. Les probabilités sont basées sur la distribution d'un poids de confiance unitaire sur les singletons du domaine des valeurs possibles. La formulation bayésienne introduit une probabilité conditionnelle ainsi qu'une probabilité a priori (le prior) révisable. Ce cadre reste solidement ancré sur la notion de probabilité. Nous ne détaillons pas cette approche car ses bases probabilistes la rendent mal appropriée à la représentation d'informations incertaines issues de l'expertise (Dubois et Prade, 2006).
- La théorie des possibilités (Dubois et Prade, 1988) fournit des outils mathématiques qui permettent de représenter de l'information incomplète, imprécise ou vague. Elle est une extension décisionnelle de la théorie des sous-ensembles flous (Zadeh, 1978) qui permet de modéliser simplement des catégories du langage naturel, des valeurs appartenant à une échelle numérique... Elle est basée sur la distribution d'un poids de confiance unitaire sur n'importe lequel des sous-ensembles emboîtés du domaine des valeurs possibles. Une description rapide de cette approche est donnée dans la partie III.4.1.
- La théorie des fonctions de croyance fournit, quant à elle, des outils mathématiques permettant de traiter de l'information de nature aléatoire et imprécise. C'est un cadre théorique qui généralise les deux cadres précédents. Les développements reposent sur la distribution d'un poids de confiance unitaire sur n'importe quel sous-ensemble du domaine des valeurs possibles. Il existe plusieurs variantes dont la théorie de Dempster-Shafer (Shafer, 1976) et le Modèle des Croyance Transférables (MCT) (Smets et Kennes, 1994).

Notre objectif étant de proposer un formalisme qui permet d'exprimer, de combiner des avis d'experts ou d'acteur en général, nous nous situons clairement dans une approche subjective. Dans la section suivante, nous présentons succinctement et positionnons les uns par rapport aux autres les trois cadres retenus, à savoir la théorie des possibilités, la théorie des fonctions de croyance et les probabilité bayésiennes.

### II.2.1.3. Positionnement des approches subjectives du traitement de l'imperfection

Théorie des fonctions de croyances, théorie des possibilités et probabilité bayésiennes sont des cadres théoriques qui permettent de prédire (avec incertitude) le résultat d'une « expérience », la réalisation d'un événement. Dans l'approche subjective, il peut aussi s'agir de la croyance (subjective) relative à une question particulière. Plutôt que de présenter en détail chaque cadre, nous avons choisi de les décrire de manière assez générale et uniforme, c'est-à-dire à partir du même cadre de représentation et des mêmes notations. En effet ces techniques ont suffisamment d'aspects communs et ne diffèrent, principalement, que sur les moyens de prise en compte et de traitement de l'incertitude. Tous ces outils partagent une base mathématique ensembliste commune qu'il est utile de préciser.

## Notations

Nous noterons  $\Omega$  le domaine de l'information considéré. Cela peut être l'ensemble des valeurs possibles d'une variable, l'ensemble des réalisations possibles d'un événement... Cet ensemble est appelé « univers des possibles » ou « référentiel ». On note  $P(\Omega)$  l'ensemble des parties de  $\Omega$  ou encore  $2^\Omega$ . Nous ne considérons, dans le cadre de ces travaux de thèse, que le cas discret. Nous avons donc pour un domaine de taille  $n$  :

$$\Omega = \{\omega_1, \omega_2, \dots, \omega_n\} = \bigcup \omega_i$$

Les  $w_i$  sont des singletons et les  $A_i$  sont des sous ensembles de  $\Omega$ . Nous notons les sous-ensembles par des lettres majuscules et les singletons par des minuscules. Nous fixons comme seule contrainte à cet ensemble discret qu'il soit exhaustif, c'est-à-dire qu'il soit composé de toutes les valeurs possibles. Cette contrainte peut être relâchée dans une certaine mesure avec le MCT (cf. partie II.2.2).

Un événement correspond à une proposition logique relative au résultat de l'expérience. Si  $A$  et  $B$  désigne deux éléments d'un événement  $W$  sous-ensemble de  $\Omega$ , alors :

- $A \cup B$  désigne la réalisation de  $A$  OU  $B$ .
- $A \cap B$  désigne la réalisation simultanée de  $A$  ET  $B$ .
- $\overline{A} = \Omega - A$  désigne le contraire de  $A$ .
- $\Omega$  représente l'événement certain.
- $\phi$ , l'ensemble vide, représente l'événement impossible (interprété différemment dans le MCT).
- Lorsque le référentiel est fini, l'ensemble d'événements  $W$  correspond à toutes les parties de  $\Omega$ .

## Positionnement des différentes approches subjectives

Les différents supports de représentation tels que les probabilités bayésiennes, les fonctions de croyance et de distributions de possibilités permettent de sélectionner des ensembles de valeurs possibles et chaque ensemble est qualifié par une mesure qui est le degré de vérité de l'affirmation que  $\omega$  est la valeur de l'information recherchée. Quelque soit la technique utilisée et en considérant des domaines de définition de l'information discret, nous pouvons nous ramener à la vue suivante : « La valuation de l'incertitude d'une information peut être exprimée par une distribution de poids de croyance sur l'ensemble des possibilités considérées ». Généralement, le poids utilisé est unitaire et une possibilité est un sous-ensemble de valeurs possibles. Les trois approches divergent sur la liberté qui est laissée quant aux choix de cet ensemble de possibilités. Les probabilités par exemple, dans le cas discret, n'autorisent à distribuer le poids (équivalent à la probabilité) que sur des valeurs précises (singletons) du domaine.

### Distribution de probabilité

La théorie la plus exploitée dans la littérature est de loin la théorie des probabilités, associée à la théorie bayésienne de la décision pour la décision dans l'incertain. L'information y est modélisée par une probabilité conditionnelle qui est révisée au fur et à mesure que de nouvelles informations sont considérées. L'avantage essentiel des méthodes probabilistes vient de ce qu'elles proposent un éventail d'outils très riche permettant aussi bien la modélisation que l'apprentissage des modèles. Elles proposent également des règles d'usage éventuellement heuristiques. Enfin, « la modélisation probabiliste, soutenue par l'interprétation objective (fréquentiste) largement répandue dans le monde de la physique et du traitement du signal, est actuellement un concept universellement partagé qui sert naturellement de base de comparaison aux autres modélisations » (Bloch et Maitre, 1994).

L'approche subjective, essentiellement exploitée par l'inférence bayésienne, est fondée sur les axiomes des probabilités. Les probabilités, si elles représentent bien l'incertain qui entache l'information (la variabilité), ne permettent pas aisément de représenter son imprécision. Elles conduisent souvent à confondre ces deux notions. Dubois et Prade notent que les probabilités subjectives, notamment l'approche bayésienne, semble confondre la représentation de l'aléatoire et la

représentation de l'incomplétude de l'information, au nom du principe de symétrie ou d'indifférence (Dubois et Prade, 2006). C'est pourquoi nous considérons que probabilités sont nativement mal adaptées à la représentation de l'incertitude liée à l'expertise.

### **Sous-ensembles flous et théorie des possibilités**

La théorie des sous-ensembles flous fournit un cadre permettant de représenter explicitement des informations imprécises, sous au moyen de fonctions d'appartenance (Zadeh, 1965). La théorie des possibilités (Zadeh, 1978) (Dubois et Prade, 1988), dérivée de la théorie des sous-ensembles flous, permet de représenter à la fois l'imprécision et l'incertitude, au moyen de distributions de possibilités, notée  $\pi$  et de deux fonctions caractérisant les événements : la possibilité, notée  $\Pi$ , et la nécessité, notée  $N$ . Ces fonctions ne souffrent pas des contraintes axiomatiques imposées aux probabilités et offrent donc une plus grande souplesse lors de la modélisation. L'inconvénient des ensembles flous est qu'ils représentent essentiellement le caractère imprécis des informations, l'incertitude étant représentée de manière implicite et accessible seulement par déduction à partir des différentes fonctions d'appartenance.

Un des intérêts de la théorie des ensembles flous et des possibilités, outre qu'elles imposent peu de contraintes au niveau de la modélisation, est qu'elles offrent une grande variété d'opérateurs de combinaison (Bloch, 2005). Parmi les principaux opérateurs de combinaison, on trouve en particulier les t-normes, les t-conormes, les moyennes, les sommes symétriques, et des opérateurs prenant en compte des mesures de conflit ou encore de fiabilité des sources. Le choix d'un opérateur peut être réalisé selon plusieurs critères, par exemple pour tenir compte du comportement de l'utilisateur. Des comportements sévères, indulgents ou prudents peuvent être traduits sous forme mathématique de conjonction, disjonction ou compromis.

C'est un cadre plus souple que le cadre probabiliste, mieux adapté à la représentation d'informations imprécises. Néanmoins, les capacités d'expression restent limitées. En effet, il est impossible de représenter une incertitude de type probabiliste.

#### **II.2.1.4. Théorie des fonctions de croyance**

La théorie des fonctions de croyance, appelée aussi théorie de l'évidence repose sur l'affectation d'une mesure subjective non-additive des chances de réalisation d'un événement. En d'autres termes, cette théorie permet d'exprimer un avis personnel basé uniquement sur la croyance ou encore l'état de connaissance d'un individu. La théorie des fonctions de croyance est très utilisée pour la fusion d'information issue de capteurs par exemple, mais elle est aussi particulièrement adaptée à l'acquisition d'information, analyses et autres connaissances fournies par un expert. Elle offre en effet une représentation des connaissances partielles, de l'ignorance totale jusqu'à la connaissance parfaite. La théorie des fonctions de croyance intègre aussi des outils permettant de combiner des informations issues de sources différentes.

Dempster, puis Shafer sont à l'origine de cette théorie. Shafer a défini les fonctions de croyances comme un cadre général de représentation des incertitudes. C'est une généralisation de la théorie des probabilités sous la forme des intervalles de probabilité que développa Dempster (Dempster, 1967). Le terme « théorie de Dempster-Shafer » (DST) est aujourd'hui souvent utilisé pour qualifier les travaux utilisant les fonctions de croyances. Or, la théorie a beaucoup évolué ces dernières années au point de distinguer deux approches dont les hypothèses de départ sont différentes, à savoir :

- Le modèle de Shafer (Shafer, 1976), la théorie des indications (Kohlas et Monney, 1995) (Hints en anglais) qui reposent sur l'acceptation d'une mesure de probabilité sur un espace sous-jacent (ensemble des hypothèses possibles). Cette probabilité n'a pas besoin d'être connue. Néanmoins, c'est une hypothèse fondamentale de ce modèle qui fut à l'origine de nombreuses critiques (Pearl, 1990).

- Le modèle des croyances transférables (MCT ou TBM pour *Transferable Belief Model*) développé par Smets est, quant à lui, une interprétation purement subjectiviste, non probabiliste, des fonctions de croyance (Smets et Kennes, 1994).

Bien que les hypothèses de départ soient fondamentalement différentes, les similitudes sont importantes entre les deux approches (conduisant parfois à des confusions). Chacune des approches possède un aspect statique en rapport avec la représentation de l'état de croyance de différentes manières et un aspect dynamique en rapport avec les différentes possibilités de fusion et de mise à jour d'un état de croyance en présence d'une nouvelle évidence. C'est essentiellement sur l'aspect dynamique que se situent les divergences (Smets, 1994).

La théorie des fonctions de croyance est un cadre riche et flexible permettant la représentation et la gestion de différentes formes d'incertitude. Elle est ainsi très bien adaptée aux problèmes dans lesquels l'information disponible est trop parcellaire pour être modélisée dans un cadre probabiliste sans hypothèses arbitraires. C'est la modélisation de l'ignorance partielle de l'information qui est apportée par la définition de la structure de croyance. Lorsqu'elle n'est définie que sur des singletons, la structure de croyance est une distribution de probabilité. Lorsque les avis exprimés sont cohérents et imprécis, l'affectation des degrés de croyance est alors effectuée sur des ensembles quelconques. C'est un formalisme plus général englobant les probabilités et les possibilités. Une contrepartie de cette richesse est le risque d'explosion combinatoire lié au raisonnement sur l'ensemble des parties lorsque la taille de  $\Omega$  est grande. Il faut alors rechercher des cadres de discernement de taille raisonnable. En pratique, beaucoup de parties possèdent une masse nulle et les calculs peuvent être exprimés sous forme matricielle.

Cette théorie peut sembler pauvre du point de vue de la fusion par rapport aux approches telles que la théorie des possibilités. En effet, il n'y a que deux opérateurs principaux, la règle de Dempster et la somme disjonctive (cf. II.2.2.2). De plus, la règle de Dempster peut avoir un comportement indésirable au voisinage du conflit total entre les sources (Dubois et Prade, 1988). Ces opérateurs imposent aussi que les sources combinées soient issues de sources d'information distinctes (au sens de l'indépendance de l'information). Or c'est une condition difficile à vérifier, par exemple deux opinions d'experts différents se basant sur une expérience commune ne sont pas des sources distinctes. Récemment, Denœux a proposé des opérateurs de combinaisons qui traitent ce cas (Denœux, 2008). Cependant, nous n'utilisons pas de combinaison d'informations en provenance de sources multiples dans le cadre de cette thèse. Il s'agit néanmoins d'une caractéristique importante car elle nous assure de pouvoir considérer (et fusionner), à terme, plusieurs avis d'experts (par exemple, la même expérience définie par des experts différents).

Dans la partie suivante, nous décrivons plus en détail le Modèle des Croyances Transférables (MCT) qui servira de base à nos travaux. En effet, le MCT semble bien adapté à la représentation de l'expertise que nous associons aux expériences et n'utilise pas la base probabiliste (comme la DST). En contrepartie, le MCT introduit un niveau de décision dit pignistique qui transforme à un instant donné l'ensemble des croyances (non-probabilistes) en une vue compatible avec les probabilités. Le MCT distingue le niveau crédal (croyance) du niveau pignistique (décisionnel) ce qui nous semble très intéressant dans le cadre du retour d'expérience (cf. Figure II-3).

## II.2.2. Le modèle des croyances transférables

Cette présentation s'appuie sur plusieurs travaux synthèses sur les fonctions de croyance et sur le modèle des croyances transférables dont (Lefevre, 2001), (Vannoorenberghe, 2003), (Bloch, 2005) et (Ramasso et al., 2008).

Le modèle des croyances transférables (MCT) interprète une fonction de croyance comme la représentation de l'opinion d'un agent rationnel relativement à une certaine question, sans référence à une probabilité ou une famille de probabilités sous-jacentes. Ce cadre repose sur l'hypothèse que

raisonnement dans l'incertain (niveau crédal) et décision (niveau pignistique) sont deux tâches cognitives de nature différentes comme cela est représenté sur la Figure II-3 où nous avons :

- le niveau crédal qui correspond à la représentation et à la manipulation des au moyen des différentes fonctions de croyance,
- le niveau décisionnel qui intervient au moment de la prise de décision.

Le passage du niveau crédal au niveau décisionnel n'est pas réversible et le résultat ne se distingue pas du modèle bayésien. En effet, afin de pouvoir s'appuyer sur le principe de maximisation de l'utilité espérée, la mesure du niveau crédal purement subjective est transformée en une mesure de probabilité. Notre préoccupation ne se situe pas vraiment au niveau de la prise de décision mais davantage sur les outils de représentation et de manipulation de l'information (combinaison), c'est pourquoi nous détaillons davantage le niveau crédal (parties II.2.2.1 et II.2.2.2). Le niveau décisionnel est rapidement décrit dans la partie II.2.2.3.

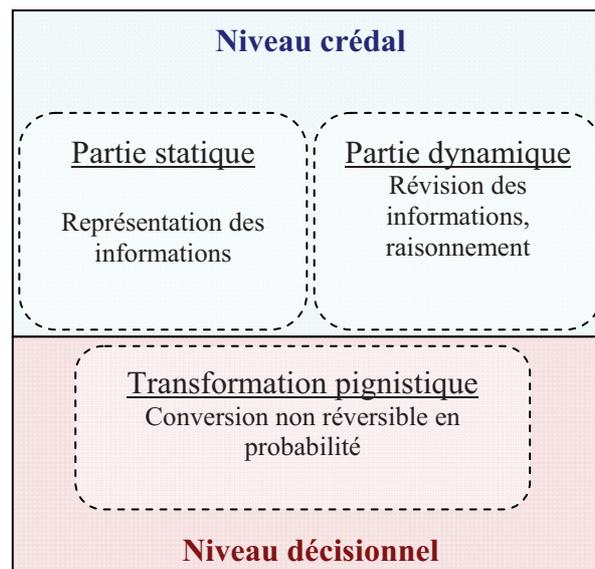


Figure II-3 - les 2 niveaux du MCT (Denœux, 2004)

### II.2.2.1. Représentation de l'information - Aspect statique

L'aspect statique des fonctions de croyance concerne la représentation de l'information incertaine. L'aspect dynamique est important dans une perspective de fusion de données et donc, dans notre domaine d'application, des avis experts. Néanmoins l'aspect statique est le support de ces traitements et nous insistons particulièrement sur cette dimension car c'est celle que nous intégrons en priorité dans le formalisme hybride de représentation des expériences que nous proposons.

#### Distribution de masse de croyance élémentaire (bba)

Dans le cadre de ces travaux de thèse, nous avons essentiellement utilisé les distributions de masse de croyance qui sont à la base des fonctions de croyance. Nous décrivons précisément cet aspect et les spécificités liées au MCT.

#### Cadre de discernement

Traiter un problème à l'aide des fonctions de croyance revient à trouver la valeur d'une variable  $\omega$ . Cette variable possède un référentiel de définition, l'univers des possibles nommé ici « cadre de discernement » (*frame of discernment*). Il correspond à l'ensemble des hypothèses du problème traité, soit l'ensemble des valeurs que peut prendre  $\omega$  et sera noté  $\Omega$ .

$$\Omega = \{\omega_1, \omega_2, \dots, \omega_n\} = \bigcup \omega_i$$

Pour les fonctions de croyance et aussi le MCT, les valeurs  $\omega_i$  du cadre de discernement doivent être mutuellement exclusives (2 valeurs ne peuvent être vrai en même temps). Par contre, contrairement aux la théorie de Dempster-Shafer l'ensemble  $\Omega$  doit être exhaustif.

### Fonction de masse

Une fonction de masse ou structure de croyance (*basic belief assignment ou bba*) est la distribution d'une masse unitaire de croyance parmi tous les sous-ensembles possibles d'états de  $\omega$ , c'est-à-dire parmi les parties de  $\Omega$  notée  $P(\Omega)$  ou  $2^\Omega$  (*power set* en anglais). Nous préférons la notation  $2^\Omega$  plus expressive car elle reflète la dimension de ce sous-ensemble (le nombre de tous les sous-ensembles possibles d'un ensemble de taille  $n$  est de  $2^n$ ).  $A$  étant un sous-ensemble de  $\Omega$

$$2^\Omega = \{A \subseteq \Omega\} = \{\emptyset, \{\omega_1\}, \dots, \{\omega_n\}, \{\omega_1 \cup \omega_2\}, \dots, \{\Omega\}\}$$

La structure de croyance traduit un état de connaissance partiel sur la valeur d'une variable  $\omega$  à valeurs dans  $\Omega$ .  $m_s^\Omega(A)$  correspond à la « part » de croyance allouée par une source  $S$  à l'hypothèse «  $\omega \in A$  » et à aucune hypothèse plus restrictive, étant donnée une base de connaissance  $B$ . La notation complète est la suivante :  $m_s^\Omega\{\omega\}[B]$ .

$$m^\Omega = \begin{cases} 2^\Omega \rightarrow [0, 1] \\ \sum_{A \in 2^\Omega} m^\Omega(A) = 1 \end{cases}$$

La possibilité d'affecter des masses aux hypothèses composées, et donc de travailler sur  $2^\Omega$  plutôt que sur  $\Omega$  constitue un des avantages de cette théorie. Elle permet en effet une modélisation très souple et très riche, en particulier de l'ambiguïté, de l'incomplétude, de l'imprécision et de l'incertitude, en analysant les capacités de chaque source à donner une information sur chaque décision possible  $m(\{1; 2; 3\}) = 1$  signifie que l'on est sûr que  $\omega$  vaut 1 ou 2 ou 3).

### Éléments focaux et Noyaux de la structure de croyance

Étant donnée une fonction de masse  $m^\Omega$ , les éléments de  $2^\Omega$  dont la masse est non nulle sont appelés « éléments focaux » de  $m^\Omega$  et constituent le noyau  $N_m^\Omega$  de la structure de croyance :

$$N_m^\Omega = \left\{ A \in \frac{2^\Omega}{m^\Omega(A)} > 0 \right\}$$

Lorsque la masse n'est définie que sur des singletons  $N_m^\Omega \subseteq \Omega$ , la structure de croyance est une distribution de probabilité (distribution de masse bayésienne).

### Hypothèse de monde fermé ou de monde ouvert

Initialement, dans la théorie développée par Shafer, le cadre de discernement représentait l'ensemble des hypothèses de façon exhaustives et exclusives (ensemble complet de solutions disjointes). Smets plus tard a relâché cette contrainte en interprétant  $\emptyset$  comme une classe de rejet, un ensemble d'hypothèses omises. C'est pourquoi sont distinguées aujourd'hui les deux hypothèses suivantes :

- L'hypothèse du « monde fermé » suppose que le cadre de discernement recense toutes les réponses possibles de la question considérée (hypothèse d'exhaustivité), ce qui signifie que  $\Omega$  désigne l'événement certain et  $\emptyset$  l'événement impossible. Nous avons  $m^\Omega(\emptyset) = 0$ .
- L'hypothèse du « monde ouvert » correspond à un cadre non exhaustif, c'est-à-dire qu'il peut exister certaines possibilités non recensées initialement. C'est pourquoi, nous avons  $m^\Omega(\emptyset) \geq 0$ . L'ensemble vide  $\emptyset$  peut alors avoir deux interprétations.

- Soit c'est une classe de rejet, c'est-à-dire qu'il correspond à toutes les hypothèses omises non représentées dans le cadre de discernement,
- soit il représente une information « impossible » qui traduit par exemple le conflit entre plusieurs sources d'informations ( $m^\Omega(\phi)$  est alors la mesure de ce conflit).

### Interprétation particulières des fonctions de masse de croyance

Les fonctions de masse de croyance peuvent prendre plusieurs formes. Voici quelques interprétations particulières :

- $m^\Omega(\Omega) = 1$  est une **fonction de masse vide** (*vacuous function* en anglais) et représente l'ignorance totale.
- $m^\Omega(\{\omega\}) = 1$  pour un  $\omega \in \Omega$  est une **fonction de masse certaine**.
- $m^\Omega(A) = 1$  pour un  $A \subseteq \Omega$ ,  $|A| > 1$  est une **fonction de masse catégorique**. Elle représente une connaissance imprécise et certaine.
- $m^\Omega(A) = 0$  pour tout  $A \subseteq \Omega$ ,  $|A| > 1$  est une **fonction de masse bayésienne** (éléments focaux singletons). Elle représente une incertitude de type probabiliste.
- Une fonction de masse  $m$  ayant des éléments focaux emboîtés (c'est-à-dire tels que  $A_1 \subseteq A_2 \subseteq \dots \subseteq A_n$ ) est une **fonction de masse consonante**. Elle représente une incertitude « possibiliste ».
- Une **fonction de masse normalisée** est une fonction de masse telle que  $m^\Omega(\phi) = 0$ . La normalisation est l'opération qui consiste à annuler et répartir ce poids entre les autres éléments focaux de  $\Omega$ .
- Une **fonction de masse non-dogmatique** est une fonction de masse telle que  $m^\Omega(\Omega) > 0$ .
- Une **fonction de masse simple** est de la forme  $m^\Omega(A) = 1 - \lambda$  et  $m^\Omega(\Omega) = \lambda$  avec  $\omega \in [0, 1]$  et  $A \subseteq \Omega$ . Elle est notée  $m = A^\lambda$ .  $A^1$  est la fonction de masse vide et  $A^0$  la fonction de masse catégorique.
- La **fonction de masse simple généralisée**  $\mu$  de  $2^\Omega \rightarrow \mathbb{R}$  est de la forme  $\mu(A) = 1 - \lambda$ ,  $\mu(\Omega) = \lambda$  et  $\mu(B) = 0$  avec  $\lambda \in [0, 1]$ ,  $A \subseteq \Omega$  et  $B \in 2^\Omega \setminus \{A, \Omega\}$ . Elle est notée  $\mu = A^\lambda$ . Si  $\lambda \leq 1$  alors  $\mu$  est une fonction de masse simple et si  $\lambda > 1$ , alors  $\mu$  n'est pas une fonction de masse, c'est une fonction de masse simple inverse.

### Affaiblissement d'une fonction de masse

Une opération fréquente sur les fonctions de masse consiste à leur attribuer un degré de confiance, pour tenir compte, par exemple, d'un niveau estimé de fiabilité des informations fournies par un expert. Cette opération appelée affaiblissement consiste à pondérer la masse des éléments focaux équivalents aux informations fournies par l'expert par  $1 - \alpha$  où  $\alpha$  est le facteur d'affaiblissement ( $\alpha$ ) et à transférer la masse résiduelle sur le cadre de discernement tout entier. Le coefficient  $\alpha$  représente une sorte de « métaconnaissance » concernant la fiabilité de la source. Ainsi, une fonction de masse affaiblie, notée  $m_\alpha$ , peut se déduire de  $m$  par :

$$\forall A \subset \Omega \quad m_\alpha(A) = \alpha \cdot m(A) \text{ et } m_\alpha(\Omega) = 1 - \alpha \cdot m(\Omega)$$

### Représentation matricielle d'une fonction de masse

Etant donné un ordre sur  $2^\Omega$ , une fonction de masse  $m$  peut être représentée par un vecteur

$$\vec{m} = (m^\Omega(A_1), \dots, m^\Omega(A_n))^t \in \mathbb{R}^n \text{ avec } n = 2^{|\Omega|}$$

L'ordre « naturel » souvent utilisé est l'ordre binaire. Dans le Tableau 2, nous donnons une illustration de cette notation avec une distribution de masse quelconque.

$A$	c	b	a	Valeur équivalente de A	$m(\cdot)$
$\phi$	0	0	0	0	0.1
$\{a\}$	0	0	1	1	0
$\{b\}$	0	1	0	2	0.3
$\{a, b\}$	0	1	1	3	0
$\{c\}$	1	0	0	4	0.1
$\{a, c\}$	1	0	1	5	0.2
$\{b, c\}$	1	1	0	6	0
$\Omega = \{a, b, c\}$	1	1	1	7	0.3

$\sum m^\alpha(A) = 1$

Tableau II-1 - Représentation matricielle d'une distribution de masse de croyance

Notons qu'il n'y a pas de techniques générales permettant l'affectation a priori de la masse de croyance aux éléments focaux, néanmoins quelques orientations sont recensées dans (Bloch, 2005).

### Fonctions associées

A partir d'une distribution de masse qui représente une croyance stricte que l'on ne peut préciser, il est possible d'obtenir de manière biunivoque des représentations équivalentes mais différentes sémantiquement. Les plus importantes sont les fonctions duales de crédibilité notée  $Cr$  et de plausibilité notée  $Pl$ . La crédibilité mesure la force avec laquelle on croit en la véracité de la proposition (croyance minimum). La plausibilité, mesure l'intensité avec laquelle on ne doute pas de  $A$  (croyance maximum). La plausibilité correspond au degré de crédibilité qui n'est pas attribué à  $\bar{A}$ ,  $\bar{A}$  étant l'événement contraire de la proposition  $A$ . C'est pourquoi nous avons :

$$\forall A \subseteq \Omega, \quad Pl(A) = Cr(\Omega) - Cr(\bar{A}) = 1 - m(\phi) - Cr(\bar{A})$$

Dans l'approche de Shafer,  $Pl$  et  $Cr$  sont respectivement équivalentes à des bornes supérieures et inférieures de probabilité alors que pour Smets, il n'y a aucun lien avec les probabilités. C'est pourquoi, au niveau décisionnel, il a introduit la « probabilité pignistique », grandeur qui consiste à transcrire de façon non réversible la structure de croyance dans une forme compatible avec les probabilités.

### Fonction de crédibilité

La crédibilité d'un ensemble d'éléments  $A$ , dans un cadre de discernement  $\Omega$ , est définie comme la crédibilité totale qui va précisément dans le sens de  $A$  ou de l'ensemble des croyances qui prouvent ou supportent cette proposition, autrement dit la vraisemblance minimale (le terme support est préféré à crédibilité dans la théorie des indications).  $Cr$  correspond au degré de croyance minimum en l'hypothèse  $\omega \in A$  compte tenu des masses de croyance affectées à toutes les hypothèses qui impliquent  $\omega \in A$  (si  $B \subset A$  alors  $B$  implique  $A$ ). Numériquement, il s'agit de la somme de toutes les masses attribuées aux éléments contenus dans  $A$ , plus la masse attribuée à  $A$  elle-même. La formule donnant le calcul est :

$$\forall A \subseteq \Omega, \quad Cr(A) = \sum_{\phi \neq B \subseteq A} m(B)$$

L'interprétation ensembliste de cette mesure est représentée sur la Figure II-4 où nous observons que la crédibilité de  $A$  correspond à la somme des poids de croyance de ses sous-ensembles et de lui-même.

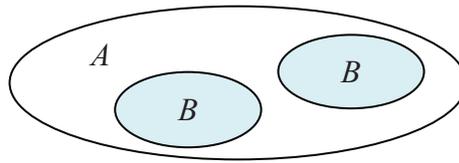


Figure II-4 - Interprétation ensembliste de la fonction de crédibilité

$m(\phi)$  n'est pas pris en compte dans le calcul du degré de croyance ( $\phi$  ne peut impliquer  $A$ ), c'est pourquoi la fonction de crédibilité est identique selon les deux approches DST et MCT

Concernant ces propriétés  $Cr$  est une mesure sur-additive et monotone :

$$\forall A_1, \dots, A_n \subseteq \Omega \quad Cr\left(\bigcup_{i=1}^n A_i\right) \geq \sum_{I \in \{1, \dots, n\}, I \neq \phi} -1^{|I|+1} \cdot Cr\left(\bigcap_{i \in I} A_i\right)$$

$$\forall A, B \in (2^\Omega)^2 \quad Cr(A) + Cr(B) \leq Cr(A \cup B) + Cr(A \cap B)$$

$$\forall A, B \in (2^\Omega)^2 \quad A \subseteq B \Rightarrow Cr(A) \leq Cr(B)$$

$$Cr(\phi) = 0 \quad \text{et} \quad Cr(\Omega) = 1 - m(\phi)$$

### Fonction de Plausibilité

La plausibilité d'un ensemble d'éléments  $A$ , dans un cadre  $\Omega$ , représente le degré maximal de croyance que l'on peut avoir dans l'hypothèse  $\omega \in A$  (conditionnellement à l'obtention de nouvelles informations). C'est la crédibilité maximale possible qui pourrait être attribuée à  $A$  si toute l'évidence future supportait  $A$ . En d'autres termes, il s'agit de toute l'évidence qui ne contredit pas  $A$ , c'est-à-dire la vraisemblance maximale et donc, numériquement, il s'agit de la somme de toutes les masses attribuées aux éléments intersectant  $A$ , plus la masse attribuée à  $A$  elle-même.

$$\forall A \subseteq \Omega, \quad Pl(A) = \sum_{B \cap A \neq \phi} m(B)$$

L'interprétation ensembliste de cette mesure est représentée sur la Figure II-5 où nous observons que la plausibilité d'un ensemble  $A$  correspond à la somme des poids de croyance des ensembles pour lesquels l'intersection avec  $A$  est non nulle (dont  $A$  lui-même).

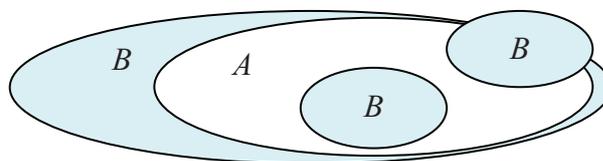


Figure II-5 - Interprétation ensembliste de la fonction de plausibilité

Ici aussi,  $m(\phi)$  n'est pas pris en compte dans le calcul du degré de croyance. En outre,  $\forall A \subseteq \Omega$  nous avons :

$$Cr(A) \leq Pl(A)$$

Cette fonction a une interprétation naturelle dans le modèle des croyances transférables où l'on considère que l'apport d'information peut permettre de transférer des croyances sur des sous-ensembles plus précis. La plausibilité représente alors la croyance maximale que l'on pourrait potentiellement affecter à un sous-ensemble  $A$  si l'on apprend par exemple que la solution se trouve dans  $A$  (toute la confiance mise dans un sous-ensemble  $B$  (intersectant  $A$ ) est alors transférée sur  $A$  afin d'affecter 0 à la confiance sur  $\bar{A}$ ).

Concernant ces propriétés  $Pl$  est une mesure sous-additive et monotone :

$$\forall A_1, \dots, A_n \subseteq \Omega \quad Pl\left(\bigcup_{i=1}^n A_i\right) \leq \sum_{I \in \{1, \dots, n\}, I \neq \emptyset} -1^{|I|+1} \cdot Pl\left(\bigcap_{i \in I} A_i\right)$$

$$\forall A, B \in (2^\Omega)^2 \quad Pl(A) + Pl(B) \geq Pl(A \cup B) + Pl(A \cap B)$$

$$\forall A, B \in (2^\Omega)^2 \quad A \subseteq B \Rightarrow Pl(A) \leq Pl(B)$$

$$Pl(\emptyset) = 0 \quad \text{et} \quad Pl(\Omega) = 1 - m(\emptyset)$$

### Mesure de degré d'incertitude

Afin de comparer plusieurs fonctions de croyance, il est nécessaire de déterminer le contenu informationnel ou degré d'incertitude d'une fonction de croyance. Pour cela, il existe deux approches :

- L'approche ordinale : il s'agit de la définition d'un ordre partiel sur l'ensemble des fonctions de croyance («  $m_1$  est plus informative (riche) que  $m_2$  »). Une telle relation doit logiquement correspondre à l'inclusion lorsque  $m_1$  et  $m_2$  sont catégoriques.
- L'approche quantitative : il s'agit de la définition de mesures d'incertitude, par exemple la non-spécificité (la spécificité correspond à la mesure du degré de précision de l'ensemble  $A$ ).

$$N(m) = \sum_{\emptyset \neq A \subseteq \Omega} m(A) \cdot \log_2(|A|)$$

### Principe d'engagement minimal (minimal commitment)

Selon ce principe, lorsque plusieurs fonctions de croyance sont compatibles avec un ensemble de contraintes, la moins informative doit être choisie. Le principe d'engagement minimal (ou information minimale) joue le même rôle que le principe de maximum d'entropie en probabilités. Il consiste à choisir la fonction de croyance la moins informative (lorsqu'elle existe) parmi l'ensemble des fonctions de croyance compatibles avec les informations disponibles. Les applications du principe d'engagement minimal sont notamment l'extension vide (opération inverse de la marginalisation), et le déconditionnement (opération inverse du conditionnement).

### II.2.2.2. Révision de l'information - Aspect dynamique

Les structures de croyance permettent la modélisation de sources d'information et la représentation explicite de l'ignorance. La révision est le traitement qui permet de mettre à jour l'information initiale correspondant à un état de croyance à un instant donné avec un nouvel élément de croyance. L'information est alors précisée (révision conjonctive) ou complétée (révision disjonctive).

Des opérateurs de combinaison permettent d'exploiter la redondance et les recouvrements de plusieurs sources d'information et donc d'accéder à une information plus précise et plus pertinente, en construisant une structure de croyance unique à partir de plusieurs structures définies sur le même univers et issues de sources d'information distinctes. Dans le cas général, les opérations de combinaison ont une complexité de traitement exponentielle. En pratique, il est rare que tous les sous-ensembles soient pris en compte, et la complexité de traitement reste souvent plus raisonnable.

L'aspect dynamique correspond aux différents traitements qu'il est possible de réaliser à partir de plusieurs fonctions de croyances exprimées, soit sur le même cadre de discernement, soit sur des cadres de discernement différents mais compatibles, soit sur des cadres totalement différents. Pour pouvoir combiner des fonctions de croyance, il faut que leurs cadres de discernement soient identiques (ou compatibles).

La suite de ce chapitre décrit les combinaisons les plus classiques. Nous dissociions les comportements de spécialisation (relatif à la conjonction) et ceux d'extension (relatif à la disjonction).

### Attitude conjonctive vs. Attitude disjonctive

Une décision fondée sur un grand nombre d'informations d'origines et de natures variées est généralement plus fiable et plus précise qu'une décision ne dépendant que d'un seul type ou d'une seule source d'information. Ces conséquences positives proviennent de :

- la redondance d'information obtenue lorsque les différentes sources exploitent les mêmes paramètres ; la fusion de ces sources permet de diminuer l'incertitude globale des informations fournies relativement à ces paramètres et elle offre également une plus grande robustesse de l'information en permettant éventuellement de pallier la défaillance de l'une des sources,
- la complémentarité, obtenue lorsque ces sources exploitent des paramètres différents, permet alors de déduire une information globale plus complète concernant certains aspects du problème qu'une source serait incapable de saisir en opérant individuellement.

Lors de la combinaison de sources distinctes d'information, nous pouvons adopter une attitude conjonctive (intersection des sources) ou disjonctive (union des sources).

- Conjonctive : la plus informative, sensible au conflit. Une attitude conjonctive correspond à un renforcement de l'information où les sources combinées sont considérées comme fiables. Elle renforce donc la focalisation, et diminue la longueur des intervalles de confiance  $[Cr, pl]$ . Deux sources trop conflictuelles ne peuvent être combinées.
- Disjonctive : insensible au conflit mais entraînant des pertes de précision. Une attitude disjonctive correspond à une attitude prudente où au moins une des sources est considérée comme fiable. L'information combinée perd donc en précision. Les règles de combinaison disjonctive sont rarement utilisées car la fusion, domaine de prédilection de la combinaison d'information, a généralement pour objectif de rendre l'information plus précise. Néanmoins, elles sont assez utiles pour réduire le temps de calcul et la mémoire nécessaire dans les réseaux de croyance (Xu et Smets, 1994).

### Définition du conflit

Considérons  $A_i$  et  $B_j$  deux éléments focaux avec la particularité  $A_i \cap B_j = \phi$ , c'est-à-dire que leur réalisation simultanée est impossible. Nous avons  $m_1(A_i) > 0$  et  $m_2(B_j) > 0$ , c'est pourquoi la valeur combinée  $m_{12}(A_i \cap B_j = C)$  est non nulle (quelque soit l'opérateur de combinaison, sauf précision particulière). Cette valeur est souvent interprétée comme le conflit entre les sources et constitue une information importante. Cette mesure de conflit n'est pas une mesure absolue. Elle dépend de la modélisation effectuée, en particulier, de la répartition des masses sur les différents sous-ensembles. Il est important de prendre en compte la valeur du conflit pour juger de la qualité de la combinaison : celle-ci peut ne pas avoir beaucoup de sens en cas de fort conflit et conduire à des décisions critiquables.

Le conflit correspond donc à l'information relayée par l'intersection des éléments focaux nuls entre les deux sources (plus il y a de sous-ensembles disjoints entre les fonctions de crédibilité, plus la quantité de masse à écarter sera grande). C'est pourquoi  $m^\Omega(\phi)$  est associée à la mesure du conflit que nous notons  $K$ .

$$K = m(\phi) = (m_1 \cap m_2)(\phi) = \sum_{B \cap C = \phi} m^{S_1}(B) \cdot m^{S_2}(C)$$

- $K = 1$  signifie que les deux sources d'information sont totalement conflictuelles. Dans ce cas, leur somme orthogonale est impossible. Il est alors nécessaire de revoir le problème.
- $K = 0$  signifie que les sources sont totalement concordantes.

## Révision conjonctive

La combinaison conjonctive, la plus utilisée, est déclinée en plusieurs méthodes. Elle a pour particularité de quantifier une mesure du conflit, que nous notons  $K$ , entre les sources et c'est justement cette grandeur et la façon dont elle est traitée qui différencie les méthodes permettant de calculer des combinaisons conjonctives.

### Combinaison conjonctive directe (Combinaison de Smets)

Soient  $S_1$  et  $S_2$  deux sources d'information indépendantes supposées totalement fiables. Les informations obtenues via ces sources sont modélisées respectivement à l'aide des fonctions de croyance  $m_1$  et  $m_2$ . La fonction de croyance résultat de la combinaison conjonctive de  $m_1$  et  $m_2$  est notée  $m_{\cap}$ .

$$m_{\cap} = m_1 \cap m_2$$

La combinaison conjonctive de ces deux fonctions est définie par :

$$\forall A \subseteq \Omega \quad m_{\cap}(A) = \sum_{B \cap C = A} m_1(B) \cdot m_2(C)$$

La règle de combinaison conjonctive précédente vérifie les propriétés suivantes. Elle est associative, commutative, possède un élément neutre ( $m(\Omega) = 1$ ) et est non idempotente. Par associativité, cette règle peut être généralisée à  $n$  sources d'information de distributions de masse élémentaire  $m_j$ , telles que  $j \in [0, n]$ . Nous avons alors :

$$m_{\cap} = \bigcap_{j=1}^n m_j$$

$$\forall A \subseteq \Omega \quad m_{\cap}(A) = \sum_{A_1 \cap \dots \cap A_n = A} \left( \prod_{j=1}^n m_j(A_j) \right)$$

En général, cette règle de combinaison produit une fonction de croyance non normalisée, c'est-à-dire que  $m_{\cap}(\phi) \neq 0$  traduit la présence de conflit. Smets (Smets, 1990) suppose des sources fiables et interprète le conflit comme provenant d'une mauvaise définition du cadre de discernement  $\Omega$  et rapporte donc cette masse sur  $\phi$ . La combinaison précédente est donc la combinaison disjonctive utilisée par Smets.

$$\forall C \in 2^{\Omega} \quad m_{\cap}^{\Omega}(C) = (m_1 \cap m_2)(C) = \sum_{(A, B) \in (2^{\Omega})^2, A \cap B = C} m_1^{\Omega}(A) \cdot m_2^{\Omega}(B)$$

$$m(\phi) = K$$

### Combinaison de Dempster ou somme orthogonale

L'opérateur de Dempster est le premier opérateur de combinaison défini dans le cadre de la théorie de l'évidence. Il s'agit de la même combinaison que la précédente, à ceci près qu'il faut renormaliser la nouvelle fonction obtenue puisque, selon Dempster,  $\Omega$  est considéré comme exhaustif selon l'hypothèse du monde fermé, ce qui impose  $m(\phi) = 0$ .

La solution pour remédier à ce problème consiste à écarter tous les produits  $m_1(A_i) \cdot m_2(B_j)$  tels que  $A_i \cap B_j = \phi$  étant donné que  $A_i$  et  $B_j$  sont contradictoires. Par conséquent, les produits restant doivent être normalisés car ils représentent une nouvelle fonction de masse et leur somme doit être égale à un. La normalisation s'obtient en multipliant chaque produit par la constante  $\kappa$  appelée facteur d'ajustement et donné par la formule suivante :

$$\kappa = \frac{1}{1-K}$$

Et la combinaison notée à l'aide de l'opérateur  $\oplus$ , s'exprime de la façon suivante :

$$(m_1 \oplus m_2)(A) = \frac{1}{1-K} \sum_{B \cap C = A} m^{S_1}(B) \cdot m^{S_2}(C) = \frac{m_{\cap}(A)}{1-K}$$

$$(m_1 \oplus m_2)(\phi) = 0$$

L'autre nom de la combinaison de Dempster est la somme orthogonale. Cette combinaison vérifie les propriétés de commutativité et d'associativité permettant des combinaisons en « cascade ».

- $m(\Omega) = 1$  est l'élément neutre de la combinaison, l'ignorance totale ne modifie pas la compréhension du système.
- $m(A) = 1$  (un seul singleton comme élément focal) est l'élément absorbant de la combinaison, une certitude est définitivement acquise ( $A$  étant un sous-ensemble de  $\Omega$ ).

Hors du cadre du MCT, cette méthode est difficile à justifier théoriquement suivant l'approche de Shafer. En effet, Dempster propose d'annuler les masses correspondantes aux intersections vides puis de renormaliser la structure obtenue en divisant par le conflit afin de ne pas contredire les hypothèses de départ. L'inconvénient principal de la combinaison de Dempster est qu'elle masque le conflit existant entre les deux structures de croyances représentant ainsi une perte d'information qui peut induire en erreur par renforcement des propositions peu crédibles mais communes dans chacune des structures de croyance combinées (Zadeh, 1986).

## Révision disjonctive

Les révisions disjonctives sont plus rarement utilisées car le plus souvent les informations à fusionner représentent le même phénomène et elles sont considérées comme provenant de sources fiables. Par contre, si nous considérons lors de la combinaison qu'au moins une des sources n'est pas fiable, alors il est plus prudent de conserver toute l'information en utilisant l'opérateur ensembliste d'union au lieu de celui d'intersection. Une fusion disjonctive est ainsi obtenue.

$$\forall C \in 2^\Omega \quad m_{\cap}^\Omega(C) = (m_1 \cup m_2)(C) = \sum_{(A, B) \in (2^\Omega)^2, A \cup B = C} m_1^\Omega(A) \cdot m_2^\Omega(B)$$

Cette fonction de combinaison est commutative et associative, elle possède un élément neutre :  $(m(\phi) = 1)$  et a pour moyenne :

$$\text{moyenne} = \frac{1}{n} \sum_{i=1}^n m_i$$

Cette combinaison ne peut pas faire apparaître de conflit. Elle élargit les éléments focaux et fournit donc une information moins précise que chacune des sources. Ce mode de fusion peut être intéressant quand les fiabilités des sources, leurs ambiguïtés et imprécisions ne sont pas connues a priori. Par exemple, si une source est focalisée sur  $A$  et une autre sur  $B$  avec  $A \cap B = \phi$ , une manière de ne pas lever le conflit est de conclure que la vérité est dans  $A \cup B$ , ce que permet la fusion disjonctive.

Toutefois, dans la plupart des applications de fusion d'informations, une fonction de masse combinée plus focalisée que les fonctions de masse initiales est recherchée. La fusion conjonctive est, en général, préférée ce qui implique de prendre en compte les imprécisions, fiabilités et ambiguïtés de chaque source à l'étape de modélisation.

## Combinaisons mixtes

Il existe aussi des combinaisons qui combinent les deux approches, à savoir la révision conjonctive et disjonctive.

### Combinaison adaptative de Dubois et Prade

Dubois et Prade (Dubois et Prade, 1998) ont proposé une combinaison alternative. Ils proposent de combiner conjonctivement les sources qui ne sont pas en conflit (intersection non vide des éléments focaux) et disjonctivement celles qui le sont (la masse de conflit est affectée à l'union des propositions à l'origine du conflit).

$$m^\Omega(C) = \sum_{A \cap B = C} m_1^\Omega(A) \cdot m_2^\Omega(B) + \sum_{\substack{A \cup B = C \\ A \cap B = \emptyset}} m_1^\Omega(A) \cdot m_2^\Omega(B)$$

Cette combinaison interprète le conflit comme une erreur d'une seule des sources d'information et conclue à l'union des deux propositions. Elle n'est pas associative.

### Transfert de masse

Dans le modèle des croyances transférables, la fonction de masse a une interprétation naturelle puisque la masse  $m^\Omega(A)$  représente le degré de croyance attribué à la proposition  $A \subseteq \Omega$  qui n'a pas pu être assignée à un sous-ensemble plus spécifique. Si maintenant, on apprend que la croyance allouée au sous-ensemble  $B \subseteq \Omega$  est totale alors la masse  $m^\Omega(A)$  initialement allouée à  $A$  est transférée au sous-ensemble  $A \cap B$  d'où le nom du modèle introduit par Smets et ce transfert est réalisé par l'opération de conditionnement. Cette opération correspond au conditionnement probabiliste après renormalisation. Soit une distribution de masse  $m^\Omega$  et  $W \subseteq \Omega$ , alors :

$$m^\Omega[W] = \begin{cases} \sum_{B \subseteq W} m^\Omega(A \cup B), & \forall A \subseteq W \\ 0 & \text{autrement} \end{cases}$$

$m^\Omega[W]$  est appelé la distribution de masse de croyance conditionnelle sachant  $W$ . Ce résultat contrairement à la règle de conditionnement introduite par Shafer n'est pas normalisée. La règle de conditionnement de Dempster est un cas particulier de la somme conjonctive.

### II.2.2.3. Niveau décisionnel

Smets, avec le modèle des croyances transférables, a introduit une distinction très nette entre la représentation des croyances (purement subjective) et la prise de décision. Cette dernière dimension doit en effet être compatible avec le formalisme des probabilités afin de garantir la cohérence des décisions (Smets, 2005). En effet, « un comportement rationnel face au risque suppose que les décisions soient prises sur la base d'une mesure de probabilité ». On doit avoir, avec  $f$  la fonction de décision :

$$f(A) + f(\overline{A}) = 1$$

Ainsi, il est possible d'appliquer le principe de maximum d'utilité (Smets et Kennes, 1994). Par contre, contrairement au calcul de croyance et de crédibilité, le calcul de la probabilité pignistique n'est pas réversible (il n'est pas possible de retrouver la distribution de masse). En effet, cette opération repose sur l'hypothèse suivante (analogue à l'équiprobabilité) : « Une masse affectée à une proposition doit être équitablement répartie entre les différentes hypothèses la composant ». Cette irréversibilité n'est pas gênante car la transformation n'est effectuée qu'au moment d'une prise de décision. La vue crédale continue d'exister en parallèle.

La construction de la probabilité pignistique est donnée par :

$$\forall H_i \in A \text{ alors } BetP(H_i) = \sum_{H_i \in A} \frac{m(A)}{Card(A)}$$

La prise de décision peut être faite selon plusieurs critères (cf. (Smets, 2002) pour davantage d'éléments concernant la prise de décision) :

- Maximum de probabilités pignistique,
- Maximum de crédibilité,
- Maximum de plausibilité.

#### II.2.2.4. Synthèse

Nous avons choisi d'utiliser le MCT comme base de représentation incertaine avant tout pour sa flexibilité de représentation de l'information imparfaite. Pour illustrer cet aspect, nous donnons une interprétation de l'incertitude qu'il est possible de véhiculer par l'intermédiaire des fonctions de croyance. Par exemple, la longueur de l'intervalle de confiance  $[Cr^\Omega(A), Pl^\Omega(A)]$  est une mesure de l'ignorance que l'on a sur un événement et son complémentaire. De même, chacun des intervalles représentés sur la Figure II-6 possède une signification traduisant un certain niveau d'incertitude.

Les aspects dynamiques des fonctions de croyance permettant entre autre la fusion et la révision d'information sont aussi très intéressants dans une perspective d'utilisation multi-experte des applications de retour d'expérience (fusion de saisie des expériences, analyse multi-experte). Néanmoins, nous ne développerons pas cet aspect que nous laissons en perspective. Nous nous consacrons à la représentation et la modélisation de l'incertitude ainsi qu'à son couplage avec le modèle objet simple de l'expérience choisi précédemment.

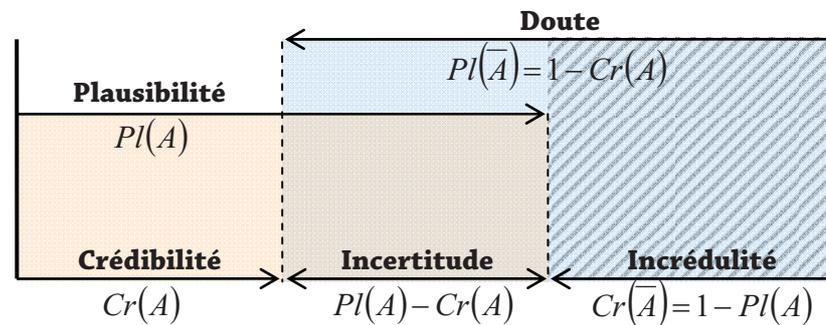


Figure II-6 - Interprétation des fonctions de croyance pour un ensemble A (Lefevre, 2001)

## II.3. Modèle hybride objet croyance de l'expérience

Dans cette partie, nous présentons la modélisation retenue pour l'expérience. Elle est exprimée dans un langage hybride qui combine la modélisation orientée objet et le modèle des croyances transférables. L'approche orientée objet sert à la représentation des entités du monde et le modèle des croyances transférables sert de support à l'expression de l'incertain et de l'imprécis. Nous détaillons dans la suite comment ces deux formalismes sont combinés pour pouvoir exprimer l'entité « expérience ». Nous montrons, dans le chapitre III, comment cette structure est exploitée dans les raisonnements et en particulier comment l'incertitude est prise en compte et propagée.

### II.3.1. Un formalisme hybride objet incertain

Nous avons choisi de combiner, dans un langage hybride, la représentation orientée objet et les fonctions de croyances pour pouvoir représenter les expériences. Nous souhaitons définir un cadre extensible basé sur des représentations orientées objet (OO) permettant d'exprimer la structure des expériences sous formes d'ensembles de couples (attribut, valeur). Les attributs peuvent être composites et contenir d'autres attributs de types différents. Les fonctions de croyance permettent de coupler la représentation de l'incertitude avec la représentation orientée objet. Nous avons choisi cette combinaison car elle nous semblait bien adaptée à notre problématique comme nous le montrons dans la suite de cette partie.

Tout d'abord, la macro-structuration de l'expérience présentée dans le chapitre I est exprimée facilement dans une organisation hiérarchique de type taxonomie nativement présente dans le formalisme objet. Nous avons en outre justifié le choix d'une approche orientée objet de représentation des connaissances, par rapport aux approches centrées objet usuelles de représentation des connaissances, principalement pour des raisons de cohérence et d'homogénéité pour l'opérationnalisation du système résultant. Les limites de l'orienté objet en termes de représentation des connaissances ne sont pas vraiment un problème dans notre cas grâce à l'approche volontairement simplifiée de représentation des connaissances que nous adoptons. Cependant, bien que n'ayons pas un besoin fondamental de justesse sémantique de la représentation des connaissances, nous avons une volonté de représenter les incertitudes présentes dans la description des situations et dans le jugement des experts. Les fonctions de croyance ont été choisies pour deux raisons principales :

- elles constituent une interface naturelle entre l'attribut et sa valeur sachant que l'attribut est une composante intrinsèque fondamentale de la notion d'objet,
- l'attribution de confiance sur des sous-ensembles du domaine de décision traduit, selon nous, très bien l'attitude des experts lorsqu'ils s'expriment.

Nous avons choisi la méthode des croyances transférables (MCT), variante de la théorie de Dempster-Shafer, d'une part, pour ses meilleures fondations théoriques puisque les critiques d'origine probabiliste adressées à la DST ne sont plus applicables (Smets, 1993) et, d'autre part, pour la possibilité d'utiliser l'hypothèse du monde ouvert (cf. partie II.2.2). Cette hypothèse permet d'attribuer une croyance sur élément inconnu, non représenté dans le cadre de discernement (autrement dit des valeurs manquantes au modèle) ce qui est généralement le cas dans les situations d'application du retour d'expérience où tous les éléments d'explication de l'apparition d'un événement ne sont pas forcément présents dans le cadre d'analyse.

#### II.3.1.1. Structuration des entités composant l'expérience

La structuration de l'expérience est modélisée selon le schéma représenté sur la Figure II-7. Il s'agit de la définition du vocabulaire utile au processus de retour d'expérience, autrement dit, l'ontologie du retour d'expérience. En parallèle, les différentes ontologies de domaine sont également représentées, c'est-à-dire le vocabulaire utilisé pour décrire les événements, les contextes, les analyses et les solutions. Ces ontologies dépendent bien évidemment du domaine d'application du retour d'expérience tandis que l'ontologie du retour d'expérience est relativement figée. Afin de préserver la généralité du retour d'expérience, nous avons choisi de développer un modèle générique, indépendant de l'ontologie du domaine.

La partie gauche de la Figure II-7 décrit l'ontologie du retour d'expérience. Il s'agit d'une taxonomie classique qui reprend les entités qui ont été isolées précédemment. Ce schéma est exprimé dans un formalisme de type UML. Nous montrons aussi les relations avec les attributs qui caractérisent l'événement et avec les attributs incertains qui caractérisent le contexte. L'ontologie de domaine n'est pas représentée sur ce schéma par contre la classe *attribut incertain* représente les points d'entrée à partir desquels les attributs du domaine seront créés par sous-classage.

Ces attributs seront nommés par la suite « descripteur simple » ou « descripteur objet » et ils sont en relation avec le domaine. Les attributs (descripteurs) sont valués au moyen de la représentation issue du modèle des croyances transférables. C'est donc l'attribut qui joue le rôle d'interface entre le langage objet et le MCT. L'analyse est une entité qui fait aussi intervenir l'incertitude dans une certaine mesure mais de manière différente à celle du contexte. L'entité analyse est une relation entre informations disponibles. Dans notre cas, il s'agit de la conjonction des attributs significatifs qui caractérisent le mieux l'occurrence de l'événement selon le point de vue d'un expert.

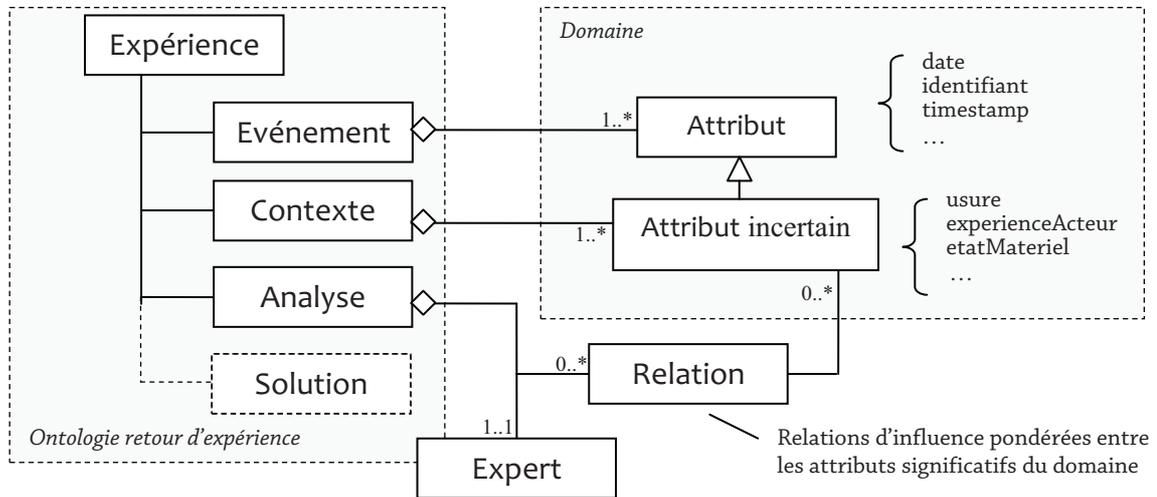


Figure II-7 - Structuration de l'expérience

L'ontologie de domaine n'est pas représentée sur la Figure II-7. Elle est construite autour de la notion d'attributs incertains que nous appelons « descripteurs ». L'ontologie de domaine est donc relativement élémentaire en termes d'expressivité. Il s'agit d'un modèle « attribut – valeur » intégrant la possibilité d'utiliser des attributs objets. Autrement dit, c'est un modèle objet simple prenant en charge la composition et l'héritage. Nous avons exploré la définition d'une ontologie plus précise du retour d'expérience qui représente plus formellement les composants du retour d'expérience et les relations possibles entre ces composants. Cette ontologie (concepts et relations entre concepts) a donné lieu à un prototype implémenté avec les graphes conceptuels (Kamsu *et al.*, 2008). Dans ce travail de thèse, l'ontologie réalisée est plus simple et plus générique. Elle ne prend pas autant d'éléments en considération, en particulier, la solution n'est pas intégrée. En contrepartie, elle permet une opérationnalisation plus directe des systèmes tout en restant flexible et évolutive du fait de sa généricité et surtout de son lien fort avec les Langages de Programmation Orientés Objet.

### II.3.1.2. Intégration de l'incertitude

Dans l'ontologie qui est présentée, il est fait référence à deux entités « incertaines », l'attribut incertain (ou descripteur) et l'analyse qui est une relation entre les descripteurs. La première intégration est assez naturelle puisqu'il s'agit de représenter des valeurs d'attributs imprécises, incertaines, voire tout simplement inconnues. Au lieu d'utiliser des valeurs précises d'attributs classiques comme cela est le cas pour l'événement, la possibilité est offerte à l'utilisateur de sélectionner parmi les valeurs possibles de l'attribut celles qui sont possibles au lieu de ne choisir qu'une valeur unique et donc précise. Pour pouvoir coupler notre représentation « objet » avec les fonctions de croyance, nous avons considéré le domaine de définition de l'attribut comme le cadre de discernement d'une fonction de croyance. La contrainte majeure induite par ce choix est que les éléments du domaine doivent être mutuellement exclusifs (deux valeurs ne peuvent être vraies simultanément). Ceci implique qu'il ne faut pas mélanger plusieurs éléments d'informations (*cf.* II.3.2.1) et donc que les choix multiples de type conjonctif sont proscrits (relation « et » entre deux valeurs d'attributs). Donner une valeur à un attribut revient alors à définir une distribution de masse sur le domaine du descripteur. Cette valuation peut cependant être réalisée de manière relativement transparente. Il est envisageable que l'utilisateur choisisse les valeurs en affectant des degrés de confiance au moyen d'une interface. Plusieurs possibilités sont alors offertes à l'utilisateur :

- Il peut choisir une valeur précise et sûre (une seule valeur avec une confiance maximale),
- Il peut exprimer l'ignorance totale (« je ne sais pas ») auquel cas l'ensemble des valeurs est sélectionné avec une confiance maximale,
- Il peut choisir plusieurs valeurs en cas de doute, toujours en associant un degré de confiance. Les valeurs choisies peuvent être disjointes. La confiance doit être répartie entre chaque groupe d'attributs,
- Enfin, il peut exprimer qu'il connaît la valeur mais que celle-ci n'est pas présente parmi les choix proposés. Cette possibilité est d'ailleurs un des apports du modèle des croyances transférables. Elle consiste à affecter la croyance sur une valeur qui ne fait pas partie du cadre de discernement (une valeur non disponible dans l'ensemble de définition).

Nous rappelons que la croyance est une masse unitaire qui est répartie entre chaque valeur ou groupe de valeurs possibles. Si l'utilisateur n'est pas sûr de la valeur choisie, il affecte un degré de confiance en relation avec sa confiance. Le reste est affecté à l'ensemble entier et traduit une information incertaine. Ce degré de confiance est équivalent à un pourcentage et pour faciliter la saisie, il est envisageable de traduire le degré de croyance à l'aide de modérateurs linguistiques (pas très sûr, moyennement sûr, quasiment sûr, sûr).

La deuxième forme d'intégration de l'incertain dans le modèle est plus complexe. Nous avons en effet choisi de traduire l'analyse des experts au moyen de relations qui sont incertaines par nature. Très souvent lorsque l'information est trop abondante, il devient nécessaire de la filtrer. Plutôt que de simplement sélectionner des attributs, nous souhaitons laisser la possibilité à l'expert d'explicitier une analyse imparfaite, avec des doutes. Par exemple, dans le cas d'un accident du travail sur une machine-outil (événement), l'expert sécurité pourrait analyser l'événement en sélectionnant les attributs traduisant que l'opérateur était inexpérimenté et que les dispositifs de protection étaient défectueux. Suivant la certitude de cette analyse. Il peut alors caractériser la confiance qu'il accorde à son analyse. Nous avons envisagé l'ajout d'une variable correspondant au degré de reproductibilité correspondant à l'estimation de l'expert quant aux chances de reproduction de l'événement en considérant son analyse. A l'image des MCT, nous avons souhaité augmenter les possibilités d'expression afin de pouvoir traduire le fait qu'une « cause » ne figure pas parmi les descripteurs de contexte disponibles. A ce stade, soit le modèle est étendu avec le nouveau descripteur, soit l'analyse qui fait intervenir l'élément « autre » est conservée. Bien que le MCT permette une telle manipulation, introduire de telles informations très imprécises a pour conséquence de répercuter dans les résultats finaux autant d'imprécision. Le principal avantage est de ne pas fausser l'analyse en considérant des données basées sur des hypothèses fausses. Il est ainsi préférable selon nous d'avoir une expérience qui ne soit pas vraiment exploitable plutôt qu'une expérience qui induise en erreur.

Nous avons aussi voulu permettre aux experts d'émettre des hypothèses disjonctives leur permettant de fournir plusieurs interprétations de l'accident. Cela pourrait être utile quand un expert hésite entre plusieurs scénarios valides pour expliquer un accident. En reprenant l'exemple précédent, nous pouvons imaginer que l'expert ne sait pas bien si l'accident sur la machine outil est dû aux causes présentées plus tôt ou tout simplement au fait que l'opérateur n'était pas suffisamment attentif. Dans ce cas, il fera une disjonction des analyses pour inscrire les deux possibilités dans la base. A nouveau, il convient de préciser que l'approche de retour d'expérience ne cherche pas à établir une vérité mais à collecter de façon incrémentale des connaissances à partir de faits. Si l'analyse d'un événement conduit à plusieurs scénarios valides, alors il est préférable de stocker chaque analyse puisque cela constituera autant de fragments de connaissance.

Nous donnons maintenant une vue plus formelle des deux entités intégrant l'incertitude dans la représentation de l'expérience à savoir le contexte et l'analyse.

## II.3.2. Formalisation du contexte

En vue de l'expression des processus de retour d'expérience qui s'appuient sur les entités de contexte et d'analyse dont la composante incertaine, il est nécessaire de préciser formellement en quoi consiste l'entité contexte, puis l'entité analyse afin de déterminer le lien avec les modèles des croyances transférables.

### II.3.2.1. La notion de descripteur

Les valeurs des attributs qui composent le contexte sont équivalentes à des éléments d'information atomique. Ces derniers sont définis comme des quadruplets (attribut, objet, valeur, confiance) (Dubois et Prade, 1988). L'« attribut » renvoie à une fonction qui affecte une « valeur » (ou ensemble de valeurs) à l'« objet » dont le nom figure dans l'élément d'information. Nous appelons cet élément d'information « descripteur ». Sa valeur correspond, en termes ensemblistes, à un sous-ensemble du référentiel  $\Omega$  lié à l'attribut. Elle peut aussi être considérée comme un prédicat. La « confiance » représente une indication sur la fiabilité de l'élément d'information. L'imprécis concerne la valeur de l'attribut tandis que l'incertain est relatif à sa vérité, et donc à la confiance. Les quatre entités peuvent être composites, c'est-à-dire composées d'entités (plusieurs attributs, plusieurs objets, prédicat *n-aire*, différents degrés de confiance).

A titre d'exemple, considérons les éléments d'information  $d_1$ , « Jean est probablement expert », ce qui peut être noté en notation canonique  $d_1$  (*niveau, Jean, expert, probable*) et  $d_2$ , « les conditions météo sur Tarbes seront bonnes demain avec une fiabilité de 80% », ce qui peut être noté  $d_2$  (*condition météo, {demain, Tarbes}, bonne, 80%*). On voit par exemple que pour  $d_2$ , l'attribut *condition météo* fait référence à deux objets du domaine, à savoir la zone correspondant à la ville de *Tarbes* et le moment symbolisé par *demain*. Il y a une différence entre l'attribut élément d'information associé à une valeur incertaine et les objets du domaine qui sont aussi d'une certaine manière une information. Or dans le deuxième exemple, il s'agit d'une forme attribut-valeur classique où la valeur est précise (lieu="Tarbes" et moment="demain"). La différence est assez subtile, les objets font partie du domaine, et sont exprimés à partir de l'ontologie du domaine alors que les éléments d'information sont des constructions (les descripteurs pour nous) basées sur les objets du domaine. Pour bien faire la différence, nous parlons de descripteurs pour les éléments d'information et d'attributs pour les variables des objets du domaine.

Nous ne nous attardons pas sur la modélisation d'objets complexes de domaines et leurs liens avec les descripteurs. Nous modélisons directement les descripteurs de l'expérience en considérant le descripteur sans préciser le lien avec les objets auxquels il fait référence.

### II.3.2.2. Le descripteur objet

Un élément d'information ou descripteur  $d$  possède un nom qui le caractérise, des objets du monde auxquels il fait référence et qui dépendent de l'ontologie du domaine, un domaine de valeurs possibles  $\Omega$  et une correspondance (*mapping*) entre ses valeurs et les objets mesurés du domaine. Cette correspondance existe toujours mais elle n'est pas obligatoirement explicitée. Enfin, le descripteur comprend une valeur  $v$  associée à une mesure de confiance. Nous avons fait le choix d'utiliser le modèle des croyances transférables pour exprimer cette valeur  $v$ . Ainsi, la valeur du descripteur et sa confiance sont codées par l'intermédiaire d'une fonction de masse sur les parties de  $\Omega$  notées  $2^\Omega$ .

Le modèle que nous souhaitons avoir du descripteur est un modèle objet simple. Nous désirons notamment pouvoir utiliser la propriété de composition des objets. Il s'agit d'un modèle classique attribut-valeur étendu au type objet (un attribut peut contenir d'autres attributs). Nous montrons dès la section suivante que cette approche peut-être ramenée à une vue attribut-valeur particulière où il faut considérer des groupes et des attributs ayant des profondeurs différentes.

A la manière d'un attribut classique, chaque descripteur possède un domaine de valeurs possibles et ces valeurs peuvent être de différents types : entier, réel, booléen, énumération (qui correspond à une extension de concept représentée par un ensemble de valeurs), objets (qui correspond à de nouveaux types). Le type objet est un type « complexe » par opposition aux autres types qui sont dits « simples ». Dans l'approche retenue, le type « objet » est équivalent à une composition de descripteurs, chacun étant soit de type simple, soit de type objet. Un contexte est donc une vue récursive composée de conteneurs de descripteurs et de descripteurs simples.

$$\text{contexte} = d_1 \times d_2 \times \dots \times d_n \text{ avec } d_i \text{ descripteur simple noté } d_i^s \text{ ou descripteur objet } d_i^o$$

Le domaine de  $d_i$  est noté  $\Omega_i$  et correspond à l'ensemble des valeurs possibles mutuellement exclusives. Cette hypothèse permet au formalisme d'être compatible avec les théories des fonctions de croyances. Elle signifie que les valeurs de domaines ne doivent pas se chevaucher. L'exhaustivité n'est plus une contrainte si l'on considère le MCT avec l'hypothèse du monde ouvert (possibilité d'affecter une croyance à l'ensemble vide) qui est avantageuse puisqu'elle permet aux utilisateurs d'exprimer qu'ils connaissent une information mais que celle-ci n'est pas présente dans le modèle qui leur est proposé. Cette possibilité pourrait être le point de départ d'un modèle adaptatif qui permettrait la modification incrémentale, en fonctionnement, du modèle.

Soit  $v$  la valeur d'un descripteur :

- Si le descripteur est de type simple, alors  $\Omega_i^s = \{a_i^1, a_i^2, \dots, a_i^n\}$  avec  $a_i^j$  une valeur atomique simple et  $v_i^s$  est une distribution de masse sur l'ensemble  $2^{\Omega_i^s}$ .
- Si le descripteur est de type objet, alors  $\Omega_i^o = \{o_i^1, o_i^2, \dots, o_i^n\}$  avec  $o_i^j$  correspondant à l'ensemble des objets instances et  $v_i^o$  est une distribution de masse sur l'ensemble  $2^{\Omega_i^o}$ .

Un descripteur objet est potentiellement composé de descripteurs simples et de descripteurs objets, si bien qu'il est possible d'exprimer l'ensemble de définition sous une forme ne comprenant que des descripteurs simples. Les définitions circulaires sont proscrites dans les représentations et par conséquent les représentations d'objets correspondent à une vue hiérarchique. Par exemple, considérons l'objet *contexte* représenté par l'arbre de la Figure II-8. Cet objet est composé de trois descripteurs simples et un descripteur objet, lui-même contenant un autre descripteur objet qui lui ne contient que des descripteurs simples. Nous utilisons comme notation  $d_{X-Y-Z\dots}$  ou X, Y et Z représente la position du descripteur dans la collection de chaque descripteur de chaque niveau. Le nombre d'identifiant de position représente le nombre de niveau ou encore la profondeur (X correspondant au niveau 1, Y au 2, Z au 3, etc.). Cette notation est utilisée dans les algorithmes de similarité (comparaison) que nous décrivons dans le chapitre III.

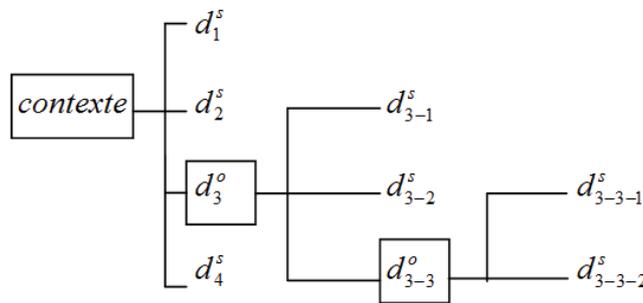


Figure II-8 - Représentation hiérarchique d'un attribut objet

Cette vue hiérarchique peut en effet être agrégée pour ne contenir que des descripteurs simples, il faudra cependant tenir compte de la profondeur de ces attributs dans la description du contexte lors de la comparaison. Pour l'exemple donné ci-dessus, la forme agrégée est la suivante :

$$\begin{aligned} \text{contexte} &= (d_1^s, d_2^s, d_3^o, d_4^s) = (d_1^s, d_2^s, d_{3-1}^s, d_{3-2}^s, d_{3-3}^o, d_4^s) \\ &= (d_1^s, d_2^s, d_{3-1}^s, d_{3-2}^s, d_{3-3-1}^s, d_{3-3-2}^s, d_4^s) \end{aligned}$$

Ainsi, par récursivité, il est possible de représenter le domaine d'un descripteur objet comme une collection d'attributs simples appartenant à des groupes différents et ayant des profondeurs différentes. Dès maintenant, nous ne considérons plus que des attributs simples. Nous pouvons aussi choisir d'exprimer le contexte sous une forme à un niveau d'attribut. Pour le même exemple que précédemment, nous renommons les attributs selon leur ordre de prise en compte. Des parenthèses servent à indiquer les différentes profondeurs :

$$\text{contexte} = (d_1^s, d_2^s, (d_3^s, d_4^s, (d_5^s, d_6^s)), d_7^s)$$

Quelque soit la sérialisation retenue, la vue hiérarchique du contexte peut donc être ramenée à un ensemble de descripteurs de type simple (entier, réel, booléen, énumération). Bien qu'il soit possible d'utiliser des valeurs continues, nous ne considérons que des valeurs discrètes pour les développements des chapitres suivants. Nous considérons uniquement des descripteurs  $d_i^s$  qui possèdent un domaine de définition de la forme  $\Omega_i^s = \{a_i^1, a_i^2, \dots, a_i^n\}$  que nous notons  $d_i$  et  $\Omega_i$  puisqu'il n'y a plus d'ambiguïté.

### II.3.2.3. Hybridation des descripteurs

Cette partie correspond à notre stratégie d'intégration et de prise en compte de l'incertitude dans le modèle du descripteur. Le *contexte* est composé de plusieurs descripteurs  $d_i$  ayant chacun un domaine de valeurs admissibles  $\Omega_i$ . Le domaine de *contexte* est noté  $\Omega$  et correspond au produit cartésien des domaines de chaque attribut qui le compose :

$$\Omega = \Omega_1 \times \Omega_2 \times \dots \times \Omega_n$$

Chaque descripteur du contexte possède une valeur qui est traduite par une fonction de masse correspondant à une distribution élémentaire de croyance par attribut. La fonction de masse correspond à la répartition d'une masse unitaire sur chaque partie de chacun des ensembles de définition  $\Omega_i^s$  de chaque attribut  $d_i^s$ . Ces distributions étant équivalentes et interchangeable avec des mesures de crédibilité et des mesures de plausibilité (voir partie II.2.1.4), nous n'utilisons par la suite que les fonctions de masses (notées *bba*) pour valuer des descripteurs puisque nous avons vu que le passage entre fonction de masse et fonction de crédibilité était réversible. Ainsi, la valeur d'un descripteur est notée :

$$v_{d_i^s} = \text{bba}(\Omega_i^s) = m_S^{\Omega_i^s}$$

Les éléments focaux représentent les différentes valeurs possibles et ils sont associés à un degré de confiance variant entre 0 et 1. Rappelons que le poids unitaire est réparti entre les sous-ensembles de valeurs possibles.

- Un poids de 1 sur une valeur atomique de l'ensemble est équivalent à une valeur certaine (fonction de masse certaine),

$$m_S^{\Omega_i^s}(\{a_i^j\}) = 1$$

- Par défaut, quand une valeur n'est pas donnée, il y a sur le domaine correspondant (c'est-à-dire sur le cadre de discernement associé), une fonction de masse vide qui traduit l'incertitude complète

$$m_S^{\Omega_i^s}(\{\Omega_i^s\}) = 1$$

Potentiellement, la masse peut être répartie entre tous ces sous-ensembles ce qui peut conduire à une explosion combinatoire. Cependant, la masse est en pratique souvent transférée sur un voire deux sous-ensembles. Très souvent, il s'agit de valeurs précises affectées sur un seul sous-ensemble contenant un seul élément (singletons).

L'incertitude est exprimée par le poids affecté. Plus ce poids décroît, plus la confiance décroît. Tout le poids non affecté (1 moins tous les poids affectés) correspond à l'incertitude restante et ce poids restant est affecté à l'ensemble complet, c'est-à-dire à  $\Omega$ .

Nous laissons aussi la possibilité d'exprimer le fait que la valeur ne fait pas partie du domaine. Dans ce cas, l'utilisateur affecte un poids à l'ensemble vide  $\emptyset$ . Il semble d'ailleurs intéressant d'envisager de permettre à l'utilisateur de saisir la valeur à laquelle il pense et qui n'est pas disponible. Cette nouvelle entrée pourrait servir d'information en cas de reconception du modèle du système de retour d'expérience. Dans le cas d'un modèle adaptatif, cette valeur pourrait servir à étendre le domaine de manière dynamique. La masse affectée à l'ensemble vide serait alors transférée sur la nouvelle valeur.

Enfin, avec cette approche et grâce aux différentes méthodes de fusion d'informations et de conditionnement, il est possible d'agrèger des valeurs saisies indépendamment par deux utilisateurs. Il est aussi possible grâce au conditionnement de venir préciser une information a posteriori. Nous n'abordons pas cet aspect de fusion d'information en relation avec la saisie multi-utilisateurs où de révision. Toutefois étant donné que notre langage d'expression des expériences est basé sur le MCT, nous pensons que ce sont des cas d'utilisation relativement qui peuvent assez naturellement être pris en compte. La principale difficulté provient du choix des méthodes de combinaison appropriées. Dans le cas de la fusion de deux saisies d'un même phénomène par deux utilisateurs, il faudrait par exemple utiliser une combinaison conjonctive (méthode de combinaison de Dempster-Shafer par exemple) ou encore la méthode hybride de Dubois et Prade.

Nous venons de voir comment représenter le contexte de l'expérience et en particulier comment il est possible de saisir des valeurs incertaines. Nous montrons dans la partie suivante comment exprimer les analyses d'experts.

### II.3.3. Formalisation de l'analyse

L'analyse est une étape primordiale dans la constitution de l'expérience. Sans cette étape, il ne s'agit pas vraiment d'expérience mais plutôt de « cas » (au sens du Raisonnement à Partir de Cas). Plusieurs types d'analyses sont bien sûr envisageables mais, dans notre approche, nous avons choisi de proposer une analyse très simple. Il s'agit d'un filtre qui permet de ne considérer que l'information utile. En effet, si les événements capitalisés ne sont pas bien bornés et bien distingués (modélisés précisément), l'application de capitalisation et de description du contexte a de fortes chances d'être trop exhaustive dans certains cas et pas assez précises dans d'autres.

La stratégie que nous proposons consiste, dans un premier temps, à collecter les informations sur le contexte sans orienter le choix de l'utilisateur quant aux descripteurs qui pourraient être importants. Inévitablement, cela conduit à ce que certaines informations présentes ne soient pas utiles en relation avec un problème particulier. De plus, même si la base contient beaucoup d'informations, tous les cas et les différents événements ne peuvent pas être imaginés et c'est pourquoi il peut manquer des informations utiles à la caractérisation d'un événement particulier. Considérant le cas très général qu'est la capitalisation d'événements sans classification a priori, nous avons alors proposé une analyse générique, indépendante du domaine, qui repose sur du filtrage et des combinaisons des informations. D'une part, les informations sont filtrées pour ne retenir que les informations significatives et d'autre part, à la manière d'une analyse causale, les informations significatives restantes sont combinées par l'intermédiaire de relations devant préciser l'événement selon un certain point de vue.

Nous avons ainsi choisi de définir des pondérations sur les attributs jugés significatifs. Les relations ainsi obtenues correspondent à des disjonctions de conjonction de descripteurs (ex : l'événement possède comme informations significatives  $A_1$  et  $A_3$  ou  $A_5$ ). Par cette méthode, nous souhaitons fournir une représentation qui met en évidence, par des relations, les entités du contexte significatives, c'est-à-dire les éléments sans lesquels l'événement ne se serait pas produit. Il y a ici deux objectifs :

- réduire la masse d'information pour ne garder que ce qui est important,
- établir d'une certaine façon l'origine de l'événement aux yeux d'un expert.

C'est d'ailleurs sur ce dernier point qu'est basée l'extraction de l'indicateur du risque présentée dans le chapitre IV. Nous avons donc choisi de proposer un mécanisme d'analyse assez simple dans le cadre d'un retour d'expérience basé sur des événements non explicités par avance (pas de classification des

types d'événements). Cette analyse correspond à une sélection d'attributs significatifs. Elle possède néanmoins une sémantique particulière par rapport à une simple sélection d'attributs. D'une part, les attributs qui ne jouent aucun rôle par rapport à l'événement sont écartés. D'autre part, avec les attributs restants, une relation le plus souvent conjonctive entre plusieurs attributs est créée traduisant le fait que l'événement n'aurait pas eu lieu (ou n'aurait pas été le même) si les attributs sélectionnés avaient eu des valeurs différentes.

Pour illustrer la représentation de l'analyse proposée, considérons une expérience  $exp_1$  relative à un événement  $ev_1$  et composée de cinq descripteurs  $d_1, d_2, d_3, d_4, d_5$ . Nous notons cette expérience de la façon suivante :  $exp_1(ev_1, contexte(v_{d1}, v_{d2}, v_{d3}, v_{d4}, v_{d5}))$  avec  $v_{di}$  la valeur du descripteur  $d_i$ , représentée par une distribution de masse de croyance. A partir de cette expérience, un expert pourrait exprimer comme analyse que l'expérience est expliquée par :

( $v_{d1}$  et  $v_{d3}$ ) ou ( $v_{d1}$  et  $v_{d5}$ ) ou encore uniquement par  $v_{d5}$

Les conjonctions d'attributs se traduisent par un « et » et les disjonctions par des « ou ». Il s'agit de la première information que nous souhaitons exprimer. Par contre, la disjonction n'a pas tout à fait le même sens que pour la valuation des attributs vue précédemment. Ici, l'analyse n'a pas pour objectif de déterminer les causes réelles de l'événement mais les attributs éventuels qui peuvent expliquer l'occurrence de l'événement considéré. Il n'apparaît, a priori, pas très intéressant d'utiliser directement les fonctions de croyance, ce qui reviendrait à répartir le poids sur les différentes explications possibles. En fait chaque possibilité est indépendante et correspond à une interprétation de l'expert. Nous ne recherchons pas la véritable cause, mais les causes possibles selon l'expert, ce qui permet de capitaliser davantage de connaissance à chaque analyse.

Disjonction et conjonction permettent ainsi d'exprimer une analyse qui semble assez intuitive. Cependant, il apparaît qu'il manque des informations importantes pour exprimer de manière plus précise et plus complète la mise en évidence des causes comme le ferait un expert. Ces informations permettent de préciser la complétude et la confiance dans les alternatives proposées :

- (a) La confiance attribuée par l'expert à chaque explication (terme de la disjonction) proposée peut ne pas être identique pour toutes les explications,
- (b) La confiance globale attribuée par l'expert à son analyse peut être plus ou moins grande.

En ajoutant une information sur la confiance de l'expert qui s'applique soit à chaque alternative soit à l'analyse globale nous pouvons intégrer ces deux aspects. Afin de séparer les deux notions, nous qualifierons de confiance le point de vue de l'expert sur l'analyse globale, et de justesse son point de vue sur une explication particulière. L'exemple précédent ( $vd1$  et  $vd3$ ) ou ( $vd1$  et  $vd5$ ) ou encore uniquement  $vd5$  deviendrait par exemple :

(confiance<sub>globale</sub>) - Analyse

- (justesse<sub>1</sub>) -  $v_{d1}$  et  $v_{d3}$
- (justesse<sub>2</sub>) -  $v_{d1}$  et  $v_{d5}$
- (justesse<sub>3</sub>) -  $v_{d5}$

La méthode retenue est donc différente d'une simple sélection/exclusion d'attributs car nous souhaitons offrir à l'utilisateur expert des moyens de mieux qualifier son analyse. Nous avons envisagé deux solutions présentées en détail dans les sections suivantes :

- l'utilisation du MCT non plus sur chaque attribut mais sur la description complète du contexte (produit cartésien des descripteurs),
- la création d'une nouvelle entité ayant une sémantique plus appropriée.

### II.3.3.1. Utilisation du MCT sur le produit cartésien

Nous avons, dans un premier temps, envisagé d'utiliser directement le modèle des croyances transférables en l'appliquant sur l'ensemble issu du produit cartésien des attributs. Cette solution aurait

un cadre de discernement égal à  $2^{2^0}$  qui correspond au domaine des analyses (toutes les analyses possibles exprimées à partir de « et » et de « ou »).

Cette approche conduit à plusieurs difficultés difficilement contournables pour pouvoir exprimer les relations voulues, en particulier les informations de confiance globale sur l'analyse et de justesse locale de chaque explication. Le poids unitaire affecté à chaque sous-ensemble possible ne peut correspondre qu'à la confiance globale de l'expert en son analyse, c'est-à-dire à l'ensemble des alternatives qu'il propose. Il n'est pas possible de distinguer simplement les trois alternatives de l'exemple ( $v_{d1}$  et  $v_{d3}$ ) ou ( $v_{d1}$  et  $v_{d5}$ ) ou encore uniquement  $v_{d5}$  pour leur affecter à chacune ce degré de confiance (ou de justesse). Il faudrait alors n'autoriser des analyses que sur une seule alternative et utiliser une combinaison disjunctive pour fusionner plusieurs analyses, bien qu'il s'agisse de la même situation.

L'utilisation des fonctions de croyances sur un tel cadre de discernement aurait pour résultat de rendre l'expression résultante de l'analyse très complexe et les traitements difficiles à mettre en œuvre. A titre d'illustration, pour l'exemple donné précédemment ( $v_{d1}$  et  $v_{d3}$ ) ou ( $v_{d1}$  et  $v_{d5}$ ) ou encore uniquement  $v_{d5}$ , l'analyse donnée  $\{(v_{d1}, v_{d3}) (v_{d1}, v_{d5}) (v_{d5})\}$  est traduite par le sous-ensemble du cadre discernement issu du produit cartésien des descripteurs. Cela semble nécessaire pour rester compatible avec les techniques de fusion de données et les opérations de combinaison classiques. Concrètement, cela signifie que, pour chaque dimension, tous les descripteurs sont présents et pas seulement les descripteurs sélectionnés par l'expert. Ainsi, pour revenir à une vue compatible, il faut associer aux descripteurs absents de l'analyse une fonction de masse vide, ce qui donnerait :

$\{(v_{d1}, v_{d2}(\text{vide}), v_{d3}, v_{d4}(\text{vide}), v_{d5}(\text{vide})) (v_{d1}, v_{d2}(\text{vide}), v_{d3}(\text{vide}), v_{d4}(\text{vide}), v_{d5}) (v_{d1}(\text{vide}), v_{d2}(\text{vide}), v_{d3}(\text{vide}), v_{d4}(\text{vide}), v_{d5})\}$

Cette solution possède l'avantage de l'uniformité et de la compatibilité avec les techniques choisies. Mais elle complexifie l'ensemble et si la technique envisagée dans cette partie est utilisée en classification et en fusion de données, elle est difficile à appliquer dans notre cadre. Pour pallier ces limites et représenter toute l'information voulue lors de la phase d'analyse, nous avons choisi de créer une relation particulière ayant une sémantique propre mais qui reste néanmoins proche des fonctions de croyance dans son principe.

La conséquence de l'utilisation d'un formalisme d'analyse propre est qu'il ne peut pas être traduit directement par des fonctions de croyances. Cela n'est pas réellement un problème puisque l'analyse est une étape intermédiaire en vue de l'extraction de connaissance. Il restera à adapter les algorithmes pour que ces derniers tiennent compte de la sémantique que nous avons donnée à l'analyse..

### II.3.3.2. Utilisation d'une relation Analyse

Le formalisme basé sur le MCT est difficilement applicable à l'analyse que nous souhaitons apporter et c'est pourquoi nous avons choisi de créer une relation adaptée à l'expression de l'analyse. Cette relation est basée sur l'utilisation des descripteurs définis par des fonctions de croyances. Nous reprenons l'exemple ( $v_{d1}$  et  $v_{d3}$ ) ou ( $v_{d1}$  et  $v_{d5}$ ) ou encore uniquement  $v_{d5}$  en le complétant des évaluations de confiance locale et de justesse des explications. Notons que nous ajoutons une information qui permet de qualifier la confiance du système en l'expert qui exprime son analyse. Cette information pourrait permettre, dans des travaux ultérieurs, d'intégrer des analyses de plusieurs experts.

L'exemple devient :

Analyse – (confiance<sub>globale</sub>= bonne) – (confiance<sub>expert</sub>= excellente)

1. (justesse<sub>1</sub>=0,75) -  $v_{d1}$  et  $v_{d3}$
2. (justesse<sub>2</sub>=0,50) -  $v_{d1}$  et  $v_{d5}$
3. (justesse<sub>3</sub>=0,25) -  $v_{d5}$

Cette analyse signifie que si dans une nouvelle situation, nous avons  $v_{d1}$  et  $v_{d3}$  alors il y a selon l'expert 75% de chance que l'événement se reproduise ; s'il y a  $v_{d1}$  et  $v_{d5}$ , il y a 50% de chance et enfin ; s'il

y a  $v_{d5}$ , seul alors il y a 25% de chance... La confiance de l'expert en son analyse est bonne et la confiance (du système) dans l'expert est excellente. Chacune de ces grandeurs est prise en compte dans les algorithmes de recherche que nous présentons dans le chapitre III. La confiance en l'expert est une information qui peut être prise en compte lors de la fusion d'analyses en provenance de plusieurs experts.

Plutôt que des pourcentages stricts pour la justesse, nous pensons autoriser le choix d'un intervalle de probabilité ou une liste de valeurs (peu de chance, possible, probable, très probable, à coup sûr). Cependant, pour la suite et pour expliquer les traitements, nous considérons une valeur numérique. L'analyse présentée ici correspond à une entité qui traduit la relation voulue. Il s'agira d'un objet à part entière qui peut être sérialisé ainsi :

$$\text{Analyse}(\text{bonne}, \text{excellent}, \{(0,75, v_{d1}, v_{d3}), (0,50, v_{d1}, v_{d5}), (0,25, v_{d5})\})$$

Contrairement à la disjonction de valeurs d'attribut vue précédemment, la disjonction d'analyse correspond à des alternatives toutes valides en même temps. C'est en particulier pour cette raison que le MCT n'a pu être appliqué directement.

### II.3.3.3. Analyses multi-expertes

La combinaison d'analyses provenant de plusieurs experts doit être envisagée comme une extension possible de nos travaux. Dans la première solution que nous avons étudiée (utilisation du MCT), elle est directement possible au moyen des différentes méthodes de combinaison des fonctions de croyances. Dans la solution que nous proposons, elle est aussi très directe puisque de nouvelles analyses correspondent à de nouvelles alternatives. Il faut en revanche faire attention à ne pas perdre la provenance de l'information, c'est-à-dire l'expert qui a donné son analyse. C'est pourquoi, de la même façon que l'on permet à un expert d'exprimer une divergence d'analyse, c'est-à-dire plusieurs analyses à partir d'un événement et d'un ensemble de descripteurs, il est important de considérer des analyses en provenance de plusieurs experts. Si l'on reprend l'analyse ( $v_{d1}$  et  $v_{d3}$ ) ou ( $v_{d1}$  et  $v_{d5}$ ) ou encore uniquement  $v_{d5}$ , on peut imaginer qu'elle correspond à la conjonction de trois analyses expertes.

- $\text{Analyse}_1(\text{expert}_1, \text{bonne}, \text{excellent}, \{(0,75, v_{d1}, v_{d3})\})$ ,
- $\text{Analyse}_2(\text{expert}_1, \text{bonne}, \text{excellent}, \{(0,50, v_{d1}, v_{d5})\})$
- $\text{Analyse}_3(\text{expert}_1, \text{bonne}, \text{excellent}, \{(0,25, v_{d5})\})$

Il y a cependant un point à éclaircir : le traitement des cas de redondance d'informations. A cause de l'aspect complémentaire des analyses, nous proposons d'utiliser une forme de combinaison conjonctive pour les analyses disjointes. Par contre, dans les cas d'analyses portant sur les mêmes attributs ou sur un sous-ensemble de ce groupe d'attributs, il semble plus approprié de transcrire cette redondance d'information. Par exemple, considérons deux analyses avec la première incluse dans la seconde :

$$(1) \{(0,75, v_{d1}, v_{d3})\} \subset (2) \{(0,5, v_{d1}, v_{d3}, v_{d5})\}$$

A partir de cet exemple, nous pensons qu'il faudrait retranscrire cette redondance d'informations. Si nous avons  $v_{d1}, v_{d3}, v_{d5}$  alors (2) donne une reproductibilité de 0,5 or,  $v_{d1}, v_{d3}$ , donne une reproductibilité de 0,75. Il nous semble alors naturel de transférer le maximum des degrés sur l'analyse qui contient l'ensemble maximum des descripteurs. Ainsi, (1) et (2) donnerait après « révision » :

$$(3) \{(0,75, v_{d1}, v_{d3}, v_{d5})\}$$

Cette proposition est une première réflexion sur la fusion des analyses. Dans le cadre de cette thèse nous nous sommes limités à la considération des analyses indépendantes, sans procédure de révision ou de fusion. Chacune des analyses apporte sa propre contribution comme nous le montrons dans les chapitres III et IV, consacrés aux mécanismes d'exploitation du retour d'expérience. Nous y développons les traitements qui permettent d'opérationnaliser le retour d'expérience à partir du formalisme de représentation hybride que nous avons proposé.

## II.4. Synthèse

Ce chapitre a été consacré aux formalismes de représentation de la connaissance. Nous avons tout d'abord présenté les approches à base de logique, à base de *frames* puis les langages hybrides qui unifient les deux, tels que les graphes conceptuels ou encore les logiques de description. Nous avons ensuite introduit les représentations « orientées objet », plus restreintes du point de vue des possibilités de représentation des connaissances que les précédentes approches mais plus flexibles et surtout suffisamment expressives pour notre objectif de représentation des expériences. Nous avons choisi un modèle attribut valeur classique étendue pour pouvoir prendre en compte des agrégations d'attributs. Ce choix de représentation assez libre nous a permis d'envisager un couplage avec des outils de représentation de l'incertitude, ce qui est une contrainte importante de notre système. Après avoir rapidement introduit les cadres de représentation de l'incertain, nous avons choisi le Modèle des Croyances Transférables que nous avons ensuite couplé avec le formalisme attribut-valeur étendu retenu précédemment. L'interface de couplage est l'attribut et plus particulièrement sa valeur. Nous nommons l'entité résultante descripteur et faisons la distinction entre descripteur simple (valeurs de type liste, numérique...) et descripteur objet (composition d'autres descripteurs).

Dans les chapitres suivants, nous détaillons les différents traitements que nous appliquons à partir du formalisme que nous venons de définir. Nous verrons en particulier comment les algorithmes de recherche tiennent compte de cette structuration de l'information, puis nous verrons les différents outils et interfaces développés.

# Chapitre III

## Exploitation, recherche et comparaison d'expériences

Les chapitres III et IV sont consacrés au volet exploitation du retour d'expérience. Nous y montrons comment il est possible, à partir du formalisme hybride de représentation de l'expérience proposé, de comparer et retrouver les expériences passées (chapitre III), puis d'extraire de la base une connaissance plus générale, à savoir un indicateur du risque relatif à l'expérience globale acquise dans la base d'expériences (chapitre IV).

Ce chapitre est donc consacré au développement des algorithmes de recherche d'expériences par rapport à un contexte courant. Nous traitons du fait que les traitements de l'expérience réalisés sans tenir compte de l'analyse experte sont similaires, dans une certaine mesure, à ceux effectués dans le cadre du Raisonnement à Partir de Cas (RàPC) et, en particulier, aux approches basées sur des représentations de cas selon un formalisme objet. C'est pourquoi nous commençons par présenter rapidement le RàPC dans ce chapitre. Nous nous attardons sur la composante principale du cycle du Raisonnement à Partir de Cas en relation avec notre problématique, l'étape de recherche par similarité. Nous présentons plusieurs approches compatibles avec la structure objet et la nature incertaine des expériences. Nous détaillons en particulier un algorithme de calcul de similarité applicable à une représentation objet intégrant des attributs et des mesures de similarité représentées dans le cadre de la logique floue et de la théorie des possibilités. Ensuite, nous présentons un algorithme de recherche adapté au cadre de notre proposition, c'est-à-dire tenant compte de l'expression des valeurs au moyen de descripteurs (objets et fonctions de croyance). Cet algorithme repose sur la proposition de mesure de similarité issue de la comparaison de fonctions de croyance en exploitant une connaissance représentée par une matrice de similarité entre les éléments du cadre de discernement. Nous montrons alors comment prendre en compte l'analyse experte associée à l'expérience au moyen d'une variante de cet algorithme.

### III.1. Du RàPC au retour d'expérience

Le Raisonnement à Partir de Cas (RàPC), ou *Case Based Reasoning* (CBR), est une méthode parente du retour d'expérience. En effet, le RàPC est un processus de résolution de problème débutant par une remémoration de situations similaires passées et se poursuivant par la réutilisation des informations et des connaissances disponibles sur ces situations afin d'élaborer une nouvelle solution au problème courant par adaptation des solutions passées retrouvées. Signalons que cette méthode est basée sur l'hypothèse d'avoir des solutions similaires lorsque les problèmes sont semblables.

D'autres méthodes, variantes du RàPC, intègrent davantage l'idée d'expérience. Nous parlons ici succinctement du Raisonnement à Partir de Trace (RàPT). Certains auteurs proposent l'idée plus générale d'*Experience Based Reasoning* (EBR) qui est un sur-ensemble de toutes les techniques de raisonnement utilisant l'expérience comme support. Dans les parties suivantes nous introduisons brièvement le RàPC, RàPT et EBR ce qui nous permet de positionner le retour d'expérience par rapport à ces méthodes.

### III.1.1.1. Raisonnement à Partir de Cas

Les bases du RàPC ont été apportées par Minsky et ses travaux sur les *frames* (Minsky, 1975) puis par Schank avec ses travaux sur la mémoire sémantique et la notion de scripts pour expliquer des situations (Schanck, 1982). Plus récemment, J. Kolodner a popularisé la méthode dans un ouvrage de référence sur le sujet (Kolodner, 1993). Le cycle, ou processus, du RàPC a été formalisé par Aamodt et Plazza (Aamodt et Plazza, 1994). Il est composé de quatre étapes : la recherche, l'adaptation, la révision et l'apprentissage. En résumé, les cas les plus proches du problème courant sont recherchés à partir d'une mémoire de cas. Ensuite, il faut réutiliser le(s) cas trouvé(s) en effectuant une adaptation pour résoudre le problème courant et enfin réorganiser la mémoire pour y intégrer le nouvel épisode. La base de cas, ou mémoire des cas, contient l'ensemble des cas passés utiles. Le lecteur intéressé pourra trouver des descriptions détaillées de ce paradigme de résolution de problème par la réutilisation de cas dans (Kolodner, 1993) (Aamodt et Plazza, 1994) ainsi que des exemples d'applications récentes dans les ouvrages (Renaud *et al.*, 2007a) (Renaud *et al.*, 2007b).

Le RàPC traduit la même logique, le même type d'inférence que le retour d'expérience à savoir réutiliser, quand cela est possible, ce qui existe déjà pour traiter une nouvelle situation. Contrairement aux méthodes d'apprentissage symbolique et logique, le RàPC est utilisé quand les phénomènes modélisés ne sont pas bien connus ce qui est aussi une caractéristique commune avec le retour d'expérience. Par contre, le retour d'expérience est une démarche plus générale qui s'inscrit dans la gestion des connaissances contrairement au RàPC qui est avant tout une technique d'apprentissage fonctionnant par analogie de cas. D'ailleurs, Bergmann le décrit comme une technique qui permet une mise en œuvre des méthodes de management de l'expérience (Bergmann, 2002). En outre, l'expérience est une entité plus cognitive que le cas du RàPC en raison de la présence de l'analyse experte. C'est ce point particulier qui nous conduit à des développements spécifiques pour intégrer l'analyse experte dans des mécanismes d'inférence analogues à ceux du RàPC.

### III.1.1.2. Raisonnement à Partir de Trace

Le raisonnement à Partir de Trace (RàPT), ou *Trace Based Reasoning* (TBR), est une évolution récente du RàPC qui vise à capturer la connaissance à partir de l'expérience (Champin *et al.* 2004). Le principe du RàPT est que l'expérience humaine, définie par des situations temporelles, est naturellement bien représentée par des enregistrements temporels appelés « traces ». Ces enregistrements décrivent le processus implicite sous-jacent et permettent de modéliser l'expérience pour en faciliter la réutilisation (Mille *et al.*, 2006).

Dans une certaine mesure, la capitalisation de traces est « transparente » puisque celle-ci est réalisée par le traçage de l'expérience (détection et capitalisation de *patterns*). Le RàPT étend les techniques des systèmes de RàPC exploitant la dimension temporelle (comportementale). CBR\*Tools est un cadre générique de modélisation de ces systèmes (Jaczynski et Trousse, 1997). Ces derniers permettent de représenter des cas sous forme d'épisodes consécutifs pour lesquels il n'est pas obligatoire d'utiliser des descripteurs de cas estampillés temporellement. Les épisodes de résolution de problème sont considérés indépendamment des contextes dans lesquels ils sont apparus. Un cas est défini avec une granularité figée, dans une temporalité spécifique et il est contraint par un vocabulaire de description. De manière intuitive, il a été proposé dans les travaux de Champin (Champin, 2003) d'exploiter l'utilisation des traces dans l'environnement d'un ordinateur comme système de capitalisation indirecte de connaissances, qui intervient lorsque l'utilisateur travaille en étant assisté par l'environnement logiciel. Ces travaux définissent la notion de trace, la manière dont les traces peuvent être représentées et les types de traitements qui peuvent être effectués de façon à retrouver des séquences d'événements utiles, ré-applicables à de nouvelles utilisations. Il est alors possible d'envisager des applications qui, suivant le contexte, s'adapteraient à l'utilisateur en reconnaissant des schémas expérimentés par d'autres utilisateurs. L'objectif est d'anticiper le besoin de l'utilisateur et de mettre à sa disposition, de manière transparente, l'expérience accumulée par les utilisateurs précédents. Dans un environnement logiciel, les traces sont des enchaînements d'interactions entre le système et l'utilisateur.

Le RàPT, peut être considéré comme une généralisation des principes du RàPC, consistant à exploiter les traces sur la base de la reconnaissance de *patterns*, autorisant ainsi certaines adaptations à de nouvelles situations (Mille, 2006). Plusieurs prototypes ont été réalisés et les recherches se poursuivent. Un cadre générique de représentation de la connaissance par l'expérience basé sur la notion de Trace a été développé et est nommé MUsETTE (Champin *et al.* 2004).

Le RàPT a pour objectif d'assister « intelligemment » un utilisateur en proposant des traces adaptées en fonction du contexte. Il est possible de considérer ces systèmes comme des adapteurs d'interfaces logicielles par des modifications de menus, des propositions de choix pertinents...). Le retour d'expérience est exploité différemment du RàPT car, le plus souvent, c'est l'utilisateur qui sollicite le système alors que le RàPT agit de façon transparente en assistant un utilisateur en fonction de son comportement (et des traces similaires qui sont extrapolées de ce comportement). Le retour d'expérience, au contraire, est plus proche dans son fonctionnement des outils classiques de capitalisation des connaissances où nous avons des phases distinctes de modélisation, de capitalisation et d'exploitation de la connaissance. Nous pensons néanmoins que cette approche (comportementale) pourrait être appliquée au retour d'expérience mais pour l'instant, nous avons fait le choix de modéliser l'expérience « statique » en collectant et structurant l'information. L'approche comportementale nécessite de modéliser la « dynamique » de la connaissance, son évolution...

### III.1.1.3. Expérience Based Reasoning

L'*Experience Based Reasoning* (EBR) est un terme introduit par Sun et Finnie (Sun et Finnie, 2004) qui désigne un cadre de modélisation logique des outils de raisonnements basés sur l'expérience. C'est en quelques sortes le sur-ensemble des approches telles que le RàPC ou encore le retour d'expérience. Par exemple, les auteurs définissent les fondations logiques du RàPC (Finnie et Sun, 2003). Nous trouvons aussi le terme d'ingénierie de l'expérience (Sun et Huo, 2006).

Les auteurs considèrent le EBR comme un paradigme de raisonnement général utilisé dans la plupart des activités humaines telles que les activités d'affaires, les activités militaires, les activités d'enseignement mais aussi dans les interactions comme le comportement, la communication, la négociation... Malgré cela, les auteurs regrettent que l'EBR n'ait pas été étudié formellement du point de vue logique et mathématique (Sun et Finnie, 2004). Ce champ de recherche consiste à formaliser n'importe quel type de raisonnement basé sur l'expérience à partir d'une plusieurs règles d'inférence qui constituent, selon les auteurs, la base des raisonnements qu'il est possible d'exprimer à partir de l'expérience. Les auteurs de cette théorie ont introduit huit règles d'inférence ensuite étendues et unifiées dans un cadre de traitements basés sur la logique floue (Sun et Finnie, 2007). Ces règles constituent les fondements théoriques de la proposition et couvrent toutes les possibilités de EBR. Cette approche, selon les auteurs, doit favoriser la recherche et le développement de systèmes de EBR, de systèmes intelligents, de systèmes flous, de système de gestion des connaissances et de gestion de l'expérience. A notre connaissance, il n'existe pas de systèmes opérationnels basés sur ces règles et ces travaux sur les aspects théoriques, restent des développements purement formels basés sur des considérations purement logiques. Le but poursuivi par ces systèmes est la justification logique d'un point de vue raisonnement de tout système raisonnant à partir de la notion d'expérience au sens large. D'ailleurs, nous pensons que les travaux sur la logique non-axiomatique, concrétisés par le développement du système NARS (Wang, 2006) (Wang, 1995), constituent une alternative plus directe et plus pragmatique. Ce système définit les valeurs de vérité et la signification à partir de l'expérience seulement et utilise une logique définie sur des bases cognitives différentes de l'approche mathématique habituelle (Wang, 2005) (Wang, 2004).

### III.1.1.4. Positionnement du retour d'expérience

Comme nous venons de le voir, le cadre du EBR peut servir de cadre générique permettant de positionner les différentes techniques ayant trait à l'expérience. Nous avons ainsi repris et enrichi un graphique proposé dans les travaux de Sun et Finnie pour y positionner le RàPT, le retour d'expérience mais aussi les systèmes de raisonnement non-axiomatique comme NARS (*cf.* Figure III-1). Sur cette

figure est représenté le fait que le retour d'expérience, le RàPC et le RàPT partagent certaines caractéristiques. NARS est un type de raisonnement logique particulier fondé sur la non obligation de cohérence (non axiomatique).

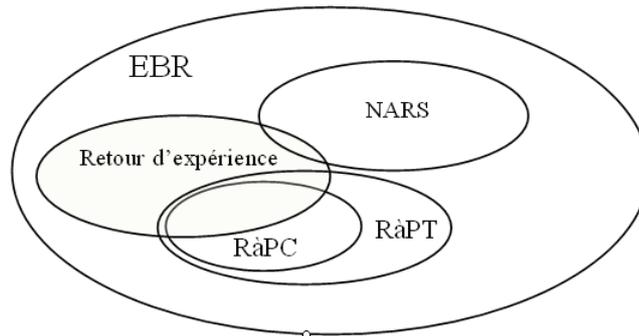


Figure III-1 - Positionnement du retour d'expérience

De notre point de vue, le retour d'expérience peut être considéré comme une extension du RàPC aux systèmes socio-techniques (Clermont *et al.*, 2007). Il s'agit d'un véritable processus métier qui implique des acteurs, des outils, de l'information et des processus. En outre, l'intégration explicite de l'analyse experte dans l'expérience forme une particularité importante du retour d'expérience. Le retour d'expérience possède donc des dimensions qui lui sont propres et que nous proposons d'étudier dans ce travail. Toutefois, les techniques utilisées en RàPC constituent un point de départ intéressant pour développer des approches utiles pour les raisonnements associés au retour d'expérience. C'est pourquoi nous consacrons la partie suivante à la description de plusieurs approches de RàPC proche de notre objectif puisqu'elles intègrent une représentation objet et/ou une représentation d'attributs à l'aide de la logique floue.

### III.1.2. Le RàPC objet et flou

Nous présentons ici quelques systèmes de RàPC basés sur l'utilisation d'un formalisme orienté objet qui mettent en œuvre des éléments de la logique floue pour prendre en compte la représentation de l'incertitude.

La notion d'objet intervient à de nombreux niveaux dans les systèmes de RàPC. Cela peut être pour la représentation des cas, qui nous concerne ici, mais aussi pour la modélisation de l'architecture interne du système d'inférence. Citons, par exemple, le *framework* CBR\*Tools qui est un cadre orienté objet extensible de modélisation du RàPC (Jaczynski, 1998) ou encore ROCADE dédié à l'identification et à la formalisation des tâches et connaissances propres au RàPC (Fuchs, 1997). Ces deux formalismes exploitent le paradigme orienté objet à la fois pour représenter les connaissances relatives au RàPC et pour la représentation interne des cas.

Le système REBECAS (pREdiction Basée sur dEs CAS) (Rougegrez-Loriette, 1994) représente les cas comme des objets et est implémenté dans le langage Smalltalk. Le système est dédié à la prédiction de l'évolution des feux de forêt. Les cas sont composés d'objets qui constituent directement la représentation du domaine. La cible est le paramètre ou l'objet qui indique le comportement du feu. Il y a, dans ce cas, un apport intéressant dans la mesure où un cas correspond à un historique de valeurs et donc à une suite d'enregistrements. Il s'agit d'un des premiers systèmes de RàPC basés sur la prise en compte d'historiques dans la représentation et la prédiction.

Plusieurs systèmes de RàPC prennent en charge les informations imprécises. Le système FLORAN (Salotti, 1992), qui signifie Filtrage fLou et Objet pour Reasonner par Analogie, possède un modèle de représentation objet et est implémenté dans SHIRKA (Rechenmann, 1985) qui est un formalisme de représentation des connaissances centré objet utilisant, notamment, la notion de facette. Un cas est ici appelé point de vue et est associé à un ou plusieurs contextes de dépendance qui précisent l'importance des attributs.

Le système CAREFUL (Case REtrieval based on FUZZY Logic) est un système de recherche de cas passés basé sur une représentation orientée objet des cas et une modélisation floue des informations imprécises (Jaczynski et Trousse, 1994). Cette modélisation floue permet d'exprimer des préférences sur les valeurs des caractéristiques des cas. Elle est exprimée par l'intermédiaire de la facette « restriction » et cette facette, qui permet de décrire les valeurs compatibles de l'attribut par rapport à son domaine, est décrite à l'aide d'un sous-ensemble flou. Les sous-ensembles flous sont aussi utilisés en relation avec des descripteurs linguistiques.

Dans le système FLORAN, des distributions de possibilité sont utilisées pour décrire les cas. On trouve par exemple, pour les attributs symboliques, des associations sous forme de couples entre un degré et un symbole. Les attributs numériques sont quant à eux représentés par des distributions de possibilité trapézoïdales. L'imprécision n'est pas uniquement utilisée dans la description des cas mais aussi dans les phases de recherche du RàPC par l'intermédiaire de « filtres flous » qui permettent de traduire l'appartenance d'un cas à un contexte selon différents degrés. Ces filtres déterminent à quel degré le cas cible appartient aux différents contextes de l'application. Les filtres flous sont définis lors de la construction du système par les experts et sont, d'une certaine manière, des règles permettant d'associer des cas à des contextes.

Le système WIND-1 est également un système qui utilise les informations imprécises pour déterminer la similarité entre deux valeurs d'un même attribut. Ce système est destiné à la prévision météorologique (Hansen, 2000). Pour chaque type de caractéristiques, des mesures de similarité sont définies par les experts et sont notamment décrites à l'aide d'ensembles flous. Sur les domaines continus, des ensembles flous sont construits pour traduire le fait que deux attributs sont très proches, proches ou encore peu proches. Pour les valeurs continues positives, l'importance d'une valeur par rapport à une autre est définie par les experts à l'aide de degrés. Enfin, pour les valeurs symboliques dont le domaine est un ensemble, les degrés de similarités sont disponibles sous la forme d'une table ou encore d'une matrice de similarité symétrique.

Citons enfin les travaux conduits par Ruet (Ruet, 2002) où sont définies des fonctions « proche de » permettant de déterminer si deux valeurs comparées sont proches au moyen d'une fonction de similarité définie comme un sous-ensemble flou. Du fait de la structure objet des cas utilisés, un processus récursif basé sur le calcul de similarité locale entre deux attributs et l'agrégation par l'intermédiaire d'une similarité globale sont utilisés. La mesure de similarité locale est définie pour un attribut sur un seul domaine alors que les similarités globales sont définies au niveau d'un objet sur le produit cartésien des domaines de ses attributs.

Nous pouvons remarquer dans ces travaux l'importance de la structuration de la connaissance (en général à l'aide d'un formalisme orienté objet) et de la représentation de l'incertitude et/ou de l'imprécision des informations lorsqu'une expertise doit être intégrée dans les cas. Ces travaux viennent donc conforter notre point de vue, constituent une source d'inspiration importante, mais ne permettent pas de traiter les problèmes que nous avons posés. C'est pourquoi nous proposons, dans la suite de ce chapitre, des développements spécifiques au cadre sélectionné notamment pour les fonctions de similarité.

### **III.1.3. Similarité, origine et calcul**

Nous nous focalisons ici sur la mesure de similarité qui doit tenir compte de la nature du contexte pour pouvoir être propagée à tous les descripteurs. Les mesures de similarité sont nécessaires afin de représenter numériquement les informations de ressemblance entre cas (Ruet, 2002). La similarité est une mesure de proximité entre des éléments du domaine qui témoigne d'une connaissance. Il ne s'agit pas nécessairement d'une distance au sens mathématique bien que les distances puissent faciliter la définition des mesures de similarité.

### III.1.3.1. Définition de la mesure de similarité

La similarité est une mesure de proximité entre deux éléments. Une mesure de similarité n'est pas obligatoirement liée à une mesure de distance bien qu'il soit possible et classique de la définir comme telle (Bisson, 1995). Selon Bergmann, distance et similarité sont des fonctions duales qui permettent d'exprimer la mesure de proximité entre les éléments d'information (Bergmann, 2002). Elles sont définies comme des fonctions  $f : \Omega \times \Omega \rightarrow [0, 1]$  ( $\Omega$  étant l'espace de caractérisation du problème ou du contexte dans notre cas). Des valeurs proches auront une similarité proche de 1, et une distance proche de 0. A l'inverse, des valeurs très éloignées auront une similarité proche de 0 et une distance proche de 1 (nous nous situons dans un espace « normalisé » prenant des valeurs entre 0 et 1). Nous notons  $sim$  la fonction de similarité et  $d$  la distance. Si  $d(c, c1) < d(c, c2)$  (ou encore si  $sim(c, c2) < sim(c, c1)$ ), alors le cas « c1 » est préféré au cas « c2 » si l'on veut réutiliser la solution associée pour résoudre le cas « c ».

Nous introduisons maintenant les différentes propriétés qui peuvent caractériser les mesures de similarité et de distance. Suivant les situations modélisées et les types d'informations qui servent à les représenter, seul un sous-ensemble de ces propriétés sera respecté. Soit  $\Omega$  l'ensemble des individus du domaine, et  $d$  l'application de  $\Omega \times \Omega$  dans  $[0,1]$ , alors  $\forall a, b, c \in \Omega$ , nous définissons les propriétés de réflexivité, de symétrie, d'inégalité triangulaire et de monotonie.

#### Réflexivité

Les mesures de similarité et respectivement de distance, est dite réflexive si  $sim(a, a) = 1$ , respectivement  $d(a,a)=0$ . De plus si  $sim(a,b)=1$ , respectivement  $d(a,b)=0$ , entraîne  $a=b$ , alors la mesure est dite fortement réflexive. La forte réflexivité est aussi connue comme la propriété d'identité. La plupart des mesures sont réflexives, ce qui signifie que l'utilité est maximale quand l'information est identique. Par contre, la réflexivité forte n'est pas souvent imposée puisqu'il existe des situations où l'expérience peut être réutilisée dans des situations différentes.

#### Symétrie

Les mesures de similarité et respectivement de distance, sont dites symétriques si  $sim(a, b) = sim(b,a)$ , respectivement  $d(a,b)=d(b,a)$ . Cependant, les mesures utilisées en pratique ne sont pas toujours symétriques. En effet, une leçon pourrait être utile pour une situation A et pour une situation B, alors qu'une leçon utile à une situation B pourrait ne pas l'être pour la situation A. La mesure ne serait alors pas symétrique.

#### Inégalité triangulaire

Les mesures de similarité et respectivement de distance respectent l'inégalité triangulaire si  $sim(a, b) + sim(b, c) \leq 1 + sim(a, c)$ , respectivement, si  $d(a, b) + d(b, c) \geq d(a, c)$ . L'inégalité triangulaire peut être vue comme une extension de la propriété de transitivité des relations. Cette propriété est dérivée de la propriété classique des distances en appliquant la transformation  $f(x)= 1 - x$ . C'est une propriété qui est souvent respectée pour les distances et qui est plus rarement vérifiée pour les mesures de similarité.

#### Monotonie

S'il existe une relation d'ordre notée  $<$  entre les éléments de l'ensemble  $\Omega$ , une mesure de similarité est dite monotone si  $a < b < c$  entraîne  $sim(a, b) \geq sim(a, c)$ . Une mesure de distance est dite monotone si  $d(a, b) \leq d(a, c)$ . La propriété de monotonie ne peut être utilisée que si une relation d'ordre existe sur les éléments mesurés du domaine. Elle signifie que la mesure est compatible avec l'ordre des éléments.

Nous étudions, dans les sections suivantes, les différentes mesures de similarité.

## III.2. Similarité attributs-valeurs et objet

De nombreuses mesures de similarité ont été proposées dans la littérature. Selon les exigences et selon la structuration des informations à comparer, nous sommes amenés à utiliser des types de mesures de similarités différents. C'est pourquoi chaque système utilise généralement sa propre mesure de similarité qui est une adaptation ou une combinaison de mesures existantes. Le quatrième chapitre de l'ouvrage de synthèse de Bergmann présente de manière détaillée les différentes approches compatibles avec les différents types de formalisation structurelle possibles pour le RàPC (Bergmann, 2002 – ch4). Nous résumons dans la suite de cette section celles qui semblent essentielles à notre approche basée sur les descripteurs objet. Nous détaillons en particulier l'approche attribut-valeur étendue puisque c'est le formalisme que nous avons choisi pour les descripteurs de l'expérience. Nous présentons aussi les similarités ensemblistes ou incertaines qui nous permettent de proposer une mesure de similarité adaptée à la nature des descripteurs de l'expérience.

L'approche attribut-valeur est la plus courante dans la représentation des cas du RàPC et aussi la plus simple des représentations structurelles. Un cas est alors défini par un ensemble d'attributs caractérisés par un type particulier auquel des valeurs sont associées. Il peut être représenté comme un vecteur de dimension  $n$ ,  $n$  désignant le nombre d'attributs. Dans la suite,  $\bar{x}$  et  $\bar{y}$  sont les deux cas que nous comparons et qui contiennent des attributs de types prédéfinis. Nous les notons sous forme vectorielle avec  $\Omega$  désignant le domaine des cas et  $\Omega^i$  le domaine de l'attribut de rang  $i$  :

$$\bar{x} = (x_1, \dots, x_i, \dots, x_n) \text{ et } \bar{y} = (y_1, \dots, y_i, \dots, y_n) \text{ avec } \bar{x}, \bar{y} \in \Omega \text{ et } x_i, y_i \in \Omega^i$$

### III.2.1. Similarité attribut-valeurs uniforme

Dans le cas simple où tous les attributs sont binaires  $\Omega^i = \{0, 1\}$ , une méthode consiste à utiliser la similarité par coefficient de correspondance simple qui provient de la distance de Hamming.

$$sim_{SMC} = \frac{1}{n} \cdot \left| \{i \mid x_i = y_i\} \right|$$

Une pratique fréquente consiste à pondérer les mesures avec des poids  $w_i \geq 0$  de sorte que  $w_1 + \dots + w_i + \dots + w_n = 1$ . Cette pondération permet d'exprimer l'importance relative des attributs dans le calcul de similarité. La mesure devient alors :

$$sim_{SMCw} = \sum_{i=1..n}^{x_i=y_i} w_i$$

Quand les attributs sont à valeurs réelles (normalisées entre 0 et 1), il est possible d'utiliser des fonctions de similarité uniformes, en particulier la distance de Manhattan, la distance euclidienne, la norme maximale ou encore la norme de Minkovski. Ces différentes mesures peuvent aussi être pondérées.

$$dist_{manhat}(\bar{x}, \bar{y}) = \frac{1}{n} \cdot \sum_{i=1}^n |x_i - y_i| \quad dist_{euclid}(\bar{x}, \bar{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

$$dist_{maxnorm}(\bar{x}, \bar{y}) = \max_{i=1}^n |x_i - y_i| \quad dist_{Minkov}(\bar{x}, \bar{y}) = \left( \frac{1}{n} \cdot \sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$$

$$dist_{MinkovPonderée}(\bar{x}, \bar{y}) = \left( \sum_{i=1}^n w_i \cdot |x_i - y_i|^p \right)^{\frac{1}{p}}$$

Enfin, il existe une extension de la distance euclidienne nommée QFD (distance de forme quadratique) qui permet d'exprimer des dépendances entre deux attributs par l'intermédiaire d'une matrice de poids  $W=[w_{ij}]$ .

$$dist_{quad-form}(\bar{x}, \bar{y}) = \sqrt{\sum_{i=1}^n \sum_{j=1}^n w_{ij} (x_i - y_j) \cdot (x_i - y_j)}$$

Toutes ces distances peuvent être transformées en mesures de similarité. Néanmoins, étant limitées aux valeurs réelles, ces mesures ne peuvent pas être appliquées directement à l'expérience dont les descripteurs sont de types différents. Les mesures présentées doivent donc être étendues pour prendre en compte des types d'attributs différents. Nous présentons dans la section suivante l'approche locale/globale qui permet de réaliser le calcul de similarité totale ou globale en utilisant un calcul récursif de similarités locales par attribut.

### III.2.2. Décomposition similarité locale/globale

Dans le cas des attributs de types différents, la modélisation de la similarité peut être simplifiée en considérant le calcul de la similarité globale, relative aux cas dans leur intégralité, comme l'agrégation de plusieurs similarités locales relatives à chaque attribut et donc au détail des cas ou de l'expérience.

#### Similarité locale

Une mesure de similarité locale est une mesure de similarité pour un attribut simple  $x_i$  défini par  $sim_{x_i} : \Omega^i \times \Omega^i \rightarrow [0,1]$  où  $\Omega^i$  correspond à l'ensemble des valeurs possibles pour le l'attribut  $x_i$ .

#### Similarité globale

Une mesure de similarité globale  $sim_\phi$  est défini par :  $sim_\phi(\bar{x}, \bar{y}) = \phi(sim_{\Omega^1}(x_1, y_1), \dots, sim_{\Omega^n}(x_n, y_n))$ . La fonction  $\phi : [0,1]^n \rightarrow [0,1]$  est appelé fonction d'agrégation et doit vérifier les propriétés suivantes :

- $\phi(0, \dots, 0) = 0$
- $\phi$  est monotone et croissante pour chaque terme

$sim_{\Omega^i}$  est une similarité locale si  $x_i$  est un attribut simple et une similarité globale définie par une fonction d'agrégation  $\phi'$  si  $x_i$  est un attribut relationnel (définition récursive). Autrement dit, pour les descripteurs de l'expérience, nous devons calculer une similarité locale pour les descripteurs simples et une similarité globale pour les descripteurs objet. La similarité du contexte est une similarité globale calculée récursivement à partir de tous les descripteurs du contexte.

### III.2.3. Fonction d'agrégation et modèle de poids

Les mesures de similarité globale sont définies comme des fonctions d'agrégation  $\phi$  appliquées sur les différentes valeurs de similarités locales des attributs. Les différentes fonctions correspondantes aux cas de similarités attribut-valeur uniformes numériques ou binaires (cf. III.2.1) peuvent être généralisées pour devenir des fonctions d'agrégation. Ces fonctions d'agrégation sont définies comme l'association d'une fonction basique d'agrégation et d'un modèle de poids exprimé sous la forme d'un vecteur de poids  $\bar{W} = [w_1, \dots, w_n]$  avec :

$$0 \leq w_i \leq 1 \text{ et } \sum w_i = 1$$

### III.2.3.1. Fonction d'agrégation standard

Il existe une multitude de fonctions d'agrégation possibles qui respectent les propriétés données précédemment. Certaines sont complexes mais, de manière générale, il est préférable que la similarité soit facile à calculer. Nous présentons ici les fonctions les plus classiques, qui peuvent éventuellement être combinées pour élaborer des fonctions plus complexes.

Soit  $s_i$  la similarité locale pour l'attribut  $i$  d'un cas  $c$  décrit par le vecteur  $\bar{x}$  à comparer avec le cas  $c'$  décrit par le vecteur  $\bar{y}$ . La similarité globale est exprimée au moyen des formules suivantes où  $\phi$  est une fonction d'agrégation :

$$\text{sim}_{\phi}(\bar{x}, \bar{y}) = \phi(\text{sim}_{\Omega^1}(x_1, y_1), \dots, \text{sim}_{\Omega^n}(x_n, y_n)) = \phi(s_1, \dots, s_n)$$

Les fonctions d'agrégation  $\phi$  les plus classiques sont formulées à partir des fonctions min, max, k-maximum, k-minimum, la fonction de Minkovski ainsi que la moyenne pondérée et le produit. Certaines de ces fonctions, toutes notées  $\phi$ , sont données ci-dessous :

$$\begin{aligned} \phi_{\text{moyenne}} &= \sum_{i=1}^n w_i \cdot s_i & \phi_{\text{max}} &= \max_{i=1}^n (w_i \cdot s_i) & \phi_{\text{min}} &= \min_{i=1}^n (w_i \cdot s_i) \\ \phi_{\text{Minkovski}} &= \left( \sum_{i=1}^n w_i \cdot s_i^p \right)^{\frac{1}{p}} & \phi_{\text{produit}} &= \prod_{i=1}^n w_i \cdot s_i \end{aligned}$$

La fonction moyenne  $\phi_{\text{moyenne}}$  est la fonction la plus courante où chacun des attributs contribue aux résultats de manière pondérée. La fonction de Minkovski  $\phi_{\text{Minkovski}}$  est une généralisation de la moyenne pondérée dans laquelle plus la valeur  $p$  est grande, plus l'influence des attributs possédant des similarités locales élevées est importante. Avec  $p=1$ , il s'agit de la moyenne pondérée ; avec  $p=2$ , il s'agit d'un équivalent de la distance euclidienne et quand le paramètre  $p$  tend vers  $\infty$ , alors le comportement tend vers la fonction maximum.

La fonction maximum  $\phi_{\text{max}}$  est apparentée à une similarité globale disjonctive puisque la similarité maximale d'un attribut parmi tous les autres est retenue. Au contraire, la fonction moyenne est qualifiée de conjonctive. La fonction minimum  $\phi_{\text{min}}$  est aussi une similarité globale conjonctive car elle tient compte de tous les attributs en ne conservant que la similarité minimum (tous les attributs ont une similarité supérieure à la similarité minimum). Enfin, la fonction produit  $\phi_{\text{produit}}$  est une fonction conjonctive dans le sens où toutes les similarités sont prises en compte et influent sur le résultat. Cette fonction est une mesure très sensible aux différences entre les attributs. De plus, une similarité locale nulle entraîne la similarité globale nulle.

### III.2.3.2. Modèle de poids

Chacune des fonctions d'agrégation est pondérée par un vecteur de poids  $w_i$  qui permet de prendre en compte l'influence relative des attributs dans le score globale de similarité. Il est possible de distinguer différents modèles en fonction de l'origine et du rôle de ces poids (Bergmann, 2002). Par exemple, le poids peut être affecté à la nature du cas et donc dépendre du domaine et de la modélisation. Ce poids est noté  $\overline{w_D}$ . Le poids peut aussi refléter une préférence de l'utilisateur pour sa recherche de cas passés. Dans ce cas il est noté  $\overline{w_U}$ . Il y a alors quatre possibilités :

- Le **poids global**, qui est le modèle le plus simple, intervient dans toutes les tâches de recherche. C'est un vecteur statique unique  $\overline{w_D}$  et donc il peut être considéré comme faisant partie de la fonction d'agrégation.
- Le **poids spécifique à une classe** correspond à une classification a priori du problème et des cas. Pour chaque classe, un vecteur de poids est déterminé et justifié par le domaine. Ces poids peuvent être combinés dans une matrice  $\Omega_D = [w_{Dij}]$  et la similarité est calculée après

classification en appliquant le vecteur de poids associé à la classe. Si le cas et le problème appartiennent à la classe  $C_j$ , alors la similarité est calculée en utilisant le vecteur de poids  $(w_{D1j}, \dots, w_{Dnj})$ .

- Le **poids associé à un cas** est un modèle plus précis qui considère le poids comme une caractéristique du cas. Un vecteur  $w_{Dc}$  est associé à chaque cas puis est utilisé lors des phases de recherche.
- Le **poids spécifié par l'utilisateur** est, quant à lui, une préférence fournie par l'utilisateur qui fait partie de la définition du problème. Il est noté  $w_U$  et peut être combiné avec une des trois formes de poids de domaine.

Pour combiner ces différents poids, la formule suivante est généralement utilisée :

$$\bar{w} = (w_1, \dots, w_n) = \frac{(w_{D1} \cdot w_{U1}, \dots, w_{Dn} \cdot w_{Un})}{(\bar{w}_D) \cdot (\bar{w}_U)^T}$$

Dans une certaine mesure, notre problème de comparaison de descripteurs tenant compte de l'analyse peut être rapproché de l'utilisation de poids associés au cas puisque l'analyse vient préciser quels ont été les attributs particulièrement significatifs aux yeux de l'expert.

### III.2.4. Mesure de similarité locale pour les attributs numériques

L'avantage des valeurs numériques réelles est qu'il existe une relation d'ordre sur les éléments du domaine ainsi qu'une notion de différence. Il est donc possible de déduire une mesure de similarité de ces distances dans l'espace normalisé en utilisant la transformation classique donnée par la formule suivante :

$$sim(x, y) = 1 - dist(x, y)$$

Les fonctions de différence typiques sont la différence linéaire standard notée  $\Delta_{std}$ , et la différence logarithmique notée  $\Delta_{log}$  :

$$\Delta_{std}(x, y) = x - y$$

$$\Delta_{log}(x, y) = \ln(x) - \ln(y) \text{ si } x, y \in \mathbb{R}^+ \text{ et } \Delta_{log}(x, y) = -\ln(-x) + \ln(-y) \text{ si } x, y \in \mathbb{R}^-$$

La différence logarithmique est généralement utilisée quand les valeurs sont étalées sur plusieurs ordres de magnitude. Il est aussi d'usage de distinguer les mesures symétriques des mesures asymétriques. Pour ramener cette mesure de distance à des similarités, nous utilisons deux fonctions  $f$  et  $g$  monotones décroissantes des réels dans  $[0, 1]$  tel que  $f(0) = g(0) = 1$ . Ces fonctions reflètent la diminution de la similarité en fonction de l'augmentation de la différence linéaire ou logarithmique. Nous utilisons la notation  $\Delta$  qui peut être la différence standard ou logarithmique.

$$sim_{sym}(x, y) = f(|\Delta(x, y)|)$$

$$sim_{asym}(x, y) = \begin{cases} f(\Delta(x, y)) \text{ quand } x < y \\ 1 \text{ quand } x = y \\ g(\Delta(x, y)) \text{ quand } x > y \end{cases}$$

Les fonctions de base qui permettent d'exprimer ces similarités, sont soit des fonctions seuils ( $f_{seuil}$ ), des fonctions linéaires ( $f_{lin}$ ), des fonctions exponentielles ( $f_{exp}$ ), ou enfin des fonctions sigmoïdes ( $f_{sig}$ ). Une illustration est donnée sur la Figure III-2 avec les fonctions associées. Le choix d'une fonction dépend de la manière d'interpréter la variation de similarité de deux valeurs.

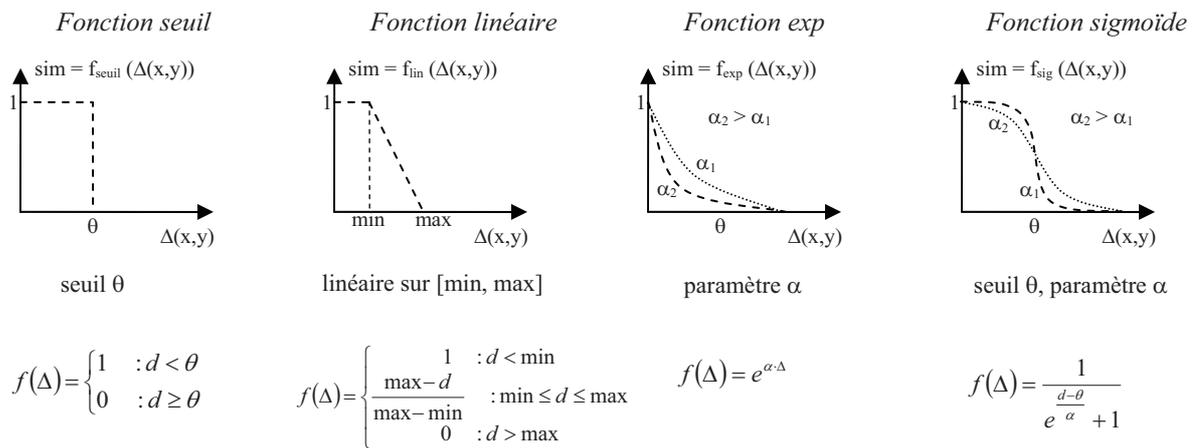


Figure III-2 - Illustration de différentes fonctions de similarité locale

### III.2.5. Mesure de similarité locale pour les listes de valeurs

Nous considérons maintenant les attributs symboliques, c'est-à-dire ceux qui prennent une valeur parmi une liste de choix possibles. La liste de valeurs  $x_i$  correspond alors au domaine noté  $\Omega$  tel que :

$$\Omega = \{x_1, \dots, x_i, \dots, x_n\}.$$

Si une relation d'ordre régulière existe (ou peut être déterminée) entre les éléments du domaine telle que pour tout  $i < j$  alors  $x_i < x_j$  (ou inversement  $x_i > x_j$ ) alors le type symbolique est dit totalement ordonné et le calcul de similarité peut être ramené au cas de similarité numérique précédemment discuté en prenant les indices ordinaux de chaque valeur. Cette approche assure que la mesure de similarité résultante est monotone par rapport à l'ordre des valeurs de l'attribut. Nous notons alors :

$$\text{sim}_{\text{symb}}(x_i, x_j) = \text{sim}_{\text{numeric}}(i, j)$$

Quand la relation d'ordre n'existe pas, il faut définir une matrice de similarité, qui est une table de correspondance entre toutes les valeurs possibles par attribut. Cette approche n'est envisageable que pour des domaines qui ne possèdent que peu de valeurs puisque l'effort nécessaire de capitalisation de connaissances pour renseigner les coefficients de la matrice  $S$  augmente avec le carré du nombre d'attributs du domaine. Pour le domaine  $\Omega = \{x_1, \dots, x_i, \dots, x_n\}$  de taille  $n$ , nous avons donc une matrice  $n \times n$  notée  $S = [s_{ij}]$  avec  $s_{ij}$  les valeurs de la matrices répondant à la condition suivante :

$$0 \leq s_{ij} \leq 1$$

Cette représentation, bien qu'elle induise un travail de capitalisation des connaissances supplémentaire, est très générale. Il est même possible de répartir des grandeurs numériques en catégories et donc de représenter leur similarité sous forme de matrice. Il faut d'ailleurs remarquer que le travail de capitalisation des connaissances, correspondant dans cette situation à la détermination des coefficients de la matrice  $S$ , n'est indispensable que dans le cas des attributs non-ordonnés. Pour les types ordonnés, cette tâche peut être automatisée avec une fonction numérique de similarité.

Dans la suite des développements, nous utilisons une représentation matricielle pour la similarité puisqu'elle permet de représenter différents types de mesures de similarité de façon homogène. Notons cependant que des pertes d'information peuvent être induites lorsqu'il est nécessaire de discrétiser les fonctions continues pour les inscrire dans ce formalisme. Nous considérons que ces pertes sont toutefois peu significatives dans le domaine considéré de l'expression de connaissances expertes.

La matrice  $S$  est symétrique avec une diagonale ne contenant que des éléments de valeur 1. Elle respecte donc toujours les propriétés de réflexivité et de symétrie.

$$S = \begin{bmatrix} s_{11} & \dots & s_{1i} & \dots & s_{1n} \\ \vdots & \ddots & & & \vdots \\ s_{i1} & \dots & s_{ii} & \dots & s_{in} \\ \vdots & & & \ddots & \vdots \\ s_{n1} & \dots & s_{ni} & \dots & s_{nn} \end{bmatrix} = \begin{bmatrix} 1 & \dots & s_{1i} & \dots & s_{1n} \\ \vdots & \ddots & & & \vdots \\ s_{1i} & \dots & 1 & \dots & s_{in} \\ \vdots & & & \ddots & \vdots \\ s_{1n} & \dots & s_{ni} & \dots & 1 \end{bmatrix}$$

Sachant que la matrice de similarité généralisée est considérée comme symétrique et réflexive, cela réduit le nombre de valeurs à saisir et réduit ainsi l'effort de saisie. Pour une matrice de taille  $n$ , le nombre de valeurs à saisir est de  $nbval$  :

$$nbval = \frac{n \cdot (n-1)}{2}$$

### III.2.6. Mesure de similarité locale pour les taxonomies

Pour les types symboliques ordonnés provenant d'une taxonomie (arbre où les nœuds intermédiaires représentent des classes d'objets réels et où les nœuds terminaux, les feuilles, représentent les objets réels), il faut considérer la structure de la taxonomie et la sémantique véhiculée par cette représentation (Bergmann, 1998). Il faut donc prendre en compte à la fois la similarité entre les objets réels (représentés par les feuilles) et la similarité structurelle qui correspond à la similarité conceptuelle traduite par la distance entre les nœuds intermédiaires communs. Pour illustrer ce principe, considérons la taxonomie simple suivante où 4 objets, a, b, c et d sont repartis selon 3 concepts comme cela est présenté sur la Figure III-3.

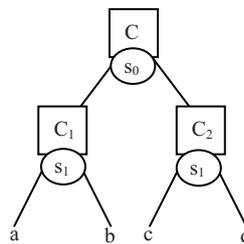


Figure III-3 - Représentation d'une taxonomie simple

Pour la comparaison d'objets réels (entre a, b, c, d), c'est-à-dire la similarité entre les feuilles de l'arbre, nous pouvons nous ramener à une forme matricielle comme dans le cas des listes non ordonnées de valeurs. Les nœuds intermédiaires correspondent à des concepts (représentés par des carrés) qui sont définis en extension par l'ensemble des objets qui descendent de ce concept (a et b sont de type  $C_1$  alors que c et d sont de type  $C_2$ ). La prise en compte de la nature des taxonomies entraîne une contrainte sur la définition de la mesure de la similarité entre les feuilles, à savoir  $sim(a, b) \geq sim(a, c)$  puisque le premier prédécesseur commun, c'est-à-dire  $C_1$ , est plus proche entre a et b, que C qui est le premier prédécesseur entre a et d. Cette contrainte implique une mesure de similarité ordinaire. Pour tenir compte de cette contrainte dans la définition de la matrice de similarité, une méthode simple consiste à associer à chaque nœud un poids de similarité qui est croissant quand on se dirige vers les feuilles et qui correspond à la similarité minimum (borne inférieure) si deux objets que l'on compare ont pour premier prédécesseur commun le nœud considéré. Nous avons ainsi :

$$s_0 < s_1, sim(a, b) \geq s_1 \text{ et } sim(a, c) \geq s_0$$

La règle précédente ne précise pas de relation entre des objets ayant le même prédécesseur. Par exemple, nous ne connaissons pas la relation entre  $sim(a, b)$  et  $sim(c, d)$ . Pour déterminer cette relation, il faut tenir compte de la hiérarchie dans l'évaluation des similarités locales. Nous ne détaillerons pas cet aspect et invitons le lecteur intéressé à lire l'article de Bergmann à ce sujet (Bergmann, 1998).

### III.2.7. Mesure de la similarité entre objets

En orienté objet, la représentation diffère de la vue attribut-valeur principalement par l'introduction des hiérarchies de classes et de l'héritage des attributs. Pour pouvoir définir et calculer la similarité entre deux objets instances de classes, il faut étendre le principe local-global au niveau objet en s'inspirant de la technique développée pour les taxonomies bien que ces dernières ne véhiculent pas tout à fait la même sémantique que les hiérarchies de classes.

La similarité entre deux objets peut être déterminée récursivement de façon ascendante. A partir de chaque attribut simple, une similarité simple, équivalente à une similarité locale, est calculée. Pour chaque attribut relationnel, équivalent à un descripteur objet, une similarité objet, équivalente à une similarité globale, est déterminée. Puis à nouveau, toutes ces mesures de similarité sont agrégées grâce à une fonction d'agrégation particulière. Cependant, cette approche a pour inconvénient de ne pas tenir compte de la position de la classe de l'objet dans la hiérarchie de classes, or cette position influence naturellement la mesure de similarité comme nous l'avons souligné dans le cas des mesures de similarité concernant les taxonomies. Bergmann et Stahl ont recensé, dans un cadre uniforme, les manières de déterminer ces similarités de classes au moyen d'une mesure à deux niveaux : la similarité intra-classe qui correspond à la comparaison des attributs communs des deux objets et la similarité inter-classes qui reflète la position des deux classes dans la hiérarchie (Bergmann et Stahl, 1998).

Il s'agit d'une technique similaire à celle utilisée pour les taxonomies même si la sémantique de la hiérarchie n'est pas tout à fait la même en particulier pour les nœuds intermédiaires. Dans une taxonomie, seules les feuilles, autrement dit les nœuds terminaux dénotent des objets réels. Les nœuds internes correspondent à des classes d'objets réels définies en extension par l'ensemble des feuilles qui sont accessibles depuis le nœud intermédiaire. Au contraire, les classes intermédiaires dans une hiérarchie orientée objet peuvent être liées à des instances réelles, c'est-à-dire des objets du monde. Les instances des sous-classes représentent des variations du premier objet. C'est d'ailleurs pourquoi, contrairement aux taxonomies où chaque nœud est associé à un symbole correspondant à une valeur, les nœuds dans une hiérarchie d'héritage objet sont décrits en intention par un ensemble de propriétés génériques non instanciées. Il s'agit des attributs (ou descripteurs dans notre cas), qui sont créés et donc valués lors de l'instanciation de leur classe. Les nœuds d'une taxonomie correspondent en extension aux objets associés (chacune des feuilles).

En conséquence, sachant que les feuilles des taxonomies représentent directement des objets alors que ce sont les instances des classes (même intermédiaires) qui représentent les objets réels (et non les classes), il est possible d'unifier les deux vues et d'appliquer les mêmes principes que ceux développés pour les taxonomies. Sur la Figure III-4, nous avons schématisé la possibilité d'étendre la hiérarchie objet de manière à rendre compatibles les mesures de similarité objet et taxonomie.

Sur la Figure III-4, une vue objet générique incluant le lien d'instanciation équivalent aux feuilles des taxonomies est représentée. Les classes abstraites peuvent être considérées comme des nœuds de taxonomies. Enfin, pour chaque classe concrète, nous pouvons ajouter un lien possédant une sémantique ensembliste et représentant la collection d'instances associées à la classe considérée. Chaque objet associé à cet ensemble peut aussi être représenté par un lien qui correspond aux feuilles des taxonomies. La similarité intra-classe représente la similarité maximum possible de deux objets. Elle est calculée sur la base des attributs communs des deux classes des objets comparées. Autrement dit il faut comparer les objets en fonction des attributs de la première superclasse<sup>1</sup> commune ( $C_{11}$  est la première superclasse commune de  $C_{111}$  et  $C_{112}$ ).

<sup>1</sup> Une superclasse est une classe dont hérite directement ou indirectement une classe donnée.

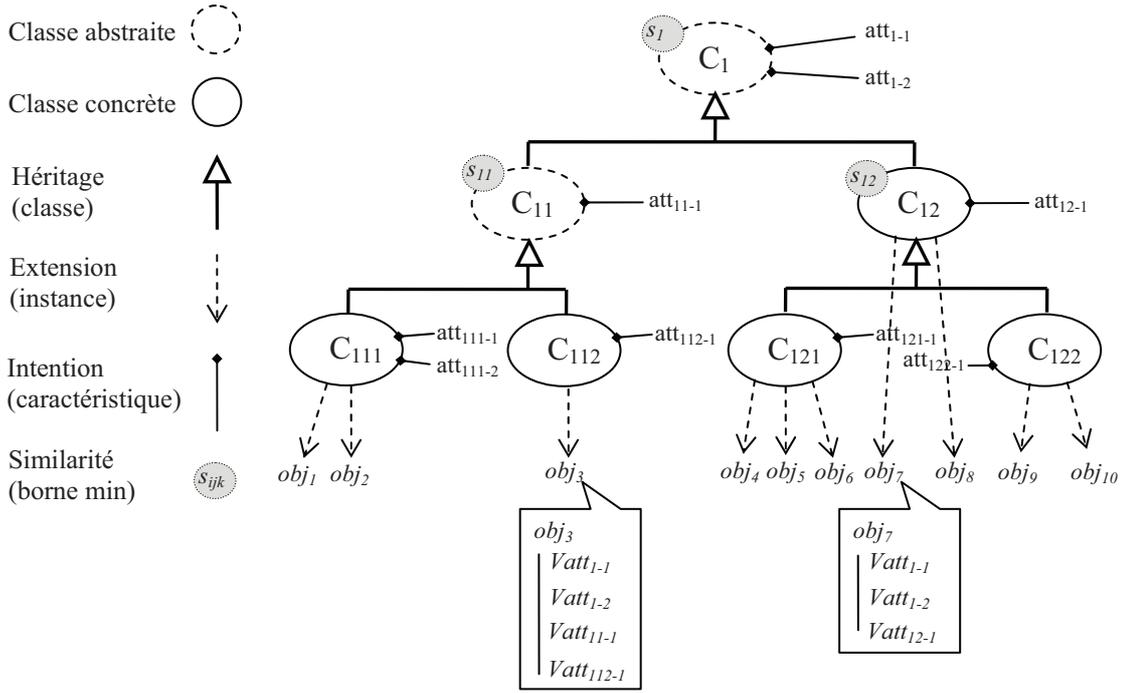


Figure III-4 - Présentation de la vue orientée objet en vue du calcul de similarité

La similarité interclasse est, quant à elle, proche des taxonomies de symboles. Ainsi, la similarité interclasse peut être définie comme la similarité des taxonomies si les classes intermédiaires sont toutes des classes abstraites (n'ayant pas d'instance). Dans le cas où il y aurait des classes concrètes intermédiaires (pour lesquelles des objets existent), une pratique pourrait consister à créer une sous-classe vide afin de rendre la super classe abstraite. Dans l'exemple donné sur la Figure III-4, il faudrait sous classer  $C_{12}$  en  $C_{123}$  par exemple, sans ajouter d'attributs, afin de rendre  $C_{12}$  abstraite et ainsi se conformer à la structure de type taxonomie. Ainsi pour calculer la similarité interclasse  $sim_{inter}$ , il faut annoter la chaque classe  $C_i$  de la hiérarchie de classe avec une valeur  $s_i$  comprise entre 0 et 1 telle que si  $C_i > C_j$  ( $C_i$  superclasse de  $C_j$ ), alors  $s_i \leq s_j$ . Cette valeur représente la valeur minimum pour la similarité intra-classe de deux instances quelconques de l'ensemble  $L_{C_i}$ .

$$sim_{inter} = \begin{cases} 1 : C_i = C_j \\ s_k : C_i \neq C_j \text{ avec } C_k > C_i, C_k > C_j \end{cases}$$

Lorsqu'on dispose de la similarité interclasse et intraclasse, il reste à combiner ces deux mesures par, en général, le produit de ces deux valeurs pour obtenir la similarité totale  $sim_{objet}$ . Ainsi pour deux objets  $o_p$  et  $o_c$  du nouveau problème  $p$  et du cas  $c$ , nous avons :

$$sim_{objet}(o_p, o_c) = sim_{Intra}(o_p, o_c) \cdot sim_{Inter}(class(o_p), class(o_c))$$

Par exemple :

$$sim_{objet}(obj_3, obj_7) = sim_{Intra}(obj_3, obj_7) \cdot sim_{Inter}(class(obj_3), class(obj_7))$$

$$sim_{objet}(obj_3, obj_7) = \phi[sim_{att_{1-1}}(obj_3, obj_7), sim_{att_{1-2}}(obj_3, obj_7)] \cdot s_1$$

Dans le cas de la comparaison d'objets, nous avons choisi, pour notre problématique, une stratégie différente. Plutôt que de considérer deux similarités distinctes, nous proposons simplement de comparer l'ensemble des attributs des deux objets sachant que certains attributs non présents seront traités au moyen du traitement de l'incertitude. Si les valeurs sont différentes, la similarité sera nulle.

### III.3. Similarité des ensembles

Le calcul de la similarité pour les représentations ensemblistes dépend de la sémantique associée à l'ensemble. Soit l'ensemble possède une sémantique conjonctive et dans ce cas tous les éléments de l'ensemble sont importants, soit la sémantique est disjonctive et alors seul un élément est la « véritable » valeur. Le premier cas correspond à une relation multi-valuée de type « et » entre les valeurs alors que le second cas correspond à une relation de type « ou » comme pour les fonctions de croyance et le cadre de discernement (cf. II.2.2.1). Dans la situation conjonctive, il faut comparer les attributs un à un alors que dans la situation disjonctive, toutes les valeurs ne doivent pas obligatoirement correspondre lors d'une comparaison par similarité. La sémantique conjonctive correspond à la muti-valuation d'attributs et la sémantique disjonctive à la valuation incertaine.

Nous nous posons ici la question de savoir quelle peut être la distance entre deux ensembles en fonction de la sémantique associée. Dans le cadre de nos travaux, nous avons choisi des attributs définis sur des domaines pour lesquels nous avons une connaissance de similarité entre chaque élément. Cette connaissance est exprimée, pour des domaines discrétisés, sous la forme d'une matrice de similarité. Nous considérons, dans un premier temps, le calcul de similarité ensembliste classique sans cette connaissance puis dans un second temps, nous introduisons une mesure de similarité ensembliste disjonctive qui tient compte de cette connaissance sur les éléments du domaine. Nous terminons en introduisant les similarités ensemblistes multi-valuées.

#### III.3.1. Similarité ensembliste sans connaissance sur les éléments

Sans considération sur la sémantique des ensembles, la distance la plus communément utilisée est une mesure du conflit inverse, qui exprime globalement la proportion d'éléments communs par rapport au nombre total d'éléments. En effet, si l'on suppose que les  $n$  objets d'un ensemble  $\Omega$  sont indiscernables et non ordonnés, la seule mesure qui permet de distinguer deux ensembles est la cardinalité. Ainsi la distance entre deux ensembles  $A, B$  de  $\Omega$  est une fonction de  $|A|$ ,  $|B|$ ,  $|A \cup B|$  et  $|A \cap B|$ .  $|A \cap B|$  mesure l'inverse du conflit puisque  $|A \cap B| = 0$  signifie que le conflit est maximum. La forme la plus fréquente de similarité d'ensembles, notée  $sim_{ensemble}$ , est donc la suivante (Jousselme *et al.*, 2001) :

$$sim_{ensemble}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Pour illustrer cette mesure, considérons l'ensemble  $\Omega = \{a; b; c; d; e\}$  et trois de ses sous-ensembles  $S_1 = \{a, b, c\}$ ,  $S_2 = \{b, c, d\}$  et  $S_3 = \{d, e\}$ . Intuitivement  $S_2$  semble plus proche de  $S_1$  que  $S_3$  puisqu'ils partagent plus d'éléments en commun et la calcul est conforme à cette intuition puisque :

$$sim_{ensemble}(S_1; S_3) = \frac{|S_1 \cap S_3|}{|S_1 \cup S_3|} = \frac{|\{\emptyset\}|}{|\{a; b; c; d; e\}|} = 0$$

$$sim_{ensemble}(S_1; S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = \frac{|\{b; c\}|}{|\{a; b; c; d\}|} = \frac{1}{2}$$

Cette mesure classique a aussi été utilisée pour simplifier des fonctions de croyance (Petit-Renaud et Denœux 1999), pour détecter des ensembles flous (intervalles imbriqués) (Setnes *et al.*, 1998) et aussi pour quantifier le conflit avec la théorie de Dempster (George et Pal, 1996).

### III.3.2. Similarité ensembliste avec connaissance sur les éléments

La mesure précédente est utilisée surtout pour les ensembles pour lesquels les éléments sont indissociables. Si, au contraire, les éléments du domaine peuvent être ordonnés ou comparés, ce principe n'est pas le plus approprié pour calculer la similarité ensembliste. Cette connaissance sur la comparaison des éléments peut être exprimée par une matrice de similarité dont il sera judicieux de tenir compte dans le calcul de la similarité de valeurs d'ensembles (avec sémantique disjonctive).

En effet, si nous considérons l'ensemble  $\Omega = \{a ; b ; c ; d ; e\}$  déjà utilisé mais que nous tenons compte cette fois-ci d'une matrice de similarité  $S$  entre les éléments de  $\Omega$ , alors la mesure précédente ne donnera pas forcément les résultats intuitifs attendus. Par exemple si la matrice donne  $sim(b ; d) = 0,6$  alors nous devrions retrouver  $sim_{ensemble}(\{b\} ; \{d\}) = 0,6$ . Or, la formule précédente donne une similarité nulle :

$$sim_{ensemble}(\{b\} ; \{d\}) = \frac{|\{b\} \cap \{d\}|}{|\{b\} \cup \{d\}|} = \frac{|\{\phi\}|}{|\{b ; d\}|} = 0$$

A notre connaissance, il n'existe pas de mesure de similarité tenant compte de ce type de connaissance dans l'interprétation d'un ensemble comme une valuation incertaine (la « vraie » valeur est une valeur de l'ensemble). Nous proposons donc une mesure de similarité pour les ensembles disjonctifs qui tient compte d'une connaissance sur les éléments du domaine exprimée sous la forme d'une matrice de similarité. Nous notons cette mesure de similarité  $sim_{ensdis}$  dans la suite.

Soit  $S = [s_{ij}]$  une matrice de similarité de taille  $n \times n$ . Considérons deux ensembles  $A$  et  $B$  tels que  $v_{d1}=A \subset \Omega$  et  $v_{d2}=B \subset \Omega$ ,  $|A| = k$ ,  $|B| = m$  et  $|\Omega| = n$  avec  $k < n$  et  $m < n$ .  $\Omega = \{w_1 ; \dots ; w_n\}$ . Soient deux descripteurs  $d_1$  et  $d_2$  auxquels sont associées les valeurs ensemblistes  $v_{d1}=A$  et  $v_{d2} = B$ .

- Si  $|A| = |B| = 1$ , alors  $v_{d1}$  et  $v_{d2}$  ont une valeur précise et la similarité des deux descripteurs est donnée directement par la matrice de similarité et vaut :

$$sim_{ensdis}(v_{d1} ; v_{d2}) = s_{ij} \text{ avec } v_{d1} = i \text{ et } v_{d2} = j$$

Dans le cas de valeurs précises, la mesure de similarité proposée doit correspondre aux valeurs de la matrice de similarité.

- Dans les cas où  $v_{d1}$  et/ou  $v_{d2}$  ne sont pas des valeurs précises, alors il faut spécifier le type de mesure de similarité à utiliser. Prenons l'exemple de deux valeurs complètement incertaines, c'est-à-dire :  $v_{d1}=v_{d2}=\Omega$
- Si la similarité est considérée comme l'égalité ensembliste des valeurs des descripteurs, telle que présentée dans la partie III.3.1, nous avons une similarité de 1 ( $|\Omega| / |\Omega| = 1$ ). Or cette approche n'est pas satisfaisante du fait de la nature des fonctions de croyance et plus particulièrement de la signification des fonctions de masse.  $v_{d1} = \Omega$  signifie que la valeur du descripteur 1 est inconnue. Nous savons cependant que la véritable valeur appartient à  $\Omega$ . Toutes les valeurs de  $\Omega$  sont possibles mais une seule est la véritable valeur. La similarité dans ce cas correspond à une valeur quelconque  $s_{ij}$  de la matrice de similarité. La similarité est donc comprise entre le plus petit élément  $s_{ij}$  de la matrice et le plus grand. Des approximations comme le minimum, le maximum, la moyenne ou encore des valeurs qui reflètent l'incertitude comme l'intervalle des valeurs extrêmes ( $[\min, \max]$ ) ou encore la liste complète des valeurs peuvent alors être utilisées. Ce dernier choix est le plus détaillé puisqu'il permet de ne pas perdre d'information.
- Lorsque les deux ensembles  $A$  et  $B$  sont quelconques, c'est-à-dire tel que  $A \subset \Omega$  et  $B \subset \Omega$ ,  $|A| = k$ ,  $|B| = m$ , nous avons les mêmes possibilités que dans le cas précédent. Il ne faut pas considérer toutes les valeurs de la matrice mais uniquement celles qui appartiennent à l'intersection des ensembles  $A$  et  $B$ . La liste des valeurs de similarité possibles est  $S_{AB}$  définie par :  $S_{ensdisAB} = \{s_{ij} \mid i \in A, j \in B\}$ .

Si  $s_{ij}$  est remplacée par la valeur de la matrice de similarité comprise entre 0 et 1, alors on obtient un système de points pondérés. Chaque valeur de la matrice de similarité peut être considérée comme un point  $P_i$  appartenant au segment  $[0, 1]$ . La pondération  $w_i$  correspond à l'occurrence de la valeur de similarité considérée pour tous les couples compatibles. Nous notons  $S_{AB_{pond}}$  la version pondérée de  $S_{ensdisAB}$  qui devient :

$$S_{AB_{pond}} = \{(P_i; w_i)\}$$

Exemple : pour  $A = \{a; b; c\}$  et  $B = \{b; c; d; e\}$ ,  $n = 3$ , et  $m = 4$  et  $\Omega = \{a; b; c; d; e; f\}$  et la matrice de similarité symétrique suivante :

<b>S</b>	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>
<b>a</b>	1	0,8	0,5	0,3	0,1	0
<b>b</b>		1	0,7	0,5	0,1	0
<b>c</b>			1	0,9	0,7	0,2
<b>d</b>				1	0,8	0,1
<b>e</b>					1	0
<b>f</b>						1

Tableau III-1 - Matrice de similarité de l'exemple

En considérant la matrice de similarité donnée par le Tableau III-1, nous avons :

$$sim_{ensdisAB} = \{s_{ab}; s_{ac}; s_{ad}; s_{ae}; s_{bb}; s_{bc}; s_{bd}; s_{be}; s_{cb}; s_{cc}; s_{cd}; s_{ce}\}$$

$$S_{AB} = \{0,8; 0,5; 0,3; 0,1; 1; 0,7; 0,5; 0,1; 0,7; 1; 0,9; 0,7\}$$

$$0,1 \leq S_{AB} \leq 1$$

$$S_{AB_{pond}} = \{(0,1; 2); (0,3; 1); (0,5; 2); (0,7; 3); (0,8; 1); (0,9; 1); (1; 2)\}$$

$$moyenne(S_{AB}) = \overline{S_{AB}} = \frac{7,3}{12} = 0,61$$

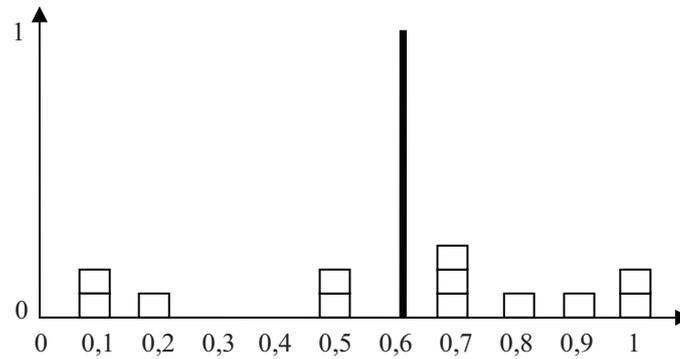
Notons que dans le cas des similarités « classiques » des ensembles, pour le même exemple nous aurions la distance entre A et B qui vaut :

$$S_{AB} = \frac{|A \cap B|}{|A \cup B|} = \frac{2}{5} = 0,4$$

Cette distance est en rapport avec la similarité ensembliste de A et de B. Si A et B sont disjoints, elle vaut 0, alors que, dans notre cas, deux ensembles disjoints n'ont pas forcément une similarité nulle. La détermination du système de points pondérés est analogue à la détermination d'une distribution de masse bayésienne (masse affectée à des sous-ensembles précis). Pour ramener le poids à des valeurs entre 0 et 1, il faut diviser chacun des poids par le nombre total de couples possibles à savoir  $|S_{AB}|$ . Pour l'exemple nous obtenons :

$$M_{AB} = \{(0,1; 1/6); (0,3; 1/12); (0,5; 1/6); (0,7; 1/4); (0,8; 1/12); (0,9; 1/12); (1; 1/6)\}$$

Nous tenons à préciser que  $S_{AB_{pond}}$  et  $M_{AB}$  sont des valeurs de prise de décision qui résultent d'une hypothèse forte. Il s'agit de la sommation des contributions de chaque couple par la distribution uniforme des chances d'avoir chaque valeur possible de similarité. Il s'agit néanmoins d'une information assez complète qui présente toutes les valeurs possibles. La moyenne pondérée permet d'estimer une valeur de synthèse tenant compte à la fois de l'imprécision et des différentes similarités. Ces deux informations sont combinées et peuvent être utiles lors de la prise de décision. Cependant, il faut retarder l'utilisation des mécanismes d'agrégation le plus possible car la nuance de l'information est alors perdue. Pour représenter cette information, nous avons choisi d'utiliser un graphe avec en abscisse les catégories (classes) de similarité et, en ordonnée, le poids de chaque catégorie. Naturellement, le choix des ensembles de discrétisation est laissé à l'appréciation de l'utilisateur. Une représentation graphique de  $M_{AB}$  est alors la suivante :

Figure III-5 - Représentation graphique de  $sim_{ens-dis}$ 

En outre, pour se conformer à la représentation par les fonctions de croyance, en accord avec les principes de cette théorie, nous aurions pu créer une nouvelle distribution de masse,  $m_{AB}$  à laquelle nous aurions associée un poids de 1 pour le sous-ensemble qui contient toutes les valeurs de similarité de couples possibles. Pour l'exemple, nous aurions eu :

$$m_{AB}\{0,1 ; 0,3 ; 0,5 ; 0,7 ; 0,8 ; 0,9 ; 1\}=1$$

Cette notation plus en accord avec l'affectation de croyance n'est cependant pas idéale car elle fait perdre la redondance d'information. En effet, certaines valeurs peuvent correspondre à plusieurs couples. C'est pourquoi nous avons préféré utiliser un système de points pondérés (analogue à une distribution bayésienne). Pour l'étape de prise de décision, nous faisons appel à la probabilité pignistique, notée  $BetP$ , ce qui revient à effectuer une transformation en distribution de probabilité uniforme sur les valeurs incertaines au moment de la décision (on retrouve un cas similaire à notre méthode avec une perte d'information) :

$$BetP(0,1)= BetP(0,3)= BetP(0,5)= BetP(0,7)= BetP(0,8)= BetP(0,9)= BetP(1)= 1/7$$

Le résultat précédent donne une moyenne de 0,615 qui est différente de la moyenne « exacte » trouvée plus tôt (0,61). Cette perte d'information, nous a conduit à utiliser une méthode différente des fonctions de croyance que nous avons utilisée jusqu'à présent. Utiliser les fonctions de croyances à ce stade aurait été intéressant si nous avions un besoin d'ajout ou encore de révision de l'information. Or, aucune nouvelle information n'intervient lors des calculs et c'est pourquoi nous raisonnons en termes de distribution bayésienne.

### III.3.3. Similarité d'attribut multi-valués

La multi-évaluation d'attributs est courante en représentation orientée objet et correspond aux ensembles possédant une sémantique conjonctive. En UML, par exemple, les associations entre objets possèdent des cardinalités facultatives qui indiquent les nombres minimum et maximum d'instances mises en relations. Dans les langages de représentation de connaissances de type *frame*, il y a des facettes « cardinalité min » et « cardinalité max » qui expriment cette contrainte.

Par exemple, si un descripteur décrit un groupe de randonneurs et que celui-ci, pour une expérience donnée, est composé de 4 personnes, cela signifie que l'attribut groupe contient une référence vers les 4 personnes. Le groupe est bien composé des quatre personnes ce qui est différent d'une évaluation incertaine où seule une valeur parmi toutes correspond à la réalisation.

Pour calculer une similarité avec ce type d'attribut, il faut définir une mesure de similarité locale pour les ensembles de valeurs. Ce problème de similarité n'est pas trivial car il dépend à la fois de la sémantique associée à la relation *n-aire* et de celle de la requête qui est posée. Cette similarité est considérée comme locale puisqu'elle caractérise un attribut. Cependant, elle s'apparente à l'utilisation de mesures d'agrégation.

Pour pouvoir donner une mesure, il faut connaître le nombre de valeurs qui importent lors de la requête qui correspond à une comparaison dans le retour d'expérience. Plusieurs cas sont possibles :

- toutes les valeurs importent (on cherche des cas avec des groupes constitués personnes présentant des similarités avec celles du groupe courant),
- une seule valeur suffit (on cherche parmi les personnes du groupe courant, les profils qui ont été analysés comme causes de problèmes),
- uniquement certaines valeurs importent (on cherche des cas où le nombre  $n$  de personnes appartenant au groupe est inférieur au nombre de personnes du groupe courant).

Par défaut, ces trois traitements doivent être envisagés pour effectuer la recherche de cas (ou d'expériences) qui contiennent un groupe de valeurs similaires, un sous-ensemble de valeurs ou une seule valeur parmi toutes. Sans précision au niveau du cas sur la sémantique associée au groupe (par exemple dans une analyse), il faut réaliser les trois types de requêtes. Ces requêtes reviennent à calculer la similarité entre tous les couples possibles de la requête (le problème) et du cas considéré lors de la recherche puis à appliquer une fonction d'agrégation sur ces valeurs.

Soient deux attributs  $a^p$  et  $a^c$  multi-valués et une mesure de similarité ensembliste conjonctive  $sim_{conj}(a^p, a^c)$  basée sur une mesure de similarité arbitraire entre les éléments individuels de l'ensemble des éléments  $sim_{elem}(u_j, v_j)$  (cette dernière valeur peut être soit une mesure de similarité simple si l'attribut contient plusieurs valeurs de type simple ou une similarité objet si les valeurs sont des objets ou attributs relationnels).

$$a^p = \{u_1; \dots; u_n\} \text{ et } a^c = \{v_1; \dots; v_m\}$$

$$sim_{conj}(a^p; a^c) = \phi \left( \begin{bmatrix} sim_{elem}(u_1; v_1) & \dots & sim_{elem}(u_1; v_m) \\ \vdots & \ddots & \vdots \\ sim_{elem}(u_n; v_1) & & sim_{elem}(u_n; v_m) \end{bmatrix} \right)$$

La fonction d'agrégation  $\phi$  dépend de la sémantique de l'ensemble. Si une seule valeur de la requête ou du cas est considérée alors cela revient à utiliser la fonction *maximum* par exemple comme fonction d'agrégation. Dans le cas de multi-valuation, cette sémantique particulière consiste à caractériser la correspondance entre les attributs multiples de la requête et ceux du cas comparé. Cette correspondance est donnée par la relation suivante :

$$\phi((s_{ij})) = \frac{1}{n} \cdot \max \left\{ \sum_{i=1}^n s_{i\gamma(i)} \mid R(\gamma) \right\}$$

Il est possible de définir plusieurs variantes de la sémantique de multi-valuation en fonction de la fonction d'exigence  $R(\gamma)$  que la correspondance doit vérifier. Les exigences suivantes sont les plus communes (Bergmann, 2002) (Bergmann et Eisenecker, 1995) :

- aucune exigence : n'importe quelle valeur dans les ensembles des deux valeurs. Par exemple, le maximum des similarités calculables peut être retenu,
- injective :  $\gamma$  est une fonction injective et donc une correspondance est considérée dans laquelle un seul élément du cas est assigné à un seul élément du problème (requête),
- surjective :  $\gamma$  est une fonction surjective et donc une correspondance est considérée dans laquelle tous les éléments du cas sont assignés à un seul élément correspondant du problème (requête),
- bijective :  $\gamma$  est une fonction bijective et donc tout élément du cas correspond à un élément unique du problème (requête).

Ces exigences peuvent servir lors de l'analyse pour, notamment, préciser l'importance des valeurs à prendre en compte dans l'ensemble de l'expérience analysée. Dans le cadre de cette thèse, nous

n'utilisons pas ce type d'attribut multi-valué car il complexifie la sémantique et surtout entrerait en conflit avec la multi-valuation (disjonctive) relative aux distributions de masse de croyance.

## III.4. Similarité sur la base de représentations incertaines

La théorie des ensembles est à la base des représentations de l'incertitude telles que les fonctions de croyance. La théorie des possibilités est basée sur la théorie des ensembles flous qui est un sur-ensemble de la théorie des ensembles classiques. Les ensembles qui servent de base à ces représentations sont de nature disjonctive. L'imprécision est exprimée comme une conjonction de valeurs possibles. La mesure de similarité doit donc être adaptée à cette situation. L'inconvénient majeur est que la mesure résultante ne reflète pas bien l'incertitude si la valeur de similarité est simple comme par exemple une valeur de similarité entre 0 et 1. Dans ce cas, la mesure de distance ou de similarité est biaisée par le degré d'incertitude des valeurs comparées. C'est pourquoi, nous préférons une mesure exprimée dans un cadre où moins d'information est perdue comme c'est le cas pour la mesure ensembliste avec connaissance où le résultat de la similarité est donné sous forme de distributions de masses bayésiennes au lieu d'une mesure agrégée qui confond l'information de distance (ou de similarité) avec l'information d'incertitude.

Nous voulons représenter le plus fidèlement possible l'information intégrant l'incertitude jusqu'au moment de la prise de décision. Nous avons vu, dans la partie III.3, comment mesurer la similarité entre deux ensembles disjonctifs avec ou sans connaissance de similarité entre les éléments de l'ensemble. Dans la méthode prenant en compte les matrices de similarité (avec connaissance), nous avons proposé une mesure qui synthétise toutes les mesures locales de similarités possibles. Cette mesure est équivalente à une distribution de masse bayésienne et tient compte des valeurs redondantes de similarité locales possibles.

Nous présentons dans cette partie des éléments relatifs aux mesures de similarités entre ensembles flous, distribution de possibilité et de nécessité, puis entre distributions de masse de croyance. Nous terminons par la proposition une évolution de la comparaison de distributions de masse pour prendre en compte la mesure ensembliste disjonctive que nous avons proposée dans la partie III.3.2.

### III.4.1. Similarité et théorie des possibilités

La théorie des possibilités, introduite par Zadeh en 1978 (Zadeh, 1978) et développé par Dubois et Prade en 1988 (Dubois et Prade, 1988), est en liaison étroite avec la théorie des sous-ensembles flous. La première permet de gérer les connaissances entachées d'incertitude alors que la seconde permet de traiter des données imprécises.

La croyance en un événement de  $\Omega$  (ensemble de référence) est ici déterminée par des degrés de nécessité et de possibilité. Les mesures de possibilité  $\Pi$  et de nécessité  $N$  sont des fonctions définies sur les parties de  $\Omega$  et à valeurs dans  $[0, 1]$ .

$$\begin{aligned} \Pi(\emptyset) &= 0 & \Pi(\Omega) &= 1 \text{ et } \forall A_i \in 2^\Omega \quad \Pi\left(\bigcup_{i=1}^n A_i\right) = \sup_{i=1}^n \Pi(A_i) \\ N(\emptyset) &= 0 & N(\Omega) &= 1 \text{ et } \forall A_i \in 2^\Omega \quad N\left(\bigcap_{i=1}^n A_i\right) = \inf_{i=1}^n N(A_i) \end{aligned}$$

Plus la valeur de la possibilité d'un événement est proche de 1, plus la réalisation de cet événement est possible. Contrairement aux probabilités, l'événement  $A$  ou son contraire  $\bar{A}$  sont tout à fait possibles, ce qui signifie :

$$\Pi(A) = 1 \text{ ou } \Pi(\bar{A}) = 1$$

Si les deux sont égales à 1, alors il s'agit d'une indétermination totale quant à la réalisation ou non de A. La mesure de nécessité permet de lever cette indétermination et revient à mesurer l'impossibilité de l'événement contraire. La valeur de nécessité indique le degré de certitude avec lequel un événement sera réalisé. Plus ce degré est proche de 1, plus il est certain que l'événement sera réalisé.

Il existe une relation clé entre les fonctions de possibilité et de nécessité :

$$\forall A_i \in 2^\Omega, N(A) = 1 - \Pi(\bar{A})$$

Le score total d'un produit de valeurs floues est le minimum de tous les scores individuels, parce que pour les logiques floues, la fonction par défaut pour représenter un « et » est le minimum. Au contraire, pour représenter un « ou », la fonction par défaut la plus courante est le maximum de toutes les valeurs individuelles. Plus généralement, les fonctions mathématiques T-normes et T-conormes sont les extensions des opérations ensemblistes classiques aux sous-ensembles flous. Il s'agit de fonctions commutatives, associatives, monotones possédant un élément neutre. 1 est l'élément neutre de la t-norme et 0 celui de la t-conorme.

Les définitions de mesures de similarité d'attributs exprimés à l'aide de la logique floue utilisent ces fonctions équivalentes aux opérations ensemblistes classiques d'union et d'intersection.

Un exemple de fonction de similarité floue est donnée par Ruet (Ruet, 2002) où sont définies des fonctions « proche de » notées  $\mu_{procheDe}$  permettant de déterminer si deux valeurs comparées sont proches au moyen d'une fonction de similarité définie par la formule suivante dans laquelle  $\Delta$  est une variable permettant d'influer sur la pente de la fonction résultante :

$$\mu_{procheDe}(a_1, a_2) = 1 - \frac{|a_1 - a_2|}{\Delta} \text{ si } |a_1 - a_2| \leq \Delta \text{ et } \mu_{procheDe}(a_1, a_2) = 0 \text{ sinon.}$$

Du fait de la structure objet des cas utilisés, Un processus récursif basé sur le calcul de similarité locale entre deux attributs et l'agrégation par l'intermédiaire d'une similarité globale est utilisé. La mesure de similarité locale est définie pour un attribut sur un seul domaine alors que les similarités globales sont définies au niveau d'un objet sur le produit cartésien des domaines de ses attributs.

Pour personnaliser les recherches, il est possible d'affecter un poids aux attributs du cas cible et ces poids sont pris en compte dans l'algorithme de calcul de similarité globale. Pour calculer la similarité locale entre attributs incertains (exprimés par des sous-ensembles flous), la mesure de possibilité et de nécessité est utilisée. Les comparaisons entre « X et R » et « X et D », où X est la proposition référence et R et D deux autres propositions, sont respectivement définies par les fonctions  $\mu_R(w)$  et  $\mu_D(w)$ :

$$\Pi(R, D) = \sup_{w \in \Omega} \min(\mu_R(w), \mu_D(w)) \text{ et } N(R, D) = \inf_{w \in \Omega} \max(\mu_R(w), 1 - \mu_D(w))$$

Pour comparer la similarité de deux valeurs d'un attribut  $x$ , Ruet propose d'utiliser la fonction de similarité  $\mu_{procheDe}$  associée à l'attribut  $x$ .  $\mu_{procheDe}$  sert de référence à la comparaison et les données des deux attributs comparés sont synthétisées avec l'opérateur de conjonction min :

$$\pi_D(a_1, a_2) = \min(\mu_{procheDe}(a_1), \mu_{procheDe}(a_2))$$

Ainsi, le degré de possibilité et de nécessité de proximité entre la valeur cible et la valeur source est donnée par :

$$\begin{aligned} \Pi_{local}(a_1, a_2) &= \sup(\min(\mu_{procheDe}(a_1, a_2), \pi_D(a_1, a_2))) \\ N_{local}(a_1, a_2) &= \inf(\max(\mu_{procheDe}(a_1, a_2), 1 - \pi_D(a_1, a_2))) \end{aligned}$$

La mesure de similarité globale est alors calculée par l'agrégation de toutes les similarités locales (dont le cas le plus complexe est donné ci-dessus). Cette mesure tient compte des poids  $w_a$  apportés aux attributs du cas cible par l'utilisateur :

$$\Pi_{global}(cas_{cible}, cas_{source}) = \min_{i=1..n} (\max(1 - w_a, S^i))$$

$$N_{global}(cas_{cible}, cas_{source}) = \min_{i=1..n} (\max(1 - w_a, S'_i))$$

$$\text{avec } S_i = \begin{cases} \Pi_{local}(a_{cible_i}, a_{source_i}) & \text{si } a_{source_i} \text{ existe} \\ 0 & \text{autrement} \end{cases} \quad \text{et } S'_i = \begin{cases} N_{local}(a_{cible_i}, a_{source_i}) & \text{si } a_{source_i} \text{ existe} \\ 0 & \text{autrement} \end{cases}$$

Cette mesure tient compte des poids ce qui permet de donner plus d'importance, dans le résultat final, aux attributs considérés comme importants. Ces deux mesures permettent de déterminer des bornes supérieures et inférieures de similarité entre le cas cible et le cas source. Trois cas sont distingués :

- $\Pi_{global}(cas_{cible}, cas_{source}) = N_{global}(cas_{cible}, cas_{source}) = 1$  alors le cas cible et le cas source sont identiques, au moins en ce qui concerne les attributs dont le poids est différent de 0,
- $\Pi_{global}(cas_{cible}, cas_{source}) = N_{global}(cas_{cible}, cas_{source}) = 0$  alors le cas cible et le cas source ne sont pas du tout similaires du point de vue des attributs importants (dont le poids est différent de 0),
- $\Pi_{global}(cas_{cible}, cas_{source}) = x$  et  $N_{global}(cas_{cible}, cas_{source}) = y$  avec  $x > y$ 
  - $x=1$  et  $y$  variable signifie que la ressemblance est possible avec un degré de certitude  $y$ ,
  - $x=1$  et  $y=0$  représente l'incertitude complète. La ressemblance peut-être totale, partielle ou nulle,
  - $1 > x > y > 0$  signifie que la ressemblance est possible à un degré  $x$  et certaine à un degré  $y$ . Il s'agit du cas le plus courant,
  - $1 > x > 0$  et  $y=0$  signifie que la ressemblance est au maximum du degré  $x$  mais elle peut être nulle.

En résumé, cette méthode permet de déterminer un intervalle de similarité objet tel que :

$$N_{global}(cas_{cible}, cas_{source}) \leq SIM_{objetFlou}(cas_{cible}, cas_{source}) \leq \Pi_{global}(cas_{cible}, cas_{source})$$

L'inconvénient de cette méthode est qu'elle représente uniquement les bornes de la mesure de similarité. En d'autres termes, elle ne tient compte, en raison de l'utilisation des fonctions min et max, que de la similarité des attributs les plus « défavorables » (pour la nécessité) et « favorables » (pour la possibilité). L'utilisation des mesures de probabilité serait encore plus pénalisante, le produit des valeurs retournant une valeur inférieure à celle de la fonction min. Les fonctions de Lukasiewicz pourraient donner un résultat plus resserré mais, dans tous les cas, le résultat est équivalent à des bornes supérieures et inférieures pour la mesure de similarité recherchée. Nous montrons dans la partie III.4.3 que l'utilisation des distributions de croyances permet de travailler avec une information plus riche contenant toutes les valeurs de similarité possibles et une distribution de type probabilité sur les valeurs possibles. Avant de présenter cette méthode, nous décrivons des fonctions de similarités de distributions de masses de croyance.

### III.4.2. Similarité de fonctions de masses

Nous avons choisi d'utiliser les fonctions de croyance avec en particulier la distribution de masse pour exprimer les valeurs incertaines d'attributs. Pour pouvoir comparer deux descripteurs incertains, il faut donc pouvoir comparer deux distributions de masse afin d'en tirer une mesure de similarité locale. Nous présentons, d'abord, deux méthodes de calcul de similarité proposées dans la littérature, puis nous proposons une extension tenant compte de la mesure ensembliste disjonctive avec connaissance de

similarité entre éléments introduite dans la partie III.4.3. En effet, la plupart des mesures existantes reposent sur une similarité ensembliste disjunctive sans connaissance de similarité.

Une fonction de masse correspond à une distribution d'un poids unitaire sur les parties d'un ensemble  $\Omega$ . Pour illustrer ces fonctions, nous utilisons une représentation en treillis de toutes les parties possibles d'un ensemble  $\Omega = \{a, b, c, d\}$  sur lequel nous positionnons deux fonction de masse  $m_1$  et  $m_2$  :

$$\begin{array}{lll}
 m_1(\Omega) = m_{1-1} = 0,3 & m_1(\{a; b\}) = m_{1-2} = 0,3 & m_1(\{d\}) = m_{1-3} = 0,4 \\
 m_2(\Omega) = m_{2-1} = 0,1 & m_2(\{b; c; d\}) = m_{2-2} = 0,6 & m_1(\{b\}) = m_{2-3} = 0,3
 \end{array}$$

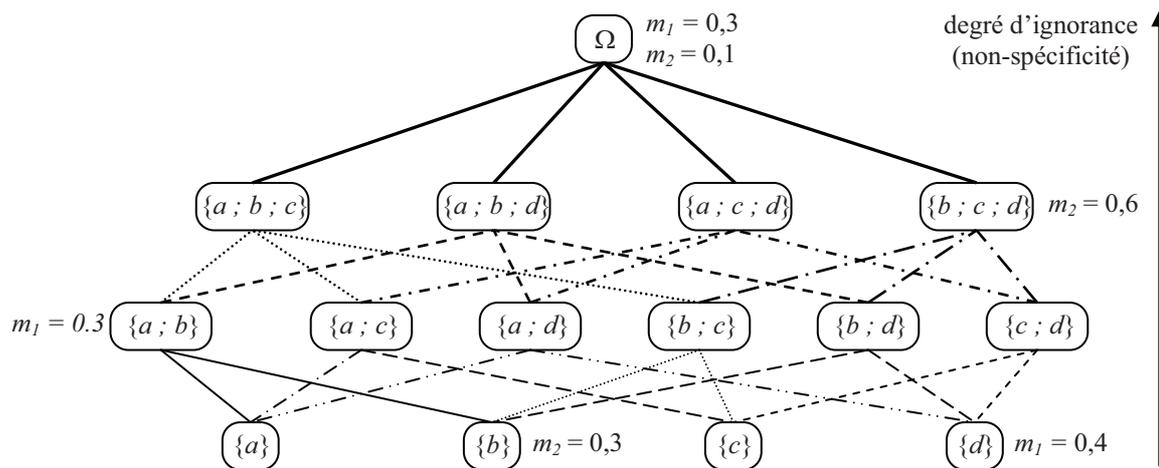


Figure III-6 - Treillis des sous ensembles de  $\{a; b; c; d\}$

Cette vue permet de bien voir les liaisons avec les sous-ensembles sur lesquels la croyance peut être transférée. Par exemple le poids de 0,3 affecté à  $\{a; b\}$  par  $m_1$  peut-être réparti sur  $\{a\}$  et/ou sur  $\{b\}$ .

Jousselme *et al.* définissent la distance entre deux distributions de masse  $m_1$  et  $m_2$  sur le même cadre de discernement  $\Omega$  de taille  $n$  en remarquant que :

$$d_{bba}(m_1, m_2) = (\vec{m}_1 - \vec{m}_2)^T \cdot \underline{\underline{D}} \cdot (\vec{m}_1 - \vec{m}_2)$$

$\vec{m}_1$  et  $\vec{m}_2$  sont des représentations vectorielles de taille  $2^n$  de  $m_1$  et  $m_2$  (chaque axe représente une partie de  $\Omega$ ). Si on prend la matrice identité pour  $\underline{\underline{D}}$ , on obtient une distance euclidienne. Cette situation n'est pas intéressante car elle donne un poids aux sous-ensembles sans tenir compte de la similarité entre les ensembles. Les auteurs ont alors proposé pour  $\underline{\underline{D}}$  la matrice de similarité ensembliste  $D_{ens}$  sans connaissance :

$$D_{ens}(A, B) = \frac{|A \cap B|}{|A \cup B|} \text{ avec } (A; B) \in (2^\Omega)^2$$

Cette mesure  $D_{ens}$  est justifiée en considérant que les  $n$  objets de  $\Omega$  sont indiscernables et non-ordonnés.  $D_{ens}$  traduit une sorte de conflit inverse (Jousselme *et al.* 2001).  $D_{ens}$  est une matrice  $2^n \times 2^n$ . Par exemple pour  $\Omega = \{a; b\}$ , nous avons :

$$D_{ens} = \begin{array}{l} \{a\} \\ \{b\} \\ \{a; b\} \end{array} \begin{array}{ccc} \{a\} & \{b\} & \{a; b\} \\ \left[ \begin{array}{ccc} 1 & 0 & 1/2 \\ 0 & 1 & 1/2 \\ 1/2 & 1/2 & 1 \end{array} \right] \end{array}$$

$D_{ens}$  n'est pas une matrice de similarité entre éléments mais une matrice qui donne le nombre d'éléments présents dans l'intersection de chaque sous-ensemble possible.

Considérant  $D_{ens}$ , la mesure de similarité entre distributions de masse devient:

$$d_{bba}(m_1; m_2) = \sqrt{\frac{1}{2} \cdot (\vec{m}_1 - \vec{m}_2)^T \cdot D_{ens} \cdot (\vec{m}_1 - \vec{m}_2)}$$

Le facteur  $\frac{1}{2}$  est nécessaire pour normaliser la distance  $d_{bba}$  et pour garantir que :

$$0 \leq d_{bba}(m_1; m_2) \leq 1$$

La mesure peut aussi être exprimée sous la forme :

$$d_{bba}(m_1; m_2) = \sqrt{\frac{1}{2} \cdot (\|\vec{m}_1\|^2 + \|\vec{m}_2\|^2 - 2 \cdot \langle \vec{m}_1; \vec{m}_2 \rangle)}$$

Avec  $\langle \vec{m}_1; \vec{m}_2 \rangle$  le produit scalaire défini par :

$$\langle \vec{m}_1; \vec{m}_2 \rangle = \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} m_1(A_i) \cdot m_2(A_j) \cdot \frac{|A_i \cap A_j|}{|A_i \cup A_j|} \text{ et } \|\vec{m}\|^2 = \langle \vec{m}; \vec{m} \rangle$$

Cette mesure de distance correspond à la somme, pour chaque sous-ensemble, des produits des poids associés aux sous-ensembles multipliés par la similarité ensembliste des deux sous-ensembles considérés. Elle ne nous satisfait pas car elle est fondée sur une similarité binaire entre les éléments et elle associe une similarité inférieure à 1 quand on compare des ensembles dont l'intersection est non nulle, alors qu'il est possible que la similarité soit de 1.

Illustrons cette remarque avec l'exemple suivant sur  $\Omega = \{a, b, c\}$ , en supposant que pour  $\Omega = \{a, b, c\}$

$$\begin{aligned} m_1(\Omega) = m_{11} = 0,3 & & m_1(\{b\}) = m_{12} = 0,7 \\ m_2(\Omega) = m_{21} = 0,1 & & m_2(\{b; c\}) = m_{22} = 0,9 \end{aligned}$$

### Application de la mesure de Jousselme

$$\begin{aligned} \langle m_1; m_1 \rangle &= m_1^2 = 0,09 + 0,21 \cdot 1/3 + 0,21 \cdot 1/3 + 0,49 = 0,72 \\ \langle m_2; m_2 \rangle &= m_2^2 = 0,01 + 0,09 \cdot 2/3 + 0,09 \cdot 2/3 + 0,81 = 0,94 \\ \langle m_1; m_2 \rangle &= 0,03 + 0,27 \cdot 2/3 + 0,07 \cdot 1/3 + 0,63 \cdot 1/2 = 0,548 \\ d_{bba} &= \sqrt{0,564} = 0,751 \end{aligned}$$

Citons comme autre exemple une approche qui permet de calculer la distance entre les éléments focaux en tenant compte de la proximité des sous-ensembles comparés au cadre de discernement (qui représente l'ignorance totale) (Yahia et Rougegrez-Loriette, 2005). Dans cette approche, plus un élément focal est proche du cadre de discernement, plus il représente l'ignorance totale et il sera donc pénalisé.  $\sqrt{x}$  est la fonction de récompense et  $x^2$  est la fonction de pénalisation.

La mesure définie par Jousselme fournit une similarité sur la « forme » des fonctions de masses. Elle ne tient pas compte de la similarité, de la distance qui peut exister entre les éléments de  $\Omega$ . Considérons l'exemple suivant sur  $\Omega = \{a; b; c\}$  avec deux valeurs d'attributs  $v_1$  et  $v_2$  données par des distributions de masse simples :  $v_1 = (m_1(\Omega) = 1)$  et  $v_2 = (m_2(\{b\}) = 1)$  :

$$\begin{aligned} \|\vec{m}_1\|^2 &= \|\vec{m}_1\|^2 = 1 \\ \langle m_1; m_2 \rangle &= 1 \cdot 1/3 \\ d_{bba} &= \sqrt{0,666} = 0,816 \end{aligned}$$

La similarité entre les distributions retournée par la formule est élevée et précise alors que la similarité des valeurs sous-tendues par ces distributions vaut 1 si  $v_1$  vaut b et 0 si  $v_1$  vaut a ou c (matrice de similarité identité). Cet exemple simple montre que nous ne pouvons pas utiliser cette mesure de similarité entre distributions comme mesure de similarité des valeurs d'attributs exprimés de manière incertaine. Nous proposons d'étendre la mesure ensembliste disjonctive pour prendre en compte une connaissance de similarité entre les éléments de  $\Omega$ . Dans l'exemple ci-dessus, si la matrice de similarité identité est considérée, la mesure que nous souhaitons développer doit retourner une mesure compatible avec celle retournée par la méthode basée sur les objets flous vue précédemment. C'est-à-dire, nous devons avoir au minimum, l'intervalle des valeurs possibles suivant :

$$0 \leq d_{bbasim}(m_1(\Omega)=1, m_2(\{b; c\})=1) \leq 1$$

En fait, la similarité disjonctive avec connaissance est une bien meilleure candidate pour le calcul qui nous intéresse puisque, si on considère une matrice de similarité identité (cas par défaut et celui de l'exemple), on obtient :

$$sim_{ensdis-AB} = \{(0; 2/3); (1; 1/3)\}$$

Dans la partie suivante, nous allons étendre cette mesure pour prendre en compte des distributions de masse de croyance complètes qui correspondent à plusieurs sous-ensembles (les ensembles focaux) alors que la mesure  $sim_{ensdis}$  permet seulement de comparer que deux ensembles.

### III.4.3. Similarité de distribution de croyance avec connaissance de similarité entre éléments

Pour pouvoir comparer les descripteurs incertains de l'expérience, nous avons besoin d'une mesure de similarité qui tienne compte à la fois des valeurs possibles et d'une matrice de similarité entre les éléments de l'ensemble considéré. Comme nous l'avons montré, les mesures classiques de distributions de masse ne sont pas bien adaptées car elles retournent une mesure précise qui traduit plutôt la « forme » des distributions alors que nous souhaitons avant tout comparer les valeurs réelles des attributs. La mesure  $sim_{ensdis}$  est à la base de celle que nous proposons car elle recense et calcule toutes les similarités possibles entre les éléments de deux sous-ensembles. Nous devons simplement étendre cette mesure pour ne plus comparer seulement deux mais autant de sous-ensembles que les distributions comparées en comptent. Pour réaliser cette mesure, nous procédons, comme pour la mesure de Jousselme, en calculant chaque couple possible de valeurs pour toutes les combinaisons possibles d'ensembles.

#### III.4.3.1. Algorithme de similarité locale de valeur exprimée par des fonctions de croyance

Soient deux informations exprimées par deux sources d'information  $S_1$  et  $S_2$  sur un même cadre de discernement  $\Omega$ , contenant n valeurs, noté  $\{a_1; \dots; a_i; \dots; a_n\}$ . Ces informations sont capturées au moyen de deux fonctions de masse notées  $m_1$  et  $m_2$  possédant respectivement les ensembles d'éléments focaux  $\{A_1; \dots; A_k\}$  et  $\{B_1; \dots; B_r\}$  (avec  $k \leq 2^n$  et  $r \leq 2^n$ ).

Soit une matrice de similarité symétrique  $S=[s_{ij}]$  associée au domaine  $\Omega$  telle que la diagonale soit égale à 1. Bien que nous utilisons le plus souvent des matrices de similarité symétriques, réflexives et même respectant l'inégalité triangulaire (voir les exemples), il n'y a pas vraiment de propriétés dont dépend l'algorithme. Un des avantages des propriétés de symétrie et de réflexivité ( $s_{ij} = s_{ji}$  et  $i=j \Rightarrow s_{ij} = 1$ ) est que la matrice de similarité de taille n n'est alors définie que par  $n(n-1)/2$  valeurs au lieu de  $n \times n$  ce qui facilite le travail de saisie des experts. Lorsque cette condition est remplie, il est aussi possible d'optimiser l'algorithme pour effectuer moins de recherches dans la matrice de similarité.

La mesure de similarité est exprimée dans l'intervalle [0 ; 1]. Afin de permettre un traitement analogue à celui proposé dans le cadre de comparaison des ensembles, et comme préconisé dans la section III.3.2., nous discrétisons cet intervalle pour définir des classes de similarité. Afin d'illustrer nos propositions, nous avons choisi de scinder arbitrairement l'intervalle [0 ; 1] en 21 intervalles de taille 0,05 centrés autour des valeurs décimales remarquables (0,1 ; 0,2 ; ...). Le premier intervalle et le dernier sont par conséquent de taille 0,025. Les intervalles sont donc :

$$1 : [0 \ 0,025[ ; 2 : [0,025 \ 0,075[ ; 3 : [0,075 \ 0,125[ ; \dots ; 20 : [0,925 \ 0,975[ ; 21 : [0,975 \ 1]$$

La mesure de similarité retournée par l'algorithme de similarité que nous présentons correspond alors à un vecteur à 21 composantes qui contient chaque valeur de similarité pour chaque intervalle. Cette taille est bien sûr variable et dépend d'une donnée d'entrée (le nombre des classes de discrétisation par exemple) que nous fixons dans un premier temps pour simplifier la présentation.

### Algorithme III-1 - Algorithme SIMBBA(m1, m2)

entrées :

$\Omega$  contenant n valeurs  $\{a_1 ; \dots ; a_i ; \dots ; a_n\}$ .

$m_1$  distribution de masse de croyance sur  $\Omega$  possédant k ensembles focaux :  $\{A_1 ; \dots ; A_k\}$  avec  $k \leq 2^n$ .

$m_2$  distribution de masse de croyance sur  $\Omega$  possédant r ensembles focaux :  $\{B_1 ; \dots ; B_r\}$  avec  $r \leq 2^n$ .

$S=[s_{ij}]$  matrice de similarité de taille  $n \times n$  associée aux éléments  $\{a_1 ; \dots ; a_i ; \dots ; a_n\}$  de  $\Omega$ .

sortie :

$SIM_{BBA} []$  = tableau de taille 21 correspondant aux intervalles dans l'ordre croissant et contenant la valeur de similarité associée.

$$1 : [0 \ 0,025[ ; 2 : [0,025 \ 0,075[ ; 3 : [0,075 \ 0,125[ ; \dots ; 20 : [0,925 \ 0,975[ ; 21 : [0,975 \ 1]$$

---


$$SIM_{BBA} [] = [0 ; 0]$$

Pour chaque ensemble focal  $A_i$  de  $m_1$

  Pour chaque ensemble focal  $B_i$  de  $m_2$

*listeTemp* ← liste vide

    Pour chaque élément  $x$  de  $A_i$

      Pour chaque élément  $y$  de  $B_i$

        Ajouter le couple  $(x ; y)$  dans *listeTemp*

      Fin pour

    Fin Pour

  Pour chaque couple  $(i ; j)$  de *listeTemp*

$$indiceTab \leftarrow \text{partieEntiere} \left( \frac{s_{ij} + 0,025}{0,05} \right)$$

$$SIM_{BBA} [indiceTab] \leftarrow SIM_{BBA} [indiceTab] + \frac{m_1(A_i) \cdot m_2(B_i)}{|A_i| \cdot |B_i|}$$

  Fin Pour

FinPour

FinPour

*récupération dans une  
liste temporaire de tous  
les couples de valeurs  
possibles*

*affectation de la part de  
similarité à chaque  
couple*

**Retourne**  $SIM_{BBA} []$

### Exemple d'application de l'algorithme

Pour illustrer le fonctionnement de ce premier algorithme, nous considérons l'exemple suivant sur  $\Omega = \{a, b, c\}$  avec les deux distributions de masses  $m_1$  et  $m_2$  et une matrice de similarité définie sur le Tableau III-2 :

$$BBA1 : m_1(\Omega) = m_{11} = 0,3 \quad m_1(\{b\}) = m_{12} = 0,7$$

$$\text{BBA2} : m_2(\Omega) = m_{21} = 0,1 \quad m_2(\{b ; c\}) = m_{22} = 0,9$$

A partir des deux fonctions de masse précédentes possédant chacune deux ensembles focaux, nous obtenons les 4 ensembles suivants avec le poids associé calculé par la multiplication des poids des ensembles d'origine. Nous utilisons le produit des masses de chaque ensemble, à l'instar des combinaisons conjonctives.

$\Omega \backslash \Omega$	<b>a</b>	<b>b</b>	<b>c</b>
<b>a</b>	1	0,6	0,1
<b>b</b>		1	0,8
<b>c</b>			1

Tableau III-2 - Matrice de similarité de l'exemple

<b>Ensembles</b>	<b>Poids associé</b>
{(a: a) (a ; b) (a ; c) (b ; a) (b ; b) (b ; c) (c ; a) (c ; b) (c ; c)}	$m_{11} * m_{21} = 0,03$
{(a ; b) (a ; c) (b ; b) (b ; c) (c ; b) (c ; c)}	$m_{11} * m_{22} = 0,27$
{(b ; a) (b ; b) (b ; c)}	$m_{12} * m_{21} = 0,07$
{(b ; b) (b ; c)}	$m_{12} * m_{22} = 0,63$
	$\Sigma = 1$

Tableau III-3 - Similarité ensembliste et connaissance - étape 1

Chaque couple correspond à une valeur de similarité (utilisation de la matrice de similarité). La matrice de similarité étant symétrique, il y a égalité entre les valeurs symétriques  $(i, j) = (j, i)$  et l'égalité sur la diagonale quand  $i = j$ . En remplaçant par les valeurs de la matrice de similarité de l'exemple puis en regroupant par valeur de similarité (0,1 ; 0,6 ; 0,8 ; 1), nous obtenons :

<b>Regroupement par niveau de similarité et par ensembles</b>	<b>Poids associé</b>
$m_{\text{sym}}(0,1) = 2/9 ; m_{\text{sym}}(0,6) = 2/9 ; m_{\text{sym}}(0,8) = 2/9 ; m_{\text{sym}}(1) = 1/3$	$m_{11} \cdot m_{21} = 0,03$
$m_{\text{sym}}(0,1) = 1/6 ; m_{\text{sym}}(0,6) = 1/6 ; m_{\text{sym}}(0,8) = 1/3 ; m_{\text{sym}}(1) = 1/3$	$m_{11} \cdot m_{22} = 0,27$
$m_{\text{sym}}(0,6) = 1/3 ; m_{\text{sym}}(0,8) = 1/3 ; m_{\text{sym}}(1) = 1/3$	$m_{12} \cdot m_{21} = 0,07$
$m_{\text{sym}}(0,8) = 1/2 ; m_{\text{sym}}(1) = 1/2$	$m_{12} \cdot m_{22} = 0,63$
	$\Sigma = 1$

Tableau III-4 - Similarité ensembliste et connaissance - étape 2

Nous obtenons ainsi autant de mesures de similarité que de combinaisons entre les ensembles focaux. Pour calculer la similarité globale, nous collectons les contributions de chacune des combinaisons sur chacun des intervalles du cadre de discernement des mesures de similarités, ainsi :

<b>Regroupement par niveau de similarité</b>	<b>Poids associé</b>
$m_{\text{sym}}(0,1) = 2/9 * 0,03 + 1/6 * 0,27$	$= 0,052$
$m_{\text{sym}}(0,6) = 2/9 * 0,03 + 1/6 * 0,27 + 1/3 * 0,07$	$= 0,075$
$m_{\text{sym}}(0,8) = 2/9 * 0,03 + 1/3 * 0,27 + 1/3 * 0,07 + 1/2 * 0,63$	$= 0,435$
$m_{\text{sym}}(1) = 1/3 * 0,03 + 1/3 * 0,27 + 1/3 * 0,07 + 1/2 * 0,63$	$= 0,438$
	$\Sigma = 1$

Tableau III-5 - Similarité ensembliste et connaissance - étape 3

Ainsi, l'algorithme présenté retournerait alors le tableau  $\text{SIM}_{\text{BBA}} [ ]$  suivant :

$$\text{SIM}_{\text{BBA}} [ ] = [0 ; 0 ; 0,052 ; 0 ; 0 ; 0 ; 0 ; 0 ; 0 ; 0 ; 0 ; 0 ; 0 ; 0,075 ; 0 ; 0 ; 0 ; 0,435 ; 0 ; 0 ; 0 ; 0,438]$$

La mesure de similarité appartient à l'ensemble {0,1 0,6 0,8 1}, la valeur moyenne est de 0,8362 et les « chances » d'avoir 0,1 et 0,6 sont faibles comparées aux chances d'avoir 0,8 ou 1. A partir d'une information précise donnant une connaissance de distance entre les éléments du domaine et d'une valuation incertaine par des distributions de masse de croyance, nous avons extrapolé une distribution de masse bayésienne sur un nouveau référentiel qui correspond à l'ensemble discrétisé des valeurs de similarité (intervalles entre 0 et 1). Cette transformation, à l'instar de la transformation pignistique,

intervient au moment de la prise de décision. Si les valeurs des attributs changent, il n'est pas possible de mettre à jour cette mesure de similarité autrement qu'en relançant son calcul. La distribution bayésienne ne permet pas de transférer une masse d'un ensemble sur un autre puisque tous les ensembles sont des singletons.

Nous notons la mesure de similarité déduite :

$$sim_{ensdis-AB} = \{ (0.1, 0.052), (0.6, 0.075), (0.8, 0.432), (1, 0.438) \}$$

La représentation graphique de  $sim_{ensdis-AB}$  est la suivante :

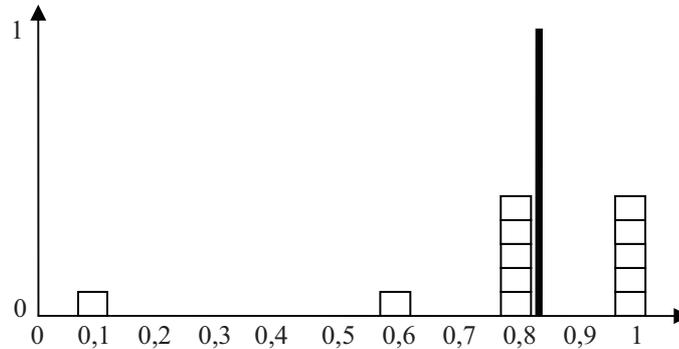


Figure III-7 - Représentation graphique de la mesure de similarité calculée pour l'exemple

L'algorithme développé permet ainsi de retourner une information plus riche que par exemple l'algorithme basé sur la théorie des possibilités présenté dans la partie III.4.1. En effet, nous avons une information supplémentaire quand à la répartition des valeurs possibles alors que l'algorithme basé sur la théorie des possibilités ne retient que les résultats min et max, ce qui donnerait ici :

$$\Pi=1 \text{ et } N=0,1 \text{ soit } \{0,1 ; 1\}.$$

## III.5. Evaluation de la similarité d'expériences

Les descripteurs de l'expérience ont une structure proche de la structure attributs-valeurs classique étendue par des attributs de type objet. Le typage objet correspond principalement à des compositions récursives d'attributs. Il ne reprend pas tous les aspects du paradigme orienté objet et en particulier nous avons choisi de ne pas considérer l'héritage. C'est pourquoi nous adoptons une vue locale-globale pour calculer la similarité de l'expérience. Les algorithmes proposés sont récursifs et basés sur le même référentiel que pour la mesure de similarité locale présentée dans la partie III.4.3. Nous allons, notamment dans cette partie, présenter différentes techniques d'agrégation de ces similarités locales.

Bien que les descripteurs simples puissent être de type différents (numérique, booléen, liste de valeur, taxonomie...), nous avons choisi de ne considérer que les listes de valeurs et de ramener chaque type à celui-ci par discrétisation lorsque c'est nécessaire. Nous avons fait ce choix pour présenter la méthode de manière uniforme, ce qui nous permet d'utiliser des similarités matricielles pour les domaines des descripteurs. Ce choix pourra facilement être remis en cause par la suite puisqu'il suffira de remplacer la fonction de similarité par une fonction de similarité locale classique vue dans la partie III.2 (seuil, ...) et de la paramétrer pour l'adapter au domaine du descripteur considéré. Ce changement ne pose pas de problème puisque toutes les méthodes de similarité retenues répondent toutes à la même spécification d'interface : retourner le degré de similarité entre deux éléments du domaine.

Dans la suite de cette partie nous commençons par la présentation du principe d'agrégation que nous avons retenu et basé sur l'agrégation de tous les vecteurs de similarité possibles. Le principe de cette mesure est le même que pour la similarité locale sauf que nous n'avons que des éléments focaux précis (distributions bayésiennes). Nous proposons une construction incrémentale du résultat permettant de limiter l'explosion combinatoire. Puis, nous présentons les différents algorithmes qui nous servent à faire une comparaison globale en considérant seulement le contexte de chaque expérience, ou en considérant les analyses associées à chacune des expériences. Nous proposons alors une évolution de la

mesure d'agrégation dans ce cas pour mieux traduire la sémantique que nous conférons à l'analyse (« et » fort).

### III.5.1. Agrégation de descripteurs simples

Nous considérons le cas de l'agrégation de deux descripteurs simples. Sur cette base, nous proposons une solution pour le cas plus général faisant intervenir  $n$  descripteurs simples. Nous étudions ensuite l'extension de la proposition à une structure objet, c'est-à-dire une structure dans laquelle certains descripteurs sont eux-mêmes composés de descripteurs.

#### III.5.1.1. Méthode d'agrégation à deux descripteurs

Pour exposer la méthode que nous proposons, nous considérons le type d'expérience  $\text{Exp}$  contenant deux descripteurs simples  $d_1$  et  $d_2$ . Soient  $\text{exp}(v_1, v_2)$  et  $\text{exp}(v'_1, v'_2)$  les deux expériences, de type  $\text{Exp}$ , à comparer.  $v_i$  correspond à une distribution de masse non bayésienne. Nous donnons sur la Figure III-8 un exemple représentant un vecteur de similarité  $\text{SIM}_{\text{BBA}}$  issu d'une comparaison descripteur à descripteur.

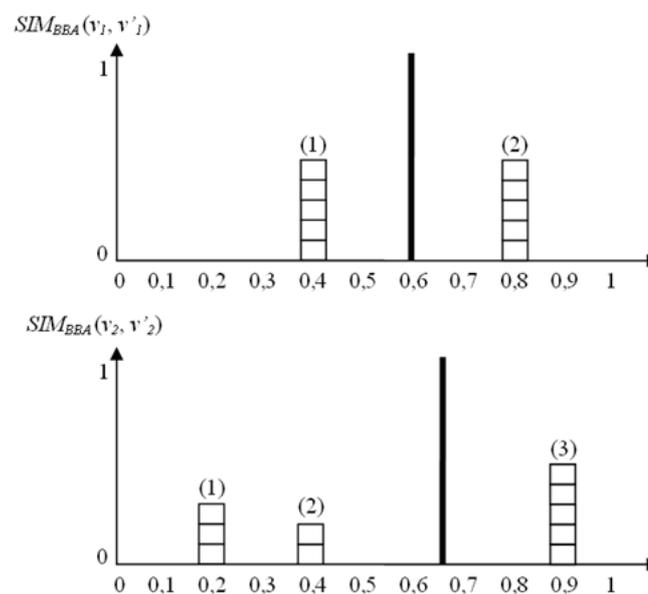


Figure III-8 - Vecteur de similarité pour deux descripteurs simples

Une première manière de réaliser l'agrégation consiste à faire la moyenne des poids niveau par niveau. Toutefois, cette méthode occulte une part importante de la sémantique que nous avons apportée avec la mesure de similarité de fonctions de masse. En effet, si l'on considère  $\text{SIM}_{\text{BBA}}(v_1, v'_1)$ , la sémantique de cette mesure signifie que la similarité entre  $v_1$  et  $v'_1$  de  $d_1$  est soit 0,4, soit 0,8 et que les combinaisons de valeurs possibles de l'information donnent autant de fois une similarité de 0,4 que 0,8. Dans l'exemple, cela correspond à un poids identique de 0,5. Etant donné que le calcul de la similarité traduit une prise de décision, nous avons opté pour une répartition prenant en compte les différentes combinaisons possibles en répartissant uniformément le poids unitaire.

Ainsi, en considérant l'exemple donné sur la Figure III-8, nous constatons que six combinaisons sont possibles, c'est-à-dire six valeurs envisageables de similarité globale entre  $V$  et  $V'$ . Puisque ces vecteurs de similarité ne sont composés que de deux attributs, les similarités envisageables peuvent être représentées par un vecteur dans le plan. Sur la Figure III-9, sont illustrés les six cas possibles de similarité. Pour chaque cas,  $C_{(i)(j)}$  représente le poids équivalent calculé sans préférence utilisateur.

En abscisse et en ordonnée nous plaçons respectivement les valeurs de similarité locale de  $s_1$  et les valeurs de similarité locale de  $s_2$ . Le vecteur résultant en trait plein correspond à la mesure globale qu'il faut quantifier par une fonction d'agrégation  $\phi$ . La similarité maximum est obtenue pour le vecteur

unité représenté en pointillé, c'est-à-dire pour  $s_1=s_2=1$ . De manière à déterminer le poids résultant, et de façon analogue à celle utilisée pour les opérations de combinaison des fonctions de croyance, nous multiplions les poids respectifs de chaque source.

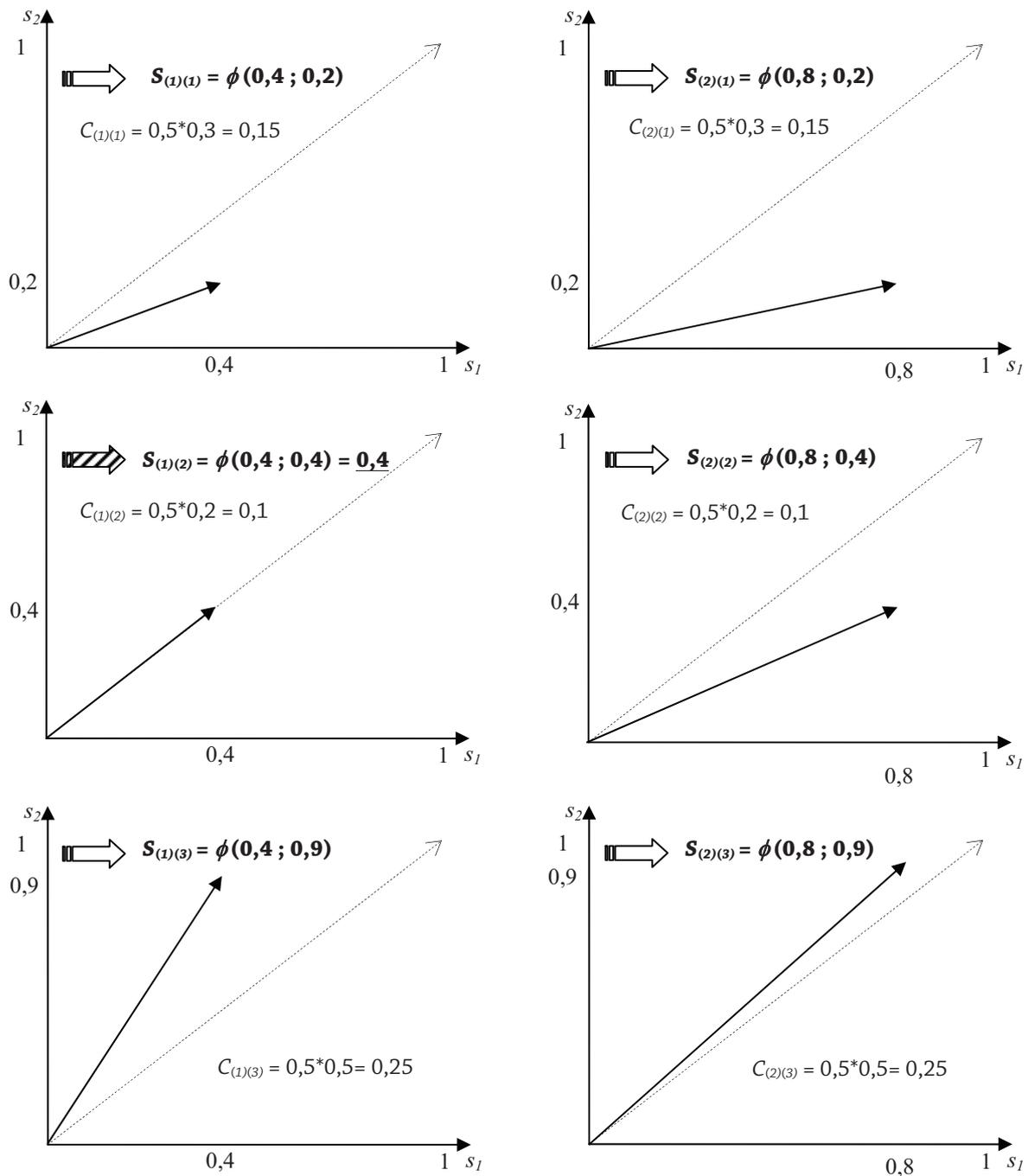


Figure III-9 - Illustration des 6 cas de similarités globale pour l'exemple

De manière générale, pour chacune des situations possibles extraites (six situations pour l'exemple), nous positionnons chaque contribution sur le référentiel en ajoutant le produit des poids associés à chacune des deux mesures individuelles. Ce poids résultant est positionné dans l'intervalle en abscisse correspondant à la valeur de similarité globale trouvée. Cette proportion est cumulée avec la valeur éventuellement présente précédemment. Par exemple, l'apport de  $S_{(1)(2)}$  dans la mesure  $SIM_{GLOBAL}$  est représenté sur la Figure III-10. En faisant l'hypothèse que la fonction d'agrégation  $\phi$  vérifie la propriété d'identité  $\phi(x,x)=x$ , nous obtenons :

- Agrégation  $\phi(0,4; 0,4)=0,4$
- Masse associée :  $s_1(\{0,4\}) \cdot s_2(\{0,4\})$  soit  $C_{(1)(2)} = 0,5 * 0,2 = 0,1$ . Donc, pour  $s_{(1)(2)}$ , on ajoute 0,1 à la valeur se trouvant à l'abscisse 0,4.

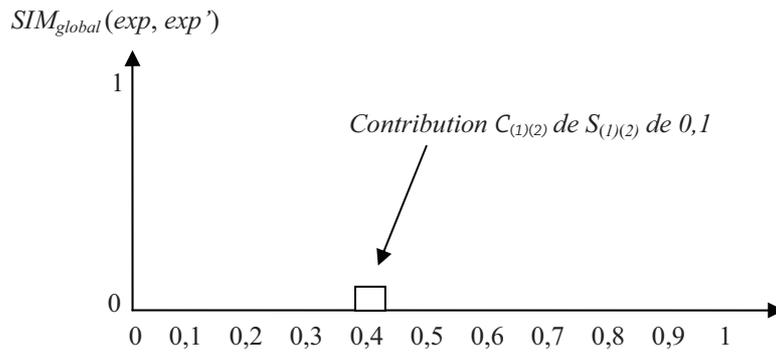


Figure III-10 - Contribution de  $S_{(1)(2)}$

Au-delà de ces calculs permettant l'agrégation des similarités locales pour construire une similarité globale, nous pensons qu'il est important de pouvoir exprimer des préférences sur certains descripteurs considérés comme importants. Pour cela, nous proposons d'ajouter un mécanisme simple de pondération des descripteurs, les poids étant fournis par l'utilisateur en amont de la phase de recherche. Pour l'exemple, on peut supposer que les descripteurs  $d_1$  et  $d_2$  possèdent respectivement des poids  $w_1$  et  $w_2$ . La masse associée sera alors calculée selon la formule  $w_1 * w_2 * s_{(i)(j)}$ . Enfin, précision que la définition d'une fonction d'agrégation  $\phi$  appropriée est discutée dans la section III.5.1.3.

### III.5.1.2. Généralisation de l'agrégation à n descripteurs

Nous proposons ici une généralisation du principe d'agrégation présenté dans la section précédente pour la prise en compte de  $n$  descripteurs. Cela signifie que l'on doit calculer toutes les combinaisons possibles, puis calculer la part de similarité associée afin de la positionner sur le référentiel de la mesure globale. Le problème de cette approche est l'explosion combinatoire qu'elle entraîne puisque les possibilités de mesure qu'il faut agréger par la fonction  $\phi$  dépendent du nombre d'éléments focaux de chaque mesure de similarité locale. Pour  $m$  intervalles de discrétisation de la similarité et pour  $n$  descripteurs du contexte courant, le nombre de combinaisons possibles est dans le pire des cas de l'ordre de  $m^n$ . Par exemple pour  $m=21$  et  $n=10$ , nous avons  $1,7 \cdot 10^{13}$  combinaisons par expérience.

Pour réduire la complexité exponentielle du calcul direct, nous proposons d'utiliser un calcul incrémental qui est basé sur le mécanisme suivant : nous calculons d'abord une similarité globale pour deux attributs et puis nous substituons la valeur résultant aux deux valeurs qui ont permis de la calculer. Pour  $n$  descripteurs, nous avons donc  $n-1$  opérations d'agrégation de deux valeurs de similarité à effectuer. Cette opération est possible si l'on dispose d'une méthode d'agrégation « décomposable », c'est-à-dire pour laquelle :

$$\phi(x_1; x_2) = x_{12} \text{ et } \phi(x_1; \dots; x_n) = \phi(x_{12}; x_{34}; \dots)$$

Nous montrons que cette décomposition est possible mais elle induit des manipulations pour pouvoir faire une retranscription du degré de similarité relatif à chaque étape. Par exemple, la fonction de Minkovski non normalisée est décomposable (cf. partie III.5.1.3).

Dans ce cas, la complexité est polynomiale et de l'ordre, de  $m^2(n-1)$ . Le principe de construction incrémentale est illustré sur la Figure III-11 où les étapes intermédiaires de calcul sont détaillées.

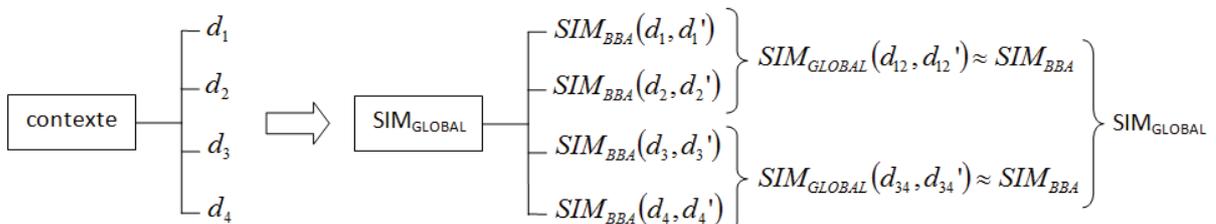


Figure III-11 - Principe de construction incrémentale

Pour  $m=21$  (la valeur que nous avons choisie) et  $n=10$ , nous avons  $21^2 \cdot (10-1)$ , soit 3969 valeurs à agréger. L'inconvénient de cette méthode est la perte d'information qui résulte de la discrétisation. Pour améliorer la précision des résultats, il est possible d'augmenter le nombre d'intervalles de discrétisation qui sert de référentiel commun pour le cadre de discernement. Par exemple, pour une précision au centième, nous devons choisir  $p=100$ , correspondant à  $100 \cdot 100 \cdot (n-1)$ , calculs ou encore 90 000 calculs pour 10 descripteurs. Cette valeur reste élevée mais est cependant acceptable sachant qu'un ordinateur classique aujourd'hui de type Athlon 64 X2 4600 peut réaliser 17400 MIPS (soit 14.7 gigaflops), c'est-à-dire  $14,7 \cdot 10^9$  opérations en virgules flottantes à la seconde. En effet, si on considère qu'un calcul de similarité nécessite 10 instructions en virgule flottantes, alors en une seconde il est possible de traiter 16 333 expériences de 10 descripteurs ( $p=100$ ) avec un Athlon 64 X2 4600. Pour la version exponentielle et  $p=21$  seulement, il faut 1 135 secondes pour traiter une expérience soit quasiment 20 minutes.

La solution itérative est donc largement préférable dans la mesure où la perte d'information due à l'itération peut être aisément compensée par une discrétisation plus précise de l'intervalle de similarité.

### III.5.1.3. Fonction d'agrégation de comparaison de contexte décomposable

Comme nous l'avons précisé dans la section précédente, afin de pouvoir utiliser un algorithme itératif et ainsi limiter la complexité exponentielle, il faut disposer d'une fonction décomposable telle que :

$$\phi(x_1; x_2) = x_{12} \text{ et } \phi(x_1; \dots; x_n) = \phi(x_{12}; x_{34}; \dots)$$

Nous avons choisi de baser la mesure sur une fonction de Minkowski puisque celle-ci est naturellement décomposable. Nous rappelons à cet effet que nous nous plaçons dans le cas où les différents descripteurs peuvent être pondérés (par des poids compris entre 0 et 1) pour signifier l'importance que l'utilisateur leur attribue dans la recherche. Pour normaliser la mesure, il faut diviser chaque poids par la somme des poids. Nous avons :

$$\phi_{Minkowski}(s_1, \dots, s_i, \dots, s_n) = \left( \frac{\sum_{i=1}^n w_i \cdot s_i^p}{\sum_{i=1}^n w_i} \right)^{1/p} = \frac{1}{\left( \sum_{i=1}^n w_i \right)^{1/p}} \cdot \left( \sum_{i=1}^n w_i \cdot s_i^p \right)^{1/p}$$

Nous définissons la fonction  $\phi_{Mink2}$  telle que :

$$\phi_{Mink2}(s_1, \dots, s_i, \dots, s_n) = \left( \sum_{i=1}^n w_i \cdot s_i^p \right)^{1/p}$$

L'intérêt de faire intervenir cette fonction est qu'elle est décomposable et nous permettra ainsi d'obtenir la récursivité pour le calcul de  $\phi_{Minkowski}$ . Ainsi, un algorithme récursif est obtenu car :

$$\begin{aligned} \phi_{Mink2}(s_1, \dots, s_i, \dots, s_n) &= \sqrt[p]{\left( \sqrt[p]{w_1 \cdot s_1^p + w_2 \cdot s_2^p} \right)^p + w_3 \cdot s_3^p + \dots + w_n \cdot s_n^p} \\ \phi_{Mink2}(s_1, \dots, s_i, \dots, s_n) &= \sqrt[p]{\left( \phi_{Mink2}(s_1, s_2) \right)^p + w_3 \cdot s_3^p + \dots + w_n \cdot s_n^p} \end{aligned}$$

La fonction  $\phi_{Mink2}$  est une fonction de Minkowski non normalisée et décomposable. Elle peut donc être employée dans l'algorithme itératif de calcul de la similarité globale pour la comparaison des contextes. Le poids de préférence ( $w$ ) n'est pris en compte que lors de la première agrégation.

À titre d'exemple, nous considérons la comparaison de 5 descripteurs simples qui fournit les valeurs de similarités locales  $s_1, \dots, s_5$ . Le fonctionnement récursif de l'algorithme est schématisé sur la Figure III-12.

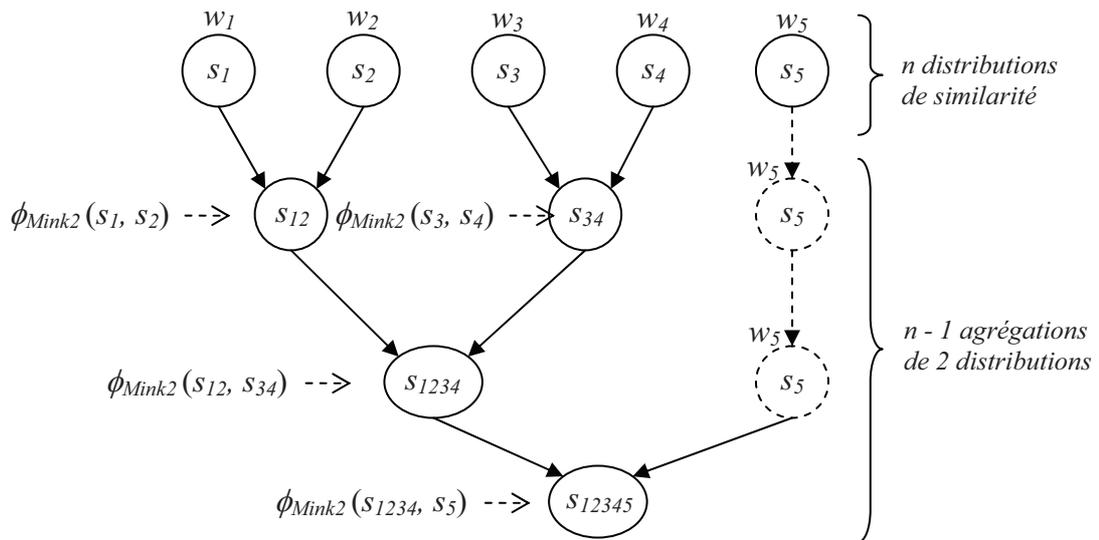


Figure III-12 - Principe d'agrégation itératif

L'ordre de prise en compte des valeurs n'a pas d'importance. Chaque valeur  $s_i$  étant une distribution bayésienne de similarité, nous obtenons  $s_{ij}$  par l'agrégation avec la fonction de Minkowski non-normalisée  $\phi_{Mink2}$  de  $s_i$  et de  $s_j$ . Dans chaque niveau de récursivité, l'algorithme en charge de classer l'agrégation prend en entrée un tableau de distributions de similarité et retourne, en sortie, une valeur qui correspond à l'agrégation complète malgré une certaine perte d'information qui est de l'ordre de grandeur de la taille des intervalles pour la fonction d'agrégation proposée. Nous présentons ci-dessous l'algorithme récursif d'agrégation des valeurs de similarité de contexte (cf. Algorithme III-2).

### III.5.2. Généralisation par récursivité pour les descripteurs objet

L'algorithme précédent permet d'agréger un ensemble de distributions de similarité. Chacune de ces distributions peut provenir de la comparaison de descripteurs simples mais aussi objets. C'est pourquoi, il nous faut maintenant à considérer comment associer une distribution de similarité à la comparaison de deux descripteurs objets. Ces derniers sont composés de descripteurs simples et/ou objets. Un tel contexte d'expérience est illustré, sur la Figure III-13, composé de 4 descripteurs dont un est objet. Ce descripteur objet est lui-même composé de 3 descripteurs dont un objet qui contient à son tour deux descripteurs simples. Les poids de préférence affecté à chaque attribut sont également notés sur la Figure III-13.

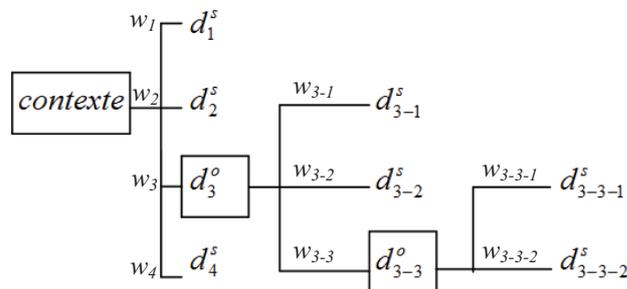


Figure III-13 - Exemple d'expérience contenant des descripteurs objets

**Algorithme III-2 - CALCUL<sub>AGREGATION-Mink</sub>(tableau  $V_{BBA}[]$ )**entrées

$V_{BBA}[]_n$  : le tableau contenant les n associations formées de distribution bayésienne de similarités à agréger, associées à un poids de préférence. Ces valeurs de similarité sont relatives à la comparaison 2 à 2 de descripteurs.

$w_i$  : un poids de préférence variant entre 0 et 1 (1 par défaut)

p : le paramètre de la fonction de Minkowski

Resultat $V_{BBA}[]_{(n/2+n/2)}$  : un tableau de taille n/2 si n pair ou n/2 + 1 si n impair

sortie

$SIM_{Global}[]$  : tableau de taille m + 1 correspondant aux intervalles dans l'ordre croissant et contenant la valeur de similarité associée.

$$1 : [0 ; 1/2m[, 2 : [1/2m ; 3/2m[, \dots, m : [1-1/2m ; 1]$$

$SIM_{temp}[] = [0, \dots, 0]_{m+1}$

n ← Taille de  $V_{BBA}[]$

Si n < 1

Condition d'arrêt de l'algorithme récursif

**Retourner** Clé de ( $V_{BBA}[1]$ )

Sinon

initialisation du tableau résultat suivant la parité de n - // est l'opération modulo

Si n//2 = 0

①

Resultat $V_{BBA}[]$  ← Nouveau Tableau de taille n/2

Sinon

Resultat $V_{BBA}[]$  ← Nouveau Tableau de taille n/2 + 1

Resultat $V_{BBA}[n/2 + 1]$  ← Clé de ( $V_{BBA}[n]$ )

remplissage résultat itération x

Pour chaque i variant de 1 à partieEntiere(n/2)

②

SimDistA[] ← Clé de ( $V_{BBA}[2*i-1]$ )

associations

SimDistB[] ← Clé de ( $V_{BBA}[2*i]$ )

$Sim_i[]_{m+1} \rightarrow w_i$

$w_A$  = Valeur de ( $V_{BBA}[2*i-1]$ )

$w_B$  = Valeur de ( $V_{BBA}[2*i]$ )

Pour j variant de 1 à m + 1

j correspond à l'intervalle de SIM, soit le degré (fonction de m)

masse<sub>1</sub> ← SimDistA[j]

Si masse<sub>1</sub> ≠ 0

Pour k variant de 1 à m + 1

masse<sub>2</sub> ← SimDistB[k]

Si masse<sub>2</sub> ≠ 0

Alors

ensemble focal i ou k correspond à la valeur de similarité s (centre de l'intervalle)

$$s_1 \leftarrow \frac{1}{m}(j-1) \text{ et } s_2 \leftarrow \frac{1}{m}(k-1)$$

$$\phi_{Mink2}(s_1, s_2) \leftarrow (w_A \cdot s_1^p + w_B \cdot s_2^p)^{\frac{1}{p}}$$

$$s_{AB} \leftarrow \phi_{Minkowski}(s_1, s_2) \leftarrow \frac{\phi_{Mink2}(s_1, s_2)}{(w_A + w_B)^{\frac{1}{p}}}$$

$$indSim \leftarrow 1 + partieEntiere\left(m \cdot s_{AB} - \frac{1}{2}\right)$$

$SIM_{temp}[indSim] \leftarrow SIM_{temp}[indSim] + masse_1 \cdot masse_2$

Fin Si

Fin Pour

fin de la première boucle d'itération sur les ensembles focaux

Fin Si

Fin Pour

Resultat $V_{BBA}[i]$  ← ( $SIM_{temp}[] \rightarrow (w_A + w_B)$ )

fin de la deuxième boucle d'itération sur les ensembles focaux

Fin Pour

**CALCUL<sub>AGREGATION-Mink</sub>(Resultat $V_{BBA}[]$ )**

③

Fin Si

appel récursif - itération x+1

Fin Si

Le contexte est donc une structure hiérarchique de descripteurs, les feuilles étant des descripteurs simples et les nœuds des descripteurs objet. Le contexte peut finalement être considéré comme un descripteur objet particulier (racine) et donc la similarité globale est simplement retrouvée en appliquant la fonction d'agrégation de manière récursive. Pour l'exemple précédent, nous donnons l'enchaînement des calculs :

Contexte est équivalent à un descripteur objet, appel de  $SIM_{GLOBAL}$

- 1) Calcul de la similarité locale  $SIM_{BBA}$  entre  $d^s_1$  et son équivalent dans une expérience.
- 2) Calcul de la similarité locale  $SIM_{BBA}$  entre  $d^s_2$  et son équivalent.
- 3)  $d^o_3$  est de type objet, appel de  $SIM_{GLOBAL}$ 
  - a) Calcul de la similarité locale  $SIM_{BBA}$  entre  $d^s_{3-1}$  et son équivalent.
  - b) Calcul de la similarité locale  $SIM_{BBA}$  entre  $d^s_{3-2}$  et son équivalent.
  - c)  $d^o_{3-3}$  est de type objet, appel de  $SIM_{GLOBAL}$ 
    - i) Calcul de la similarité locale  $SIM_{BBA}$  entre  $d^s_{3-3-1}$  et son équivalent.
    - ii) Calcul de la similarité locale  $SIM_{BBA}$  entre  $d^s_{3-3-1}$  et son équivalent.
    - iii) Agrégation des valeurs
  - d) Agrégation des valeurs
- 4) Calcul de la similarité locale  $SIM_{BBA}$  entre  $d^s_4$  et son équivalent.
- 5) Agrégation des valeurs

Le calcul de similarité présenté dépend de l'ordre des arguments de la fonction. La comparaison est donc guidée par la structure du contexte. Dans le formalisme retenu, la structure des expériences, les ensembles de descripteurs ne sont pas obligatoirement les mêmes. Soit les expériences ont la même structure, et on compare alors les descripteurs un à un, soit la structure est différente, et la recherche est alors effectuée depuis le contexte courant en balayant tous les descripteurs un à un et en cherchant le descripteur équivalent (représentant la même information). Pour chaque valeur, si le descripteur équivalent existe, la comparaison est effectuée et le processus continue.

Quand un descripteur du contexte est absent, nous pouvons envisager plusieurs façons de traiter le cas. Soit la similarité est nulle si l'information n'existe pas, soit elle existe et elle est considérée comme inconnue, c'est-à-dire totalement incertaine. Etant dans une phase décisionnelle, nous proposons de représenter une proposition totalement incertaine par une répartition uniforme de la masse unitaire sur les éléments du référentiel de similarité (modélisation analogue au principe d'équiprobabilité). Dans le cas où l'on cherche à comparer deux expériences  $e_1$  et  $e_2$  de la base indépendamment l'une de l'autre, nous suggérons de considérer les deux mesures  $SIM_{Global}(expérience\ e_1, expérience\ e_2)$  et  $SIM_{Global}(expérience\ e_2, expérience\ e_1)$ .

### III.5.3. Prise en compte des analyses

La différence entre une similarité de contexte telle que celle présentée dans les sections précédentes et la similarité propre au processus de retour d'expérience est principalement due à la présence des analyses expertes ajoutées à chaque expérience. Cette association attribue une nouvelle sémantique à l'information globale contenue dans l'expérience en précisant la/les conjonction(s) d'attributs significatifs contrairement au contexte qui ne représente qu'une disjonction d'attributs. La pertinence de l'étape de recherche, centrale dans le processus de retour d'expérience dépend en conséquence fortement de la capacité du mécanisme de recherche à intégrer l'analyse.

Le premier algorithme d'agrégation ne peut pas être utilisé directement dans la similarité d'expérience (avec analyses) car la sémantique qui lui est associée est celle d'une recherche basée sur la disjonction d'attributs. C'est pourquoi les analyses, qui pourraient être vues simplement comme un sous-ensemble de descripteurs de l'expérience, sont davantage qu'un simple filtre comme le sont les distributions de poids de préférence. L'élément sémantique important à modéliser est le « et », qui vient contraindre fortement la sémantique. Par exemple l'analyse suivante  $Analyse_i(v_{d1}, v_{d3})$  signifie que les valeurs des descripteurs 1 et 3 sont significatives aux yeux de l'expert. Autrement dit, sans l'une des valeurs présentes dans l'analyse, la similarité globale qui traduit l'intérêt de l'expérience pour le

contexte courant est nulle. La similarité sera maximum si tous les descripteurs présents dans l'analyse sont égaux à ceux présents dans l'expérience mémorisée. Il faut donc prévoir une mesure d'agrégation pénalisante en cas d'écart.

### III.5.3.1. Fonction d'agrégation prenant en compte les analyses

La fonction d'agrégation  $\phi$  doit vérifier la condition conjonctive suivante :  $\phi(0, x) = 0$ . En effet, cette condition conjonctive, qui implique qu'une seule similarité locale nulle entraîne une similarité globale nulle, provient de l'interprétation du « et » des analyses. Pour que la similarité globale soit différente de 0, il faut qu'au moins tous les descripteurs aient un degré de similarité non nul.

La fonction de Minkowski et ses dérivées (moyenne et max) sont mal adaptées car elles la présence d'une similarité locale nulle n'entraîne pas une similarité globale nulle. Les fonctions produit et minimum sont en revanche deux bons candidats puisqu'une similarité locale nulle entraîne la similarité globale nulle. Nous avons décidé de développer une mesure d'agrégation spécifique car la fonction min fait perdre trop d'information et la fonction produit rassemble trop les informations autour de 0.

Deux fonctions ont été étudiées, l'une basée sur l'interprétation géométrique et l'autre basée sur une variante du produit des similarités. Elles sont présentées dans les deux parties suivantes.

### III.5.3.2. Mesure d'agrégation géométrique

La première approche que nous avons envisagée est basée sur l'interprétation géométrique vectorielle de la fonction d'agrégation en utilisant la norme et l'angle entre le vecteur que l'on cherche à comparer et le vecteur correspondant à la similarité maximum. Nous notons cet angle  $\alpha$ . Nous souhaitons que la mesure, notée  $\phi_\alpha$ , respecte les mêmes conditions que celles définies précédemment. Nous souhaitons aussi retranscrire le « et » et pour cela nous proposons de considérer que plus le vecteur de similarité est éloigné du vecteur unité, plus la mesure globale est pénalisée. Sur le schéma de la Figure III-14 les trois vecteurs a, b, c sont de même longueur et  $\alpha_a < \alpha_b < \alpha_c = 45^\circ$ . Pour cela, il faut que  $SIM_{GLOBAL}(a) > SIM_{GLOBAL}(b) > SIM_{GLOBAL}(c)$ . De plus, nous souhaitons avoir  $SIM_{GLOBAL}(c) = 0$  puisque la similarité de la deuxième composante est nulle. La fonction  $\cos(2\alpha)$  est une fonction de pénalisation candidate. Nous avons en effet :

$$-45^\circ < \alpha < 45^\circ$$

$$\cos(-90^\circ) = \cos(90^\circ) = 0 < \cos(2\alpha) < \cos(0) = 1$$

Nous proposons donc comme mesure  $\phi_\alpha$  :

$$\phi_\alpha(s_1, s_2) = \phi_{Minkowski}(s_1, s_2) \cdot \cos(2\alpha)$$

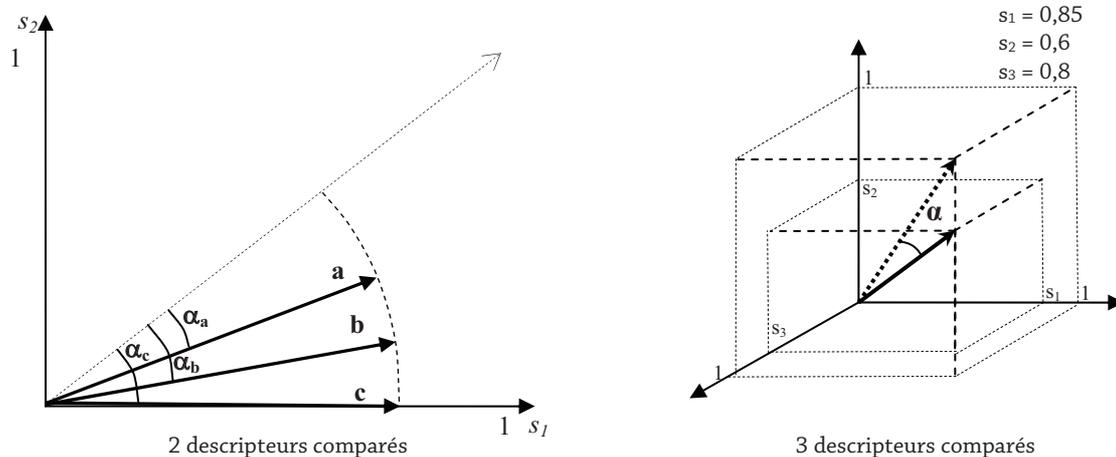


Figure III-14 - Interprétation géométrique de l'agrégation de 2 et 3 descripteurs

Pour pouvoir être utilisée, cette mesure doit être généralisée au cas de vecteurs à  $n$  composantes. La version généralisée consiste à calculer l'agrégation de Minkowski en utilisant la méthode de construction incrémentale proposée dans les sections précédentes puis à multiplier le résultat par la fonction de pénalisation. Cette fonction est proportionnelle à l'angle qui sépare le vecteur unité de dimension  $n$  du vecteur de similarité. Or cette valeur d'angle n'est pas toujours comprise entre  $-45^\circ$  et  $45^\circ$ . Par exemple, sur la Figure III-14, dans le cas d'agrégation de 3 descripteurs, l'angle peut dépasser la valeur de  $45^\circ$ . En fait, plus la dimension des vecteurs augmente, plus la valeur d'angle augmente. Pour une dimension  $n$ , l'angle alpha maximum est de :

$$\alpha_{\max} = \arccos\left(\frac{1}{\sqrt{n}}\right)$$

Nous ne pouvons directement multiplier  $\phi_{Minkowski}$  par une valeur proportionnelle à cette valeur de  $\alpha$  car nous obtenons un résultat nul seulement lorsque  $n-1$  composantes du vecteur de similarité sont nulles. Or nous souhaitons disposer d'une mesure de similarité nulle dès que l'une des similarités des composantes est nulle. En conséquence de quoi, cette approche ne correspond pas exactement à nos besoins.

Une alternative consiste à calculer puis agréger tous les angles  $\alpha$  en considérant les descripteurs par paires. Pour des raisons de performance, en relation avec la complexité de l'algorithme, nous souhaitons avoir une fonction d'agrégation décomposable. Ainsi, le principe est d'agréger les distributions de masse deux à deux comme dans l'algorithme vu précédemment (cf. Algorithme III-2) puis de pénaliser le résultat par la fonction cosinus. Nous n'avons, cependant, pas trouvé de fonction satisfaisante. Nous avons tenté d'associer à chaque paire l'angle  $\alpha$  associé. Cette approche s'est avérée mauvaise car les fonctions cosinus rendant impossible la multiplication de chaque similarité intermédiaire par le  $\cos(2\alpha)$  correspondant. En effet, dans ce cas, l'ordre de prise en compte des valeurs a une influence sur le résultat retourné ce qui n'est pas conforme à la sémantique du traitement.

Enfin, une dernière possibilité envisagée consiste à calculer tous les angles  $\alpha$  par projection du vecteur de similarité et du vecteur unité sur chaque plan de l'espace de dimension  $n$  ( $n$  étant le nombre de descripteurs comparés) puis à multiplier la mesure d'agrégation de contexte par le cosinus de 2 fois cet angle.

$$\phi_{\alpha}(s_i) = \phi_{Minkowski}(s_i) \cdot \cos(2 \cdot \max_{\alpha})$$

$$\text{avec } \max_{\alpha} = \max_{\substack{j=n; k=n \\ j=1; k=1; j < k}} \left( \arccos\left( \frac{s_j + s_k}{\sqrt{2 \cdot (s_j^2 + s_k^2)}} \right) \right)$$

Cette solution n'est pas entièrement satisfaisante non plus car elle nécessite de calculer tous les angles  $\alpha$  possibles et induit, à nouveau, un problème de complexité pour les algorithmes si les distributions de masses sont considérées comme valeurs de similarités.

Nous conservons malgré tout cette proposition, car dans le cas de la recherche avec analyses, en pratique le nombre de combinaisons possibles n'est pas très important. Quand les contextes seuls sont comparés, il faut comparer tous les attributs de l'expérience. Quand les analyses sont prises en considération, seul un sous-ensemble en général assez réduit des attributs importe et toutes similarités locales nulles entraînent une similarité globale nulle. Il est donc possible d'utiliser la dernière proposition basée sur la détermination de l'angle  $\alpha$  malgré la complexité sous-jacente. Néanmoins, il nous a semblé préférable de proposer une mesure différente basée sur une construction incrémentale et itérative comme cela est le cas pour le calcul de la similarité de contexte et cette mesure est introduite dans la partie suivante.

### III.5.3.3. Mesure d'agrégation conjonctive retenue

Pour traduire le « et », c'est-à-dire la conjonction de descripteurs significatifs, les deux fonctions habituellement utilisées sont le « produit » et le « minimum ». L'inconvénient majeur de la fonction « minimum » est qu'elle ne tient pas compte des contributions de l'ensemble des similarités or nous préférons traduire l'influence relative de chaque attribut. Par exemple, afin d'ordonner les résultats, nous souhaitons que :

$$\phi(0,2, 0,2)=0,2 < \phi(0,2, 0,8) < \phi(0,8, 0,8)=0,8$$

ou encore  $\phi(0,2, 0,2, 0,2) = 0,2 < \phi(0,2, 0,8, 0,8) < \phi(0,2, 0,8, 1)$

La fonction « produit » traduit mieux l'influence de chaque argument et mais ne vérifie pas la propriété d'identité puisque :

$$\phi(x, x)=x^2 \text{ ou par exemple } \phi(0,2, 0,2)=0,04 \neq 0.2$$

De plus, étant donné que toutes les valeurs de similarités sont inférieures à 1, le produit est une fonction très pénalisante. Au bout de quelques itérations les valeurs ont tendance à être regroupées autour des faibles valeurs (inférieures à 0,1) ce qui rend difficile l'exploitation d'un référentiel discrétisé.

En conséquence, afin d'avoir une fonction qui réponde à toutes nos contraintes, nous proposons d'utiliser une fonction basée sur le produit mais qui vérifie la propriété d'identité par l'utilisation de la fonction racine. Pour décrire cette fonction, nous utilisons un paramètre  $n$  qui correspond au nombre de valeurs agrégées retournées par la mesure. La fonction proposée, notée  $\phi_{ET}^n$ , est donnée par la formule suivante :

$$\phi_{ET}^n(s_1, \dots, s_i, \dots, s_n) = \left( \prod_{i=1}^n s_i \right)^{\frac{1}{n}}$$

Cette fonction présente tous les avantages de la fonction produit tout en préservant l'identité ( $\phi(x, x)=x$ ). De plus elle est décomposable et peut ainsi être utilisée dans un algorithme récursif car :

$$\begin{aligned} \phi_{ET}^n(s_1, \dots, s_i, \dots, s_n) &= \left( \prod_{i=1}^{n-1} s_i \cdot s_n \right)^{\frac{1}{n}} \\ \phi_{ET}^n(s_1, \dots, s_i, \dots, s_n) &= \left[ \left( \prod_{i=1}^{n-1} s_i \right)^{\frac{1}{n-1}} \right]^{n-1} \cdot s_n \Bigg)^{\frac{1}{n}} \\ \phi_{ET}^n(s_1, \dots, s_i, \dots, s_n) &= \left( \left( \phi_{ET}^{n-1}(s_1, \dots, s_i, \dots, s_{n-1}) \right)^{n-1} \cdot s_n \right)^{\frac{1}{n}} \end{aligned}$$

L'inconvénient dans une approche récursive de fonction utilisant le produit de valeurs est que la précision ne sera pas aussi bonne que pour une agrégation par sommation comme c'est le cas pour la similarité de contextes. Le produit, contrairement à la somme, accentue les écarts. Il faudra par conséquent utiliser des intervalles de discrétisation plus petits que pour l'agrégation de Minkowski. L'algorithme que nous proposons pour réaliser l'agrégation sur cette base est donné ci-après.

Dans cet algorithme, les poids de préférence ne sont pas utilisés dans la recherche portant sur les analyses. En effet, ceux-ci n'auraient aucune utilité puisque c'est l'analyse qui vient biaiser la recherche et non les poids de préférence comme pour la recherche simple, basée sur le contexte.

**Algorithme III-3 - CALCUL<sub>AGREGATION-ET</sub>(tableau  $V_{BBA}[]$ )**entrées :

$V_{BBA}[]_n$  la tableau contenant les  $n$  associations formées de distribution bayésienne de similarités à agréger associé à un poids de préférence. Ces valeurs de similarités sont relatives à la comparaison 2 à 2 de descripteurs.

$V_{BBA}[i]$  = Association $BBA_i$  = ( $Sim_i []_{m+1} \rightarrow p_i$ ) une association entre une distribution de similarité et du poids associé.  $Sim_i []_{m+1}$  est la clé de l'association  $V_{BBA}[i]$  et  $p_i$  sa valeur

$Sim_i []_{m+1}$  une distribution bayésienne de similarité de taille  $m+1$  (discrétisation de l'intervalle unité.

$p_i$  un entier correspondant au nombre de distribution agrégée pour la mesure  $BBA_i$ )

**Resultat**  $V_{BBA}[]_{(n/2+n//2)}$  Tableau de taille  $n/2$  si  $n$  pair ou  $n/2 + 1$  si  $n$  impair

sortie:

$SIM_{Global} []$  = tableau de taille  $m + 1$  ( $m = 10, 21$  ou  $100$  par exemple) correspondant aux intervalles dans l'ordre croissant et contenant la valeur de similarité associée.

1 :  $[0 ; 1/2m[$ , 2 :  $[1/2m ; 3/2m[$ , ...,  $m$  :  $[1-1/2m ; 1]$

$SIM_{temp} [] = [0, \dots, 0]_{m+1}$

$n \leftarrow$  Taille de  $V_{BBA}[]$

Si  $n = 1$

**Retourner** Clé de  $V_{BBA}[1]$

Sinon

Si  $n//2 = 0$

**Resultat**  $V_{BBA}[] \leftarrow$  Nouveau Tableau de taille  $n/2$

Sinon

**Resultat**  $V_{BBA}[] \leftarrow$  Nouveau Tableau de taille  $n/2 + 1$

**Resultat**  $V_{BBA}[n/2 + 1] \leftarrow$  Clé de  $V_{BBA}[n]$

Pour chaque  $i$  variant de 1 à  $partieEntiere(n/2)$

**SimDistA**  $[] \leftarrow$  Clé de  $V_{BBA}[2*i-1]$

**SimDistB**  $[] \leftarrow$  Clé de  $V_{BBA}[2*i]$

$p_A =$  Valeur de  $V_{BBA}[2*i-1]$

$p_B =$  Valeur de  $V_{BBA}[2*i]$

Pour  $j$  variant de 1 à  $m + 1$

$masse_1 \leftarrow$  **SimDistA**  $[j]$

Si  $masse_1 \neq 0$

Pour  $k$  variant de 1 à  $m + 1$

$masse_2 \leftarrow$  **SimDistB**  $[k]$

Si  $masse_2 \neq 0$

$$s_1 \leftarrow \frac{1}{m}(j-1)$$

$$s_2 \leftarrow \frac{1}{m}(k-1)$$

$$s_{AB} \leftarrow \phi_{ET}(s_1, s_2) \leftarrow (s_1^{p_a} \cdot s_2^{p_b})^{\frac{1}{p_a+p_b}}$$

$$indSim \leftarrow 1 + partieEntiere\left(m \cdot s_{AB} - \frac{1}{2}\right)$$

$SIM_{temp} [indSim] \leftarrow SIM_{temp} [indSim] + masse_1 \cdot masse_2$

Fin Si

Fin Pour

Fin Si

Fin Pour

**Resultat**  $V_{BBA}[i] \leftarrow (SIM_{temp} [] \rightarrow (w_A + w_B))$

Fin Pour

**CALCUL<sub>AGREGATION-ET</sub>(Resultat  $V_{BBA}[]$ )**

Fin Si

Condition d'arrêt de l'algorithme récursif

initialisation du tableau résultat suivant la parité de  $n - //$  est l'opération modulo

remplissage résultat itération  $x$

associations

$Sim_i []_{m+1} \rightarrow w_i$

$j$  correspond à l'intervalle de  $SIM$ , soit le degré (fonction de  $m$ )

ensemble focal  $i$  ou  $k$  correspond à la valeur de similarité  $s$  (centre de l'intervalle)

fin de la première boucle d'itération sur les ensembles focaux

nouvelle association

fin de la deuxième boucle d'itération sur les ensembles focaux

appel récursif - itération  $x+1$

## III.6. Synthèse

Nous avons défini dans ce chapitre les mécanismes élémentaires d'exploitation de la base d'expérience. Ces mécanismes sont décrits par des algorithmes de similarité qui ont été adaptés à la nature objet incertain du formalisme de représentation des expériences. Il découle de ces fonctionnalités deux cas d'utilisation majeurs :

- Une recherche générique paramétrable par des poids de préférence qui permet d'effectuer des requêtes variées sur la base d'expérience (*cf.* Algorithme III-2)
- Une recherche biaisée par l'analyse qui est effectuée par les experts sur les expériences en phase de capitalisation. Cette analyse, nous le rappelons, agit comme un filtre écartant l'information non significative. Ainsi, cette étape de recherche définie par l'Algorithme III-3 permet, en tenant compte de la sémantique de l'analyse, de retrouver les expériences réellement significatives pour un contexte donné. Selon nous, c'est cette étape qui matérialise le processus de retour d'expérience cognitif.

Dans le chapitre suivant, nous donnons un exemple particulier d'utilisation de la base d'expériences fondée sur l'Algorithme III-3 qui vise à extraire une connaissance particulière, à savoir un indicateur du risque par l'expérience.

# Chapitre IV

## Extraction de connaissance

### Indication du risque

L'objectif de ce chapitre est de proposer un mécanisme spécifique d'extraction de connaissance utile dans le contexte de la prévention des risques. Nous illustrons ainsi le principe énoncé qui présente l'expérience comme un vecteur potentiel de création de connaissance.

Ce travail de thèse a été réalisé dans le contexte d'un projet européen pour lequel nous avons spécifié et développé une application de retour d'expérience dédiée à la prévention des risques liés à la pratique d'activités en montagne (cf. partie V.5). Nous avons choisi d'étudier la possibilité d'élaborer, à partir des expériences capitalisées, une mesure ou plutôt un indicateur de risque pour les utilisateurs qui prévoient une activité en montagne et qui la décrivent dans le logiciel.

L'idée sous-jacente est assez intuitive : A partir de la description du contexte courant, si dans des conditions similaires il y a déjà eu des accidents alors l'utilisateur est averti et ainsi il renforce sa vigilance. L'application ou plutôt l'algorithme de recherche présenté dans le chapitre III apporte déjà, dans une certaine mesure, des éléments de réponse à la prévention du risque en présentant des expériences significatives, à un moment donné, en fonction d'un contexte particulier. Néanmoins, nous avons tenté d'aller plus loin en créant un indicateur de risque qui agrège et interprète la connaissance implicite de la base pour informer de manière plus complète le ou les utilisateurs.

Avant de proposer un axe de développement d'un indicateur de risque basé sur l'architecture de retour d'expérience, nous définissons plus précisément la notion de risque et présentons les référentiels de risques. Nous introduisons ensuite quelques méthodes et outils industriels de prévention et gestion des risques. Ensuite, nous montrons le lien et les rapprochements proposés entre risque et retour d'expérience.

## IV.1. Définition générale du risque

Le risque est généralement perçu comme la conjonction d'un aléa et d'une vulnérabilité. En d'autres termes, le risque est le produit des chances d'occurrence d'un événement dommageable par le degré de gravité pressenti associé à cet événement et correspondant généralement aux pertes potentielles (Favre *et al.*, 1998). Une avalanche dans un site de montagne reculé où il n'y a aucune habitation et aucune fréquentation représente un risque quasi-nul par exemple.

### IV.1.1. Formulation générale du risque

Le risque est souvent perçu comme la probabilité de l'occurrence d'un événement non désiré, et le degré de gravité des pertes potentielles induites. Le danger peut être considéré comme un état et le risque comme sa mesure (Bourrelier *et al.*, 2000). De manière très générale, le risque correspond aux définitions suivantes :

- Situation potentielle non souhaitée ayant des conséquences négatives résultant de la survenue d'un ou plusieurs événements dont l'occurrence est incertaine,
- Tout événement redouté qui réduit l'espérance de gain et/ou d'efficacité dans une activité humaine.

L'évaluation du risque est souvent donnée par la formule suivante, où l'aléa représente les chances d'occurrence d'un événement et la vulnérabilité les pertes potentielles associées :

$$\text{Risque} = \text{Aléa} \cdot \text{Vulnérabilité}$$

Cette définition constitue une simplification car le risque est une notion bien plus complexe. Par exemple, aléa et vulnérabilité sont rarement indépendants et souvent il existe des risques pour lesquels la victime est aussi responsable de l'événement. Une avalanche peut-être déclenchée par un skieur auquel cas, le skieur, en relation avec la vulnérabilité est aussi responsable de l'occurrence de l'événement tout en étant victime s'il a été emporté par celle-ci. C'est ce type de situation que la formule donnée ci-dessus ne permet pas de prendre en compte de manière pertinente. Néanmoins, cette formule reflète la vue communément admise du risque comme étant une relation entre fréquence et gravité.

Une définition plus précise du risque associé aux risques naturels est donnée par Asté (Asté, 2001). Dans cette définition, le risque est associé à des pertes, dommages, préjudices ou dysfonctionnements potentiels susceptibles d'être générés directement ou indirectement par un phénomène pressenti. La notion de risque exprime toujours la notion d'aléa et de vulnérabilité mais, cette fois-ci, la relation entre les deux est indéfinie comme le traduit la formulation « susceptibles d'être générés directement ou indirectement ». Il ne semble pas réaliste aux yeux de l'auteur de décider d'une mesure du risque qui correspondrait à une relation entre la probabilité d'occurrence d'un aléa et les mesures de vulnérabilité.

### IV.1.2. Vision causes conséquences

Une autre (mais complémentaire) vue classique du risque est liée à la perception causes/conséquences. Dans ce cadre, le risque est lié à l'occurrence d'un événement non souhaité aux conséquences dommageables suite à l'apparition d'évènements déclencheurs appelés « causes ». Le risque est alors perçu, d'un point de vue macroscopique, comme une combinaison causes/situation/conséquences et peut être représenté comme dans la Figure IV-1, à l'instar de Gouriveau dans ses travaux en rapport avec les risques industriels (Gouriveau, 2003).

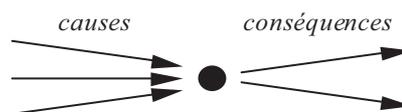


Figure IV-1 - Vue Causes - Conséquences des risques

L'avantage de cette formulation réside à la fois dans son caractère général et dans sa facilité de mise en œuvre dans les systèmes informatiques. En revanche, la caractérisation des causes et des conséquences reste souvent complexe. Dans de nombreux cas, la méconnaissance des phénomènes est telle qu'il n'est pas possible de déterminer et de quantifier avec certitude les différents facteurs entraînant un événement particulier. Il est donc assez rare de donner des mesures d'occurrence justes et précises d'un événement ou d'un phénomène.

### IV.1.3. Une modélisation imparfaite et subjective

Le risque peut aussi être considéré comme une donnée subjective, une construction sociale qui renvoie à des aspects purement sociologiques. Quantifier un niveau de risque est avant tout un travail de modélisation et donc d'approximation du réel. « Le risque est une notion floue puisque cela correspond précisément à ce que l'on ne connaît pas, ce que l'on ne peut qu'imaginer, de façon toujours trop subjective, en fonction de sa propre sensibilité ou fragilité » (Asté, 2001). En outre, un paradoxe

classique en sociologie du risque explique que l'acceptabilité humaine du risque est en réalité peu corrélée à l'intensité du risque. La perception du risque est caractérisée à la fois par une sensibilité exacerbée et par une apparente irrationalité (Asté, 2001). Ceci explique que nous redoutons davantage les catastrophes aériennes, industrielles, chimiques, naturelles par rapport aux accidents de la route et aux accidents domestiques qui sont pourtant plus fréquents. L'homme accepte plus facilement un risque choisi par rapport à un risque subi car un mécanisme de dénégation est mis en œuvre, s'appuyant sur la croyance erronée d'une capacité individuelle à maîtriser le risque (ANAES, 2003).

Par conséquent, prévenir les risques revient avant tout à faire prendre conscience de l'existence objective du risque. Naturellement, indiquer un niveau de risque s'avère souvent insuffisant. Toutefois, si cette information s'appuie sur des données réelles et expertisées (expériences) sa pertinence en sera renforcée et par voie de conséquence sa crédibilité auprès des utilisateurs sera augmentée. Il est important de préciser également que la mesure du risque, l'identification et la quantification ne sont pas suffisantes mais font partie d'un processus plus vaste souvent appelé processus de « management des risques ».

#### IV.1.4. Les différentes classes de risques

La notion de risque est donc source de nombreuses confusions tant ce terme est utilisé de façon abusive et dans des domaines variés. Les vues générales proposées généralisent certains aspects comme la notion d'aléa et de gravité ou de causes/conséquences. Afin de mieux cerner cette notion de risque différentes catégories de risques ont été proposées. Les catégories que nous retrouvons le plus souvent sont les suivantes :

- Les risques majeurs qui peuvent correspondre à des catastrophes et sont en rapport avec des événements exceptionnels et marquants :
  - les catastrophes naturelles (avalanches, inondations, incendies, éruptions volcaniques...),
  - les catastrophes technologiques (Seveso, AZF, Tchernobyl, transport de matières dangereuses, rupture de barrage...),
  - les risques sanitaires (crise de la vache folle, légionellose...).
- Les risques financiers relatifs à la prise de décision en particulier dans le secteur de la Bourse.
- Le risque d'échec d'un projet qui sont étudiés et pris en compte dans tous les projets. Ce management basé sur les risques est particulièrement utilisé dans le domaine du génie logiciel et pour les projets industriels de grande envergure.
- Le risque individuel d'accidents, d'atteinte à l'intégrité physique. Ce risque est omniprésent dans la vie de tous les jours. La prise de risque est en effet indissociable de toute activité humaine. Il est possible de distinguer dans cette classe les accidents domestiques, du travail, de la route ainsi que ceux relatifs à la pratique d'activités sportives ou de loisir.

Dans notre travail, nous nous sommes davantage focalisés sur le dernier type de risque, à savoir le risque lié à la pratique d'activités individuelles ou en groupes restreints.

## IV.2. Hiérarchisation du risque

La quantification d'un niveau de risque passe par la hiérarchisation et le positionnement relatif des différents risques. Le niveau de risque est en relation avec les possibilités d'occurrence d'un événement ainsi qu'avec les pertes potentielles associées. Ces pertes potentielles sont regroupées sous le terme de vulnérabilités. Or, quand un événement a réellement lieu, celui-ci est caractérisé par une gravité qui est la mesure réelle selon une échelle, le plus souvent, celle des pertes véritables associées à l'événement considéré.

## IV.2.1. Notion de gravité et pyramide des risques

La mesure des pertes pour un événement qui s'est réellement passé peut être ramenée à une mesure de gravité. Un même événement peut en effet entraîner des pertes totalement différentes, des plus bénignes au plus graves. C'est pourquoi le risque est très directement lié à la notion de gravité et le type d'événement ne constitue alors qu'un critère de classification. Ainsi, en reliant la notion de gravité à la notion d'événement, il est possible de considérer plusieurs catégories selon le niveau gravité. La hiérarchie des risques induit une hiérarchisation des traitements et priorités. Il est bien sûr indispensable et prioritaire d'étudier les accidents et les événements graves qui se produisent. Mais lorsque ceux-ci deviennent rares, l'étude des événements de gravité moindre peut se révéler utile pour analyser le système et faire progresser la sécurité en permettant d'agir avant qu'un accident ne se produise.

Une représentation fréquente des événements selon le niveau de gravité est donnée par la pyramide des risques ou des événements (Bird, 1974) qui peut être déclinée pour chaque secteur où le risque est étudié selon les niveaux de gravité rencontrés. Cette pyramide est une vue hiérarchique qui représente les différents niveaux de conséquences associées aux événements indésirables. Elle permet, d'une part, de visualiser la fréquence des événements puisque les événements situés à la base sont plus fréquents que ceux des niveaux supérieurs et, d'autre part, de constater qu'il y a des interactions entre les différents niveaux.

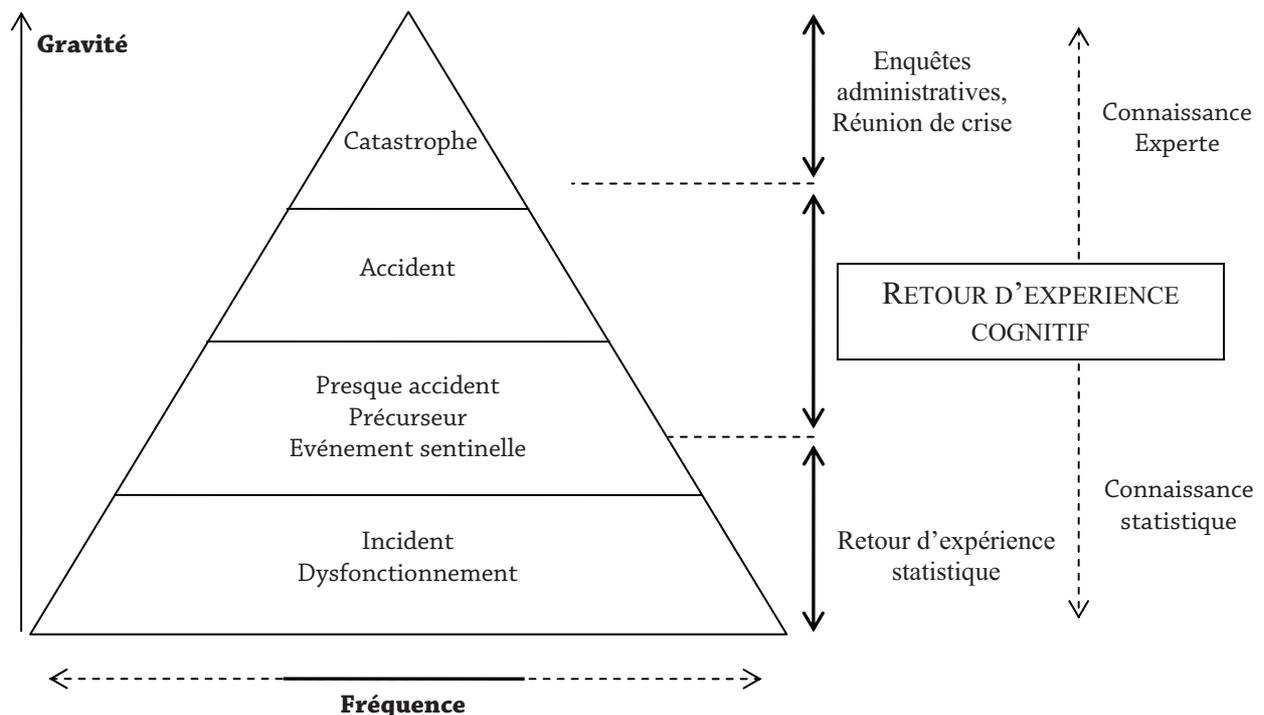


Figure IV-2 - Pyramide des risques

Sur la Figure IV-2, nous avons représenté une pyramide des risques générique typique (extraite d'un rapport thématique de l'ANAES (ANAES, 2003)). Cette pyramide est composée des éléments suivants :

- « Les incidents et dysfonctionnements : un incident correspond à un événement fortuit, peu important en soi et n'ayant pas engendré de conséquences ou alors des conséquences bénignes. Les dysfonctionnements sont des problèmes au regard d'un fonctionnement normal : on peut distinguer : la non-conformité, correspondant à une non-satisfaction d'une exigence spécifiée, l'anomalie, écart entre une situation existante et une situation attendue, le défaut, non-satisfaction d'une exigence ou d'une attente raisonnable. Le recueil et l'analyse de ces événements indésirables présentent deux intérêts

*majeurs : déterminer les défaillances qui, restant isolées, n'auraient pas de conséquences graves mais qui, associées, peuvent conduire à un accident ».*

- « *Les presque accidents (« near miss »), précurseurs et événements sentinelles : ces événements correspondent à des risques avérés. Leur analyse est particulièrement instructive pour améliorer la sécurité. Le presque accident est une situation qui aurait conduit à l'accident si des conditions favorables n'avaient pas permis de l'éviter. Le précurseur correspond à tout événement critique qui peut conduire à l'accident avec une probabilité importante ».*
- « *Les accidents : un accident correspond à un événement aléatoire qui conduit à des pertes humaines, matérielles ou environnementales ».*
- « *Les catastrophes : une catastrophe est un événement brutal, d'origine naturelle ou humaine, ayant généralement la mort et la destruction à grande échelle pour conséquence. Une catastrophe est donc un accident particulier car de grande ampleur, c'est-à-dire entraînant des pertes très importantes ».*

Cette représentation permet, notamment, de positionner les différents types de retour d'expérience selon leur approche de prise en compte du risque. En effet, les catastrophes et autres accidents graves sont en principe rares et sont suivis d'enquêtes souvent très approfondies pour en étudier les causes, les responsabilités et aussi pour mettre en place des mesures afin d'éviter que ces événements se reproduisent ou afin, au moins, d'en limiter leurs conséquences. Nous sommes ici dans une approche de retour d'expérience de type enquête administrative (cf. partie I.2.3) ou retour d'expérience de crise. Au contraire, les événements très fréquents ont en principe une gravité modérée ou acceptable. Leur grand nombre permet d'utiliser les nombreux outils mathématiques, statistiques et probabilistes pour les cerner, les contenir et les prédire. Nous associons à cette partie les retours d'expérience de type statistique. Enfin, nous attribuons la qualification de retour d'expérience cognitif à la catégorie intermédiaire concernant des risques en rapport avec les événements qui ne sont ni suffisamment fréquents, ni suffisamment graves pour être traités soit par des enquêtes très approfondies, soit par des moyens statistiques puisqu'il n'y a pas suffisamment de données. Nous pensons, avec le retour d'expérience cognitif, pouvoir traiter des risques de cette catégorie de manière structurée et dans une certaine mesure automatisée.

De la même manière que nous avons positionné le type d'expérience par rapport à la gravité, le niveau de gravité induit aussi une distinction entre les types de connaissances qui sont mises en œuvre dans les différentes approches de management du risque. Nous voyons que la base de la pyramide est en relation avec des connaissances statistiques alors que plus la gravité augmente, plus les connaissances explicitées sont de type « connaissances expertes ».

La création de pyramide intervient en principe dans le cadre de l'identification des risques a posteriori, c'est-à-dire lorsqu'il y a une bonne connaissance des risques encourus. La limite entre chaque catégorie n'est pas figée et les catégories peuvent aussi être légèrement superposées. Il faut également replacer le référentiel dans le contexte du risque étudié. Par exemple, les accidents de la route sont suffisamment nombreux et peuvent donc être traités statistiquement. Au contraire, les accidents aériens sont le plus souvent des catastrophes mais de fréquence moindre. C'est pourquoi chaque domaine d'étude du risque peut utiliser une variante de la pyramide des risques présentée sur la Figure IV-2. Par exemple, la Figure IV-3 illustre un exemple particulier de pyramide des risques associée aux risques d'accidents du travail.

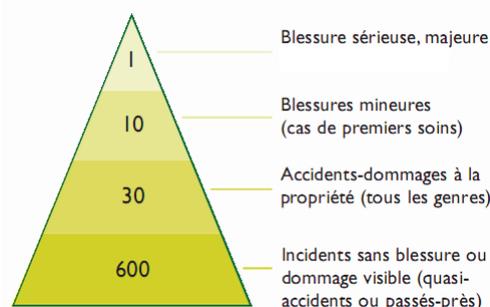


Figure IV-3 - Pyramide des risques d'accidents de travail

La pyramide des risques est utile pour positionner le retour d'expérience mais ne nous permet pas d'exprimer une mesure ou un indicateur du risque. Pour cela, nous utilisons un référentiel du risque et cette notion est introduite dans la partie suivante.

## IV.2.2. Les référentiels de risque

La mesure la plus fréquente pour exprimer un risque est une valeur échelonnée. Par exemple, le risque d'avalanche, selon la norme européenne, est un risque qui varie de 1 (risque modéré) à 5 (risque maximum). Le risque nul n'existant pas, la première catégorie de risque est généralement le risque minimum ou faible. Le risque maximum signifie que le risque encouru est très important, ce qui ne signifie pas qu'un accident surviendra mais uniquement que les conditions les plus favorables sont réunies pour l'occurrence d'une avalanche.

Le référentiel de risque est un outil qui permet de relier aléa, vulnérabilité (ou gravité) et niveau de risque. La hiérarchisation des événements entraîne naturellement la prise en compte d'un référentiel du risque. Intuitivement, nous comprenons que des événements de gravité modérée ou faible peuvent être tolérés dans une certaine mesure contrairement aux événements graves et aux catastrophes. L'idée du référentiel du risque est justement de pouvoir relier ces grandeurs. Un événement possiblement grave mais peu fréquent pourra conduire au même niveau de risque qu'un événement bénin se produisant quasiment à chaque réalisation des conditions favorables. Sachant que le risque peut être exprimé en fonction de l'aléa ou occurrence de l'événement et de la gravité résultante ou impact, nous pouvons utiliser un référentiel qui considère le niveau de risque  $R$  donné par une fonction de  $P$  (occurrence) et  $I$  (Impact). L'association occurrence/impact peut alors être matérialisée par le positionnement dans le plan  $(P, I)$  d'un point représentatif de cette situation (voir Figure IV-4).

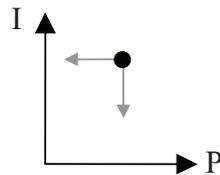


Figure IV-4 - Situation de risque

Pour pouvoir déterminer des classes de risques, il est alors possible de mettre en évidence les niveaux de risque sur une courbe 3D pour obtenir un « modèle d'expertise du risque ». Cette représentation est le plus souvent projetée sur un plan grâce à l'utilisation de courbes iso-risque. Il ne reste alors plus qu'à reporter la situation du risque sur ce référentiel pour apprécier le niveau de risque. L'illustration de la Figure IV-5 utilise un référentiel donné par la formule  $R=P \times I$ . Les zones ainsi créées permettent de définir des ordres de priorité dans le traitement des risques et donc des niveaux d'acceptabilité du risque. En effet, si un risque est potentiellement dangereux mais aussi très rare, il est peut-être finalement peu pertinent de le considérer en priorité. De même, si un risque qui survient fréquemment n'a que des conséquences bénignes, il est peut-être préférable de ne pas le traiter. Enfin, notons que les courbes seuils ne sont pas figées. Elles dépendent du référentiel choisi et du rapport entretenu avec le risque. Ces courbes peuvent être modulées pour répondre à des problématiques particulières.

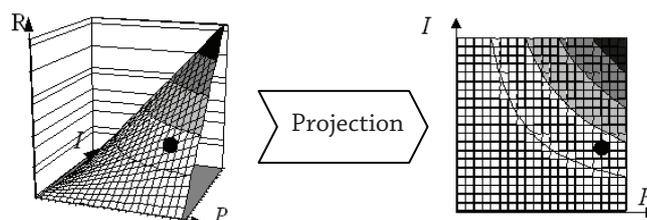


Figure IV-5 - Représentation 3D et 2D du risque (Gouriveau, 2003)

En ce qui nous concerne, nous avons choisi d'utiliser un référentiel de même type excepté qu'au lieu de positionner des probabilités pour I, nous positionnons une distribution de masse bayésienne. Nous précisons dans la suite de ce mémoire comment est construite cette mesure mais nous illustrons, sur la Figure IV-6, le type de référentiel que nous utilisons. Il s'agit d'un référentiel 2D possédant des courbes iso-risque qui sont des variantes de la fonction inverse. Nous plaçons en abscisse les chances d'occurrence et en ordonnées le niveau de risque qui, par défaut, est compris entre 1 et 5 mais qui par la suite peut être discrétisé en autant de catégories et de niveaux correspondants.

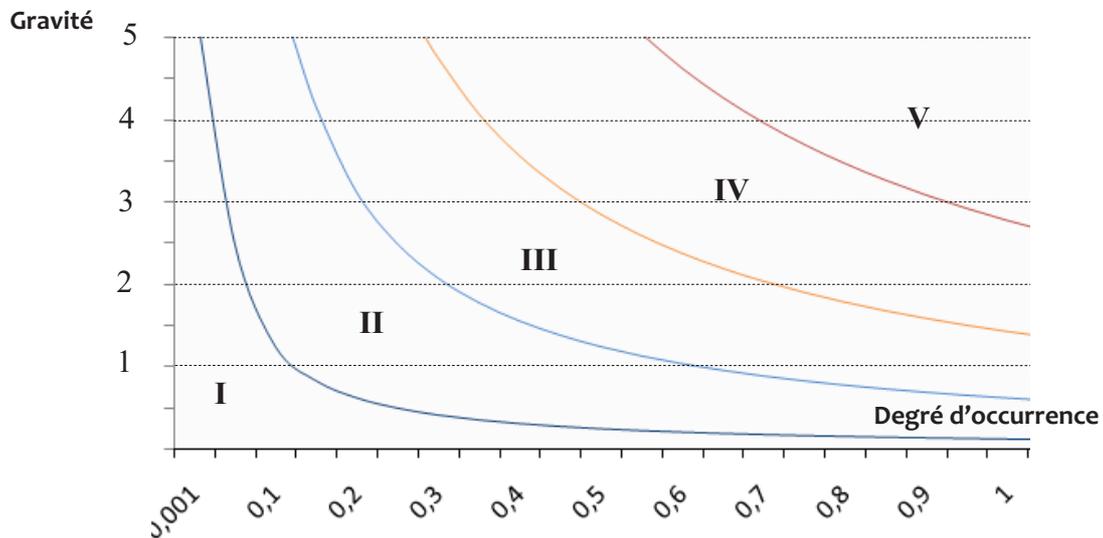


Figure IV-6 - Référentiel du risque pour les expériences

Chaque catégorie ou niveau de risque correspond à une zone ajustée par les fonctions d'iso-risque. Si nous considérons la formule  $R=P \times I$  alors les 5 courbes iso-risque suivantes dérivées de  $I=R/P$  avec R variant entre 1 et 5.

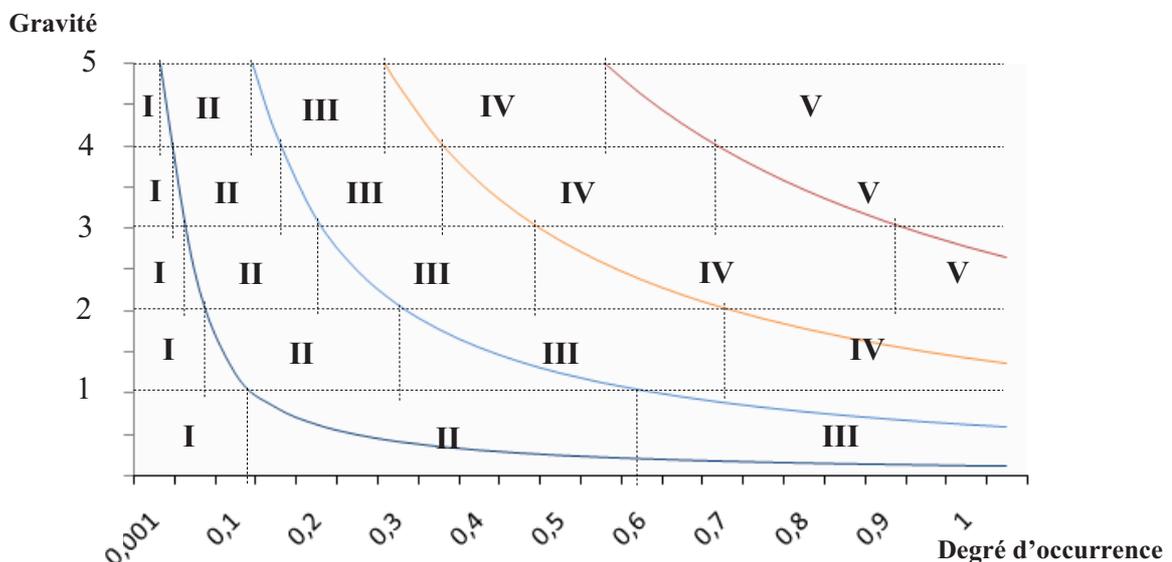


Figure IV-7 - Les différentes partitions de niveaux de risque

Le principe de construction de l'indicateur du risque consiste à positionner sur ce graphe chaque contribution d'expérience pondérée par son degré de similarité global. Après avoir effectué ce positionnement sur le graphe, un peu comme un changement de repère, nous pouvons établir une nouvelle distribution de masse sur l'ensemble  $\{I, II, III, IV, V\}$  qui correspond aux différents niveaux de risque (I : risque acceptable, V : risque inacceptable). Nous expliquons dans la partie IV.3.2 comment nous réalisons la correspondance entre degré d'occurrence et la similarité des expériences. La

correspondance entre niveau de risque discrétisé et gravité est donnée sur la Figure IV-7 où les contributions sur le référentiel de risque par niveau de gravité y sont représentées.

Nous voyons sur ces représentations que chaque niveau de gravité est coupé par les courbes iso-risque définissant des partitions. Ainsi, il est possible de construire une mesure du risque ayant ses valeurs dans l'ensemble {I, II, III, IV, V}. Nous observons, sur l'exemple, que les événements de gravité 1 contribuent seulement aux niveaux de risque I, II, et III. Or, avec la formule générique  $R=P \times I$ , les courbes iso-risque sont éloignées et, par exemple, les événements de gravité 1 ne contribuent plus qu'au niveau 1 du risque. Nous proposons d'utiliser une variante de cette formule suggérée par Gouriveau (Gouriveau, 2003). Cette formule est paramétrable et utilise des paramètres a et b variant entre 0 et 1 permettant d'ajuster les courbes iso-risque en fonction de la représentation désirée. Les courbes sont données par les fonctions :

$$y = \frac{R - a \cdot x}{b + (1 - a - b) \cdot x}$$

En pratique, il sera plus simple de déterminer les points d'intersection de chaque axe de gravité avec la probabilité voulue telle que le niveau de risque soit cohérent. Par exemple, nous pourrions supposer, dans un contexte donné, que tout événement de gravité 5 ayant une probabilité supérieure à 0,5 est inacceptable auquel cas, nous lui associons le niveau de risque maximum.

Le référentiel du risque peut alors être ramené à un tableau comme ci-dessous où nous avons choisi des valeurs décalées entre chaque niveau :

Gravité\Risque	I	II	III	IV	V
5	0,025	0,05	0,1	0,2	0,5
4	0,05	0,1	0,2	0,5	0,8
3	0,1	0,2	0,5	0,8	1
2	0,2	0,5	0,8	1	1
1	0,5	0,8	1	1	1

Tableau IV-1 - Probabilité supérieure du niveau de risque par niveau de gravité

### IV.2.3. Synthèse

Les définitions générales que nous avons présentées sont à la base de la compréhension de la notion du risque. Tout risque est en rapport avec la survenue d'un événement non souhaité, aléatoire dont les effets sont dommageables. La solution habituelle pour déterminer un risque revient à déterminer fréquence et gravité d'un événement. Cette approche présente l'avantage de pouvoir être intégrée dans des référentiels du risque qui permettent notamment la hiérarchisation des risques en vue de prioriser leur traitement.

Etant donné que l'expérience présentée capitalise un « événement négatif », c'est-à-dire aux conséquences dommageables, le lien expérience-risque existe indéniablement. Cependant, dans le cadre proposé, nous ne pouvons déterminer la fréquence ou la probabilité d'occurrence d'un événement donné. En effet, s'il est possible d'attribuer un niveau de gravité à chaque nouvel événement recensé, la probabilité d'occurrence au sens fréquence ne peut pas être, en règle générale, connue a priori. Cela

nécessite un nombre suffisamment grand d'expériences pour chaque type d'événement possible, ce qui n'est pas envisageable dans le cadre du retour d'expérience cognitif.

## IV.3. Proposition d'une mesure du risque par retour d'expérience cognitif

Dans le retour d'expérience cognitif tel que nous l'avons défini, les événements sont dits « négatifs », ils représentent une situation à risque avéré, c'est-à-dire ayant déjà eu des conséquences dommageables. Nous avons choisi de mettre en œuvre un processus d'exploitation du retour d'expérience peu développé qui est le cycle de prévention (cf. partie I.2.5.2) par opposition au cycle de résolution de problème. L'idée sous-jacente de ce processus est de rechercher, parmi les expériences capitalisées, celles qui correspondent le mieux avec le contexte courant d'une activité à un instant donné ou sur un intervalle de temps donné. A partir de chaque résultat de recherche d'expériences significatives par rapport au contexte courant, il est possible, dans une certaine mesure, d'envisager une évaluation des événements dommageables potentiels afin de les anticiper ou de les éviter. Bien sûr, pour que cette utilisation particulière ait un sens, le modèle de l'expérience et les analyses doivent être suffisamment précises et surtout représentatives de l'événement décrit. Nous considérons ce dernier point acquis car il est de la responsabilité du concepteur du modèle de l'expérience et des experts qui réalisent les analyses de pertinence.

Dès lors, il apparaît naturellement un lien évident mais non formalisé entre l'ensemble d'expériences similaires retourné et une mesure du risque encouru relative aux événements potentiels. Nous proposons, dans les sections suivantes, le principe d'une mesure du risque par l'expérience qui ne repose pas sur des outils statistiques et l'algorithme associé.

### IV.3.1. Association entre expérience cognitive et risque

Comme nous venons de l'exposer, il existe une relation implicite entre l'évaluation du risque et l'expérience. La recherche d'expériences pertinentes par rapport à un contexte courant permet d'obtenir un ensemble d'expériences négatives significatives pour ce contexte particulier. C'est pourquoi, nous avons décidé d'explicitier cette relation implicite en utilisant comme support les analyses expertes consignées dans les expériences.

Contrairement à une approche « classique » de mesure du risque, nous ne pouvons pas nous appuyer sur des approches statistiques. Il n'est en effet pas possible de choisir un indicateur du risque qui soit basé en partie sur le rapport entre le nombre d'événements problématiques (dans certaines conditions) et le nombre total d'événements (dans les mêmes conditions). C'est pourquoi nous nous éloignons des approches classiques d'évaluation du risque reposant en grande partie sur la fréquence (nombre d'événements dommageables divisé par le nombre total d'événements). La fréquence est cependant une information importante et il serait intéressant, dans les prolongements de ces travaux, d'envisager le couplage entre données statistiques et données expertes (les expériences étant au centre de ce couplage). Ici, nous ignorons cette information fréquentielle et nous souhaitons donner seulement un aperçu des différents risques encourus. Autrement dit, le rôle premier de l'indicateur que nous élaborons est de donner une synthèse de la connaissance disponible dans les expériences. Celui-ci doit pouvoir synthétiser l'ensemble des accidents et donner un accès à leur description détaillée si nécessaire. En aucun cas, il ne doit être considéré comme une mesure prédictive indiscutable mais plutôt comme une information ciblée qui dépend du contenu de la base de connaissance.

Dans la mesure où nous n'utilisons pas les fréquences des événements, notre approche s'éloigne du cadre habituel d'évaluation des risques. Nous avons néanmoins souhaité développer un mécanisme qui permet d'utiliser un référentiel du risque similaire. La mesure que nous proposons est donc basée sur

l'hypothèse suivante : lors d'une confrontation à une nouvelle situation, nous considérons que si un événement s'est produit dans des conditions parfaitement similaires, alors la croyance qu'un événement de même type peut se produire est maximum. Les mesures obtenues seront naturellement d'autant plus significatives que la base sera peuplée d'un grand nombre d'expériences.

Pour pouvoir déterminer la mesure du risque par l'expérience, il est nécessaire de définir un référentiel qui servira de support à l'indicateur mais il faut également compléter le modèle de l'expérience afin d'intégrer dans ce modèle l'évaluation de la gravité des événements passés. La gravité des événements passés est une information importante pour pouvoir ordonner les différents événements et ainsi permettre la hiérarchisation des risques. Cette information n'appartient pas au contexte mais directement à l'événement. Cette entité servait auparavant uniquement à donner des informations génériques et ne possédait pas d'élément sémantique particulier. Nous ajoutons donc un attribut permettant de décrire la gravité. Cette nouvelle information est reliée à l'entité événement de l'expérience. Nous avons simplement choisi de représenter la gravité comme une valeur sur une échelle croissante allant de 0 (problème sans gravité) à  $MAX_{gravité}$ ,  $MAX_{gravité}$  étant un entier positif.

Concernant le support de la mesure du risque, nous avons utilisé un référentiel de risque classique adapté aux mesures de similarité que nous avons proposé. Ce référentiel présente le niveau de gravité en ordonnée et une information analogue à une mesure de probabilité d'occurrence (utilisée habituellement) en abscisse.

#### IV.3.1.1. Lien entre les analyses, les causes et le risque

Pour pouvoir caractériser le risque de manière acceptable, il faut établir un lien entre les constituants de l'expérience et le risque. L'ajout de l'information de gravité des événements est un premier pas en vue de cet objectif. Il reste à trouver un lien entre l'expérience et les causes de l'événement.

Nous rappelons qu'une expérience est définie par un contexte et dans une certaine mesure, si ce contexte est suffisamment général, il englobe les conditions de réalisation de l'événement. Bien sûr cette retranscription des causes n'est pas aussi précise qu'elle peut l'être dans le cas dans un arbre des causes où les causes sont les entités principales. Cependant, à partir d'un contexte suffisamment détaillé, nous pensons que l'analyse (qui est une pondération caractérisant l'influence des attributs dans la réalisation de l'événement) constitue une approximation valable de l'identification des causes de l'événement.

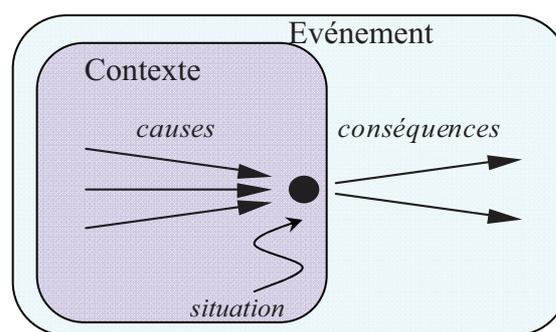


Figure IV-8 - Composantes d'un risque dans un cas

Nous représentons, sur la Figure IV-8, le fait que l'analyse d'un contexte inclue la projection des causes de l'événement décrit. Idéalement, nous pensons qu'il serait préférable de construire le contexte comme l'ensemble des informations relatives aux causes de chaque événement. Dans le cadre de cette thèse, nous avons choisi une construction plus simple, en partant du contexte général en allant vers la construction de sous-ensembles causaux par l'intermédiaire des analyses. C'est pourquoi, la construction de l'indicateur s'appuie sur l'utilisation des algorithmes de recherche prenant en compte l'analyse (cf. partie II.3.3) afin de trouver les expériences les plus significatives du point de vue du risque.

### IV.3.2. Construction de la mesure

L'indicateur que nous proposons est construit à partir de l'information retournée par la recherche par similarité entre les analyses d'événements et le contexte courant. A partir de cet ensemble d'expériences similaires, comme cela est représenté sur la Figure IV-9, il faut détailler les étapes qui vont permettre d'obtenir la mesure exprimée sur un référentiel classique de risque (partie droite de la figure).

Le principe de l'extraction de la mesure du risque pourrait être assimilé à un problème de classification visant à prédire l'événement qui pourrait se produire. Nous souhaitons proposer aux utilisateurs une synthèse des cas passés en fonction du contexte présent. La mesure du risque doit refléter tous ces cas « déjà vus » et les agréger dans une mesure globale.

L'expression de la mesure (le résultat) que nous envisageons est une représentation de type distribution de masses de croyance exprimée sur le cadre de discernement qui contient les différents niveaux de risque. Nous rappelons que cet indicateur exprime un principe suggéré par le retour d'expérience et qui énonce que dans des conditions parfaitement similaires, un événement de même nature a de fortes chances de se reproduire.

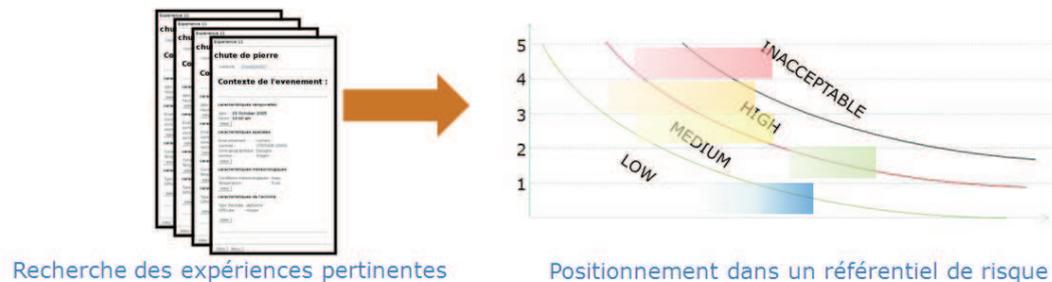


Figure IV-9 - Principe général de l'indicateur du risque par l'expérience

La mesure du risque proposée est construite selon trois étapes :

- Tout d'abord, un ensemble d'expériences similaires à un contexte courant (avec prise en compte des analyses dans la recherche) est collecté.
- Ensuite, à partir de ces expériences, la mesure du risque est construite en agrégeant la contribution de chacune des expériences similaires et en tenant compte de la gravité et de l'analyse.
- Enfin, une fois cette agrégation réalisée, un changement de référentiel est effectué afin de passer d'une famille de distributions de similarité (prenant ses valeurs entre 0 et 1) par niveaux de gravité à une distribution bayésienne sur le référentiel du risque. Ce transfert est effectué à partir des courbes iso-risque définies pour le domaine considéré.

### IV.3.3. Algorithme d'extraction de la mesure du risque

Dans cette section, nous transposons en algorithme les étapes préconisées précédemment sur les structures de données que nous avons définies et utilisées jusqu'à présent.

L'algorithme proposé comporte quatre grandes étapes illustrées sur la Figure IV-10 où nous présentons un cas simple de recherche. Tout d'abord, à partir de la recherche par rapport à un contexte courant, nous obtenons un tableau de distributions de masses bayésiennes de similarité. Chacune est associée à un niveau de gravité. Nous connaissons également les éléments qui traduisent l'analyse de l'expert. Dans l'exemple proposé, afin de présenter la construction de la mesure, nous ne présentons que deux analyses significatives et donc deux mesures de similarité correspondantes. Chaque événement associé possède une gravité de niveau 4. Lors de la deuxième étape, qui consiste à agréger par niveau de gravité chacune des masses des distributions pour chaque ensemble focal, il faut pour l'exemple agréger les deux mesures de similarités précédentes (en tenant compte des analyses qui sont différentes). Une

fois agrégées, ces mesures sont positionnées sur le référentiel du risque en fonction de la gravité. Ainsi, chacune possède une liste de valeurs seuil (donnée par une matrice comme par exemple sur le Tableau IV-1). Dans l'étape 4, à partir de chaque agrégation de mesure et de ses seuils associés qui définissent les limites de niveaux de risque, la famille de distribution (par niveaux) est combinée en une seule distribution sur le référentiel du risque, ce qui après normalisation fournit l'indicateur du risque.

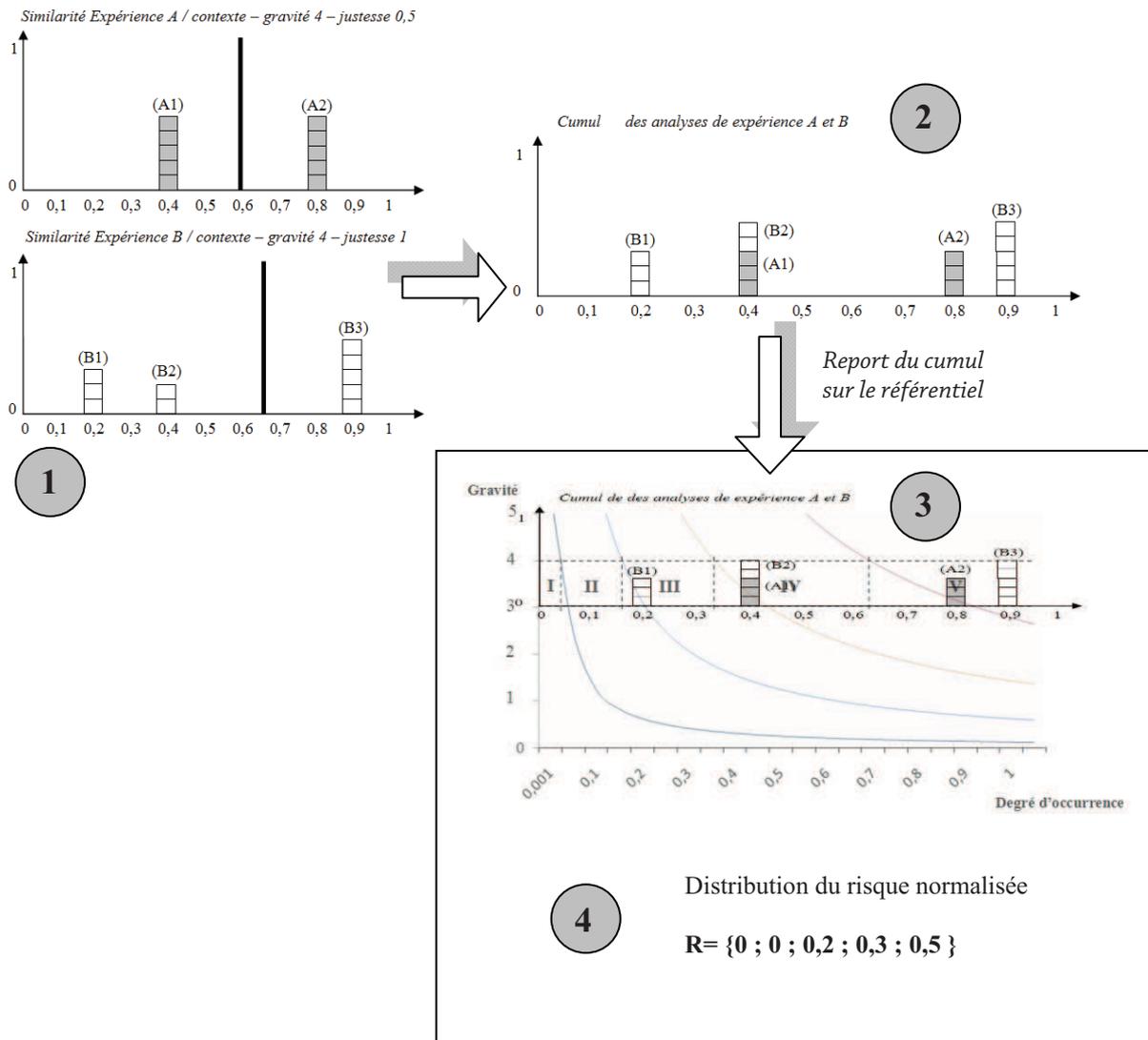


Figure IV-10 - Les 4 étapes de la construction de l'indicateur du risque

Il convient de remarquer que l'étape d'agrégation des résultats (niveau de risque par expérience) pour toutes les expériences n'a pas été abordée en profondeur ici. Nous avons envisagé d'utiliser une distribution finale normalisée qui permet d'obtenir une distribution de masse valide mais qui présente l'inconvénient de donner une mesure lissée ne reflétant pas bien le niveau de risque possible. Il serait intéressant de proposer un indicateur qui retienne par niveau le poids maximum pour toutes les expériences. Cette mesure serait alors une vue pessimiste du risque ce que nous pensons être préférable dans une optique de prévention. En contrepartie, la distribution résultante ne serait pas équivalente à une distribution de masse mais ceci n'est pas gênant étant donnée que la mesure n'est pas traitée a posteriori. Cette procédure d'agrégation des distributions de risque par expérience fera l'objet de travaux futurs.

Nous présentons l'Algorithme IV-1 (page 123) qui permet de calculer l'indicateur du risque d'après le principe défini et décrit pour l'exemple de la Figure IV-10.

**Algorithme IV-1 - INDICATEUR<sub>RISQUE</sub>(contexteCourant c[])**entrées :

c[] le contexte courant contenant un ensemble de descripteurs [ $d_{c1}$  ; ... ;  $d_{ci}$ , ... ;  $d_{cn}$ ]

Nombre g de niveau de gravité

Nombre r de niveau de risque

SIM<sub>agreg</sub>[] un tableau contenant g distributions de masses agrégées

Le tableau SIM<sub>agreg</sub> [] contient g tableaux de similarité Sim<sub>i</sub> [] = [0 ; ... ; 0]<sub>m+1</sub> qui sont des distributions de masse bayésiennes (non normalisées).

N l'indice fonction du nombre d'expérience (analyse) prises en compte et de leur justesse

SEUIL<sub>risque</sub>[][] matrice g x r contenant les valeurs seuils de chaque niveau de risque pour chaque niveau de gravité

sortie :

RISQUE[ ]<sub>r</sub> un tableau représentant le risque par une distribution de masse bayésienne sur  $\Omega = \{1 ; \dots ; r\}$

TableauSIM ← **COMPAREContexteCourant()**

N ← 0

Pour chaque valeur ( $a_j \rightarrow$  SIMbbaj) de TableauSIM

gravité ←  $a_j$ .expérience.evenement.gravité

SIMbba<sub>temp</sub>[] ← SIM<sub>agreg</sub>[gravité]

coefficient ←  $a_j$ .justesse

N ← N + coefficient

Pour chaque indice x de SIMbbaj[]

SIMbba<sub>temp</sub>[x] ← SIMbba<sub>temp</sub>[x] + SIMbbaj[x] \* coefficient

Fin Pour

SIM<sub>agreg</sub>[gravité] ← SIMbba<sub>temp</sub> []

Fin Pour

Pour chaque indice k de SIM<sub>agreg</sub> []

index ← 1

Limite ← SEUIL<sub>risque</sub>[k][index]

SimK[] ← SIM<sub>agreg</sub>[k]

SommeTotale ← 0

temp ← 0

Pour chaque indice<sub>i</sub> de SimK[]

SimTemp ← SimK[indice<sub>i</sub>]

Si SimTemp ≠ 0

Si Limite ≤ (indice<sub>i</sub> - 1)/m

index ← index + 1

Limite ← SEUIL<sub>risque</sub>[k][index]

temp ← 0

Fin Si

temp ← temp + SimTemp

Risque[index] ← Risque[index] + temp

SommeTotale ← SommeTotale + SimTemp

Fin Si

Fin Pour

Pour chaque indice<sub>i</sub> de Risque []

Risque [indice<sub>i</sub>] ← Risque [indice<sub>i</sub>] / SommeTotale

Fin Pour

**Retourne Risque[ ]<sub>r</sub> et N**

❶

❶ Appel d'algorithme de recherche dans la base basé sur l'algorithme : CALCUL<sub>AGREGATION-ET</sub>()

❷

❷ agrégation

TableauSIM est un tableau d'association ( $a_j \rightarrow$  SIMbba)

mémoriser les exp concernées ou analyses

❸

❸ construction de la distribution de risque

pour la normalisation

optimiser si simtemp = 0

Limite.suivante

correspondance entre les intervalles et la limite - a vérifier

❹

❹ Normalisation de la mesure du risque

## IV.4. Synthèse

Dans ce chapitre, nous avons montré comment extraire une connaissance de la base d'expériences. Nous avons proposé une approche originale d'élaboration d'un indicateur de risque relatif à la connaissance contenue dans la base d'expériences. Cet indicateur est aussi la concrétisation du cas d'utilisation à titre préventif du retour d'expérience que nous avons développé en complément des cas d'utilisation de type « réutilisation de solution » plus fréquents et assez proches du RàPC. La mesure du risque ou l'indicateur du risque retourné par la méthode proposée est une distribution de masses de croyance bayésienne sur l'ensemble des niveaux de risque possibles.

Dans la mesure où peu d'évènements sont disponibles, l'utilisation d'approches statistiques n'est pas envisageable. Toutefois, nous pouvons profiter de la richesse cognitive contenue dans les expériences et matérialisée par les analyses des experts afin de construire une autre mesure de risque. Naturellement, la mesure proposée est basée uniquement sur la connaissance contenue dans la base d'expériences et cette connaissance est par nature très incertaine et incomplète. C'est pourquoi les résultats obtenus doivent être considérés avec précaution. Il serait d'ailleurs intéressant, pour des travaux ultérieurs, d'envisager la qualification de la connaissance contenue dans la base afin de quantifier l'incertitude sur l'indicateur obtenu, ou encore de ne pas tenir compte des scores déduits à partir de trop peu d'expériences.

# Chapitre V

## Application générique de retour d'expérience

L'objectif de ce chapitre est de présenter le développement d'une architecture logicielle qui met en œuvre les principes et techniques développés dans les chapitres précédents. Les applications intégrées dans cette architecture doivent remplir plusieurs fonctions : l'opérationnalisation du processus de retour d'expérience dans un domaine d'application particulier, la persistance des modèles et des expériences, la mise à disposition à des utilisateurs d'interfaces graphiques et la proposition d'outils de déploiement des applications.

De plus, comme nous l'avons déjà précisé, la définition du modèle de l'expérience nécessite plusieurs itérations et ajustements afin de modéliser correctement le domaine concerné. Connaissant ce besoin de conception itérative du modèle de l'expérience, il nous est apparu judicieux de développer un cadre générique d'applications de retour d'expérience et non une application dédiée. Nous souhaitons ainsi baser l'opérationnalisation des applications de retour d'expérience sur la définition d'un modèle de l'expérience. Un des points importants du formalisme retenu est qu'il permet de prendre en compte des valeurs d'information (des descripteurs) incomplètes et incertaines, conformément aux principes retenus dans le chapitre II. Il faudra donc tenir compte de ce point dans le modèle de l'expérience réalisé.

Dans ce chapitre nous introduisons et justifions le besoin d'un développement générique puis nous détaillons les principes de fonctionnement des applications de retour d'expérience réalisées. Nous détaillons les choix des composants de l'architecture, le moyen de persistance, l'articulation entre les différents modules... Ensuite, nous présentons les techniques de modélisation générique usuelles pour le développement logiciel jusqu'à choisir et présenter l'approche retenue. Enfin, nous terminons ce chapitre par l'application du cadre générique au domaine de la sécurité en montagne en présentant la réalisation effectuée pour le projet européen à l'origine de ces travaux de thèse.

### V.1. Une implémentation dirigée par le modèle de l'expérience

Comme cela a été indiqué dans le Chapitre I, le retour d'expérience est un système à base de connaissance particulier. Le choix d'un formalisme orienté objet a été motivé, outre la simplicité et l'expressivité de la représentation, par la fluidité qu'il permet lors du passage de la modélisation à l'opérationnalisation. Nous souhaitons, pour opérationnaliser une application de retour d'expérience, définir uniquement le modèle de l'expérience, c'est-à-dire les ensembles de descripteurs, leurs domaines de valeurs et fonctions de similarités associées. Il est utile pour atteindre cet objectif de préciser comment les modèles sont intégrés dans le développement des logiciels et quelles sont les évolutions de leur prise en compte.

Les modèles sont des vues conceptuelles abstraites qui permettent de représenter, à l'aide d'un langage ou d'une notation, des idées, des processus mais aussi les éléments de code, une organisation des composants (Le Moigne, 1987). Le langage UML<sup>1</sup> (Unified Modeling Language) est un exemple très connu d'un tel langage qui fournit les éléments nécessaires à la modélisation de nombreux types de systèmes. UML est né de la généralisation de plusieurs langages de modélisation et est construit selon le paradigme orienté objet bien que ses possibilités de modélisation dépassent le cadre des applications orientées objet. Un des avantages d'UML par rapport à d'autres supports de modélisation est son niveau d'abstraction, en particulier son métamodèle qui est suffisamment général et permet à UML de s'auto-définir. Cette caractéristique permet d'étendre UML au moyen de profils UML ou encore de stéréotypes. Ainsi, UML peut être adapté à des domaines particuliers. Par exemple, alors que les modèles de processus (BPM – Business Process Model) ne font pas partie des diagrammes standard d'UML, il existe des extensions standardisées qui permettent leur représentation.

De manière plus générale, il est possible de différencier les niveaux d'abstraction d'un modèle et en particulier les modèles d'analyse, de conception et d'implémentation. Ces trois formes font partie de Rational Unified Process (RUP) qui est un processus de conception de logiciel souvent couplé à UML. Le modèle d'implantation est le plus concret et le plus proche du code qui est développé. Les deux autres modèles sont abstraits et indépendants des détails d'implémentation (dont ceux relatifs aux plateformes support).

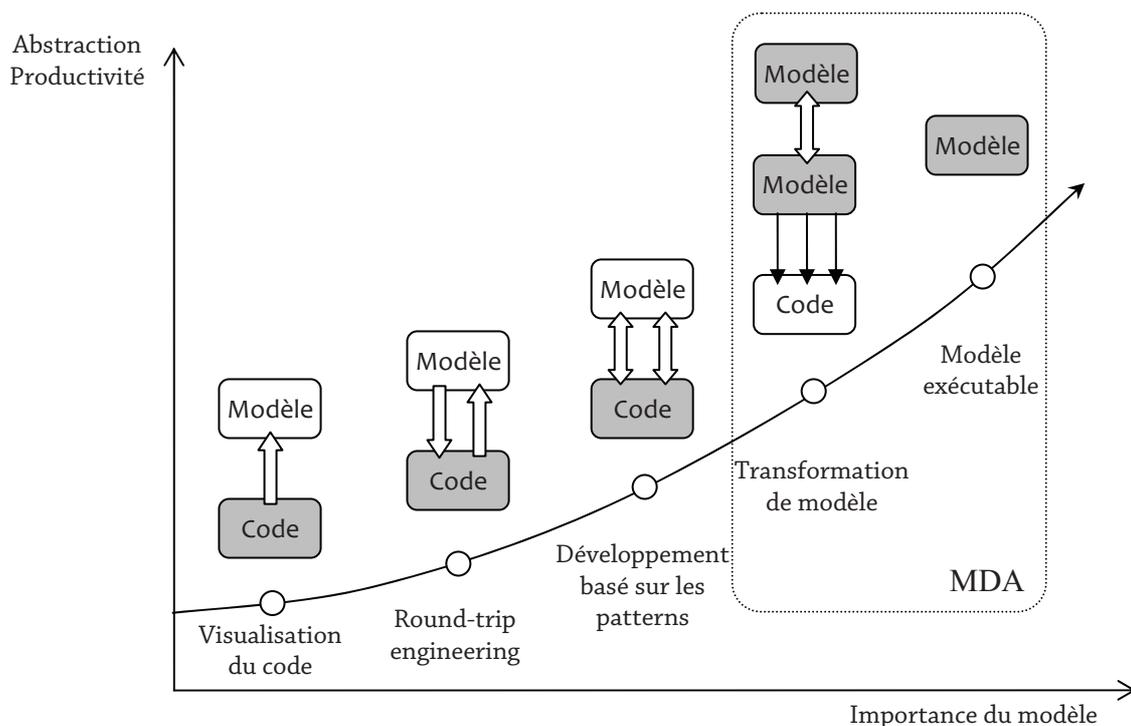


Figure V-1 - Ingénierie des modèles - les différents rôles des modèles (Bezivin et Gerbe, 2001)

Le rôle des modèles a évolué avec le temps et la tendance en ingénierie des modèles est d'aller vers plus d'abstraction afin d'augmenter la portée des applications réalisées. Cette évolution est présentée sur la Figure V-1 où sont représentés les différents rôles des modèles en fonction de leur niveau d'abstraction et de l'importance qu'ils jouent dans le processus de conception des logiciels. Nous distinguons les approches suivantes :

- Le rôle premier des modèles est de servir de moyen de visualisation du code et donc principalement de moyen de communication et de documentation. C'est encore l'usage le plus classique de la modélisation.
- Le *round-trip engineering* consiste à automatiser certains aspects, à savoir la génération d'une partie de code à partir des modèles et l'ingénierie inverse qui consiste à extraire des modèles depuis le code. Les outils typiques de cette relation avec les modèles sont les CASE Tools

<sup>1</sup> <http://www.uml.org>

(Computer-aided Software/System Engineering). Ils font référence aux outils et méthodes dédiés au courant d'ingénierie associant le développement des Systèmes d'Information avec des outils qui automatisent, dans une certaine mesure, ce processus (Loucopoulos et Karakostas, 1995).

- Le développement basé sur les *patterns* assure une correspondance réciproque entre le code et la conception puisque les *patterns* sont des référentiels qui, pour un problème de conception donné, associent une ou plusieurs méthodes d'implémentation. Les *patterns* sont donc à la fois une solution technique et un moyen de documentation.

Dans le premier cas, c'est le code qui occupe la place la plus importante dans la relation modèle-code. Le modèle est asservi au code, de manière plus ou moins automatisée. Au contraire, dans les deux derniers cas, c'est le modèle qui asservit la création de code et qui devient l'élément de base de la conception. Actuellement, le code est généré (en partie) par transformations successives de modèles. L'objectif à terme est la réalisation de modèles exécutables, c'est-à-dire totalement réversibles. L'Ingénierie Dirigée par les Modèles (IDM) et le MDA (Model Driven Architecture) en particulier sont des approches qui visent à contourner l'utilisation de middleware (passerelles entre systèmes hétérogènes). Cette initiative n'est pas nouvelle car elle reprend un cadre connu sous le nom de Cycle en Y (Bezevin et Gerbe, 2001). C'est l'OMG qui conduit le développement des standard MDA et la fondation Eclipse qui en conduit l'implémentation. MDA (Kleppe *et al.*, 2003) apporte une approche globale standardisée car basée sur des standard mis en relations :

- UML (Unified Modeling Language), le langage de modélisation unifié qui permet de décrire le domaine du problème et l'architecture de la solution (Booch *et al.*, 1999),
- MOF (Meta Object Facility) qui est le standard permettant de décrire et de manipuler les modèles standard ou spécifique ainsi que les métalangages (MOF, 2004),
- XMI (XML Model Interchange) qui permet d'échanger les informations sur les modèles (et métamodèles) dans un format XML (eXtensible Markup Language) (XMI, 1998),
- CWM (Common Warehouse Model) qui permet de décrire les correspondance entre les données stockées et les schémas des bases de données présentent dans un entrepôt de données (data warehouse) (CWM, 2003).

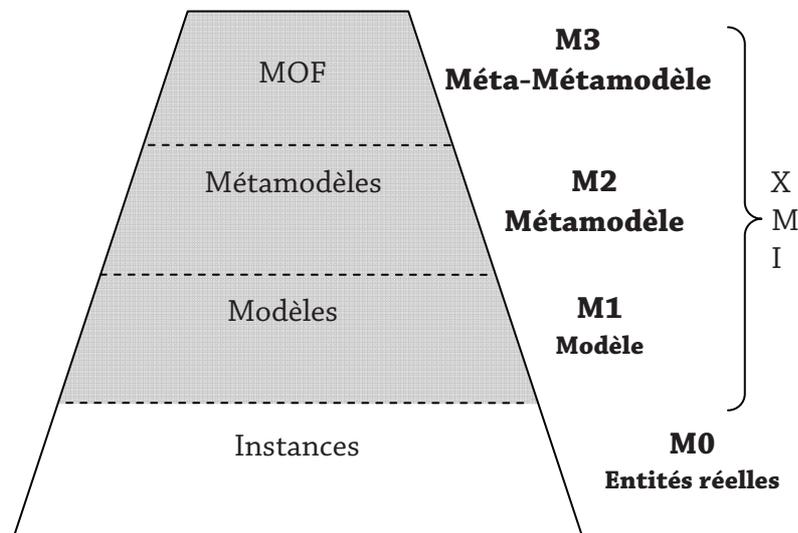


Figure V-2 - Architecture de métamodélisation MOF à 4 niveaux

Il est cependant intéressant à ce stade de présenter l'architecture à 4 niveaux que MOF définit car elle permet de bien situer les différents niveaux d'abstraction des modèles (Figure V-2). Le niveau M0 (ou instance) correspond au monde réel. Ce sont les informations réelles de l'utilisateur, instances du modèle de M1. Le niveau M1 (ou modèle) est composé de modèles d'information. Il cadre les informations de M0. Les modèles UML par exemple appartiennent à ce niveau. Les modèles M1 sont des instances de métamodèle de M2 (relation « est-un » descriptive). Le niveau M2 (ou métamodèle), définit le langage de modélisation et la grammaire de représentation des modèles M1. Le métamodèle

UML qui est décrit dans le standard UML, et qui définit la structure interne des modèles UML, fait partie de ce niveau. Les profils UML, qui étendent le métamodèle UML, appartiennent aussi à ce niveau. Les métamodèles sont des instances du MOF. Le niveau M3 (ou méta-métamodèle) est composé d'une unique entité, le MOF, qui permet de décrire la structure des métamodèles, d'étendre ou de modifier les métamodèles existants. Le MOF est réflexif, il se décrit lui-même, et donc, il n'est pas nécessaire d'avoir des niveaux supplémentaires.

La métamodélisation définie par MOF est une métamodélisation descriptive (*cf.* V.3.3). Elle est très utile dans une perspective d'interopérabilité de modèles hétérogènes et aussi pour l'analyse de code et de systèmes. Nous souhaitons avoir des modèles qui contraignent d'autres modèles. Les différents modèles et en particulier UML n'étant pas encore exécutables, nous n'avons pas choisi cette approche *meta* à 4 niveaux qui en outre aurait conduit à plus de complexité (Demeyer *et al.*, 1999).

L'IDM et en particulier l'initiative MDA de l'OMG sont fondées sur la séparation entre d'une part, les modèles indépendants des plateformes (PIM - Platform Independent Model) et d'autre part, les modèles spécifiques aux plateformes (PSM - Platform Specific Model). La démarche MDA (Model Driven Architecture) est jalonnée d'une série de transformations de modèles, depuis un haut niveau d'abstraction (CIM - Computation Independent Model), en passant par des niveaux intermédiaires (PIM - Platform Independent Model et PSM - Platform Specific Model), puis vers des modèles spécifiques de l'implémentation en fonction de son support. La transformation de modèle est donc à la base de l'approche MDA.

Ces tendances dans le domaine de l'ingénierie des logiciels nous montrent tout l'intérêt d'une approche basée sur la transformation de modèles et d'une approche générique dont l'intérêt est double à nos yeux. Tout d'abord, elle nous permet de faire abstraction de problèmes de modélisation incrémentale qui sont une difficulté majeure dans la mise en œuvre de systèmes de retour d'expérience. Nous pouvons ainsi modifier le modèle de l'expérience tant que celui-ci n'a pas convergé vers un modèle stable et approprié. En outre, la généricité permet de réutiliser les applications dans d'autres domaines sans se préoccuper des aspects liés directement au développement logiciel. Par exemple, le logiciel réalisé doit alors prendre en charge de manière la plus automatisée possible les interfaces graphiques (de présentation et de saisie des informations), la persistance du modèle et des instances (les expériences) et le déploiement ainsi que les traitements génériques spécifiques au retour d'expérience tel que la recherche basée sur les différentes fonctions de similarité (*cf.* partie III.5), l'extraction du risque (*cf.* partie IV.3.3). La contrepartie de la généricité est que le développement du cadre applicatif devient plus complexe. Nous présentons dans la partie V.3 les différents outils et méthodes en rapport avec les développements génériques. Dans la section suivante, nous présentons une vue macroscopique des applications que nous souhaitons élaborer.

## V.2. Vue d'ensemble des applications de retour d'expérience

Avant d'entrer plus en détail dans les considérations de généricité, il est nécessaire de donner une vision plus précise des applications que nous souhaitons réaliser. Nous définissons leurs différentes caractéristiques et en particulier les différents cas d'utilisation ciblés. Précisons que le cœur du retour d'expérience est l'implémentation du modèle et des algorithmes de recherche développés dans les chapitres précédents qui reposent sur la sémantique particulière des expériences (*cf.* partie II.3).

### V.2.1. Architecture client serveur web

Nous avons choisi de développer des applications basées sur une architecture accessible à partir d'un navigateur (client) internet, architecture que nous qualifierons d'architecture « web ». Le serveur

d'application contient la base d'expériences et de connaissances du système. Ce choix présente l'avantage de centraliser toute l'information dans une base de données unique, ce qui garantit une information actualisée et a priori cohérente. L'utilisation d'une architecture permet d'assurer une très bonne accessibilité aux applications de retour d'expérience. Ainsi, le déploiement des applications est localisé sur un seul poste et, pour accéder aux applications, il suffit d'utiliser un navigateur internet disponible non seulement sur les ordinateurs mais aussi, de plus en plus, sur les terminaux mobiles de type téléphone portable ou P.D.A. (Personal Digital Assistant).

Une utilisation itinérante des applications de retour d'expérience serait particulièrement utile en particulier pour le processus de prévention. Pour l'instant, nous nous limitons à un fonctionnement en mode connecté qui implique d'avoir une connexion internet à disposition. Il est cependant envisageable de prévoir un mode non connecté qui permettrait de pallier cette limitation. La « norme » de développement des applications web est aujourd'hui le mode « connecté » mais des initiatives récentes telles que Google Gear<sup>1</sup> laissent envisager la possibilité de disposer de navigateurs adaptés à cet usage non connecté dans un futur proche (voir à ce sujet (Goncalves et Leitao, 2007)).

Pour les interfaces graphiques, nous souhaitons disposer de composants génériques et réutilisables. Ces composants sont affichés par les navigateurs internet. Il s'agit principalement d'exploiter des formulaires ainsi que les possibilités de développement d'applications actuelles qui dépassent le cadre initial de présentation documentaire du Web. Nous disposons d'un ensemble de composants qui réalisent les fonctions suivantes :

- définition du modèle de l'expérience
  - édition du modèle des descripteurs
    - édition des domaines de descripteurs
    - édition différentes matrices de similarités
- édition de la valeur des instances des descripteurs
  - édition de l'expérience
    - édition de l'événement, du contexte et de l'analyse
  - édition du contexte courant
- présentation des résultats
  - tableau paramétrable avec score de similarité associé
  - présentation de la mesure du risque
  - composant d'affichage d'une expérience
- interface administrateur
  - gestion des utilisateurs et des droits
  - gestion des expériences, suppression, modification
- contrôle du processus de retour d'expérience
  - initialisation et paramétrage de la recherche
  - filtrage des résultats et sauvegarde
  - interface d'interrogation de la base d'expériences (filtrage, requêtes...)

Les composants les plus bas niveau sont ceux d'expression et de paramétrage des descripteurs qui, nous le rappelons, ne sont pas de simples valeurs d'attribut. C'est pourquoi, il ne suffit pas d'utiliser des champs textes pour collecter des valeurs d'attributs précises (entier, flottant, chaîne de caractères) ou des listes à choix simple ou multiple. Dans notre cas, nous avons dû définir un composant spécifique qui permet de saisir des distributions de masse. Il s'agit en fait le plus souvent de listes de valeurs à choix multiples où à chaque saisie de valeur est associé un poids. L'ergonomie de la saisie de ces valeurs est importante à considérer pour que la saisie des valeurs paraisse naturelle aux utilisateurs. Nous pouvons par exemple proposer une interface basculant entre un mode simple et avancée. Le premier mode est utilisé pour la saisie de valeurs précises ou relativement simples, ne mettant en jeu qu'un seul ensemble focal. Bien que conscient du fait que l'ergonomie joue un rôle fondamental dans l'acceptation du système, nous ne cherchons pas, dans cette phase de développement, à rendre l'utilisation des

---

<sup>1</sup> <http://gears.google.com/>

applications de retour d'expérience la plus intuitive possible, mais c'est évidemment un aspect à considérer par la suite. Nous présentons la proposition d'interface en détails dans la partie V.5.2.1.

## V.2.2. Persistance et traitement de l'information

Pour assurer le bon fonctionnement des applications il faut pouvoir sauvegarder à la fois les expériences et le modèle selon lequel elles sont décrites. L'application réalisée est basée sur un modèle particulier de l'expérience qui permet d'opérationnaliser l'application. Le modèle peut être codé directement dans le système et, dans ce cas, le code exécutable assure la persistance du modèle. Il peut aussi être stocké sous une forme sérialisée (dans un fichier par exemple).

Suivant le nombre d'expériences, il est possible d'effectuer les sauvegardes simplement dans des fichiers ou bien d'utiliser une base de données dédiée qui peut-être relationnelle ou objet. Nous nous sommes contentés, pour le développement d'un prototype, de l'utilisation d'une sauvegarde dans des fichiers dédiés n'autorisant pas les accès concurrents. Ce choix de sauvegarde permet de prendre en charge de l'ordre de 10 000 enregistrements d'expériences.

Toutefois, s'il peut être simple et pratique, dans un premier temps, de ne pas utiliser de base de données et de sauvegarder les expériences dans des fichiers, il est clair qu'à terme, quand les expériences seront très nombreuses et qu'il faudra gérer des accès concurrents, il sera nécessaire d'utiliser une véritable base de données. Notre préférence va vers les bases de données objet en liaison avec la nature des expériences et des choix de langages que nous avons faits. Il est malgré tout possible d'utiliser des bases de données relationnelles par l'intermédiaire d'un ORM (Object Relationnal Mapper) qui permet de préciser la correspondance, le *mapping*, entre les objets sauvegardés et les relations (tables) d'une base de données relationnelle.

## V.2.3. Principe des applications et cas d'utilisation

Sur la Figure V-3 est présentée l'architecture client-serveur ainsi que le positionnement des différents éléments et modules des applications de retour d'expérience. Les différents acteurs ainsi que les différents flux d'informations y sont également représentés.

A gauche de la figure sont représentées les différentes bases de données, celle de l'expérience et celle des connaissances qui sont dissociées afin de refléter le fait qu'il ne s'agit pas du même type d'information (expériences d'un côté et règles, par exemple, de l'autre). A droite, les clients sont représentés (PORTAIL WEB). Ce sont de simples navigateurs web, l'application étant accessible depuis un portail Internet. Les utilisateurs peuvent être de trois types :

- Les utilisateurs « classiques » qui ont pour rôle d'utiliser les interfaces de recherche et éventuellement de soumettre de nouvelles expériences sans les analyser,
- Les utilisateurs « experts » qui sont en charge de réaliser les analyses des expériences déjà capitalisées,
- Les utilisateurs cogniticiens, non représentés sur la figure, qui définissent uniquement le modèle de l'expérience avant la mise en œuvre du système. Si le système autorisait la modification en fonctionnement du modèle de l'expérience, ces utilisateurs procéderaient aux modifications et contrôlèrent les répercussions sur les expériences déjà capitalisées.

Trois flux d'informations sont distingués sur la Figure V-3, à savoir :

- Le flux d'expérience qui consiste d'une part en la saisie de l'événement, sa gravité et son contexte et, d'autre part, en l'interrogation de la base par les algorithmes de recherche par similarité de contexte,
- Le flux d'expertise qui consiste à associer aux expériences une analyse de pertinence des informations significatives. Ce flux d'expertise est ensuite fusionné avec le flux d'expérience lors des phases de recherche tenant compte de l'analyse,

- Le flux de connaissance qui consiste à exploiter le flux d'expérience, en effectuant par exemple des recherches avec ou sans analyse, pour extraire des invariants qu'il est possible de traduire sous la forme de règles ou encore de classes. Ces invariants peuvent être intégrés dans la base de connaissances. L'extraction de la mesure de risque est un exemple d'extraction de connaissance qui n'est pas a priori sauvegardée puisque l'indication utile à un moment donné n'est pas une connaissance qui sera utile et réutilisable par la suite (sauf s'il s'agit par exemple de l'élaboration de cartes de risques).

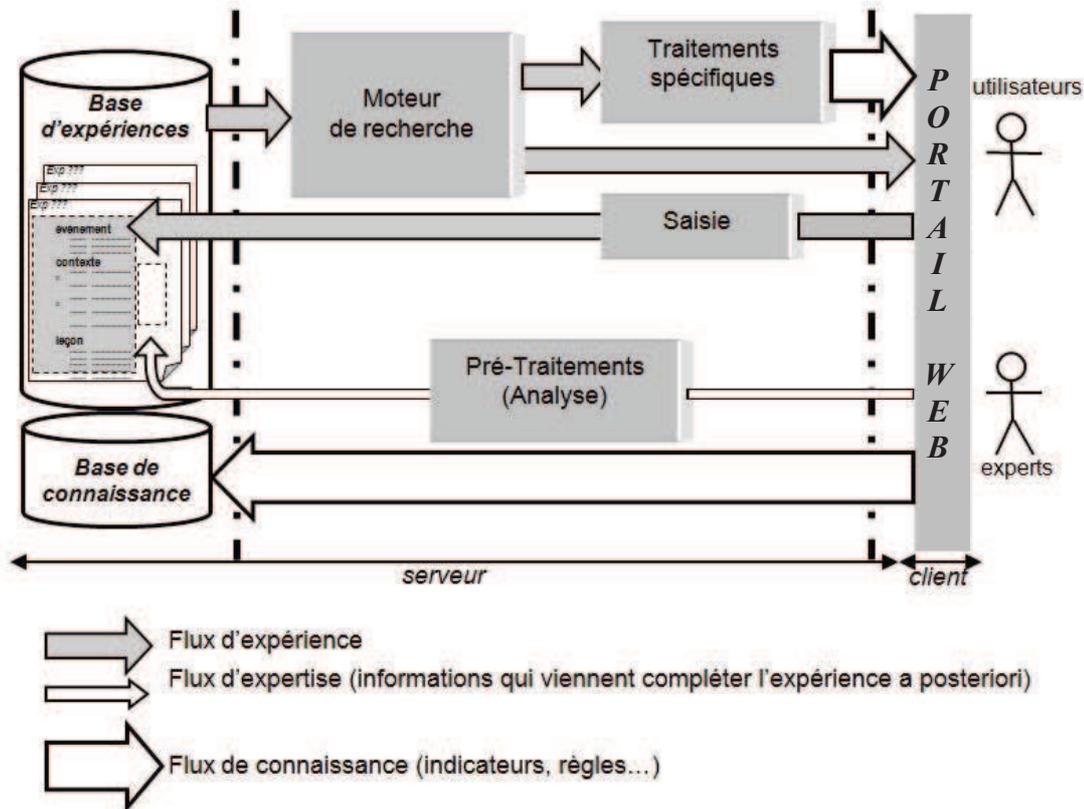


Figure V-3 - Principe et architecture des applications de retour d'expérience

Les quatre blocs représentés sur la figure correspondent à des fonctionnalités majeures qui sont elles-mêmes associées à des composants standard des applications :

- Saisie : il s'agit d'une interface de type formulaire qui permet de saisir l'événement et le contexte d'une expérience. Ce composant est distinct de celui d'analyse des expériences.
- Analyse : il s'agit d'un composant qui permet de sélectionner parmi le contexte un ensemble d'attributs significatifs. Cette opération peut être renouvelée (pour fournir plusieurs explications possibles) et il est possible de préciser la confiance globale, la justesse des explications proposées selon l'expert et d'autres informations spécifiques à l'expérience le cas échéant.
- Moteur de recherche : il s'agit d'un composant qui implémente les deux algorithmes de similarité globale. L'interface d'entrée correspond à un formulaire qui permet de décrire le contexte courant et ainsi d'initier la recherche basée sur les analyses. En ce qui concerne les recherches de type requêtes, interrogation de la base, il faut prévoir une autre interface autorisant des requêtes simples sur les expériences contenues dans la base (expériences concernant un groupe d'acteurs, survenues depuis une date donnée...). Bien qu'il soit indéniablement utile, ce dernier aspect « pratique » n'a cependant pas été développé en profondeur car il correspond au développement de traitements classiques. Le résultat des différentes « recherches » est présenté sous la forme d'un tableau d'expériences qui peut être ordonné suivant toutes ses caractéristiques dont principalement le nom de l'événement, la gravité ou bien le score de similarité. Ce tableau présente un résumé (paramétrable) et permet

d'accéder à une description plus complète de l'expérience en sélectionnant la ligne correspondante (composant d'affichage de l'expérience).

- Traitements spécifiques : il s'agit de l'ensemble des traitements orientés extraction des connaissances qui sont utilisés en complément du moteur de recherche. Nous trouvons à ce niveau, par exemple, l'algorithme d'extraction de la mesure du risque. Dans la perspective d'autres extractions de connaissances, c'est aussi à cet endroit que seront recensés les outils permettant de trouver des invariants dans les expériences. Ce bloc comprend aussi les outils d'adaptation de solution permettant de réutiliser des solutions. En sortie de ce bloc, nous avons des composants permettant de présenter la ou les nouvelles solutions, de la ou les modifier et de la ou les accepter ou non (insertion dans la base d'expériences). Dans notre cas, puisque nous n'avons pas pris en considération la composante « solution » des expériences, il y a simplement la présentation de la mesure du risque, sous forme d'une distribution de masses ainsi que quelques informations complémentaires comme le nombre d'expériences prise en compte dans la mesure.

Aucun bloc en relation avec la modélisation de l'expérience n'est représenté sur la Figure V-3 car il s'agit d'un schéma des applications en fonctionnement. Nous supposons que les tâches de modélisation interviennent avant la mise en service des applications. Il s'agit de la configuration du système par la définition du modèle de l'expérience, c'est-à-dire l'échelle de gravité des événements, l'ensemble des descripteurs (domaine et fonction de similarité associée pour chaque descripteur). A l'avenir, ces étapes de modélisation devraient être réalisées au moyen d'interfaces graphiques dans une partie spécifique d'administration du système.

## V.3. Modélisation générique

La généricité est un objectif constant dans le monde du développement logiciel permettant d'automatiser certaines tâches répétitives dans la réalisation d'un système particulier. Plusieurs techniques existent pour favoriser la généricité et nous présentons celles qui nous semblent être les principales, à savoir l'utilisation de *patterns*, les *frameworks*, la métamodélisation liée au domaine de l'ingénierie des modèles et enfin la métaprogrammation.

### V.3.1. Patterns

Les *patterns* (patron de conception logiciel en français) sont des architectures particulières qui peuvent être appliquées pour résoudre une classe de problèmes récurrents. Ce sont des collections de bonnes pratiques accumulées au fil du temps. Nous pouvons d'ailleurs faire un parallèle avec le retour d'expérience puisque les *patterns* sont des catalogues de solutions pour des familles de problèmes particulières. Les concepteurs, à force de résoudre des problèmes récurrents, ont fait émerger des architectures et des modèles génériques apportant des solutions à ces problèmes. Il existe plusieurs familles de *patterns*, dont ceux de conception (Gamma *et al.*, 1995) (Alpert *et al.*, 1998) ou encore d'analyse (Fowler, 1997). Les *patterns* sont des modèles d'implémentation ou d'analyse réutilisables. Contrairement aux autres techniques, il ne s'agit pas ici de code exécutable mais uniquement de modèle abstrait de conception ou d'analyse récurrent. Les *patterns* visent une certaine généricité et surtout sont des outils d'experts pour résoudre des problèmes liés à du code existant afin d'implémenter des solutions élégantes et extensibles. Cependant, l'usage des *patterns* n'est pas à généraliser dès les premières conceptions. Ils peuvent rendre les systèmes difficiles à comprendre et parfois trop complexes pour la tâche à réaliser. Il est toujours préférable de partir d'une solution simple quitte à la reprendre et à la reconcevoir avec l'appui d'un *pattern* par la suite (Fowler, 1997).

Afin de faciliter la compréhension et la recherche d'information, nous conservons les noms anglais des différents *patterns* cités (ces noms sont notés en italique).

### V.3.1.1. Patterns de généricité pour les modèles dynamiques

Il existe plusieurs *patterns* permettant de modéliser des modèles adaptables ou dynamiques et ceux-ci sont organisés dans des architectures génériques qui sont, d'une certaine manière, des *patterns* plus complexes (composites). Un des plus généraux est AOM<sup>1</sup> (Adaptive Object Model) (Foote et Yoder, 1998) (Yoder *et al.*, 2001) (Yoder et Johnson, 2002). Les AOM correspondent à des architectures génériques, basées sur des *patterns* classiques, qui visent à servir de support à la construction de systèmes dynamiques et adaptables. Les auteurs utilisent les métadonnées et les modèles objets adaptables pour décrire des systèmes adaptables en utilisant des structures (modèles) dynamiques, des règles métiers et des vues variables.

Nous ne présentons ici qu'un sous-ensemble de cette architecture, à savoir le Dynamic Object Model (DOM) (Riehle *et al.*, 2000) (Riehle *et al.*, 2005) et les *patterns* qui le composent car c'est cet aspect qui nous intéresse. Les AOM apportent, par l'utilisation de *patterns* supplémentaires, un support pour la modélisation des règles et des processus métiers (toujours de façon dynamique et adaptable). Nous n'avons pas un réel besoin d'adaptabilité des processus de recherche du retour d'expérience car il n'existe pour l'instant qu'un seul type de traitement (recherche par similarité). C'est pourquoi nous nous limitons à l'aspect dynamique des modèles définissant les entités et les propriétés qui leur sont associées.

Le DOM est représenté sur le schéma de la Figure V-4. Le cœur du DOM est composé des classes Composant, ComponentType, Property et PropertyType, ainsi que de leurs clients respectifs.

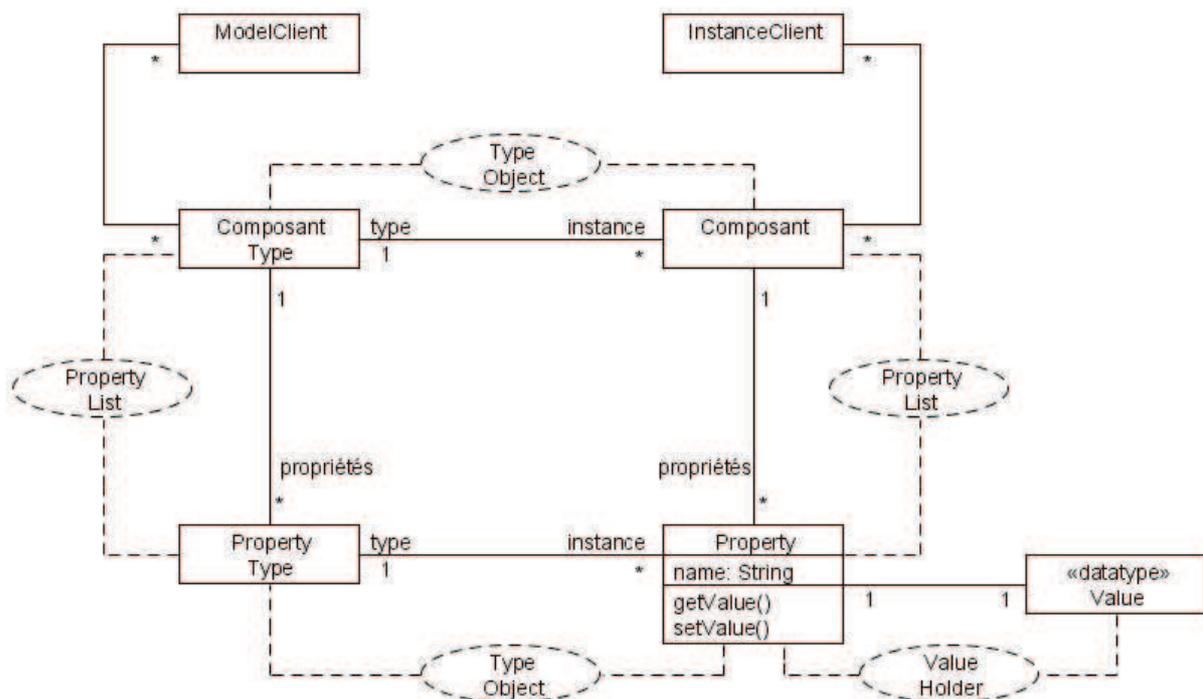


Figure V-4 - Dynamic Object Model (Riehle *et al.*, 2000)

Le DOM utilise les *patterns* suivants :

- le *pattern Type Object* (Johnson et Woolf, 1997) permet à un objet de posséder un ensemble d'objets « Property » qui sont accessibles par un client et qui agissent comme autant de variables d'instance, une variable d'état classique. Les objets « propriété » sont obtenus et créés/modifiés dynamiquement, en utilisant typiquement des chaînes de caractères, leur nom, comme moyen d'accès. Ce *pattern* réifie la relation d'instanciation en la déclarant comme une simple association entre deux classes dont une représente l'objet (l'instance) et l'autre son type (la classe).

<sup>1</sup> <http://www.adaptiveobjectmodel.com/>

- Les *patterns Property List* définis dans (Riehle1997a), (Foote et Yoder, 1998) permettent de regrouper plusieurs propriétés d'un objet dans une seule variable d'instance qui les contient toutes. Ce type de *pattern* est très approprié pour les classes qui contiennent potentiellement de nombreuses propriétés pas toujours définies.
- Le *pattern Value Holder* (Gamma *et al.*, 1995) qui permet à un client de récupérer un objet valeur à partir d'un objet *Value Holder*. Un objet *Value Holder* sert d'adaptateur pour la valeur qu'il rend disponible au client en utilisant une interface homogène (`getValue()` et `setValue()` par exemple). Les objets *Value* peuvent être primitifs ou non-primitifs. (Cunningham, 1995) (Baumer *et al.*, 1998).

Nous constatons la double utilisation des *patterns TypeObject* et *PropertyList*. L'idée majeure de cette architecture est de réifier (modéliser) le lien « instance de » entre une classe et son instance (entre le niveau M1 et M0 du MOF) par une simple association entre deux objets (chacune de niveau M0), l'un représentant la classe et l'autre l'objet (l'instance). Le *pattern Value Holder* est simplement associé à la classe *Property*. Ces *patterns* utilisent tous la séparation entre niveau « connaissance » et niveau « opérationnel » présentée dans (Fowler, 1997). Une partie du diagramme représente le modèle et l'autre ses instances. La partie gauche de la Figure V-4 correspond au niveau « connaissance » (ou modèle) et la partie droite au niveau « opérationnel » (ou instance).

Il est possible d'appliquer le *pattern Property List* seul mais l'interprétation du type de données est laissée libre. C'est pourquoi, pour réaliser une vérification de type, le *pattern Type Object* est appliqué deux fois. Le DOM étant une composition de *patterns*, il en résulte un ensemble de contraintes qui sont imposées à chaque *pattern* :

- contrairement au *pattern Type Object* seul, les types sont maintenant décrits explicitement par la structure de ces instances,
- contrairement au *pattern Property List* seul, les propriétés sont maintenant contraintes par leurs types.

### V.3.1.2. Critique des patterns AOM et DOM

Ces solutions sont intéressantes car elles confèrent aux modèles une grande flexibilité. En contrepartie, il y a des limitations au niveau des performances et de la complexité. Dans, la suite de cette section, nous listons succinctement les avantages et inconvénients des compositions de *patterns* présentées (Riehle *et al.*, 2000).

Les avantages de ces approches dynamiques et adaptables sont notamment :

- la réduction de la complexité d'un système qui contient de nombreux types qui ne diffèrent que selon certains attributs (présence de moins de classes),
- l'adaptation à des évolutions fréquentes du logiciel, les utilisateurs du logiciel résultant peuvent créer de nouveaux types de données dont ils ont besoin sans avoir à modifier le code source (création des classes du modèle puis de ses instances en fonctionnement),
- la possibilité de définir un DSL (Domain Specific Language), un langage métier spécifique et indépendant.

Les inconvénients sont :

- le modèle conceptuel implémenté est plus difficile à comprendre. La structure logique du modèle créé dynamiquement n'est pas explicite. A cause de la complexité additionnelle, les développeurs peu expérimentés peuvent avoir des problèmes de compréhension conceptuelle (Yoder et Johnson, 2002). Les classes ne sont plus des entités monolithiques, elles correspondent à des assemblages d'objets.
- une classe est une instance de *ComposantType* et ses propriétés (variables d'instance) sont des instances de *PropertyType*.
- un objet (« instance » de la classe ci-dessus) est une instance de *Composant* et ses propriétés (variables d'instance) sont des instances de *Property*.

- Il faut relier le modèle dynamique aux traitements, c'est-à-dire :
  - il faut créer des outils de développement adaptés. Le modèle dynamique, car instancié, n'est plus accessible directement dans l'IDE du langage de programmation, seul son métamodèle (cf. partie V.3.3.1) est accessible.
  - Il faut adapter les traitements que l'on veut appliquer aux modèles. Autrement dit, les comportements des objets, habituellement définis dans les classes monolithiques, doivent tenir compte de la structure dynamique. Il convient alors d'utiliser un ensemble de *patterns* standard comme *Chain of Responsibility*, *Interpreter*, ou *Observer* dont les éléments d'articulation sont détaillés dans (Johnson et Woolf, 1997).
- Les performances et l'occupation mémoire sont moins bonnes qu'avec une implémentation directe à cause des indirections d'exécution induites par la prise en compte des modèles de modèles lors du fonctionnement du système. Si les performances sont un problème, il vaut mieux alors envisager des techniques de métaprogrammation (décrites dans la partie V.3.3.2) qui génèrent le modèle et n'ont pas à l'interpréter en permanence. Typiquement, la vérification du *typage* nécessaire et l'accès aux propriétés peuvent être optimisés en utilisant des identifiants uniques et des tables de hachage (Riehle *et al.*, 2000).

## V.3.2. Framework

Les bibliothèques logicielles et les *patterns* sont parfois composés dans des *frameworks*, de façon à constituer un ensemble cohérent et complémentaire. Un *framework* (cadre d'application ou cadriceil en français) est une infrastructure logicielle orientée objet qui est suffisamment générique pour pouvoir être appliquée (on dit spécialisée) dans plusieurs domaines permettant ainsi de réaliser des applications voisines à moindre coût. Il s'agit en quelque sorte d'un squelette d'application qui peut être étendu et précisé par un développeur afin de répondre à une problématique donnée (Johnson, 1997). Ce sont des applications réutilisables incomplètes qui sont spécialisées et complétées pour créer des applications adaptées particulières (Johnson et Foote, 1988). Comme les bibliothèques logicielles, un *framework* aide le développeur à réutiliser l'existant sous la forme de briques logicielles pré-établies et génériques qu'il faut assembler et étendre selon les possibilités offertes par le *framework*. Une grande partie du travail est donc déjà fait si l'application que l'on souhaite développer est conforme aux possibilités du *framework*.

### V.3.2.1. Une approche modulaire, réutilisable et extensible

Les bénéfices des *frameworks* sont multiples et dépassent ceux des bibliothèques de fonctions et des *patterns* en permettant une approche modulaire, réutilisable, extensible et basée sur l'inversion de contrôle (Fayad et Schmidt, 1997) aussi connu sous le nom de principe d'Hollywood (c'est le *framework* qui « contacte » les applications spécialisées et non l'inverse). Les *frameworks* améliorent la modularité en encapsulant les détails d'implémentation volatiles derrière des interfaces stables. Ces interfaces améliorent aussi la réutilisabilité en définissant des composants génériques qui permettent d'améliorer la productivité des développeurs mais aussi la qualité globale, les performances, la fiabilité et l'interopérabilité du code réalisé. L'extensibilité est obtenue au moyen des méthodes d'ancrage explicites (hook method) ou points d'entrée qui permettent aux développeurs de proposer des variations de traitements tout en respectant les interfaces stabilisées (Pree, 1994). Cette caractéristique est essentielle pour permettre l'adaptation d'un *framework* à un cadre spécifique d'application (par exemple, définir une gestion des erreurs adaptée). D'autres classes et méthodes sont internes et généralement non amenées à évoluer sauf en cas d'extension du *framework* lui-même. L'inversion de contrôle est une caractéristique forte du fonctionnement de l'architecture basée sur un *framework* (Riehle, 2000) qui définit une différence fondamentale avec les bibliothèques. Les bibliothèques sont à la disposition des programmes qui les utilisent, il n'y a pas vraiment d'ordre à respecter, ni de restriction puisqu'il suffit d'appeler les fonctions quand c'est nécessaire. En revanche, les *frameworks* contraignent le programme réalisé car celui-ci doit respecter des conventions dictées par le principe d'inversion de contrôle. Ce principe qui semble assez contraignant a pour objectif de limiter les problèmes de dépendances en

empêchant les composants de haut niveau (les composants du *framework*) de dépendre des composants de bas niveau qui sont ceux de l'application spécialisée effective.

L'inversion du contrôle est réalisée, le plus souvent, par des classes particulières à sous classer et des méthodes particulières à étendre. Nous parlons alors de points d'entrée du *framework*. Par exemple dans le *framework* web Seaside<sup>1</sup>, que nous avons utilisé pour la réalisation des interfaces, pour créer un composant, il faut sous classer la classe générique `WComponent` et l'affichage est réalisé par l'appel de la méthode `MyComponent>>renderContentOn` : qui est redéfinie dans le composant créé. Dans les cycles de rendus, cette méthode sera appelée à un moment donné par le *framework* web. Aussi, la méthode d'initialisation des composants doit être étendue (et non redéfinie) sans quoi les composants ne seraient pas bien configurés.

### V.3.2.2. Les limites des frameworks

Utilisé en conjonction avec les *patterns*, les bibliothèques et les composants, les *frameworks* peuvent considérablement accélérer le développement d'une application et en améliorer la qualité. Néanmoins, plusieurs limites existent. Tout d'abord, du point de vue utilisateur, pour un *framework* donné, il faut que l'application cible soit compatible avec le périmètre du *framework*. Il faut donc nécessairement que les points d'entrée soient suffisamment nombreux et appropriés pour pouvoir adapter le *framework* à son contexte applicatif. Si ce n'est pas le cas, ce qui est assez fréquent, il faut étendre le *framework* et cette tâche est souvent complexe car il faut bien comprendre les objectifs et choix technologiques du *framework* considéré. Les *frameworks* étant relativement complexes du fait de la généralité mise en œuvre et de l'utilisation fréquente de *patterns*, il n'est pas simple de bien en comprendre les limites. L'apprentissage d'un *framework* demande un investissement et un effort considérable qui ne sont pas toujours justifiés par rapport à la complexité de l'application escomptée (Fayad et Schmidt, 1997). Si en plus le cadre ne correspond pas exactement au besoin, alors le *framework* devient très difficile à utiliser car il faut avant tout l'étendre et cette tâche est souvent très complexe et parfois impossible pour un coût donné.

Il existe un *framework* orienté objet générique implémentant des applications de RàPC, CBR\*Tools (Jaczynski et Trousse, 1997) mais nous n'avons pas pu l'utiliser directement car il n'accepte comme type de données que des types standard. Ce cadre vise tout particulièrement des domaines d'application nécessitant une réutilisation de cas devant être indexés par des situations comportementales, ce qui n'est actuellement pas notre préoccupation majeure. Il nous aurait donc fallu étendre le système pour tenir compte des descripteurs incertains ce qui aurait été complexe et sans garantie de résultat. Un autre inconvénient des *frameworks* est la relative perte de performances induites par les différents niveaux d'indirection employés pour permettre l'extensibilité. Ensuite, du point de vue du concepteur de *framework*, il y a aussi des contraintes importantes. Un effort de développement et d'abstraction plus important que pour le développement d'applications complexes est nécessaire. Il faut aussi bien cerner les limites du système, c'est-à-dire déterminer ce qui doit être rendu générique et extensible. D'ailleurs, la réalisation d'un *framework* est rarement immédiate. En principe, un tel développement est la conséquence de la réalisation d'applications plus traditionnelles présentant suffisamment de similarités pour envisager la généralisation de certains aspects. De la même façon qu'il faut reléguer les considérations de performance des applications à des stades de développement ultérieurs, il ne faut pas tenter de réaliser des développements génériques de type *framework* en première approche. L'intégration entre *frameworks* différents est aussi difficile à réaliser car il faut prévoir des passerelles et donc bien connaître les contraintes des différents *frameworks* mis en relation. Enfin, la maintenance est aussi difficile car les exigences évoluent et les applications basées sur des *frameworks* doivent évoluer avec elles.

Les techniques présentées dans cette partie sont couramment utilisées dans le domaine du Génie Logiciel. Nous présentons maintenant des techniques plus abstraites connues sous le nom de métamodélisation ou encore métaprogrammation.

---

<sup>1</sup> <http://www.seaside.st>

### V.3.3. Les approches *Meta*

Le préfixe *meta*, d'origine grecque et qui signifie « après », « au-delà de », est souvent employé aujourd'hui dans le monde de la programmation pour exprimer la réflexivité. De la même manière que la métaphysique est la « physique de la physique », une méta-information est une information sur l'information, un métamodèle est un modèle de modèle... Nous pouvons distinguer à ce niveau deux approches principales : la métaprogrammation et la métamodélisation. Il semble que ces techniques de métaprogrammation et métamodélisation sont les directions que le développement logiciel prend aujourd'hui pour succéder à l'orienté objet largement utilisé dans des domaines très variés. Les deux techniques ne sont pas nouvelles mais elles connaissent aujourd'hui un regain d'intérêt certainement lié au besoin d'adaptation aux contextes changeants auxquels les entreprises doivent faire face.

Bien qu'elles reposent souvent sur l'utilisation des *patterns* de généricité présentés dans la partie V.3.1., ces deux techniques ont cependant des propriétés différentes :

- La métamodélisation permet :
  - de favoriser l'interopérabilité (approche IDM ou MDA – cf. V.1) en fournissant une alternative au Middleware<sup>1</sup> pour faire coexister des systèmes différents et ainsi les rendre interopérables,
  - de créer des modèles dynamiques exécutables (approches AOM – cf. partie V.3.1).
- La métaprogrammation permet de générer des modèles par l'interprétation de métadonnées permettant d'automatiser certaines tâches,

La métamodélisation et métaprogrammation sont donc des techniques qui semblent particulièrement bien adaptées à notre problématique d'opérationnalisation. Nous allons détailler les détailler dans les sections suivantes. En ce qui concerne la métamodélisation, nous nous sommes focalisés sur un type de métamodèle particulier qualifié de métamodèle exécutable qui vise une opérationnalisation simplifiée.

Dans cette partie, nous présentons à la fois les métamodèles « exécutables » et la métaprogrammation.

#### V.3.3.1. Métamodélisation

Concernant la réalisation de métamodèle « exécutables », nous établissons un rapprochement avec la partie concernant les *patterns* de généricité. En effet, l'idée de base est de reprendre les principes de l'AOM. Nous utilisons un exemple de métamodèle défini dans (Knublauch, 2002) pour illustrer ce propos. Ce métamodèle généralise le métamodèle d'OKBC (OKBC) et représente le cœur d'un système de représentation des connaissances. Ce métamodèle présente, entre autres caractéristiques, la capacité de modélisation des facettes qui sont des attributs particuliers. Ceci est représenté sur le diagramme de classe UML de la Figure V-5.

Sur le diagramme de ce métamodèle, deux utilisations du *pattern Type Object* (entre Instance et Class ainsi que entre RefProperty et Class) ainsi que l'utilisation de *pattern Property List* (entre Class et Property) sont représentées. Ce modèle (dynamique) permet aussi de représenter des ontologies (au sens représentation des connaissances de domaine) simples basées sur le paradigme orienté objet (notion de superclasse unique, agrégation de propriétés, ...). Ce modèle permettrait d'obtenir un langage plus adapté à la représentation des connaissances de domaine. Nous nous sommes, cependant, limités à des objets simples, de type liste de valeurs alors que ces valeurs pourraient être des éléments de connaissance non-atomique. La différence entre ce modèle et le modèle que nous proposons (cf. V.4) est que ce dernier « métamodélise » l'entité expérience et la notion de descripteur incertain. Le modèle présenté sur la Figure V-5, nous permettrait de pouvoir modéliser et associer au modèle d'expérience des entités de domaine plus précises.

<sup>1</sup> Système passerelle qui fait la jonction entre deux systèmes hétérogènes mais compatibles

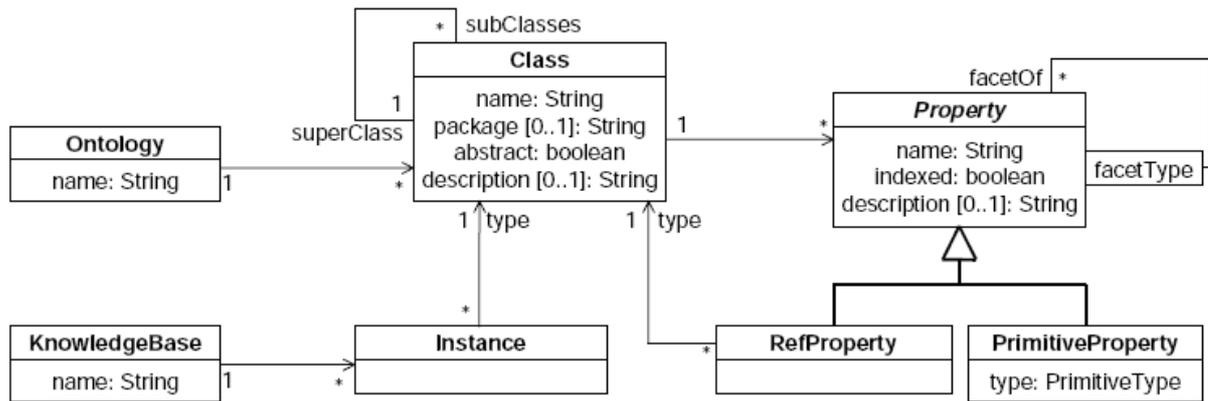


Figure V-5 - Métamodèle de K-Beans (Knublauch, 2002)

### V.3.3.2. Métaprogrammation

La métaprogrammation consiste à écrire des programmes qui écrivent ou manipulent d'autres programmes (ou eux-mêmes), considérant ces derniers comme leurs propres données. Quand un langage ou un formalisme de modélisation peut être défini ou redéfini par sa propre syntaxe et ainsi « agir » sur lui-même, il est dit réflexif (Maes, 1987). Les différentes implémentations du langage de programmation Smalltalk comme VisualWorks ou Squeak (Black *et al.*, 2007), sont des systèmes réflexifs. Smalltalk permet en effet de récupérer des informations sur lui-même (introspection) et il peut être modifié (intercession, modification de la structure de base). Cette caractéristique favorise la métaprogrammation et la métamodélisation. Dans la plupart des cas, cela permet aux programmeurs d'automatiser la génération du code qu'ils auraient autrement dû écrire. Le langage avec lequel le métaprogramme est écrit est nommé métalangage. L'idée est de représenter le langage de base à partir du métalangage (Wuyts, 2001). Ainsi, la métaprogrammation permet de générer ou de modifier en fonctionnement des programmes ou des fonctionnalités.

Il existe plusieurs types de métaprogrammation basées sur :

- la génération de code,
- le protocole à méta-objet (MOP : Meta Object Protocol, consiste à exposer les composants internes de l'exécution, du fonctionnement d'un langage au moyen d'API spécifiques) (Kiczales et al, 1991)
- l'interprétation de métadonnées

L'inconvénient majeur de la technique de génération de code est que les systèmes générés sont assez difficiles à maintenir et à garder synchronisés, bien qu'avec des MOP adaptés, cette limitation est moindre. L'approche interprétative est plus flexible. Si nous développons dès le départ le système avec des annotations, les possibilités du système obtenu sont intéressantes et la séparation aspect statique du code et *meta* est bien réalisée. Par contre, pour utiliser les annotations sur des systèmes existants, la tâche est plus complexe puisque la correspondance entre le métamodèle et le code objet doit être faite manuellement. En effet, contrairement au MOP, il n'y a pas exactement de connexion causale entre les annotations et le code (Renggli *et al.*, 2007).

Avec des langages dynamiques qui sont introspectifs (les programmes sont capables d'observer leur propre exécution) et réflexifs comme LISP ou Smalltalk, il est possible d'agir directement sur la structure des programmes en utilisant le Méta-Object Protocol (MOP). Les langages à typage statique comme Java et C++, permettent difficilement de telles interventions sur la structure des programmes (Riehle et, 2005). Alors que le MOP agit directement (et éventuellement en fonctionnement) sur les entités statiques et dynamiques du langage, les métadescriptions fonctionnent par interprétation à l'exécution du programme. Les tâches de métaprogrammation réalisées avec le MOP sont performantes mais parfois assez difficiles à comprendre. De plus, la sémantique du langage *meta* est relativement

limitée (par exemple, si l'héritage multiple n'est pas une caractéristique du langage, il n'est pas possible de « métaprogrammer » cet aspect). Le principe des annotations est plus simple. Il s'agit finalement d'un ensemble de métadonnées associé à des entités particulières du modèle et c'est l'interprétation de ces métadonnées qui permet d'automatiser certains aspects. Cet ajout est moins coûteux que d'avoir à développer toutes les tâches qui peuvent ainsi être réalisées automatiquement à partir de leurs interprétations. C'est une technique particulièrement utile pour la configuration d'interfaces utilisateurs personnalisées, la sérialisation des données, le contrôle de validité... Ces annotations, reliées à des définitions de méthodes, n'influencent pas le comportement de ces méthodes. Elles permettent de réaliser des requêtes avec le langage de programmation, ce qui les rend particulièrement utiles pour coupler la programmation déclarative avec la programmation impérative des langages de programmation orientés objet tel Java, Smalltalk, C++... Les requêtes sur les annotations permettent également de faire de la métaprogrammation sur des systèmes existants (*legacy system*)... (Renggli *et al.*, 2007).

Nous présentons dans la partie suivante le modèle du système de retour d'expérience retenu.

## V.4. Modélisation et implémentation

Au cours des développements réalisés au cours de ces travaux de thèse, nous avons eu à faire un compromis entre les possibilités offertes par les représentations « meta » et les objectifs de représentation générique. Il nous a semblé préférable alors de développer un environnement dynamique à un niveau « meta » basée sur l'approche de métadescriptions et à lui associer des *patterns* de généralité. La définition de ces métadescriptions est finalement le modèle de l'expérience. En fait, c'est à partir de la définition des attributs, de leurs domaines respectifs et des fonctions de similarités associées que le modèle de l'expérience d'un domaine particulier est défini.

Par conséquent, le système qui est proposé peut-être utilisé sur plusieurs domaines. Les experts peuvent même faire évoluer leur définition du domaine par contre, nous n'avons pas abordé la possibilité de modifier le modèle en fonctionnement (à partir du moment où des expériences ont été définies, il n'est pas possible de modifier le domaine). Nous pensons cependant qu'il est réaliste d'envisager à terme la modification de modèle en fonctionnement. Par exemple, l'ajout d'un nouveau descripteur ne devrait pas poser de problème, les expériences existantes auraient par exemple une incertitude totale sur la valeur associée. Les cas plus complexes sont ceux de fusion d'information ou de précision. Nous pensons cependant que dans ces cas les techniques de transformation de modèles pourraient être utiles de même que les possibilités d'extension ou de réduction de cadre de discernement qu'offre la théorie des fonctions de croyance.

### V.4.1. Choix d'implémentation

Cette partie est consacrée à la présentation et à la justification des différents choix en rapport avec les langages et outils que nous avons utilisés.

#### V.4.1.1. Smalltalk, un langage tout objet

Smalltalk est un langage objet typé dynamiquement (*late binding*), réflexif et introspectif (Goldberg et Robson, 1983). Les langages dynamiquement typés souvent décriés à tort pour leur inefficacité et le manque de sécurité connaissent aujourd'hui un renouveau, en particulier lié au développement de Ruby et du *framework* Ruby on Rails. Ils offrent une grande flexibilité et permettent des développements rapides (Nierstratz *et al.* 2005). Smalltalk est un langage reconnu pour le prototypage. Son modèle objet est simple (héritage simple, variables d'instance privées, méthodes publiques...) et très uniforme puisque dans Smalltalk tout est objet. D'ailleurs, l'écriture d'applications consiste souvent en l'extension

ou la réutilisation de parties d'applications existantes. Squeak<sup>1</sup> (Ingalls *et al.* 1997), (Black *et al.* 2007) est une implémentation libre et portable de langage et de l'environnement de programmation Smalltalk, basée sur Smalltalk-80 et intégrant des possibilités multimédia. Squeak en outre est écrit en lui-même, ce qui permet de dépasser les limites du MOP puisqu'il est possible d'étendre le modèle objet et de modifier la machine virtuelle en rapport. Il faut aussi remarquer qu'un métamodèle coexiste implicitement, une métaclasse est créée automatiquement à chaque création de classe.

Contrairement aux autres langages où la notion de fichier est centrale, cette dernière est quasiment absente de Smalltalk, c'est simplement un support d'entrée/sortie. L'ensemble de l'environnement, des classes, des objets, des méthodes est contenue dans une « image » qui, bien que matérialisée par un fichier (associé à un fichier source et un fichier change) abstrait complètement la notion de fichier source présent dans les langages de programmation plus classiques<sup>2</sup>. L'image est un espace « vivant », où les objets co-existent et interagissent. Il est plus simple de concevoir l'image comme une image virtuelle d'un système d'exploitation exécuté virtuellement. Avec Smalltalk la frontière est en fait très étroite entre langage/environnement de programmation et système d'exploitation. Il existe d'ailleurs une variante de Squeak, un dialecte Smalltalk open-source qui fonctionne comme un système d'exploitation (c'est-à-dire sans passer par un système d'exploitation classique) : SqueakNOS<sup>3</sup>.

### V.4.1.2. Adéquation de Smalltalk à notre problématique

Les notions d'« image » et d'approche « tout objet » relatives à Smalltalk (Goldberg et Robson, 1983) sont certainement le frein le plus important à l'apprentissage de ce langage. Le changement par rapport à des langages de programmation plus conventionnels est délicat mais, une fois effectué, il apparaît clairement que Smalltalk est un outil idéal pour la réalisation de prototype et démonstrateurs. L'un des avantages de cette approche par image est qu'il n'y a pas de besoin de sérialisation des objets. Sauvegarder l'image revient à prendre un instantané de l'environnement que l'on retrouve exactement lorsque l'application est à nouveau lancée (même positionnement des fenêtres, même état pour les objets...).

Nous avons vu plusieurs avantages dans l'utilisation de Smalltalk pour notre objectif de réalisation d'application :

- Dans Smalltalk tout est objet, il n'a pas de typage fort des variables. Le typage faible ou « late binding » (typage au dernier moment lors de l'exécution) permet de ne pas avoir à prévoir les objets qui seront contenus dans les ensembles, les tableaux... ce qui nous a semblé être un avantage important par rapport aux langages statiques tel que Java. Il est vraiment plus aisé en Smalltalk de réaliser des collections, des tableaux d'objets hétérogènes. Comme nous pouvons le voir dans les algorithmes présentés au chapitre III, il y a beaucoup de traitements basés sur des tableaux qui représentent des distributions, tableaux qui peuvent être mis en association avec des expériences, le tout à nouveau stocké dans des tableaux. Sans un langage au typage dynamique, il aurait été très complexe de déclarer de tels types d'entités.
- Les itérateurs en Smalltalk sont nombreux et puissants. Ces sont des éléments du langage qui permettent d'itérer sur des collections d'objets. Ils permettent entre autre d'exprimer facilement des contraintes proches du langage OCL. Par exemple, l'expression ci-dessous permet de sélectionner les objets appartenant à une collection dont l'information date (une propriété) est postérieure au 12 décembre 2007.
 

```
aCollection select: [:element | element date > '12/12/2007' asDate]
```
- La nature réflexive de Smalltalk (Maes, 1987) permet a un programme en cours d'exécution de changer, d'ajouter ou de retirer des parties certains aspects.

<sup>1</sup> <http://www.squeak.org>

<sup>2</sup> Il existe des dialectes de Smalltalk (ex : GNU Smalltalk) n'utilisant pas d'image nativement.

<sup>3</sup> <http://sourceforge.net/projects/squeaknos/>

### V.4.1.3. Métaprogrammation et Réflexivité

Dans cette partie, nous illustrons certains aspects de métaprogrammation relatifs à Smalltalk, à savoir les capacités réflexives du langage et les possibilités de métaprogrammation. Enfin nous introduisons un *framework* de meta-description appelé Magritte (Renggli, 2006) (Renggli *et al.*, 2007).

#### Réflexivité

Les capacités réflexives intrinsèques de Smalltalk sont très performantes. Comme nous l'avons déjà spécifié, Smalltalk est un langage dynamiquement typé (late binding), et ses capacités réflexives permettent d'inspecter et d'interroger le code (introspection), mais aussi d'agir sur ce même code en modifiant, créant de nouvelles classes ou méthodes par exemple (introspection). Il est donc relativement direct en Smalltalk de créer de nouvelles classes et ainsi, par exemple, d'étendre un *framework*.

Pour illustrer les capacités réflexives, voyons une méthode communément utilisée et très puissante qui permet d'envoyer un message de manière générique (en Smalltalk, les messages sont les méthodes). La méthode `Object>>perform:` (ou `Object>>perform:withArgument:`) permet d'interpréter un message réifié. Par exemple si le prototype de la méthode utilisée pour comparer deux objets par similarité est :

```
anObject
  compareTo: anotherObject
  using: aSimilarityFunction
```

Alors, l'équivalent après réification du message est :

```
anObject
  perform: #compareTo:using:
  withArguments: #(anotherObject aSimilarityFunction)
```

Cette possibilité du langage est très puissante et illustre les capacités réflexives du langage. Pour illustrer les aspects relatifs à l'intercession (modification du code) et plus particulièrement à la métaprogrammation, nous présentons un exemple tiré d'une des solutions que nous avons envisagées et qui met en œuvre des techniques de métaprogrammation.

#### Métaprogrammation, génération de code

Nous avons, à un moment donné pensé réaliser le cadre générique sous forme de *framework*. Cependant, les *frameworks*, pour être opérationnalisés, nécessitent d'étendre le système en des points définis (les points d'entrée – cf. V.3.2). Dans la solution réalisée, nous devons sous-classer une classe descripteur générique (nommée `EFDescriptor`) autant de fois que le modèle le nécessitait. Il fallait en outre redéfinir la méthode `EFDescriptor class>>domaine` pour qu'elle retourne la liste des éléments du domaine et `EFDescriptor class>>similarityTable` pour qu'elle retourne la matrice de similarité. L'inconvénient de cette solution est qu'il fallait directement coder ces nouvelles classes dans l'environnement de programmation.

Nous montrons comment il est possible en Smalltalk de générer ce code particulier. Une illustration donnée sur la Figure V-6 permet de créer 3 sous-classes de descripteurs (`EFSUPMeteo`, `EFSUPCondition` et `EFSUPExperience`) avec leurs domaines respectifs sans matrice de similarité pour ne pas alourdir l'exemple. Nous observons 3 principaux éléments sur l'exemple de cette méthode d'illustration :

- L'introduction des paramètres d'entrée (ligne 5) qui, en pratique, se trouverait dans un fichier externe ou dans une base de données. Nous avons fait le choix sur cet exemple d'utiliser des associations entre un symbole représentant le nom du nouveau descripteur et un tableau contenant les valeurs du domaine.
- La définition d'une fonction (ligne 10) qui permet de sous-classer `EFDescriptor` en prenant comme paramètre un symbole (le nom du descripteur créé). Nous avons utilisé ici un bloc (Closure) pour encapsuler dans une fonction le message de création de classe. Une meilleure

conception serait certainement de créer une nouvelle méthode dans la classe `EFDescriptor` qui prendrait en charge la création de nouvelles classes de descripteurs, par exemple, `EFDescriptor>>createDescriptorNamed:domain:similarity:`

- Le traitement effectif et la génération du code sont réalisés seulement dans la troisième et dernière partie de l'exemple. Le système itère sur la liste de paramètres d'entrée, crée la classe en appelant le bloc (fonction abstraite) défini au-dessus, et enfin, accède à la nouvelle classe créée et compile la méthode qui retourne le domaine.

Cet exemple simple démontre bien selon nous les capacités de métaprogrammation de Smalltalk. Remarquons simplement pour finir que la réflexivité, l'introspection et l'intercession sont utilisées de pair. Par exemple, à la ligne 20 de l'exemple, nous voyons un exemple d'introspection, (`Smalltalk at: #aClassName`) permet d'accéder directement à la classe `aClassName` et donc, de lui envoyer des messages, ce que nous faisons avec le message `#class`. qui nous permet en fait d'accéder à la classe de la classe `aClassName`. La méthode `#domaine` est une méthode de classe (méthode statique en Java) car elle est générique et commune à toutes les instances.

```

1  REXDemo>>metaprogTest
2  "génération de classes et de méthodes"
3  | descripteurNames creatorFunction |
4
5  descripteurNames := {
6      #EFSUPMeteo -> #('excellente' 'bonne').
7      #EFSUPCondition -> #('cool' 'bof') .
8      #EFSUPExperience -> #('expert' 'petit scarabe').
9
10     creatorFunction := [:symbol | EFDescriptor
11         subclass: symbol
12         instanceVariableNames: ''
13         classVariableNames: ''
14         poolDictionaries: ''
15         category: 'Rex-SUP'].
16
17     descripteurNames
18         do: [:association |
19             creatorFunction value: association key.
20             (Smalltalk at: association key) class
21                 compileName: #domaine body: (association value asString) classified: #domain].

```

Figure V-6 - Exemple de métaprogrammation

Bien qu'intéressante et moins complexe que dans d'autres langages, nous n'avons pas retenu cette méthode de génération de classes, bien qu'elle possède des avantages intéressants comme la possibilité de sauvegarder le modèle créé dans un package classique. L'inconvénient de cette approche est que les évolutions du modèle sont assez difficiles à maintenir.

## Metadescription

La solution qui nous a semblé être bien adaptée à notre problématique est l'utilisation de métadescriptions (annotations dans le code – cf. V.3.3.2). Nous avons utilisé cette technique pour les développements réalisés dans le projet SUP (cf. V.5.1). Nous avons envisagé l'utilisation de Magritte (Renggli, 2006) qui est une implémentation en Smalltalk du concept des métadescriptions. Un des points intéressants de ce cadre est qu'il existe des correspondances avec le *framework* de développement d'application internet Seaside.

Magritte est un *framework* de métadescriptions qui permet de décrire des classes de domaine et leurs attributs respectifs, leurs relations ainsi que certaines contraintes. Magritte est développé en Smalltalk et augmente le métamodèle réflexif de Smalltalk (Rivard, 1996) avec des moyens additionnels qui permettent de raisonner sur la structure et le comportement des objets par l'interprétation de métadescriptions. La description relative à une propriété (un attribut) contient plusieurs informations, une concernant le type, une concernant le moyen d'accès à cette information pour les principales puis

des informations optionnelles comme le label, des relations et des contraintes servant par exemple à la validation.

Bien que très flexible et extensible, Magritte s'est révélé ne pas être réellement adapté à notre problématique de représentation et ce, principalement à cause de la valuation incertaine des descripteurs. Il est possible de définir de nouveaux types de description mais cette extension est assez délicate et pas forcément en phase avec les objectifs du *framework* qui possède des descriptions atomiques pour les types de données classiques. Nos descripteurs étant exprimés comme des distributions de masse associés à la composition hiérarchique, il n'était pas naturel selon nous, de les intégrer dans l'architecture de Magritte.

#### V.4.1.4. Persistance, interface graphique et intégration

Dans la partie V.2, nous avons présenté les aspects macroscopiques des applications qui doivent être généralisées. Nous présentons ici plus précisément les choix et développements qui ont été réalisés.

### Interfaces graphiques

Nous avons choisi de réaliser des applications Internet et nous avons utilisé comme *framework* applicatif, Seaside<sup>1</sup> (Ducasse *et al.*, 2007) qui propose une méthode de développement des applications web similaire au développement d'applications traditionnelles. La notion de page HTML est totalement abstraite au profit d'un modèle basé sur des composants. Nous avons créé une série de composants standard qui peuvent être intégrés dans toutes applications de retour d'expérience. De plus Magritte peut être conjointement utilisé avec Seaside, ce qui nous a permis de nous affranchir de certaines tâches telles que la génération des interfaces et l'application de contrainte de validations. Nous voyons d'ailleurs sur la Figure V-7 des exemples de composants générés organisés dans des composants générique réutilisables. Il s'agit des premières expérimentations qui ont été réalisées dans le cadre du projet SUP. Magritte prévoit des mécanismes d'extension et l'utilisation des CSS (Cascading Style Sheet) ce qui permet d'appliquer, le cas échéant, des chartes graphiques plus attrayantes. Notons que les feuilles de style CSS sont définies indépendamment et donc ne sont pas intrusives dans le code des applications (voir la Figure V-10 pour une illustration utilisant les CSS).

Figure V-7 - Composants générés

L'interfaçage avec le modèle des expériences est principalement réalisé par un ensemble de « formulaires dynamiques » proposant des choix multiples, des zones libres... Les différentes fonctionnalités du module de retour d'expérience correspondent à des interfaces graphiques spécifiques. La phase de Capitalisation met en œuvre des interfaces de saisie et de modification basées sur des

<sup>1</sup> <http://www.seaside.st>

formulaires (cf. Figure V-7 et Figure V-11). La phase d'Exploitation utilise une interface de présentation (listing, filtrage...) (cf. Figure V-12) ainsi qu'une interface pour effectuer la recherche par similarité (interface de saisie de contexte avec l'affectation de poids ainsi qu'une interface de présentation des résultats).

Dans les exemples présentés, la nature incertaine des opérateurs n'était pas prise en compte. Nous avons maintenant un composant standard qui permet d'éditer ce type de valeurs. Nous présentons une version élémentaire de ce composant sur la Figure V-8.

Un composant important permet la représentation sous forme de tables. Nous n'avons pour l'instant défini qu'un composant relativement simple de ce type, mais idéalement, il faudrait le décliner en plusieurs versions entièrement paramétrables (choix des informations affichées, des actions applicables, etc...).

Figure V-8 - Composant d'édition des descripteurs incertains

Figure V-8 - Composant d'édition des descripteurs incertains

## Persistence

La persistance du modèle et des expériences peut être réalisée de plusieurs manières. Le choix dépend aussi de l'implémentation du modèle des applications de retour d'expérience. Voici une liste non exhaustive allant des plus simples au plus performantes :

- sauvegarde dans l'environnement de programmation,
- sauvegarde binaire des objets,
- utilisation d'une Base de Données Relationnelle,
- utilisation d'une Base de Données Relationnelle (conjointement avec un ORM),
- utilisation d'une Base de Données Objet,
- utilisation d'un dialecte Smalltalk persistant et concurrent (Gemstone<sup>1</sup>).

La sauvegarde des expériences est, pour l'instant, effectuée directement dans l'environnement de programmation ainsi que par l'intermédiaire de fichiers externes (sauvegarde binaire). Une fois le prototype validé, la persistance sur base de données objet ou relationnelle sera étudiée. D'autres « sorties » sont prévues, des impressions papiers de « fiches de Retour d'Expérience », en parallèle à la version en ligne ainsi que des fichiers d'échange pour pouvoir communiquer avec d'autres systèmes informatiques.

## Intégration globale et administration

Pour intégrer et opérationnaliser les applications de retour d'expérience, il faut administrer les utilisateurs et intégrer les différents composants. Nous avons fait le choix d'intégrer les applications dans un CMS/Wiki (Content Management System) et en particulier dans PierCMS<sup>2</sup> qui présente l'avantage de pouvoir inclure des applications Seaside et utilisant Magritte (Renggli, 2006). De plus, ce

<sup>1</sup> <http://www.gemstone.com/>

<sup>2</sup> <http://www.piercms.com/>



Le modèle correspond au schéma de principe central des applications de retour d'expérience génériques. La spécificité de l'information traitée (utilisation de la valuation par distribution de masse de croyance) nous a amené à développer une architecture reposant sur une variante de modèle adaptatif (ou dynamique). En effet, on peut retrouver une similitude avec le *pattern* Type Object et Property List. Les objets instanciés par la classe `REXExperience` contiennent en effet, pour le contexte, une liste d'objets `ContextValue` et un modèle du contexte `ContextModel`. Un point intéressant, à nos yeux, est que la définition de ce modèle contraint la valuation des descripteurs du contexte. Concrètement, pour réaliser une application de retour d'expérience appliquée à un unique domaine, il n'y aura qu'un seul objet instance `REXContextModel` (singleton). Toutes les expériences qui sont ensuite instanciées possèdent une référence vers cet objet modèle et c'est l'interprétation de ce dernier qui permet d'accéder ou de modifier les véritables valeurs de l'information contextuelle. Cette information est instanciée à partir de la classe `REXDescriptorValue` (*pattern* Value Holder). Cet objet est valué par une distribution de masse de croyance instanciée à partir de la classe `REXBba`. Il garde aussi une référence vers le modèle du descripteur simple associé. Il ne nous a pas semblé utile de créer des Value Holder pour les descripteurs objets. En effet, leur valeur est une composition d'autres valeurs de descripteurs. Nous aurions pu reproduire la structure composite employée pour la définition du modèle de contexte mais cette information de composition peut aussi être reconstruite par l'interprétation du modèle de contexte associé à chaque expérience.

## V.5. Application à la sécurité en montagne

La montagne est un milieu naturel hostile. Elle est soumise aux risques naturels comme des avalanches, des glissements de terrain, des tremblements de terre mais aussi aux risques relatifs à la pratique d'activités ou à l'orographie. Les conditions météorologiques particulières qui y règnent ont une influence importante sur les accidents (phénomènes météorologiques violents et changeants, accessibilité difficile, franchissements dangereux, activités à risques...). En outre, l'augmentation de la fréquentation de ce milieu par des personnes n'ayant pas toujours une bonne connaissance de celui-ci est un facteur d'augmentation du nombre des interventions en montagne et, par conséquent, il est important de disposer d'outils de prévention de ces risques.

Le financement de cette thèse a été apporté par un projet européen INTERREG IIIa pour lequel nous devons étudier l'intégration d'un module de retour d'expérience orienté prévention des risques dans le démonstrateur du Système d'Information global. Nous présentons ce projet européen SUP (Sécurité Urgences Pyrénées) dans la partie V.5.1. Les réalisations effectuées dans le cadre du projet SUP. Les réalisations effectuées dans le cadre du projet SUP se sont appuyées sur des développements issus de ces travaux de thèse. Cette réalisation nous a permis d'étudier et de proposer une approche générique de développement des applications basées sur les métadescriptions. Nous présentons néanmoins ces réalisations effectuées dans le cadre du projet SUP car elles constituent un exemple de la génération des interfaces et des différents modules introduits dans la partie V.2.

### V.5.1. Le projet SUP

Le projet SUP (Sécurité Urgences Pyrénées) est un projet INTERREG IIIA France-Espagne dont l'objectif principal a été d'étudier et de proposer des outils basés sur les Nouvelles Technologies de l'Information et de la Communication (NTIC) afin d'améliorer la sécurité en montagne. Ce projet démontre la volonté de la France et de l'Espagne de mettre en commun leurs ressources humaines et techniques pour améliorer la sécurité des activités pratiquées dans le massif pyrénéen. Il s'agit d'un projet multipartenaires (franco-espagnol) dont le MEDES (Institut de médecine et de physiologie spatiale de Toulouse) a été le chef de file.

L'objectif principal du projet SUP est de proposer aux personnes exerçant des activités dans les Pyrénées, un système fiable permettant :

- des échanges de données médicales pour les professionnels du secours en montagne,

- la diffusion d'information sur les activités réalisées et les risques encourus.

Pour atteindre cet objectif, une unification des ressources, des expériences et compétences a été nécessaire ainsi que la création de nouveaux outils intégrés dans un système d'informations et d'aide à la décision commun et accessible à tous les professionnels de la montagne concernés des deux pays. Les NTIC ont été au centre des développements et l'une des contraintes majeures imposée par le projet a été de réaliser conjointement notre système avec les autres modules du projet (Système d'Information Géographique, base de données de prévention et d'intervention...). Initialement prévu pour durée 24 mois, le projet SUP a débuté fin 2003 et a pris fin le 31 avril 2007. Le projet SUP d'un coût total de 2,354 M€ reçoit le support financier de la Commission Européenne à travers l'INTERREG IIIA France Espagne (contribution à 47.5 % soit 1,12 M€) puis un cofinancement coté français et espagnol.

Le projet a été découpé en deux volets principaux, ce qui a défini deux orientations dans les développements et les réalisations proposées qui sont l'intervention et la prévention.

L'intervention est en rapport avec :

- le traitement des appels d'urgence,
- la préparation des interventions,
- la télé-assistance médicale.

La prévention, qui nous concerne davantage, est en rapport avec :

- l'appréciation des risques associés à la pratique d'activités dans le massif pyrénéen,
- la gestion des informations relatives à la sécurité en France et en Espagne,
- la prévention d'un certain nombre de risques,
- la sécurisation des activités.

Les livrables du projet sont principalement :

- Un Système d'Information global principalement dédiée au volet prévention mais aussi servant de portail pour les aspects intervention. C'est une application Web développée autour d'un Système d'Information Géographique couvrant la chaîne des Pyrénées dans son ensemble. Elle permet un accès simplifié aux informations de prévention pour l'appréciation des risques en moyenne et haute montagne par la gestion des comptes-rendus d'activité ainsi que par la mise à disposition des informations météorologiques. Notre dispositif de retour d'expérience a été intégré à ce système et est accessible depuis le menu principal.
- Concernant le volet intervention, les livrables ont été des outils transportables de télé-médecine pour les acteurs des opérations sanitaires et de secours (en particulier pour le versant espagnol qui est davantage étendu et désertique que le versant français). Il y a eu aussi la mise en place d'un système de gestion des échanges d'informations médicales entre le personnel du secours en montagne qui est sur le terrain et les équipes d'assistance médicale situées dans leur(s) centre(s) sanitaire(s) de référence. Les données peuvent être échangées à un niveau basique ou plus élaboré, grâce à des équipements portables simples et robustes de communications, de localisation et biomédicaux pouvant être utilisés par les professionnels impliqués.

Les applications issues du projet ont été opérationnalisées pour des utilisateurs qui sont l'ensemble des professionnels de la montagne : guides, commerçants, exploitants forestiers, bergers... mais aussi médecins, forces de l'ordre, pompiers... Ces utilisateurs sont actuellement conviés à tester le démonstrateur<sup>1</sup> mis en œuvre sur une zone restreinte du massif comprenant l'Ariège, les Hautes Pyrénées et la Haute Garonne ainsi que la province de Huesca, une partie de la Catalogne et du Val d'Aran. Sur cette zone restreinte des Pyrénées, 1750 interventions de secours sont recensées par an, soit une moyenne d'environ 6 interventions par jour.

Les utilisateurs ont comme rôle de recueillir les données destinées à améliorer la connaissance du système. Ils peuvent rédiger pour cela des comptes-rendus de sortie, décrire des expériences... Après chaque sortie, tout utilisateur ayant une fonction associée à la sécurité en montagne est

---

<sup>1</sup> <http://infosup.eu/>

systématiquement invité à saisir un compte-rendu alimentant une base de connaissances grâce à des formulaires simples reflétant les pratiques actuelles. Il sera complété par les informations contenues sur les serveurs (de météorologie, de cartographie...) en vue d'être stocké pour du retour d'expérience.

Le Laboratoire Génie de Production (LGP) a été sollicité pour ses compétences dans le domaine du retour d'expérience industriel afin de fournir un moyen d'exploiter les informations capitalisées dans l'entrepôt de données du système d'information dans un but de prévention des risques inhérents aux activités se déroulant dans le massif pyrénéen. Dans cette optique, notre travail a consisté en la réalisation d'une application de retour d'expérience. Il faut remarquer que les compétences du LGP ne sont pas spécifiques au domaine de la prévention. Nous nous situons à mi-chemin entre les aspects prévention et intervention du projet. Une grande partie du travail réalisé a consisté à de l'intégration afin que le module réalisé cohabite avec les bases de données de prévention et d'intervention et aussi avec le SIG qui nous a servi de support pour positionner les expériences. Nous ne détaillons pas ces aspects techniques qui n'entrent pas dans le cadre du travail de thèse. Nous montrons simplement les interfaces graphiques web générées directement à partir du modèle de l'expérience et qui mettent en œuvre les composants génériques développés (éditeur, tables, ...).

## V.5.2. Application réalisée

Le modèle de l'expérience relative aux activités en montagne a été long à stabiliser et c'est justement cet aspect qui nous a conduits à réaliser un cadre applicatif générique, paramétrable par le modèle de l'expérience. Nous nous sommes inspirés des comptes-rendus d'intervention de secours en montagne. L'avantage est qu'il s'agit d'une information a priori disponible car chaque intervention donne lieu à un compte-rendu d'intervention et un compte-rendu médical. Nous avons déterminé un modèle minimal par le croisement des formulaires existant mais aussi en contactant les personnes impliquées dans les secours (protection civile (SIDPC), SMUR, CRS et PGHM)). Malheureusement, nous avons observé des limitations importantes :

- cette information n'est pas suffisante pour décrire un accident dans l'optique d'un retour d'expérience orienté « prévention des risques ». Il y a trop d'information non directement exploitable et il n'y a pas assez d'informations en rapport avec l'expression des causes de l'événement,
- bien que cette information soit disponible sous forme de comptes-rendus, il n'est pas possible d'automatiser les saisies des formulaires d'expérience et il est encore moins envisageable de demander aux organismes de saisir un formulaire supplémentaire. L'idéal serait de convenir de fiches type généralisées, qui pourraient servir aux comptes-rendus d'intervention et d'expérience à la fois en France et en Espagne mais cette démarche nécessite d'être conçue et imposée par les autorités des régions et pays concernés.

Le modèle qui a servi à la réalisation du prototype est donc une sélection (réalisée de manière progressive) des informations (parmi les informations disponibles) qui nous ont semblé utiles d'un point de vue du risque. Il ne s'agit naturellement pas d'un modèle idéal et stabilisé mais du fait de l'approche de métamodélisation que nous avons développée, il est assez simple de faire évoluer le modèle et par conséquent l'application qui l'opérationnalise. Il sera d'ailleurs davantage envisageable de limiter l'application à des domaines plus restreints. Par exemple, en fin de projet, nous avons eu un retour positif d'un responsable sécurité du club alpin français (FFCAM - Fédération Française des Clubs Alpins et de Montagne) qui souhaite mettre en œuvre le système réalisé dans le cadre des activités proposées par cette organisation.

### V.5.2.1. Intégration et composants graphiques de l'application

Pour le projet SUP, il a fallu intégrer notre application dans un système existant. L'application de retour d'expérience est donc accessible depuis le menu du SI de prévention (partie gauche de la Figure V-10). Nous avons aussi réutilisé le SIG développé (partie droite de la Figure V-10) pour positionner les nouvelles expériences mais aussi pour représenter l'ensemble des expériences déjà sauvegardées.

L'application réalisée propose trois menus, celui de saisie de l'expérience, celui de recherche et enfin celui de consultation. L'interface graphique de l'application est traduite en français et en espagnol. Selon les droits de l'utilisateur, celui-ci aura plus ou moins de possibilités d'interaction avec la base d'expérience. Par exemple, le profil « expert » permet d'accéder aux interfaces d'analyses des expériences. Le profil « administrateur » permet de modifier et de réorganiser la base d'expériences (ajout/suppression d'expérience, validation d'analyse, changement de valeurs, etc.).



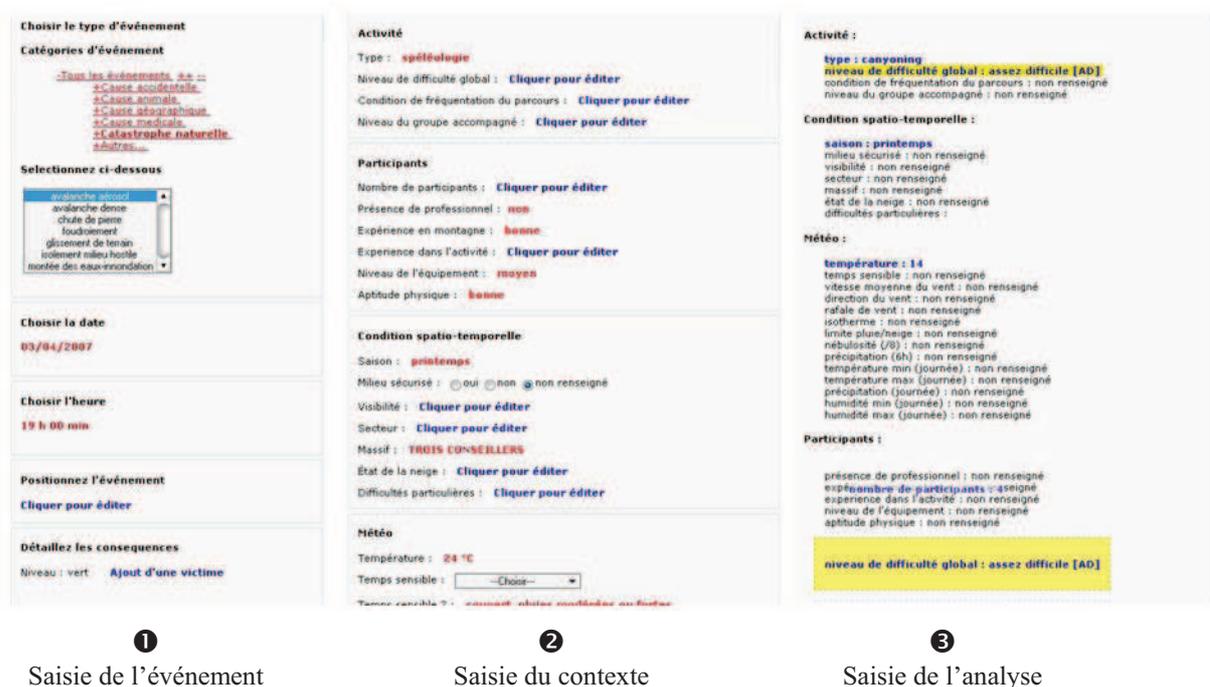
1  
Accès à l'application de retour d'expérience

2  
SIG de l'application prévention

Figure V-10 - Intégration dans le SI Global du projet SUP

### V.5.2.2. Processus de capitalisation

Les expériences sont saisies par l'utilisation de trois formulaires comme cela est représenté sur la Figure V-11. Tout d'abord, un premier formulaire sert à la saisie de l'événement, de la gravité associée mais aussi quelques informations génériques comme la date, l'heure et le lieu qui est directement saisi à l'aide du SIG du projet (partie droite de la Figure V-10). Le deuxième formulaire permet de réaliser la saisie du contexte de l'expérience.



1  
Saisie de l'événement

2  
Saisie du contexte

3  
Saisie de l'analyse

Figure V-11 - Les différentes interfaces de Saisie

C'est le même formulaire qui est utilisé pour la saisie du contexte courant qui sert ensuite à effectuer la recherche dans la base d'expérience. Enfin, un troisième formulaire permet d'analyser l'expérience. Nous voyons sur l'exemple (cf. Figure V-11-③) que les attributs « analysés », c'est-à-dire significatifs, apparaissent en jaune.

Chacune de ces interfaces est générée à partir de l'interprétation de métadescriptions qui sont dépendantes du modèle de l'expérience. Nous avons visé une certaine ergonomie d'interface en utilisant des attributs éditables en ligne (un par un), ce qui permet de masquer les listes de valeurs possibles et par conséquent de ne pas alourdir les formulaires. Il faut d'abord cliquer sur le descripteur ou l'attribut que l'on souhaite saisir pour voir apparaître les différentes possibilités. Ainsi, il n'est pas nécessaire de valider un formulaire pour sauvegarder les valeurs, chaque valeur entrée est directement stockée sur le serveur. De plus, nous utilisons les techniques de développement AJAX (Asynchronous JavaScript and XML) qui permettent de mettre à jour seulement certaines portions de l'interface sans avoir à recharger la totalité de l'interface.

Concernant l'interface d'analyse, nous avons choisi de présenter le contexte de l'expérience, c'est-à-dire une liste de descripteurs valués ou non, et de l'associer avec un cadre d'analyse. Il suffit alors pour l'expert de déplacer avec la souris (*drag and drop*) les descripteurs significatifs dans la zone d'analyse. Une fois qu'un descripteur fait partie de l'analyse, il est surligné en jaune dans la liste correspondant à l'expérience. Notons que pour le démonstrateur du projet SUP, nous n'avons pas pris en compte les estimations de « reproductibilité » et de confiance de l'expert en son analyse (cf. partie II.3.3.2).

### V.5.2.3. Processus d'exploitation

Avant de réaliser une sortie en montagne, l'utilisateur interroge la base en décrivant l'activité qu'il a prévue de réaliser. Le système retourne alors une liste d'accidents s'étant produits dans des conditions similaires et pouvant être visualisés sur le SIG commun. N'ayant pas implémenté la valuation incertaine, nous n'avons pas non plus implémenté les algorithmes présentés dans les chapitres III et IV. L'algorithme de recherche est ici simplement basé sur la valuation précise et l'association éventuelle d'une matrice de similarité. La description de l'activité, autrement dit du contexte courant qui est le paramètre d'entrée de l'algorithme de recherche, est réalisée par l'intermédiaire du même type d'interface que celle de saisie du contexte de la Figure V-11-②.

En phase d'exploitation des résultats, la base d'expériences est consultable indépendamment des phases de recherche. Il y a alors deux possibilités de consultation (cf. Figure V-12), soit par l'intermédiaire du SIG où les expériences sont représentées par des icônes en fonction de leur niveau de gravité (vert, jaune, rouge), soit par l'intermédiaire d'un tableau qui contient l'ensemble des expériences de la base. Ces deux interfaces servent aussi à présenter les résultats issus des différents types de recherche.



Figure V-12 - Interfaces graphiques de consultation des expériences

Toutefois, la mesure du risque n'a pu être intégrée ou, plus exactement, elle n'a pas pu être utilisée dans le cadre du démonstrateur SUP. D'une part, la valuation incertaine sur laquelle cet indicateur repose n'est pas implémentée et, d'autre part, la mesure nécessite un calibrage et une validation qui aujourd'hui n'est pas possible (la base possède trop peu d'expériences). Comme il s'agit d'une vue assez abstraite puisque l'ensemble des risques encourus est synthétisé en une seule mesure, cela rend assez difficile son interprétation. Quoi qu'il en soit, la représentation spatiale des accidents à partir d'une recherche en contexte basée sur la sélection pertinente des attributs assure en soi un très bon retour d'expérience. De plus, le résultat de la recherche par rapport à un contexte courant est dans une certaine mesure une indication du risque encouru qui s'est avérée suffisante dans le cadre du projet SUP. Pour le prototype, nous avons proposé de retourner le résultat sous la forme de tableaux d'expériences comme celui de la Figure V-12 (partie droite) avec une colonne supplémentaire représentant le degré de similarité.

### V.5.3. Synthèse sur le projet SUP

Le projet SUP dans son ensemble aura rempli son objectif principal, il aura démontré, pour le massif pyrénéen français et espagnol, que les NTIC peuvent contribuer efficacement à des actions de prévention des risques et aider les équipes de secours et de sauvetage dans l'accomplissement de leurs missions.

Concernant le retour d'expérience, la définition d'un modèle réaliste de l'expérience en rapport avec les différentes pratiques d'activités fut complexe. Ceci est dû en grande partie à l'étendue du domaine et c'est pourquoi il est difficile, voire impossible, d'avoir un modèle qui puisse servir de base à l'expression de tous les risques encourus en montagne. Néanmoins, en spécialisant l'application sur des domaines plus restreints, nous pensons qu'il est possible d'appliquer les principes théoriques de quantification du risque développé. Nous avons d'ailleurs poursuivi notre travail avec le responsable sécurité du club alpin français (section Haute-Garonne). A terme, nous pensons aussi qu'un tel système devrait être accessible au grand public, dans son processus de consultation.

Les travaux menés ont permis d'identifier deux grands axes de recherche. Tout d'abord, l'étude de l'intégration de l'aspect spatial dans les applications de retour d'expérience, abordé de manière relativement sommaire dans notre modèle. Nous pourrions plus particulièrement étudier l'intégration des aspects spatiaux comme la caractérisation, la similarité de zone géographique... L'autre axe concerne la « dynamique » des modèles. Il est selon nous nécessaire de poursuivre nos travaux dans ce domaine pour permettre en particulier la migration d'instance quand les modèles changent. Le cadre réalisé permet de construire assez rapidement des applications à partir de la définition du modèle d'expérience. Par contre, à chaque modification, les expériences qui ont été saisies ne sont pas conservées. Pour pouvoir conserver les informations déjà capitalisées, il faudra envisager des approches comme les techniques de transformations de modèles, ou encore les techniques de grossissement et de raffinement de cadre de discernement définies dans le MCT.

Parmi d'autres perspectives de développements, nous pensons qu'il serait intéressant d'imaginer une application de prévention unifiée reposant sur une interface cartographique telle que celles proposées aujourd'hui sur Internet comme Google Map<sup>1</sup>, Yahoo Maps<sup>2</sup>, Live Search Maps<sup>3</sup> ou encore IGN GeoPortail<sup>4</sup>. Ces supports permettraient d'intégrer les accidents, les caractéristiques particulières de zone, les randonnées effectuées et leurs remarques, les itinéraires fortement empruntés, les lieux particuliers (refuge et autres) ainsi que les prévisions météorologiques zone par zone... Ces différents systèmes de cartographie, bien que d'origines différentes, utilisent les mêmes abstractions. Il existe d'ailleurs une bibliothèque JavaScript appelé Mapstraction<sup>5</sup> qui propose un API générique qui permet d'interchanger le système de cartographie. Mapstraction ne couvre pas IGN portail, mais IGN a

---

<sup>1</sup> <http://maps.google.com/>

<sup>2</sup> <http://fr.maps.yahoo.com/>

<sup>3</sup> <http://maps.live.com/>

<sup>4</sup> <http://www.geoportail.fr/> et <https://api.ign.fr/geoportail/>

<sup>5</sup> <http://www.mapstraction.com/>

récemment étendue son offre en proposant une API. Il existe aussi des systèmes 3D qui permettent d'améliorer l'ergonomie et la représentation par rapport aux approches 2D citées (par exemple Google Earth<sup>1</sup> et Virtual Earth 3D<sup>2</sup>).

Les SIG de nouvelle génération sont de très bons média de présentation d'une information à composante géographique. Ces médias n'existaient pas encore au moment du lancement du projet SUP. Ils seront accessibles très prochainement depuis les terminaux mobiles de type téléphone. Nous montrons sur la Figure V-13 un exemple d'intégration que nous avons réalisé de notre système avec le système de cartographie de Google Map.



Figure V-13 - Intégration dans les SIG type Google Map

<sup>1</sup> <http://earth.google.fr/>

<sup>2</sup> <http://maps.live.com/>

# Conclusion Générale

La problématique des systèmes de retour d'expérience cognitif abordée dans cette thèse constitue un enjeu important dans le cadre de l'amélioration continue des organisations industrielles et, plus largement des organisations socio-techniques. Pourtant, il existe finalement peu d'outils informatiques traitant l'expérience comme une entité cognitive et non comme une simple information sur laquelle peuvent être construites des analyses statistiques. Nous sommes partis de l'hypothèse que la connaissance des experts, si elle est inscrite dans l'expérience lors de la résolution des problèmes, peut permettre de compenser le faible nombre d'expériences qui empêche une analyse statistique brute. Nous avons fait un rapprochement entre le retour d'expérience cognitif et les systèmes à base de connaissances, le retour d'expérience cognitif ayant pour particularité notable que la connaissance est développée de manière incrémentale, par l'addition d'analyses sur des cas capitalisés progressivement.

L'objectif de ce travail était de proposer un cadre pour le développement d'un générateur d'applications de retour d'expérience cognitif, intégrant la possibilité de rechercher des expériences par similarité et la proposition d'une mesure du risque relative à l'ensemble des expériences. Nous avons étudié la nature précise des expériences qui doivent être capitalisées et nous avons proposé une représentation globale de ces expériences sous la forme : événement, contexte, analyse, solutions, leçons apprises. Nous avons aussi montré que les expériences étant pour partie le résultat d'un travail d'interprétation (du contexte), il est intéressant de prendre en compte l'incertitude liée à cette expertise. C'est pourquoi nous avons proposé d'intégrer dans notre approche, de manière explicite, l'incertitude associée aux informations sur le contexte et à l'analyse réalisée par les experts. Après avoir envisagé plusieurs moyens de représentation, nous avons sélectionné le modèle des croyances transférables comme support de modélisation. En outre, et afin de faciliter la structuration des connaissances puis le passage à un système opérationnel, nous avons considéré que les connaissances seraient architecturées selon le paradigme de l'orienté objet. Ainsi, nous proposons un formalisme hybride de représentation des connaissances basé sur l'orienté objet et le modèle des croyances transférables.

Une fois cette structure posée comme base de représentation des expériences, nous avons cherché à élaborer des algorithmes qui permettent d'exploiter un ensemble d'expériences. Le premier algorithme permet une recherche d'expérience à la manière du raisonnement à partir de cas, c'est-à-dire par similarité de contexte. Le deuxième algorithme permet quant à lui de tenir compte de l'expertise apportée par les experts et on peut considérer que cet algorithme est typique de la notion de retour d'expérience cognitif. Enfin, nous avons tenté de généraliser les expériences contenues dans une base d'expériences en élaborant un indicateur de risque. Dans ce cas, plusieurs expériences peuvent être prises en compte, ainsi que l'analyse experte qui leur est rattachée.

Enfin, dans la dernière partie de ce mémoire, nous avons présenté les premiers développements concrets réalisés en essayant de mettre en lumière les choix que nous avons effectués quant aux techniques d'opérationnalisation choisies. Une première application a été construite dans le cadre du projet européen Interreg III A SUP (Sécurité Urgences Pyrénées).

Le travail réalisé dans le cadre de cette thèse est le premier, au Laboratoire Génie de Production de l'École Nationale d'Ingénieurs de Tarbes, avec un objectif d'opérationnalisation d'un système de retour d'expérience cognitif générique. Il s'inscrit dans le prolongement de plusieurs travaux davantage orientés sur les aspects méthodologiques et sur la capitalisation des expériences. Naturellement, à l'issue de ce travail beaucoup de travaux restent à conduire et nous avons identifié sept voies de développement. Certaines peuvent être prises en compte à court terme car elles concernent le cadre

applicatif réalisé comme celles évoquées dans les points 1, 2, 3, 4 et 5. D'autres nécessitent des développements complémentaires plus conséquents comme celles évoquées dans les points 6 et 7.

### **(1) Descripteur numérique continu**

Concernant, le formalisme de représentation de l'expérience, nous pensons qu'il serait utile de pouvoir définir des descripteurs simples de différents types dans les modèles de l'expérience basés sur la modélisation attribut-valeur incertaine. Nous avons développé, dans cette thèse, uniquement des types de descripteurs simples (symboliques ou valeurs discrétisées) car ceci nous semblait être une vue suffisamment générale dans la perspective du retour d'expérience. En effet, contrairement à un système d'information classique, tous les types de données classiques (entier, flottant, chaînes de caractères, date...) ne sont pas nécessaires. En d'autres termes, l'information véhiculée est souvent plus significative quand elle est catégorisée ou discrétisée. Par exemple, concernant l'âge d'une personne, il est plus utile de savoir si la personne est jeune, adulte ou âgée plutôt que de connaître précisément son âge. Néanmoins, il existe certains cas où la valeur réelle est plus informative que la valeur discrétisée (par exemple si une dimension de pièce usinée doit être représentée). Même si l'information utile est discrétisée, il est parfois plus simple de saisir une valeur numérique réelle et de laisser le système discrétiser cette information. En conséquence de quoi, nous pensons qu'il est nécessaire d'étendre le système pour tenir compte de descripteurs continus numériques. Pour cela, il faudra :

- intégrer un nouveau type de descripteur numérique continu,
- créer des passerelles entre descripteur numérique et symbolique,
- ajouter une interface numérique continue pour la saisie de descripteurs symboliques,
- ajouter un descripteur dérivé du descripteur numérique par l'association d'une quantité (unité de mesure).

### **(2) Descripteur ontologique**

Nous pensons aussi qu'il serait très utile de développer un descripteur dit « ontologique », dont le domaine correspond à des entités définies formellement par une ontologie. Pour l'instant avec les descripteurs symboliques, nous disposons simplement de listes de valeurs qui, dans une certaine mesure, peuvent être associées à des éléments du domaine. Nous pensons rendre possible l'utilisation des entités définies dans un modèle de domaine indépendant et autonome. Ceci permettrait de coupler les systèmes de retour d'expérience à des ontologies existantes, ceci étant très utile dans une perspective d'intégration et d'interopérabilité.

Le rôle de l'ontologie dans un système de représentation des connaissances est de contraindre la capitalisation des instances de connaissance. Elle établit aussi les règles sur lesquelles seront basées les règles d'inférence qui permettent de générer un corps de connaissance plus complet. Pour réaliser le couplage entre un modèle de domaine indépendant et le modèle de l'expérience, une grande partie du travail à réaliser consisterait à adapter les algorithmes de similarité pour tenir compte de l'entité ontologique. Il faudrait aussi considérer les techniques de similarité entre objets et taxonomies présentées dans les parties III.2.7 et III.2.6.

### **(3) Utilisations multi-experts du système**

Un intérêt majeur des fonctions de croyance réside dans la possibilité de réaliser une fusion d'information issue de sources variées. Or, dans ces travaux, nous n'avons pas exploité cette possibilité bien que nous l'ayons toujours considérée comme d'intérêt. Nous souhaitons en effet autoriser les analyses de cas identiques par plusieurs experts dans la phase de saisie des expériences et dans celle d'analyse.

- Pour la saisie multi-experts du contexte des expériences, nous pensons que cette possibilité est relativement directe grâce à l'utilisation des fonctions de croyance. Il reste néanmoins à la développer et à choisir ou à proposer une fonction de combinaison de l'information appropriée.
- Pour la saisie multi-experts des analyses des expériences, il est déjà possible d'envisager la fusion des analyses de cas identiques par plusieurs experts puisque nous offrons déjà la possibilité à un expert d'exprimer plusieurs. Il est nécessaire de généraliser cet aspect pour tenir compte, notamment, d'une mesure de confiance en chaque expert.

#### **(4) Expression d'une analyse plus précise**

Nous souhaitons, à un stade précoce des travaux que nous avons présentés dans ce mémoire, utiliser directement les possibilités des fonctions de croyance pour exprimer les analyses en utilisant des techniques de classification usuelles telles que la méthode généralisée des plus proches voisins afin d'estimer une mesure du risque que nous pensions être un problème de classification sur le cadre de discernement d'expression de la gravité. Nous avons finalement estimé que les techniques de classification sont mal adaptées à notre problématique. C'est pourquoi nous avons créé une relation d'analyse dédiée.

Cette expression actuelle de l'analyse souffre de certaines limites. Elle est une première approche qui nous a permis d'aller jusqu'à la définition d'un indicateur du risque par l'expérience. Néanmoins elle est assez éloignée des techniques usuelles d'identification et d'analyse des risques. Dans une certaine mesure, l'analyse proposée peut être considérée comme une vue « à plat » d'une analyse causale qui est, par nature, arborescente (*cf.* (Villemeur, 1988) pour une introduction aux méthodes de type arbre des causes et des conséquences). Il faudrait, dans le futur, tendre vers ce type d'analyse du risque plus usuelle et reposant sur des fondements théoriques établis.

Nous pensons que d'autres possibilités pour l'expression de l'analyse sont à explorer. En effet, la version généralisée du théorème de Bayes compatible avec le modèle des croyances transférables ainsi que la définition de réseaux de croyance pourraient être très intéressants dans la perspective d'une approche de prévention des risques par l'expérience.

#### **(5) Généralisation de l'hypothèse du monde ouvert**

Nous avons constaté la difficulté d'obtenir un modèle stable de l'expérience pour lequel il manque souvent des informations lors de l'analyse. C'est pourquoi, un peu de manière similaire à l'hypothèse du monde ouvert, nous pensons qu'il serait utile d'autoriser la possibilité de faire référence à des informations non présentes dans le modèle de l'expérience. Nous distinguons deux possibilités :

- l'utilisation de valeurs de descripteurs non présents dans le cadre de discernement considéré : cette possibilité est autorisée en considérant l'hypothèse du monde ouvert du modèle des croyances transférables,
- l'utilisation, dans les analyses, de descripteurs non définis dans le modèle de l'expérience : nous souhaitons laisser la possibilité à l'expert de faire référence à des éléments d'information externes.

#### **(6) Vers un modèle réellement dynamique et incrémental**

Nous avons choisi de définir des applications de retour d'expérience en fonction du modèle de l'expérience ce qui est déjà un gain de temps considérable puisqu'il suffit de changer le modèle pour modifier les applications. Les interfaces graphiques sont, en particulier, automatiquement adaptées. Malheureusement, à chaque modification de modèle les expériences capitalisées sont perdues. Ceci n'est pas réellement gênant en phase de prototypage et de définition de l'application, par contre cela rend difficile la modification du modèle quand le système est opérationnel depuis un certain temps car il faudrait prévoir la migration des expériences existantes, ce qui n'est pas le cas dans l'application que nous avons développée. Nous pensons que cet aspect lié à la dynamique des modèles est fondamental car les évolutions du modèle, aussi minimales soient-elles, sont inévitables. Nous pensons que les choix de conception, modèle dynamique, fonction de croyance et langage de programmation dynamiquement typé sont des bons choix dans cette perspective.

Cette vision dynamique du modèle de l'expérience serait alors en accord avec une construction incrémentale du modèle de l'expérience basée sur les analyses. En effet, il serait utile d'envisager une construction incrémentale du modèle de l'expérience comme la collection des descripteurs définis par les étapes successives d'analyse. Il s'agit d'une approche différente car ce n'est plus le contexte qui contraint l'analyse. En fait, contrairement à l'approche définie dans ce mémoire de thèse, le modèle du contexte serait vide initialement. Le premier expert qui modéliserait une analyse concernant un

événement donné définirait les premiers descripteurs, et ainsi de suite, d'analyse en analyse, le modèle serait construit. Le modèle de contexte de l'expérience correspondrait alors à l'ensemble des descripteurs présents et définis dans chaque analyse).

Nous ne savons pas si une telle modélisation peut aisément converger mais c'est une idée plus en phase avec la vue que nous avons du retour d'expérience, c'est-à-dire un type de gestion des connaissances ascendante et incrémentale. Cependant, il est certainement plus réaliste de définir un ensemble de descripteurs initiaux pour définir un cadre minimal, tout en autorisant par la suite l'ajout de descripteurs dans le modèle.

### **(7) Une approche couplant modèle dynamique et métaprogrammation**

Un dernier point plus technique concerne la mise en œuvre des applications de retour d'expérience. Comme nous l'avons indiqué, l'utilisation des *patterns* AOM, DOM, est intéressante en raison de leur flexibilité mais induisent certaines pertes de performances relatives aux indirections liées à l'interprétation des modèles qui sont des instances de métamodèle et non pas des entités logiques (des classes et des méthodes « visibles ») du langage de programmation. Il existe des moyens d'améliorer ces performances en optimisant l'implémentation (*cf. Rielhe et al., 2000*). Cependant, nous pensons qu'une voie de progrès consisterait à coupler métaprogrammation (génération de code) et modèle dynamique. Nous pensons, qu'à ces fins, Smalltalk est un langage idéal. L'aspect génération de classes n'a pas été retenu car ce modèle semblait difficile à faire évoluer. Néanmoins, les propriétés de Smalltalk, comme l'introspection et l'intercession, pourraient faciliter de telles évolutions.

# Table des figures

Figure I-1 – Distinction donnée/Information/Connaissance	11
Figure I-2 - Positionnement de l'expérience dans le triptyque Donnée/Information/Connaissance	12
Figure I-3 - Dynamique des connaissances, d'après (Baumard, 1996)	13
Figure I-4 - Vecteur de production de connaissance	18
Figure I-5 - Vue d'ensemble des applications de retour d'expérience	22
Figure I-6 - Méthodologie, processus et composants support du retour d'expérience	26
Figure I-7 - les différents cycles/processus et activités de capitalisation	28
Figure I-8 - les différents cycles/processus et activités d'exploitation	29
Figure I-9 - Intégration Processus Capitalisation et Exploitation	29
Figure II-1 - Primitives de modélisation ontologie et orientée objet (Knublauch, 2002)	40
Figure II-2 Modélisation UML de connaissance (exemple : famille)	42
Figure II-3 - les 2 niveaux du MCT (Denœux, 2004)	49
Figure II-4 - Interprétation ensembliste de la fonction de crédibilité	53
Figure II-5 - Interprétation ensembliste de la fonction de plausibilité	53
Figure II-6 - Interprétation des fonctions de croyance pour un ensemble A (Lefevre, 2001)	59
Figure II-7 - Structuration de l'expérience	61
Figure II-8 - Représentation hiérarchique d'un attribut objet	64
Figure III-1 - Positionnement du retour d'expérience	74
Figure III-2 - Illustration de différentes fonctions de similarité locale	81
Figure III-3 - Représentation d'une taxonomie simple	82
Figure III-4 - Présentation de la vue orientée objet en vue du calcul de similarité	84
Figure III-5 - Représentation graphique de $sim_{ens-dis}$	88
Figure III-6 - Treillis des sous ensembles de { a ; b ; c ; d }	93
Figure III-7 - Représentation graphique de la mesure de similarité calculée pour l'exemple	98
Figure III-8 - Vecteur de similarité pour deux descripteurs simples	99
Figure III-9 - Illustration des 6 cas de similarités globale pour l'exemple	100
Figure III-10 - Contribution de $S_{(1)(2)}$	101
Figure III-11 - Principe de construction incrémentale	101
Figure III-12 - Principe d'agrégation itératif	103
Figure III-13 - Exemple d'expérience contenant des descripteurs objets	103
Figure III-14 - Interprétation géométrique de l'agrégation de 2 et 3 descripteurs	106
Figure IV-1 - Vue Causes - Conséquences des risques	112
Figure IV-2 - Pyramide des risques	114
Figure IV-3 - Pyramide des risques d'accidents de travail	115
Figure IV-4 - Situation de risque	116
Figure IV-5 - Représentation 3D et 2D du risque (Gouriveau, 2003)	116
Figure IV-6 - Référentiel du risque pour les expériences	117
Figure IV-7 - Les différentes partitions de niveaux de risque	117
Figure IV-8 - Composantes d'un risque dans un cas	120
Figure IV-9 - Principe général de l'indicateur du risque par l'expérience	121
Figure IV-10 - Les 4 étapes de la construction de l'indicateur du risque	122
Figure V-1 - Ingénierie des modèles - les différents rôles des modèles (Bezivin et Gerbe, 2001)	126
Figure V-2 - Architecture de métamodélisation MOF à 4 niveaux	127
Figure V-3 - Principe et architecture des applications de retour d'expérience	131
Figure V-4 - Dynamic Object Model (Riehle <i>et al.</i> , 2000)	133
Figure V-5 - Métamodèle de K-Beans (Knublauch, 2002)	138
Figure V-6 - Exemple de métaprogrammation	142
Figure V-7 - Composants générés	143
Figure V-8 - Composant d'édition des descripteurs incertains	144
Figure V-9 - Modèle du retour d'expérience	145
Figure V-10 - Intégration dans le SI Global du projet SUP	149
Figure V-11 - Les différentes interfaces de Saisie	149
Figure V-12 - Interfaces graphiques de consultation des expériences	150
Figure V-13 - Intégration dans les SIG type Google Map	152



# Bibliographie

**(Aamodt and Plaza, 1994)**

Aamodt A., et E. Plaza, Case-based reasoning : foundational issues, methodological variations, and system approaches, *AI Communications*, vol. 7 (1), pp. 39--59, 1994

**(Aha et al., 2001a)**

Aha D.W., R. Weber, et I. Beccerra-Fernandez, Intelligent lessons learned systems, *International Journal of Expert Systems Research & Applications*, vol. 20 (1), 2001

**(Aha et al., 2001b)**

Aha D.W., R. Weber, H. Munoz, L. Breslow, et K. Gupta, Bridging the Lesson Distribution Gap, In : *IJCAI 01 (Seattle, WA, Aug 2001)*, M. Kaufmann Publishers, Inc., pp. 987--992, 2001

**(Alpert et al., 1998)**

Alpert S., K. Brown et B. Woolf, *The Design Patterns SmallTalk Companion*, Addison-Wesley, 1998

**(ANAES, 2003)**

ANAES, Principes méthodologiques pour la gestion des risques en établissement de santé, rapport thématique, Agence Nationale d'Accréditation et d'Évaluation en Santé (ANAES), janvier 2003

**(Asté, 2001)**

Asté J.P., Géomatique et gestion des risques urbains : vers des progress attendus, pp. 249--283, IN *Gestion spatiale des risques*, sous la direction de G. Brugnot, Ed. Hermes Lavoisier, ISBN : 2-7462-0319-7, 2001

**(Atkinson et Kuhne, 2001)**

Atkinson C., et T. Kuhne, The essence of multilevel metamodeling, In Martin Gogolla and Cris Kobryn, editors, *Proceedings of the 4 International Conference on the UML 2000*, Toronto, Canada, LNCS 2185, pp. 1933, Springer Verlag, October 2001

**(Baget et Mugnier, 2002)**

Baget J-F., et M-L. Mugnier, Extensions of Simple Conceptual Graphs : the Complexity of Rules and Constraints, *Journal of Artificial Intelligence Research (JAIR)*, vol. 16, pp. 425--465, 2002

**(Basili et al., 1994)**

Basili V.R., G. Caldiera, et H. D. Rombach, The Experience Factory, In *Encyclopedia of Software Engineering*, vol. 1, pp. 469--476, edited by J.J. Marciniak, J. Wiley & Sons, 1994

**(Baudrit et Dubois, 2005)**

Baudrit C., et D. Dubois, Comparing Methods for Joint Objective and Subjective Uncertainty Propagation with an example in a risk assessment, *International Symposium on Imprecise Probabilities and Their Application (ISIPTA05)*, Pittsburg (USA, Pennsylvanie), pp. 31--40, 2005

**(Baudrit, 2006)**

Baudrit C., Représentation et propagation de connaissances imprécises et incertaines: Application à l'évaluation des risques liés aux sites et aux sols pollués, thèse de doctorat, Université Paul Sabatier, Toulouse, octobre 2005.

**(Baumard, 1996)**

Baumard P., *Les Organisations Déconcertées – La Gestion Stratégique de la Connaissance*, Editions Masson, 1996

**(Bäumer et al., 1998)**

Bäumer D., D. Riehle, W. Siberski, C. Lilienthal, D. Megert, K-H. Sylla, et H. Züllighoven, Values in Object Systems, Ubilab Technical Report 98.10.1, Zurich, Switzerland : UBS AG, 1998

**(Bergmann et Eisenecker, 1995)**

Bergmann R., et U. Eisenecker, Fallbasiertes Schließen zur Unterstützung der Wiederverwendung objektorientierter Software : Eine Fallstudie, In M. Richter, et F. Maurer, editors, Proceedings der 3, Deutschen Expertensystemtagung, XPS-95, pp. 152--169, 1995, Infix

**(Bergmann et Stahl, 1998)**

Bergmann R., et A. Sthal, Similarity measures for object-oriented case representations, 4th European Workshop on Case-Based Reasoning, EWCBR, 1998

**(Bergmann, 1998)**

Bergmann R., On the Use of Taxonomies for Representing Case Features and Local Similarity Measures, In Proceedings of the 6th German Workshop on Case-Based Reasoning (GWCBR 98), 1998

**(Bergmann, 2002)**

Bergmann R., Experience Management, Springer-Verlag Berlin Heidelberg, 2002

**(Bezivin et Gerbe, 2001)**

Bezivin J., et O. Gerbe, Towards a precise definition of the OMG/MDA framework, In Proceedings Automated Software Engineering (ASE 2001), pp. 273--282, Los Alamitos CA, IEEE Computer Society, 2001

**(Bickford, 2000)**

Bickford J., Sharing lessons learned in the Department of Energy, AAAI-00 Intelligent Lessons Learned Systems Workshop, Austin, Texas, USA, 31 juillet 2000

**(Bird, 1974)**

Bird F., Management Guide to Loss Control, Intl Loss Control Inst, 1974

**(Bisson, 1995)**

Bisson G., Why and how to define a similarity mesure for object-based representation systems, 2nd international Conference on Building and Sharing Very Large-Scale Knowledge Bases, KBKS, Enschede (NL), 10-13 avril, Eds N. Mars, IOS Press, Amsterdam (NL), pp. 236--246, 1995

**(Black et al., 2007)**

Black A., S. Ducasse, O. Nierstrasz, D. Pollet, D. Cassou et M. Denker, Squeak by Example, Square Bracket Associates, 2007, <http://SqueakByExample.org/>

**(Bloch et Maitre, 1994)**

Bloch I., et H. Maître, Fusion de données en traitement d'image : Modèles d'informations et décisions, traitement du signal, vol. 6 (11), 1994

**(Bloch, 2005)**

Bloch I., Fusion d'informations numériques : panorama méthodologique, JNRR 05, Guidel, Morbihan, 5-7 octobre 2005

**(Booch et al., 1999)**

Booch G., J.E. Rumbaugh, et I. Jacobson, The Unified Modeling Language User Guide, Addison-Wesley, Reading, MA, 1999

**(Bourrelie et al., 2000)**

Bourrelie P.H., G. Deneufbourg, et B. de Vanssay, Les catastrophes naturelles, le grand cafouillage, Ed. Osman Eyrolles, Santé & Société, ISBN : 2-7464-0163-0, 2000

**(Brachman et Levesque, 1985)**

Brachman R.J., et H.J. Levesque, editors Readings in Knowledge Representation, Morgan Kaufmann, Los Altos, CA, 1985

**(Brachman et Schmoltze, 1985)**

Brachman R.J., et J.G. Schmolze, An overview of the KL-ONE knowledge representation system, *Cognitive Science*, vol. 9 (2), pp.171--216, April 1985

**(Carré, 1989)**

Carré B., Méthodologie orientée objet pour la représentation des connaissances, Concepts de point de vue, de représentation multiple et évolutive d'objets, Thèse de Doctorat : Informatique, Lille, 1989

**(Champin et al., 2004)**

Champin P.A., J. Laflaquière, A. Mille, et Y. Prié, MUSETTE : a *framework* for Knowledge Capture from Experience, 12ème Atelier de Raisonnement à Partir de Cas, 2004

**(Champin, 2003)**

Champin P.A., Ardeco : an assistant for experience reuse in Computer Aided Design, In WS5 : From Structured Cases to Unstructured Problem Solving Episodes for ExperienceBased Assistance, Workshop at ICCBR 03, 2003

**(Chaudhri et al., 1998)**

Chaudhri V.K. , A. Farquhar, R. Fikes, P.D. Karp, et J. Rice, OKBC : A programmatic foundation for knowledge base interoperability, In Proc. of The AAAI-98, Madison, WI, 1998

**(Chein et Mugnier, 1992)**

Chein M., et M.L. Mugnier, Conceptual graphs : fundamental notions, *Revue d'Intelligence Artificielle* 1992, vol. 6 (4), pp. 365--406, 1992

**(Clermont et al., 2007)**

Clermont Ph., C. Béler, H. Rakoto, X. Desforges, et L. Geneste, Capitalisation et exploitation du retour d'expérience : un raisonnement à partir de cas étendu aux systèmes socio-techniques, in *Raisonnement à partir de cas, conception et configuration de produits*, vol. 1, série Informatique et Systèmes d'Information, collection Hermès, pp. 249--277, janvier 2007

**(Collins et Quillian, 1970)**

Collins A.M., et M.R. Quillian, Facilitating retrieval from semantic memory : The effect of repeating part of an inference, In A.F.Sanders, editor, *Acta Psychologica* 33 Attention and Performance III, North-Holland, Amsterdam, Holland, 1970

**(Cranefield et Purvis, 1999)**

Cranefield S., et M. Purvis, UML as an ontology modeling language, In Proc. of the IJCAI-99 Workshop on Intelligent Information Integration, Stockholm, Sweden, 1999

**(Cuneo, 2003)**

Cuneo A., *Le maître de Garamond*, Stock, 2003

**(Cunningham, 1995)**

Cunningham W., The Checks Pattern Language of Information Integrity, In *Pattern Languages of Program Design*, pp. 147--162, Addison-Wesley, 1995

**(CWM, 2003)**

Common warehouse metamodel, Technical report, Object Management Group, 2003

**(Dalkir, 2005)**

Dalkir K., *Knowledge Management In Theory and Practice*, Amsterdam, Boston, Elsevier/Butterworth Heinemann, June 2005

**(Dekker, 1994)**

Dekker L., FROME : Représentation multiple et classification d'objets avec points de vue, Thèse de Doctorat : Informatique, Lille, 1994

**(Demeyer et al., 1999)**

Demeyer S., S. Ducasse, et S. Tichelaar, Why Unified is not Universal, UML Shortcomings for Coping with Round-trip Engineering, In Proceedings UML 99 (The Second International Conference on The Unified Modeling Language), Kaiserslautern, Germany, October, LNCS, vol. 1723, Springer-Verlag, pp. 630--644, 1999

**(Dempster, 1967)**

Dempster A.P., Upper and Lower Probabilities Induced by a Multivalued Mapping, *Annals of Mathematical Statistics*, (38), pp. 325--339, 1967

**(Dencœux, 2004)**

Dencœux T., Théorie des fonctions de croyance et classification, ASTRID seminar, Heudiasyc, Compiègne, Juin 2004

**(Dencœux, 2008)**

Dencœux T., Conjunctive and Disjunctive Combination of Belief Functions Induced by Non Distinct Bodies of Evidence, *Artificial Intelligence*, vol. 172, pp. 234--264, 2008

**(Dubois et al., 2003)**

Dubois D., H. Prade, et P. Smets, A definition of subjective possibility, *Operations Research and Decisions* 2003, vol. 4, pp. 7--22, 2003

**(Dubois et al., 2006)**

Dubois D., H. Fargier, H. Prade, et R. Sabbadin, Critères qualitatifs pour la décision dans l'incertain, Dans : Concepts et méthodes pour l'aide à la décision - risque et incertain, D. Bouyssou, D. Dubois, M. Pirlot, et H. Prade (Eds.), Lavoisier, (4), pp. 99--137, vol. 2, *Traité IC2*, 2006

**(Dubois et Prade, 1988)**

Dubois D., et H. Prade, *Théorie des possibilités*, 2<sup>de</sup> édition, Masson, Paris, ISBN : 2-225-81273-X, 1988

**(Dubois et Prade, 2006)**

Dubois D., et H. Prade, Représentations formelles de l'incertain et de l'imprécis, Dans : Concepts et méthodes pour l'aide à la décision - outils de modélisation, D. Bouyssou, D. Dubois, M. Pirlot, et H. Prade (Eds.), Lavoisier, (3), pp. 111--171, vol. 1, *Traité IC2*, 2006

**(Ducasse et al., 2007)**

Ducasse S., A. Lienhard, et L. Renggli, Seaside : A Flexible Environment for Building Dynamic Web Applications, In *IEEE Software*, Los Alamitos, CA, USA, vol. 24, IEEE Computer Society, pp. 56--63, 2007

**(Ducournau et al., 1998)**

Ducournau R., J. Euzenat, et A. Napoli, Langages et modèles à objets, État des recherches et perspectives, ISBN : 2-7261-1131-9, 1998

**(Duizabo et Guillaume, 1996)**

Duizabo S., et N. Guillaume, Les enjeux du transfert des connaissances, *Cahiers du GRES*, (9601), Université Paris Dauphiné, janvier 1996

**(Dupuis-Hepner, 1999)**

Dupuis-Hepner N., Les hommes et la mémoire vive de l'entreprise, *L'Art du Management de l'Information*, dossier (6), supplément gratuit du 5-6 novembre, *Les Echos*, 1999

**(Duribrux et al., 2000)**

Duribrux M., P. Caulier, et B. Houriez, Application industrielle d'une approche mixte de modélisation des connaissances, In J. Charlet, M. Zacklad, G. Kassel, et D. Bourigault (eds.) (Ed.), *Ingénierie des connaissances : évolutions récentes et nouveaux défis*, Eyrolles et France Télécom R&D, Paris, France, 2000

**(Easterby-Smith et Lyles, 2003)**

Easterby-Smith M., et M.A. Lyles (editors), *The Blackwell Handbook of Organizational Learning and Knowledge Management*, Oxford, Blackwell Publishing, 2003

**(Eck et al., 2001)**

van Eck P.A.T., J. Engelfriet, D. Fensel, F. van Harmelen, Y. Venema, et M. Willems, A survey of languages for specifying dynamics : A knowledge engineering perspective, *IEEE Transactions on Knowledge and Data Engineering*, 2001

**(Ermine et al., 1996)**

Ermine J.-L., M. Chaillot, P. Bigeon, B. Charreton, et D. Malavieille, MKSM : Méthode pour la gestion des connaissances, *Ingénierie des systèmes d'Information, AFCET-Hermès*, vol. 4, (4), pp. 540--575, 1996

**(Euzenat, 1999)**

Euzenat J., Représentations des connaissances, de l'approximation à la confrontation, Dossier d'habilitation à diriger des recherches, Université Joseph Fourier, Grenoble, France, 1999

**(Evans et Kent, 1999)**

Evans A.S., et S. Kent, Meta-modelling semantics of UML : the pUML approach, In : 2nd International Conference on the Unified Modeling Language, Editors : B.Rumpe, et R.B. France, Colorado, LNCS 1723, 1999

**(Faure et Bisson , 2000)**

Faure A., et G. Bisson, Gérer les retours d'expérience pour maintenir une mémoire métier, étude chez PSA Peugeot Citroën, Journées Francophones d'Ingénierie des Connaissances IC 2000, Toulouse, mai 2000

**(Faure et Bisson, 1999)**

Faure A., et G. Bisson, Modelling the Experience Feedback Loop to improve Knowledge Base reuse in industrial environment, KAW 99, 12th Workshop on Knowledge Acquisition, Modelling and Management, Alberta, Canada, 16th - 21st October 1999

**(Favre et al., 1998)**

Favre J.L., G. Brugnot, J.M. Gresillon, et M. Jappiot, Evaluation des risques naturels, une approche probabiliste ?, les techniques de l'ingénieur, *Risque et décision* : C3295, 1998

**(Fayad et Schmidt, 1997)**

Fayad M., et D. Schmidt, Object-Oriented Application Frameworks, guest editorial of for the Communications of the ACM, Special Issue on Object-Oriented Application Frameworks, vol. 40 (10), (<http://www.cs.wustl.edu/~schmidt/CACM-frameworks.html>), 1997

**(Fensel et al., 2000)**

Fensel D., I. Horrocks, F. van Harmelen, S. Decker, M. Erdmann, et M. C.A. Klein, Oil in a nutshell, In Proc. of the 12th Int. Conf. on Knowledge Engineering and Knowledge Management (EKAW-2000), Juan-les-Pins, France, 2000

**(Fensel, 2001)**

Fensel D., *Ontologies : Silver Bullet for Knowledge Management and Electronic Commerce*, Springer-Verlag, ISBN : 3540416021, Berlin, 2001

**(Finnie et Sun, 2003)**

Finnie G., et Z. Sun, A logical foundation for the CBR Cycle, *Int J Intell Syst*, vol. 18 (4), pp. 367--382, 2003

**(Foote et Yoder, 1998)**

Foote B., et J. Yoder, Metadata and Active Object-Models, Collected papers from the PLoP 98 and EuroPloP 98 Conference, Technical Report #wucs-98-25, Dept. of Computer Science, Washington University, Sept 1998

**(Fowler, 1997)**

Fowler M., Analysis Patterns : Reusable Object Models, Addison-Wesley, 1997

**(Fridman-Noy et al., 2000)**

Fridman-Noy N., R.W. Ferguson, et M.A. Musen, The knowledge model of Protege-2000 : Combining interoperability and flexibility, In 2nd International Conference on Knowledge Engineering and Knowledge Management (EKAW2000), Juan-les-Pins, France, 2000

**(Fuchs, 1997)**

Fuchs B., Représentation des connaissances pour le raisonnement à partir de cas, Le système ROCADE, Thèse de Doctorat : Informatique, Saint-Etienne, 1997

**(Gaillard, 2005)**

Gaillard I., Etat des connaissances sur le retour d'expérience industriel et ses facteurs socioculturels de réussite ou d'échec, cahier de l'ICSI, vol. 2005-02, 2005

**(Gamma et al., 1995)**

Gamma E., R. Helm, R. Johnson et J. Vlissides, Design Patterns : Elements of Reusable Software Architecture, Addison-Wesley, 1995

**(Genesereth et Fikes, 1992)**

M. R.Genesereth, et R. Fikes, Knowledge Interchange Format, version (3.0), reference manual, Technical Report, Logic-92-1, Computer Science Department, Stanford University, 1992

**(George et Pal, 1996)**

George T., et N.R. Pal, Quantification of Conflict in Dempster-Shafer *framework* : A New Approach, Int. Journ. Of Gen. Systems, vol. 24 (4), pp. 407--423, 1996

**(Goldberg et Robson, 1983)**

Goldberg A., et D. Robson, Smalltalk80 : the Language and its Implementation, Addison Wesley, Reading, Mass., May 1983

**(Goncalves et Leitao, 2007)**

Goncalves E., et A.M. Leitao, Offline execution in workflow-enabled Web applications, Quality of Information and Communications Technology, QUATIC 2007, 6th International Conference on the Quality of Information and Communications Technology, pp. 204--207, 12-14 Sept. 2007

**(Gouriveau, 2003)**

Gouriveau R., Analyse des risques, formalisation des connaissances et structuration des données pour l'intégration des outils d'étude et de décision, Thèse de Doctorat : Systèmes Industriels, Institut National Polytechnique de Toulouse, 2003

**(Grelot et Guillaume, 2004)**

Grelot F., et B. Guillaume, Retour d'expérience, une façon d'écrire l'histoire ?, Les inondations du Gard de septembre 2002 ou l'expertise scientifique collective au service d'un retour d'expérience, colloque Pollutec 04, 2004

**(Grosso et al., 1999)**

Grosso W., H. Eriksson, R. Ferguson, J. Gennari, S. Tu, et M. Musen, Knowledge modeling at the millennium (the design and evolution of Protégé-2000), In Proc. of the Knowledge Acquisition Workshop (KAW-99), Banff, Canada, 1999

**(Gruber, 1993)**

Gruber T.R., A Translation Approach to Portable Ontology Specifications, Knowledge Acquisition, vol. 5 (2), pp. 199--220, 1993

**(Hansen, 2000)**

Hansen B.K., Weather Prediction Using Case-Based Reasoning and Fuzzy Set Theory, Master of Computer Science Thesis : Technical University of Nova Scotia : Halifax, Nova Scotia, Canada, 2000

**(Hayes, 1977)**

Hayes P.J., In defence of logic, In IJCAI-77, pp. 559--565, 1977

**(Hayes, 1980)**

Hayes P.J., The logic of *frames*, In Metzging 1980, pp. 46--61, 1980

**(Hedlund et Nonaka, 1993)**

Hedlund G., et I. Nonaka, Models of knowledge management in the West and Japan, In P. Lorange, B. Chakravarthy, J. Roos, et A. van de Ven (Eds.), *Implementing Strategic Processes : change, learning and co-operation*, Blackwell, London, 1993

**(Hermosillo et al., 2005)**

Hermosillo Worley J., H. Rakoto, B. Grabot, et L. Geneste, A competence approach in the experience feedback process, *Integrating Human Aspects in Production Management, Series : IFIP International Federation for Information Processing*, vol. 160, pp. 220--235, Ed. Springer, 0-387-23065-3, 2005

**(Ingalls et al., 1997)**

Ingalls D., T. Kaehler, J. Maloney, S. Wallace, et A. Kay, Back to the future : the story of Squeak, a practical Smalltalk written in itself, *Proceedings of the 12th ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, Atlanta, Georgia, United States, A. M. Berman, Ed. OOPSLA 97, ACM Press, New York, NY, pp. 318--326, 1997

**(Jaczynski et Trousse, 1994)**

Jaczynski M., et B. Trousse, Fuzzy logic for the retrieval step of a case-based reasoner, 2nd European Workshop on Case-Based Reasoning, EWCBR 94, Chantilly, France, pp. 313--320, 1994

**(Jaczynski et Trousse, 1997)**

Jaczynski M., et B. Trousse, CBR\*Tools : une bibliothèque objet pour l'indexation des cas par situations comportementales, *Rapport de recherche de l'Institut National de Recherche en Informatique et en Automatique (INRIA)*, (3215), 1997

**(Jaczynski, 1998)**

Jaczynski M., *Modèle et plate-forme à objets pour l'indexation des cas par situations comportementales : application à la navigation sur le Web*, Thèse de Doctorat : Informatique, Nice-Sophia, Antipolis, 1998

**(Johnson et Foote, 1988)**

Johnson R., et B. Foote, Designing Reusable Classes, *Journal of Object-Oriented Programming*, SIGS, 1-5 July, pp. 22--35, 1988

**(Johnson et Woolf, 1997)**

Johnson R., et B. Woolf, Type Object, *Pattern Languages of Program Design 3*, R. Martin, D. Riehle, et F. Buschmann, eds., Addison-Wesley, Reading, MA., pp. 47--66, 1997

**(Johnson, 1997)**

Johnson R., *Frameworks : Patterns, Components, Communications of the ACM*, vol. 40, pp. 10--17, 1997

**(Jousselme et al., 2001)**

Jousselme A.L., D. Grenier, et E. Bosse, A new distance between two bodies of evidence, *Information Fusion*, vol. 2, pp. 91--101, 2001

**(Kamsu et al., 2008)**

Kamsu Foguem B., T.Coudert, C.Béler, et L.Geneste, Knowledge Formalization in Experience Feedback Processes : An Ontology-Based Approach, *Journal on Computers in Industry*, (5), vol. 59, Ed. Elsevier, 2008

**(Kiczales et al, 1991)**

Kiczales G., J. Rivieres et D.G. Bobrow, *The Art of the Metaobject Protocol*, 1991

**(Kiczales et al., 1997)**

Kiczales G., J. Lamping, A. Mendhekar, C. Maeda, C. Lopez, J.-M. Loingtier, et J. Irwin, Aspect-oriented programming, In ECOOP 97, vol. 1241 of Lecture Notes in Computer Science, pp. 220--242, Springer-Verlag, 1997

**(Kifer et al., 1995)**

Kifer M., G. Lausen, et J. Wu, Logical foundations of object-oriented and frame-based languages, Journal of the ACM, vol. 42 (4), pp. 741--843, 1995

**(Knublauch, 2002)**

Knublauch H., An Agile Development Methodology for Knowledge-Based Systems Including a Java *framework* for Knowledge Modeling and Appropriate Tool Support Dissertationsschrift (PhD thesis), University of Ulm, 2002

**(Kohlas et Monney, 1995)**

Kohlas J., et P.A. Monney, A Mathematical Theory of Hints : An Approach to Dempster-Shafer Theory of Evidence, Lecture Notes in Economics and Mathematical Systems, (425), Springer-Verlag, 1995

**(Kolb, 1984)**

Kolb D., Experiential learning : experience as the source of learning and development, Englewood cliffs, Prentice-Hall, New Jersey, 1984

**(Kolodner, 1993)**

Kolodner J.L., Case-Based Reasoning, Morgan Kaufmann Publishers, Inc., ISBN : 1-55860-237-2, 1993

**(Le Moigne, 1987)**

Le Moigne J.L., Qu'est-ce un modèle ? Publié dans Les modèles expérimentaux et la clinique (AMRP 1985), Confrontations Psychiatriques, numéro spécial consacré aux modèles, 1987

**(Lebowitz, 1999)**

Lebowitz J., eds., Knowledge Management Handbook, CRC Press, 1999

**(Lefevre, 2001)**

Lefevre E., Fusion adaptée d'informations conflictuelles dans le cadre de la théorie de l'évidence, thèse de doctorat en Physique, INSA de Rouen, Laboratoire Perception Systèmes et Information (PSI, FRE 2645 CNRS), novembre 2001

**(Leroy, 2001)**

Leroy P., Utilisation du retour d'expérience pour l'utilisation de la maintenance, Convergence et divergence des pratiques des retours d'expérience technique et humain, Collection de l'Institut de Sûreté de Fonctionnement, Paris, 2 janvier 2001

**(Loucopoulos et Karakostas, 1995)**

Loucopoulos P., et V. Karakostas, System Requirements Engineering, Mc Graw-Hill, Inc., New York, NY, USA, 1995

**(Mac Gregor et Burstein, 1991)**

Mac Gregor R., et M.H. Burstein, Using a description classifier to enhance knowledge representation. IEEE Expert, vol. 6 (3), pp. 41--46, 1991

**(Maes, 1987)**

Maes P., Computational Reflection, Thèse de doctorat : Informatique, Bruxelles, Belgique, Bruxelles (Vrije Universiteit), 1987

**(Malvache et al., 1994)**

Malvache P., C. Eichenbaum, et P. Prieur, La maîtrise du retour d'expérience avec la méthode REX, Performances humaines et Techniques, (69), pp. 6--13, mars-avril 1994

**(Masterman, 1962)**

Masterman M., Semantic message detection for machine translation, using an interlingua, In Proceedings of the 1961 International Conference on Machine Translation of Languages and Applied Linguistic Analysis, pp. 438--475, London, UK, 1962

**(Mille et al., 2006)**

Mille A., M. Caplat, et M. Philippon, Faciliter les activités des utilisateurs d'environnements informatiques : quoi, quand, comment ?, INTELLECTICA, vol. 2 (44), pp. 121--143, ISSN : 0769-4113, 2006

**(Mille, 2006)**

Mille A., From case-based reasoning to traces-based reasoning, Annual Reviews in Control, vol. 30 (2), pp. 223--232, Elsevier, ISSN : 1367-5788, 2006

**(Milton, 2008)**

Milton N.R., Knowledge Technologies, Milan, Polimetrica, 2008, [http://eprints.rclis.org/archive/00012800/01/Knowledge\\_Technologies\\_reading.pdf](http://eprints.rclis.org/archive/00012800/01/Knowledge_Technologies_reading.pdf)

**(Minsky, 1975)**

Minsky M., A *framework* for representing knowledge, In P. Winston, editor, The Psychology of Computer Vision, pp. 211--277, Mc Graw-Hill, New-York, 1975

**(MOF, 2004)**

Meta object facility (MOF) 2.0 score final adopted specification, Technicalreport, Object Management Group, 2004

**(Moore, 1982)**

Moore R.C., The role of logic in knowledge representation and common sense reasoning, In AAAI-82, pp. 428--433, 1982

**(Nebel, 1990)**

Nebel B., Reasoning and Revision in Hybrid Representation Systems, Lecture Notes in Artificial Intelligence, Springer-Verlag, 1990

**(Nierstrasz et al. 2005)**

Nierstrasz O., A. Bergel, M. Denker, S. Ducasse, M. Gaelli et R. Wuyts, On the Revival of Dynamic Languages, Proceedings of Software Composition 2005, Thomas Gschwind and Uwe Aßmann (Eds.), vol. 3628, LNCS 3628, 2005, pp. 1—13, Invited paper.

**(Nonaka et Takeuchi, 1995)**

Nonaka I., et H. Takeuchi, The Knowledge-Creating Company : How Japanese Companies Create the Dynamics of Innovation, Oxford and New York, Oxford University Press, 1995

**(Nonaka, 1969)**

Nonaka I., A Dynamic Theory of Organisational Knowledge Creation, Organization Science, vol. 5 (1), 1969

**(Pearl, 1990)**

Pearl J., Reasoning with Belief Functions : an Analysis of Compatibility, Intern. J. Approx. Reasoning, vol. 4, pp. 363--390, 1990

**(Petit-Renaud et Dencœux, 1999)**

Petit-Renaud S., et T. Dencœux, Handling different forms of uncertainty in regression analysis : a fuzzy belief structure approach, In A. Hunter et S. Pearsons (Eds), Symbolic and quantitative approaches to reasoning and uncertainty (ECSQARU 99), pp. 340--351, London, June 1999

**(Piaget, 1950)**

Piaget J., Introduction à l'épistémologie génétique, Tomes I, II et III, Presses universitaires de France, Paris, 1950

**(Polanyi, 1962)**

Polanyi M., *Personal knowledge : towards a post critical philosophy*, London, Routledge, 1962

**(Prax, 1997)**

Prax J.Y., *Le guide du knowledge management, Concepts et pratiques du management de la connaissance*, p. 261, Ed. Dunod 1997

**(Pree, 1994)**

Pree W., *Design Patterns for Object-Oriented Software Development*, Addison-Wesley, Reading, MA, 1994

**(Quillian, 1966)**

Quillian M.R., *Semantic Memory*, PhD thesis, Carnegie Institute of Technology, Pittsburgh, PA, BBN Report AFCRL-66-189, Bolt, Beranek, and Newman Inc., October 1966

**(Rakoto et al., 2002)**

Rakoto H., J. Hermosillo Worley, et M. Ruet, *Integration of experience based decision support in industrial processes*, International Conference on Systems, Man and Cybernetics, IEEE SMC 02, In A. El Kamel, K. Mellouli, et P. Borne (Eds.), *Bridging the Digital Divide, Cyber-development, Human Progress, Peace and Prosperity*, Hammamet, Tunisie, ISBN : 2-9512309-4-X, 6-9 October 2002

**(Rakoto, 2004)**

Rakoto H., *Intégration du Retour d'Expérience dans les processus industriels - Application à Alstom Transport* (in French), PhD thesis, National Polytechnic Institute of Toulouse (France), October 2004

**(Ramasso et al., 2008)**

Ramasso E., M. Rombaut et D. Pellerin, *Modèle des Croyances Transférables : Représentation des connaissances, Fusion d'informations*, RAPPORT INTERNE GIPSA-lab, Département Images et Signal, Université Joseph Fourier, 2008

**(Ramasso, 2007)**

Ramasso E., *Reconnaissance de séquences d'états par le Modèle des Croyances Transférables et application à l'analyse de vidéos d'athlétisme*, thèse de doctorat en Electronique, Electrotechnique, Automatique et Traitement du Signal, Université Joseph Fourier, décembre 2007

**(Rechenmann, 1985)**

Rechenmann F., *SHIRKA : Mécanismes d'inférences sur une base de connaissances centrée-objet*, 5ème congrès Reconnaissances des Formes et Intelligence Artificielle AFCET-ADI-INRIA, novembre 1985

**(Renaud et al., 2007a)**

Renaud J., B. Chebel-Morello, B. Fuchs, et J. Lieber, *Raisonnement à partir de cas 1 : conception et configuration de produits*, *Traité IC2, série Informatique et systèmes d'information*, Lavoisier, 2007

**(Renaud et al., 2007b)**

Renaud J., B. Chebel-Morello, B. Fuchs, et J. Lieber, *Raisonnement à partir de cas 2 : surveillance, diagnostic et maintenance*, *Traité IC2, série Informatique et systèmes d'information*, Lavoisier, 2007

**(Renggli et al., 2007)**

Renggli L., S. Ducasse, et A. Kuhn, *Magritte - A Meta-Driven Approach to Empower Developers and End Users*, In *ACM/IEEE 10th International Conference On Model Driven Engineering Languages And Systems (MODELS)*, September, LNCS, Springer-Verlag, 2007, To appear

**(Renggli, 2006)**

Renggli L., *Magritte meta-described web application development*, Masters thesis, University of Bern, 2006

**(Rexao, 2008)**

Groupement Rexao, <http://www.rexao.org>, 2008

**(Richens, 1958)**

Richens R.H., Interlingua machine translation, *Computer Journal*, vol. 1 (3), pp. 144--147, October 1958

**(Riehle et al., 2000)**

Riehle D., M. Tilman et R. Johnson, Dynamic Object Model, In E. Wallingford et A. Garrido, editors, *Proceedings of the 7th Annual Conference on Pattern Languages of Programs (PLoP2000)*, Monticello, Illinois, USA, August 2000

**(Riehle et al., 2005)**

Riehle D., M. Tilman et R. Johnson., Dynamic Object Model, In *Pattern Languages of Program Design 5*, Edited by D. Manolescu, M. Völter, et J. Noble. Reading, MA : Addison-Wesley, 2005

**(Riehle, 1997)**

Riehle D., A Role-Based Design Pattern Catalog of Atomic and Composite Patterns Structured by Pattern Purpose, Ubilab Technical Report 97-1-1, Zürich, Switzerland : Union Bank of Switzerland, 1997

**(Riehle, 2000)**

Riehle D., *framework Design : A Role Modeling Approach*, *Softwaretechnik-Trends*, vol. 20 (2), 2000

**(Rivard, 1996)**

Rivard F., Smalltalk : a reflective language, In : *Proceedings of REFLECTION 96*, pp. 21--38, 1996

**(Rougegrez-Loriette, 1994)**

Rougegrez-Loriette S., Prédiction de processus à partir de comportements observés : le systèmes REBECAS, Thèse de Doctorat : Informatique, Paris, 1994

**(Ruet, 2002)**

Ruet M., Capitalisation et réutilisation d'expériences dans un contexte multiacteur, thèse de doctorat en Systèmes Industriels, Institut National Polytechnique de Toulouse, 2002

**(Salotti, 1992)**

Salotti S., Filtrage flou et représentation centrée objet pour raisonner par analogie : le système FLORAN, Thèse de Doctorat : Informatique, Paris, 1992

**(Schank, 1982)**

Schank R.C., *Dynamic Memory*, Cambridge, Cambridge University Press, 1982

**(Schreiber et al., 1993)**

Schreiber A.T., B.J. Wielinga, et J.A. Breuker, editors, KADS, A Principled Approach to Knowledge-Based System Development, vol. 11 of *Knowledge-Based Systems Book Series*, Academic Press, London, ISBN : 0-12-629040-7, 1993

**(Schreiber et al., 1994)**

Schreiber A.T., B.J. Wielinga, R. de Hoog, H. Akkermans, et W. van de Velde, CommonKADS : A Comprehensive Methodology for KBS Development, *IEEE Expert*, 1994

**(Séguéla, 2001)**

Séguéla P., Construction de modèles de connaissances par analyse linguistique de relations lexicales dans les documents techniques, Thèse de doctorat, Université Paul Sabatier, Toulouse, 2001

**(Setnes et al., 1998)**

Setnes M., R. Babuska, U. Kaymak, et H.R. van Nauta Lemke, Similarity measures in fuzzy rule base simplification, *IEEE Transactions on Systems, Man and Cybernetics - part B : Cybernetics*, vol. 28, pp. 376--386, 1998

**(Shafer, 1976)**

Shafer G., *A Mathematical Theory of Evidence*. Princeton University Press, 1976

**(Smets et Kennes, 1994)**

Smets P., et R. Kennes, The transferable belief model, *Artificial Intelligence*, vol. 66, pp. 191--234, 1994

**(Smets, 1990)**

Smets P., Constructing the pignistic probability function in a context of uncertainty, *Uncertainty in Artificial Intelligence 5* (M. Henrion *et al.*, Eds.), North-Holland, Amsterdam, pp. 29--39, 1990

**(Smets, 1993)**

Smets Ph., No Dutch Book can be built against the TBM even though update is not obtained by Bayes rule of conditioning, *SIS, Workshop on Probabilistic Expert Systems*. R. Scozzafava (ed.), Roma, Italy, pp. 181-204, 1993

**(Smets, 1994)**

Smets P., What is Dempster-Shafer's model ? *Advances in the Dempster-Shafer Theory of Evidence*, R.R.Yager, M. Fedrizzi, et J. Kacprzyk (eds.), Wiley, pp. 5--34, 1994

**(Smets, 1997)**

Smets P., Imperfect information : Imprecision - Uncertainty, *Uncertainty Management in Information Systems, From Needs to Solutions*. A. Motro et P. Smets (eds.), Kluwer Academic Publishers, pp. 225--254, 1997

**(Smets, 2002)**

Smets Ph., Decision Making in a Context where Uncertainty is Represented by Belief Functions. *Belief Functions in Business Decisions*, Srivastava R. and Mock, T.J. (ed.) Physica-Verlag, Heidelberg, Germany, pp. 17-61, 2002

**(Smets, 2005)**

Smets Ph., Decision Making in the TBM: the Necessity of the Pignistic Transformation. *Int. J. Approximate Reasoning*, 38, pp. 133--147, 2005.

**(Sowa, 1984)**

Sowa J.F., *Conceptual Structures : Information Processing in Mind and Machine*, Addison-Wesley Publishing Company, Reading, MA, 1984

**(Studer *et al.*, 1998)**

Studer R., V.R. Benjamins, et D. Fensel, *Knowledge Engineering : Principles and Methods*, *Data and Knowledge Engineering* 25, pp. 161--197, 1998

**(Sun et Finnie, 2004)**

Sun Z., et G. Finnie, Brain-like architecture and experience-based reasoning, In: *Proc, 7th Joint Conf on Information Sciences (JCIS)*, Sept 26-30, 2003 Cary, North Carolina, USA., pp. 1735--1738, 2004

**(Sun et Finnie, 2007)**

Sun Z., et G. Finnie, A fuzzy logic approach to experience-based reasoning : *Research Articles*, *International Journal of Intelligent Systems* archive, vol. 22 , Issue (8), pp. 867--889, ISSN : 0884-8173, August 2007

**(Sun et Huo, 2006)**

Sun Z., et H. Huo, The Engineering of Experience, *Intelligent Systems Design and Applications*, ISDA apos. 06, *Sixth International Conference on Digital Object*, vol. 2, pp. 1114--1117, oct. 2006

**(Vannoorenberghe, 2003)**

Vannoorenberghe P., Un état de l'art sur les fonctions de croyance appliquées au traitement de l'information, *Revue I3*, 3(1):9-45, 2003

**(Verot, 2001)**

Vérot Y., Retour d'Expérience dans les industries de procédé, *Techniques de l'Ingénieur*, AG 4 610, juillet 2001

**(Villemeur, 1988)**

Villemeur A., Sûreté de fonctionnement des systèmes industriels, Ed.Eyrolles Paris, 1988

**(Vogel, 1988)**

Vogel C., Génie Cognitif, Masson, 1988

**(Wang, 1995)**

Wang P., Non-Axiomatic Reasoning System : Exploring the Essence of Intelligence, PhD thesis, Indiana University, 1995

**(Wang, 2004)**

Wang P., Cognitive Logic versus Mathematical Logic, Working Notes of the Third International Seminar on Logic and Cognition, Guangzhou, May 2004

**(Wang, 2005)**

Wang P., Experience-Grounded Semantics : A theory for intelligent systems, Cognitive Systems Research, vol. 6 (4), pp. 282--302, 2005

**(Wang, 2006)**

Wang P., Rigid Flexibility, The Logic of Intelligence, Springer Netherlands, ISBN : 978-1-4020-5044-2 (Print) 978-1-4020-5045-9, 2006 (on line)

**(Warmer et Kekleppe, 1998)**

Warmer J., et A. Kekleppe, The Object Constraint Language : Precise Modeling with UML, Addison-Wesley, Reading, MA, 1998

**(Weber et al., 2001)**

Weber R.O., D.W. Aha et I.Becerra-Fernandez, Intelligent lessons learned systems, Expert Systems with Applications, vol. 20 (1), pp. 17--34, janvier 2001

**(Weil-Barais, 1994)**

Weil-Barais A., L'homme cognitif, PUF, 1994

**(Woods, 1975)**

Woods W.A., What's in a link : Foundations for semantic networks, In D.G. Bobrow et A.M. Collins, editors, Representation and Understanding : Studies in Cognitive Science, pp. 35--82, New York, NY, Academic Press, 1975

**(Wuyts, 2001)**

Wuyts R., A Logic Meta-Programming Approach to Support the Co-Evolution of Object-Oriented Design and Implementation, Thèse de Doctorat : Informatique, Belgique, Bruxelles (Vrije Universiteit), 2001

**(XMI, 1998)**

XML Metadata Interchange (XMI), OMG, <http://www.omg.org/cgi-bin/doc?ad/98-10-05>, 1998

**(Xu et Smets, 1994)**

Xu H., et P. Smets, Evidential Reasoning with Conditional Belief Functions, Uncertainty in AI 94, D. Heckerman, D. Poole, et R. Lopez de Mantaras (eds), M. Kaufmann, San Mateo, California 1994, pp. 598--606, 1994

**(Yahia et Rougegrez-Loriette, 2005)**

Yahia S.O., et S. Loriette-Rougérez : Fuzzy Querying of Evolutive Situations : Application to Driving Situations, JAC III, vol. 9 (1), pp. 13--17, 2005

**(Yoder et al., 2001)**

Yoder J., F. Balaguer et R. Johnson, Architecture and design of adaptive object-models, ACM SIGPLAN Notices, vol. 36, pp. 50--60, 2001

**(Yoder et Johnson, 2002)**

Yoder J. et R. Johnson, The adaptive object model architectural style, In Proceeding of The Working IEEE/IFIP Conference on Software Architecture 2002, (WICSA302), Août 2002

**(Zadeh, 1965)**

Zadeh L.A., Fuzzy sets, Information & Control, (8), pp. 338--353, 1965

**(Zadeh, 1978)**

Zadeh L.A., Fuzzy Sets as a basis for a theory of possibility, Fuzzy Sets and Systems, (1), pp. 3--28, 1978

**(Zadeh, 1986)**

Zadeh L.A., A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination, AI Mag. vol. 7, pp. 85--90, 1986

**(Zorn et Taylor, 2004)**

Zorn T.E., et J.R. Taylor, Knowledge management and/as organizational communication, In D. Tourish et O. Hargie (Eds.), Key Issues in Organizational Communication, London and New York, Routledge, 2004