Structuration des connaissances et des savoir-faire pour l'amélioration du système de production

Thierry Erbeja

Université
Louis Pasteur
de Strasbourg

Ecole Nationale Supérieure
des Arts et Industries
de Strasbourg

Thèse

présentée pour obtenir le titre de Docteur de
l'Université Louis Pasteur de Strasbourg
le 21 décembre 2001

option Sciences pour l'Ingénieur
spécialité Sciences et Technologie Industrielle

par

Thierry ERBEJA

Structuration des connaissances et des savoir-faire pour l'amélioration du système de production

Membres du jury :

M. Bernard Mutel	Directeur, Professeur à l'Ecole Nationale de Arts et Industries de Strasbourg
M. Roland De Guio	Professeur à l'Ecole Nationale de Arts et Industries de Strasbourg
M. Jacques Richard	Rapporteur externe, Professeur à l'Université Henri Poincaré de Nancy
M. Jerzy Korczak	Rapporteur interne, Professeur à l'Université Louis Pasteur de Strasbourg
M. Denis Gien	Rapporteur externe, Professeur à l'Institut Français de Mécanique Avancée de Clermont-Ferrand

Ce travail bien que référencé sous le seul nom de son auteur, est néanmoins redevable à de nombreuses personnes :

Je tiens tout d'abord à remercier Mr Roland De Guio et Mr Bernard Mutel pour qui ont su m'amener au terme de ce travail.

J'exprime ma gratitude aux membres du jury, qui se sont penchés sur ce mémoire pour l'améliorer.

Un grand merci à toute l'équipe du L.R.P.S que j'ai côtoyée pendant mes années de doctorat. J'ai beaucoup appris à leur contact.

Et le mot de la fin, mais non le moindre pour mes proches qui ont su me soutenir, m'encourager et me donner le courage d'aller jusqu'au bout.

A Céline

Table des matières

Chapitre I

Introduction

I.1 Les problèmes de classification pour l'organisation industrielle

Face à la mondialisation des marchés dans le cadre d'une concurrence accrue où l'offre est supérieure à la demande, les entreprises doivent pouvoir répondre à de multiples critères concernant les coûts, les délais, la qualité et les services. C'est notamment le cas des entreprises manufacturières produisant par lots de moins de 50 unités. Concevant et produisant des produits variés en petites et moyennes séries répétitives, d'une durée de vie de plusieurs années, elles représentent plus de la moitié des entreprises manufacturières et constituent notre domaine d'investigation. Pour améliorer les critères de performance précédents, deux concepts dominants ont émergé au cours de ces dernières décennies : le CIM, acronyme de Computer Integrated Manufacturing dont une traduction serait « production automatisée », et le JIT-TQC, acronyme de Just in Time-Total Quality Control, le « juste à temps » incluant la qualité totale. Le concept CIM vise à l'amélioration de la production par l'informatisation des technologies et l'intégration des activités en les liant par des moyens informatiques et des bases de données. Le JIT-TQC, quant à lui, est un concept d'ordre organisationnel qui affirme que l'amélioration des performances de l'entreprise passe par l'élimination des gaspillages et des opérations inutiles, par l'implication des employés et par la qualité du système logistique. La Technologie de Groupe (TG) est un concept nettement plus ancien qui permet de faire le lien entre le CIM et le JIT-TQC. Aussi peut-on utiliser les principes de la Technologie de Groupe pour réorganiser et améliorer les flux de production (l'un des objectifs du JIT), et constituer des îlots de fabrication. Une fois créés, ces îlots sont de bons candidats à l'automatisation avec les concepts CIM. Un autre raisonnement consiste à partir des applications du CIM pour aboutir au JIT. Les systèmes de CFAO automatisent les tâches de conception classiques sans intégrer les concepts du JIT. Par l'application des concepts de la TG on peut créer des outils d'analyse des similarités qui permettent d'extraire des bases de données CFAO les produits similaires à ceux de la commande client traitée, apportant ainsi des gains de temps substantiels (objectif du JIT) dans les phases précédant la production.

La Technologie de Groupe (Kusiak 1987; Askin 1993; Vakharia 1994) est un concept qui propose de rationaliser la production en tirant profit des analogies entre les éléments du système de production. La classification est une des méthodes appropriées à l'analyse des ces analogies. La TG s'applique à tous les niveaux de l'entreprise, du bureau d'études aux ateliers.

En conception, les objectifs de la classification sont de réduire le nombre d'articles en éliminant la variété inutile et en standardisant les formes et dimensions des pièces. Les familles servent aussi à capitaliser le savoir-faire relatif aux articles existants. On utilisera les connaissances relatives à une famille pour traiter les projets de pièces similaires à cette famille. Dans la majorité des cas, l'objectif concret de la classification sera de réaliser un nombre limité d'esquisses paramétrables, représentatives des variantes existantes. La classification des pièces s'appuiera donc sur des similitudes morphologiques, dimensionnelles et fonctionnelles. Considérons par exemple, le bureau d'étude d'une entreprise fabricant des réducteurs de puissance. Les réducteurs sont composés de trois types de produits : les bâtis, l'habillage de la machine (couvercles, cales, visserie) et les produits mobiles (arbres, pignons, etc.). Le bâti et l'habillage sont spécifiques à chaque commande, par contre, la diversité des produits mobiles est susceptible d'être rationalisée. C'est notamment le cas des rotors. Pour cette analyse, le groupe de travail qui mène le projet de classification va regrouper les rotors en familles, dans l'idée de supprimer les variantes inutiles ou anti-productives. Par variante nous n'entendons pas les variantes du produit fini vu du client mais les variantes de conception objectivement injustifiées pouvant être supprimées sans nuire à la qualité du produit et du service rendu au client. La standardisation joue ici le rôle de mémorisation d'un savoir-faire et de rationalisation du processus de conception. Chaque famille sera caractérisée par une esquisse paramétrable qui codifie le savoir-faire de l'entreprise par rapport à cette famille de pièces.

En fabrication, la classification a pour objectif de réduire la diversité des processus de traitement (les gammes), de les standardiser et de réutiliser l'existant pour accélérer l'élaboration des nouvelles gammes (De Guio 1990). On cherchera donc à définir des familles de produits dont les processus de fabrication sont semblables, et à associer une gamme standard à chaque famille. En reconnaissant la famille d'une nouvelle pièce, il est possible d'identifier sa gamme de fabrication, c'est le principe de la méthode des variantes. Les critères de classification sont différents de ceux utilisés en conception. Deux pièces de même forme mais de matières différentes, l'une à fabriquer en petite série et l'autre en grande série n'auront pas les même gammes de fabrication (Mutel 1993). Les critères de classifications doivent donc décrire le processus de fabrication des pièces étudiées. Il y a principalement deux façons d'aborder le problème, soit en utilisant des critères de fabrication, soit en utilisant des critères de gamme (Sedqui 1995) ; les plus simples étant les critères de fabrication. Ils décrivent, les opérations nécessaires à l'élaboration de la pièce ainsi que certaines caractéristiques de la pièce liées au processus de fabrication. Les critères de fabrication ne tiennent pas compte de l'ordre entre les opérations. Les critères de gammes sont plus complexes. Ils comprennent (Nadif 1985; Mutel 1992) les opérations, l'ordre entre les opérations, la différence de rang entre les opérations (permutations) et le nombre d'opérations.

En production, la classification permet de décomposer l'atelier en sous-systèmes indépendants : les îlots. Un îlot est un groupe de machines dédiées à la fabrication d'une famille de produits. Le problème de la définition des îlots de production (« cell formation » en anglais) a donné lieux à de nombreux travaux (De Guio 1998). Selon les objectifs des utilisateurs, les critères à optimiser peuvent être différents. Les trois objectifs principaux sont :

- Définir des sous-systèmes de production indépendants : il s'agit de concevoir des familles de pièces qui utilisent les mêmes ressources. Les critères de classification sont donc les ressources nécessaires à l'élaboration de la pièce.
- Optimiser la charge des machines dans les îlots : il faut tenir compte des ressources mais aussi de la charge de travail que nécessite le traitement des articles. Ce sont les critères de charge.
- Optimiser les déplacements dans l'atelier : il faut tenir compte des ressources, mais aussi des quantités transportées d'une machine à l'autre, du poids ou des volumes, ce sont les critères de flux.

En gestion de production, une rationalisation des fournisseurs diminue les écritures, diminue les zones de stockage, permet d'obtenir de meilleurs coûts et délais par groupement des achats.

L'implantation de la TG, débute par une étude de faisabilité qui comprend trois volets : l'analyse des flux d'informations, la sensibilisation du personnel et une étude de la faisabilité technique (Feltz 1993). L'étude de faisabilité technique détermine la pertinence de l'utilisation des concepts de la TG dans l'entreprise et définit les principaux jalons organisationnels du projet. C'est une étape d'étude et de validation technique sur un ensemble d'apprentissage. Elle est réalisée par une équipe qui comprend des experts de l'entreprise, (principalement des membres du service concerné) et un spécialiste de la TG. L'étude de faisabilité technique comprend quatre étapes : définir les objectifs, définir un système de codification, réaliser puis valider les familles. La figure I.1 résume cette démarche.

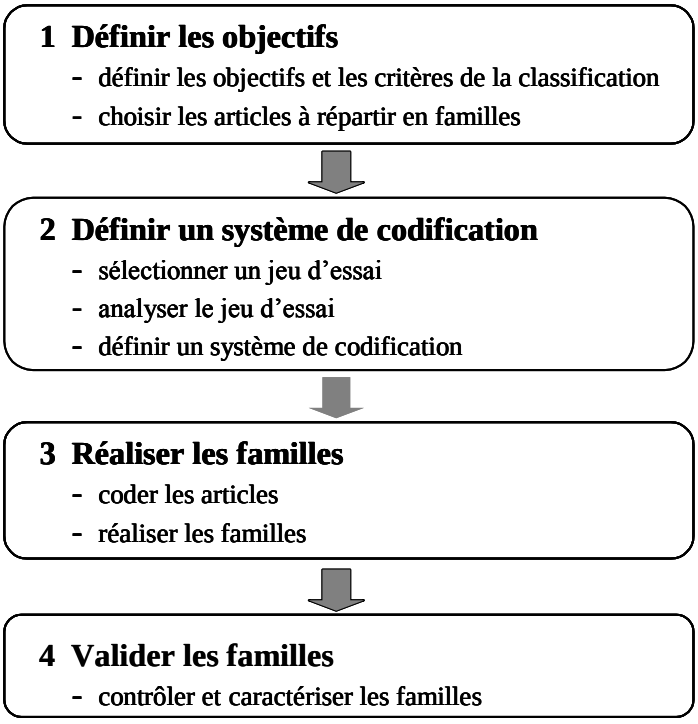


Fig. I.1 Les quatre étapes de l'étude technique d'implantation de la TG.

La méthode de mise en œuvre traditionnelle de la TG est presque entièrement « manuelle ». Les familles sont directement réalisées par les membres de l'équipe. Lors de la première étapes, ils commencent par récolter un échantillon des pièces produites en fonction de l'objectif de l'étude et du type de critère de classification (morphologique, dimensionnel, et fonctionnel en conception, critères de gammes ou de fabrication en fabrication, critères de ressources en production). Au cours de la deuxième étape, ils élaborent un code descriptif. Ce dernier sera le plus souvent construit sur la base de codes TG très généraux comme le code OPITZ ou le code MULTICLASS, mais peut tout aussi bien être spécifique à l'entreprise. Pendant la troisième étape, les familles de produits sont obtenues par analyse du code et ajustements manuels. Il n'y a pas a priori de méthode systématique pour analyser le code et construire les familles. Les experts s'aident éventuellement d'outils de tri tels que ceux qui se

trouvent dans tout système de gestion de base de données pour lister les familles qu'ils conçoivent et vérifier la qualité des concepts qui leur sont liés. L'étape finale de validation repose sur l'hypothèse que les membres du groupe sont capables d'évaluer la pertinence d'une classification par rapport aux objectifs de la TG. Les membres du groupe sont des experts de leur métier et ce métier n'a pas été modélisé. Ils sont généralement les seules personnes à même de juger si les familles obtenues caractérisent l'activité de leur service et répondent aux objectifs de la TG au niveau plus générale de l'entreprise considérée dans sa globalité. L'étape de validation se déroule en deux temps (Mutel 1993). Dans un premier temps, les experts évaluent la pertinence des familles par rapport aux objectifs de la TG. Dans un deuxième temps, il faut déterminer l'intention des familles, c'est à dire une condition nécessaire et suffisante d'appartenance à la famille. L'intension permet de déterminer si un nouvel objet doit être rangé ou non dans la famille considérée. Les intentions sont utilisées sur un autre jeu de pièces pour tester la stabilité des familles.

L'étude de faisabilité comprend généralement de nombreux retours en arrière. C'est une démarche empirique. A l'issue de l'étape (4) de validation, les experts sont souvent amenés à modifier le jeu d'articles utilisé et le système de codification. Toute la difficulté de la démarche réside dans la recherche de critères de classification pertinents par rapport aux objectifs de la TG. Les experts connaissent implicitement un critère d'évaluation de la pertinence d'une partition, mais ils ne sont pas capables de le formuler en terme de critères de classification qui permettra à partir de la description des pièces de les classifier. C'est en examinant des partitions successives qu'ils affinent les critères de classification. Les délais qu'implique cette méthode presque entièrement « manuelle » dépendent beaucoup des qualités de synthèse et de savoir-faire des experts de l'entreprise. Ils varient entre 3 et 6 mois. La conception des familles mobilise des membres clés de l'entreprise durant cette période. De plus la méthode s'applique difficilement à une production changeante. C'est pourquoi, malgré une grande variété d'applications potentielles, peu d'entreprises s'appuient rigoureusement sur les principes de la TG. Les procédures connues de formation de familles de pièces sont ressenties comme longues et fastidieuses au point de décourager leur application par les ingénieurs et de placer les chercheurs dans une impasse (De Guio 1998). Afin d'accélérer l'étude d'implantation de la TG, la majorité des contributions proposent des techniques de classification basées sur des méthodes d'analyse typologique (Mutel 1984; Han 1986; Seifoddini 1986; Nadif 1987; Offodile 1991; Sharker 1996), excepté pour la formation des îlots de production, où les méthodes sont beaucoup plus diversifiées. Les outils d'analyse typologique permettent de classifier un ensemble d'objets en fonction de leur description et sont a priori tout indiqués pour résoudre l'étape de construction des familles (étape (3) de la figure I.1). Cependant, il existe très peu d'études qui évaluent dans quelle mesure, leur emploi permet d'accélérer, non pas l'étape de classification proprement dite, mais l'ensemble du processus d'implantation de la TG. Après une brève introduction aux méthodes d'Analyse Typologique, nous évaluerons dans quelle mesure elles contribuent à réduire les délais de l'étude d'implantation de la TG.

I.2 Approche par analyse typologique

Les outils d'analyse typologique sont conçus pour décomposer un ensemble d'objets décrits par de nombreuses variables, en groupes homogènes. Les groupes doivent être les plus homogènes possibles et les moins nombreux possibles. Ce sont des méthodes descriptives qui visent à rendre explicite la structure interne des données avec le maximum d'objectivité et le minimum d'hypothèses de la part de l'utilisateur (Chandon 1981). En l'absence d'informations supplémentaires, les méthodes d'analyse typologique rassemblent les objets qui se ressemblent en les comparant sur un ensemble fixe de caractéristiques considérées comme d'égale importance. La démarche standard de mise en œuvre comporte 4 étapes : (1) la construction des données, (2) la définition d'un indice de similarité et le calcul du tableau de similarité, (3) la formation des groupes et (4) l'évaluation ou la validation des résultats (cf. figure I.2).

Les méthodes d'analyse typologique imposent des contraintes assez fortes sur la représentation des objets. Ils sont décrits dans un langage dit attributs-valeurs. Chaque objet est représenté par un ensemble d'attributs descriptifs. On trouvera en annexe II.1 le détail des concepts relatifs à la codification des objets en Analyse Typologique. Les données sont considérées comme simples ; elles vérifient les propriétés d'homogénéité, de régularité et de monovaluation (Bouchon-Meunier 1990). Les objets doivent être **homogènes** : il n'existe pas a priori de relations entre les objets à classer. Il n'est pas possible de classifier un ensemble d'objets qui mélange des concepts et des instances. Le code utilisé doit être **régulier**. Les objets à classer sont tous évalués sur les mêmes variables. Par exemple, un code qui comprend la variable diamètre n'est pas régulier sur un ensemble de pièces cylindriques et de pièces carrées. Les variables descriptives doivent être **monovaluées**, c'est à dire qu'elles ne peuvent prendre qu'une seule valeur pour un objet donné. L'étape de classification proprement dite nécessite une bonne connaissance des techniques d'Analyse Typologique. Il faut définir ou choisir la mesure de similarité, car il n'existe pas de mesure de similarité générale qui conviendrait à tous les problèmes (Choi 1991; Offodile 1991). La plupart des algorithmes de formation des groupes nécessitent le réglage de paramètres numériques qu'il est parfois difficile d'interpréter. Il est rare que les membres du groupe de travail possèdent les compétences spécifiques nécessaires à l'exploitation de ces méthodes. Le cas échéant, il faudra inclure un spécialiste de l'analyse typologique dans l'équipe qui mène le projet d'implantation de la TG.

1 Les objets à classifier (wi) sont décrits par un ensemble de variables ou attributs (vi). Les différentes valeurs que peuvent prendre ces variables sont appelées modalités. Cette description constitue le tableau de données : Td.

Td	V1	v2	v3	v4
w1	1	1	1	2
w2	2	2	1	2
w3	2	2	1	1
w4	1	1	1	1
w5	2	2	1	2
w6	2	2	2	1

Tableau de données : Td

2 La mesure de similarité ou par abus de langage la similarité, exprime (en %) le degré de ressemblance entre deux éléments décrits dans Td. Appliquée sur toutes les paires d'éléments de Td, la similarité sert à construire le tableau de similarités : Ts. La similarité est symétrique par définition. Aussi, est-il d'usage de ne représenter que la diagonale supérieure de Ts.

Ts	w1	w2	w3	w4	w5	w6
w1	100	50	25	75	50	0
w2		100	75	25	100	50
w3			100	50	75	75
w4				100	25	25
w5					100	50
w6						100

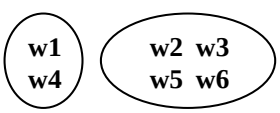
Tableau de similarités : Ts

3 L'algorithme de formation des classes, regroupe les éléments ayant une forte similarité et sépare les autres pour obtenir une partition. Les techniques sont nombreuses. Pour cet exemple, les objets sont regroupés si leur similarité est supérieure à un seuil fixé.

si is(wi, wj) > 60 alors
wi et wj appartiennent à
la même famille

Algorithme de formation des classes

4 L'outil de d'analyse typologique fournit une partition des données initiales. Les utilisateurs interprètent les classes obtenues.



Partition de l'ensemble initial et interprétation des résultats

Fig. I.2 Les 4 étapes d'une Analyse Typologique

I.3 La validation du résultat et le cycle de classification

L'évaluation par les experts, des familles issues de l'étape de classification (étape (4) de la figure I.1) se ramène à deux cas de figure :

1 Dans le premier cas, les familles obtenues sont directement validées ou du moins jugées acceptables à quelques modifications près par les membres du groupe d'expert. Dans ce cas, l'outil d'analyse typologique aura permis d'accélérer le processus de définition des familles.
2 Dans le deuxième cas, les familles ne sont pas jugées interprétables.

L'expérience des chercheurs du L.R.P.S nous permet de constater que les familles obtenues directement à l'aide des méthodes d'Analyse Typologique, ne sont généralement pas validées par les experts de l'entreprise (Frey 1990; Feltz 1993; Laget 1994; De Guio 1998; De Guio 1999). Le processus de classification prend alors une forme cyclique. Les experts du domaine et l'analyste agissent sur les données et les paramètres du logiciel d'analyse typologique par essai erreur, jusqu'à l'obtention d'un résultat satisfaisant. On trouvera en I.3.1 les actions correctives les plus courantes. La figure I.3 présente les différentes étapes du cycle de classification liées à l'implantation de la TG. Chaque étape du processus est accompagnée de la liste des compétences nécessaires à son exécution.

On retrouve les étapes (1) à (4) liées à l'implantation de la TG (cf. figure I.1.) Les étapes (5), (6) et (7) décrivent le processus de correction qui au terme de plusieurs itérations, engendre une classification validée par les experts.

- L'étape (5) consiste à recueillir les « critiques » que les experts de l'entreprise émettent lorsqu'ils évaluent la pertinence d'une classification. Ces « critiques » sont des connaissances supplémentaires qui permettent de simplifier le problème de classification.

- L'étape (6) a pour objectif de valider ces « critiques » en confrontant les points de vue des experts de l'entreprise et celui du spécialiste en Analyse Typologique.

- L'étape (7) est une re-formulation des « critiques » validées, en termes d'actions sur les différents paramètres et entrées du logiciel d'Analyse Typologique.

Selon le domaine d'application : production, conception ou fabrication, le processus de classification varie légèrement. Nous décrivons ci-après les deux cas les plus courants.

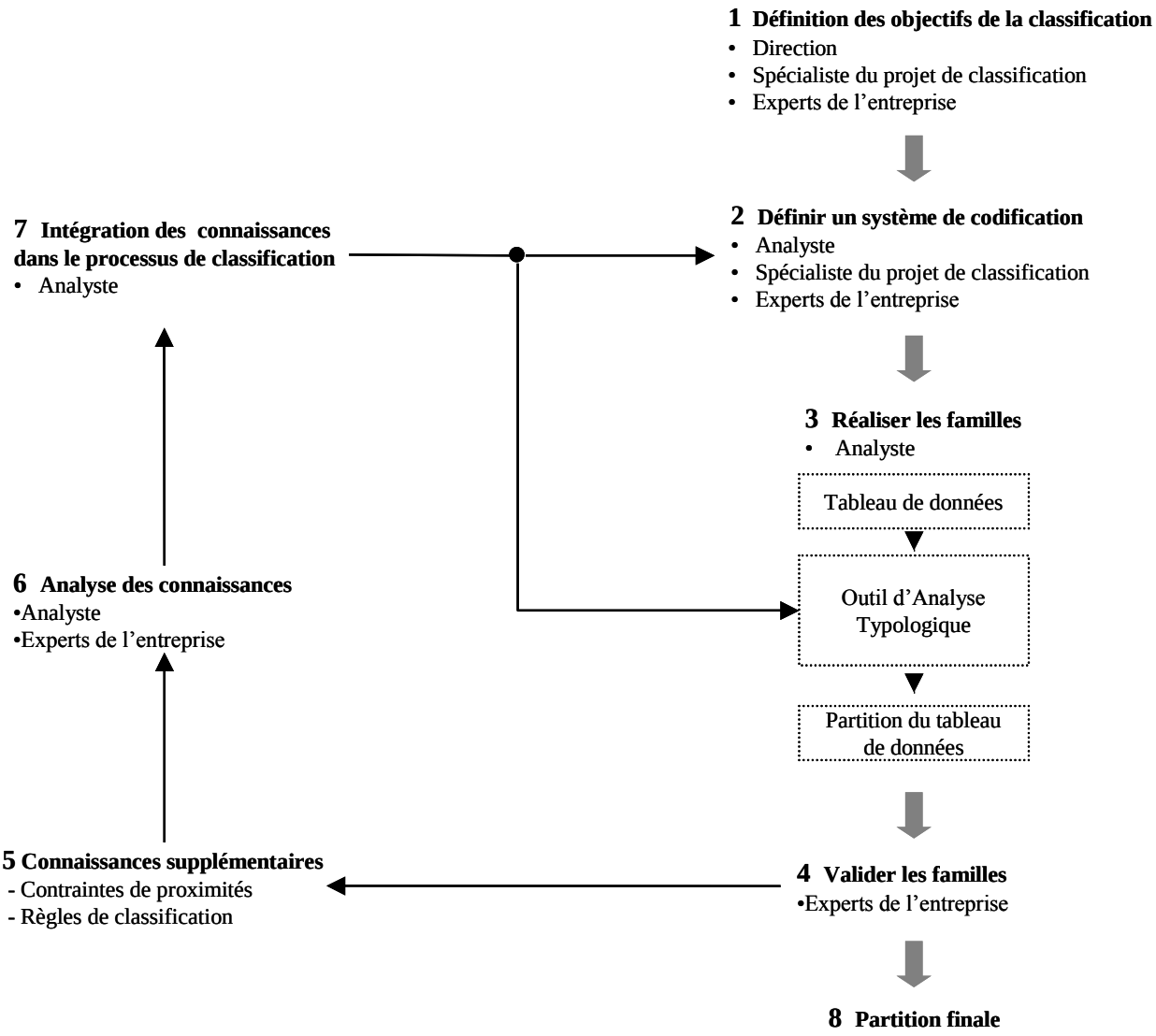


Fig. I.3 Processus de classification des données techniques pour l'implantation de la TG à l'aide d'un outil d'Analyse Typologique

I.3.1 Le cycle de classification en conception et en fabrication

En conception et en fabrication, les attributs descriptifs initiaux sont définis par les experts ou issus d'un code TG existant. Il est très rare que l'application directe d'une méthode de d'analyse typologique sur les données ainsi construites aboutisse directement à une partition pertinente pour les objectifs de la TG. L'expérience est pourtant constructive. A l'issue de l'étape de classification (3), les experts de l'entreprise évaluent la classification obtenue (4) et formulent des connaissances supplémentaires dans le but de corriger la partition élaborée par le logiciel d'analyse typologique (5). Ces connaissances se présentent généralement sous la forme de règles précisant la façon dont les objets devraient être rassemblés ou comparés pour former une « bonne » partition. Nous les appellerons des règles de classification. L'exemple ci-après est représentatif des règles de classification pour une application en conception.

Exemple I. 1 Base de règles de classification

La base de règles composée des règles de classification R1 à R8 ci-dessous, est définie par rapport aux variables descriptives L/D, FE, FI US et FO. Ces règles sont issues d'un corpus présenté en Annexe IV.1. Les variables descriptives sont détaillées au chapitre V.

R1 Deux pièces ayant des modalités L/D différentes ne peuvent pas appartenir à la même famille.

R2 Si L/D=0 alors FE=2 ou FE =5 est plus important que FE=2 ou FE=4 pour comparer deux pièces

R3 Si L/D=0 alors FE=2 ou 5 est plus important que FI=1 ou 4 pour déterminer la similarité de deux pièces.

R4 Si L/D=0 alors les modalités FI=1 et FI=4 peuvent être groupées en une seule FI=1ou4 pour comparer deux pièces

R5 Si L/D=0 alors les pièces telles que FE =1 ou 4, FE=0, FI=2 et FO=5 ne peuvent pas cohabiter au sein d'une même famille

R6 Les pièces telles que L/D=0 et FE=0 et FI=1 et US=0 et FO=0 définissent une famille.

R7 (FE=7) et (L/D=0) est une fonction discriminante pour les familles.

R8 Les pièces telles que L/D=0 et FO=6 et FE=7 doivent appartenir à la même famille.

L'exploitation de ces règles (étapes (6) et (7)) est délicate. Elles sont exprimées en langage naturel. Leur interprétation est difficile car les notations ne sont pas rigoureuses et prêtent à confusion. On remarquera que les règles de l'exemple I.1 qui sont déjà le fruit d'une première formalisation, ne sont pas forcément triviales à utiliser pour classifier un ensemble d'objets. Il est souvent nécessaire d'examiner de nombreux exemples avant d'être certain d'avoir saisi le sens exact d'une règle. Elles ne sont pas sûres, certaines règles ne s'appliquent sur aucun objet, d'autres sont contradictoires. Les experts sont généralement capables de corriger ces incohérences une fois qu'elles sont mises en évidence. L'idéal est de tester les règles sur l'ensemble des objets disponibles, avant de les prendre en compte. Chaque règle correspond plus à un essai qu'à une connaissance établie et inébranlable. La liste des critiques émises par les membres du groupe pilote ne permet généralement pas de définir une partition correcte à partir des attributs initiaux. Il faut plusieurs itérations du cycle (2), (3), (4), (5), (6) et (7) avant d'obtenir une partition entièrement pertinente par rapport aux objectifs de la TG (Mutel 1993). Cette boucle comprend un cycle de structuration des connaissances des experts en parallèle de la structuration des données. A chaque itération, les experts et l'analyste affinent les attributs descriptifs et les connaissances supplémentaires. La connaissance nécessaire à l'élaboration de la partition prend forme progressivement.

Comme nous l'avons dit au paragraphe 1.1, toute la difficulté de la démarche réside dans la recherche de critères de classification pertinents par rapport aux objectifs de la TG. Les experts connaissent implicitement un critère d'évaluation de la pertinence d'une partition, mais ils ne sont pas capables de le formuler en terme de critères de classification. L'outil d'analyse typologique utilise un critère de classification général qui est la similitude des objets par rapports aux variables descriptives. En d'autres termes, il rassemble les objets qui ont des descriptions semblables. En l'appliquant sur les données, on met en évidence les lacunes des variables descriptives exploitées en terme de ressemblance par rapport au critère d'évaluation implicite des experts. Ce n'est qu'en examinant des partitions successives que les experts vont affiner leurs critères de classification et apprendre à le formuler en terme de similitude des objets par rapport aux variables descriptives.

Le dialogue entre les experts de l'entreprise et l'analyste (étapes (6)) n'est pas toujours facile. Chacun possède une vue précise de son métier, mais ne partage pas les compétences de son homologue. D'un côté, les experts de l'entreprise essayent de construire une partition dont ils n'ont qu'une idée imprécise, et cela par l'intermédiaire d'un outil d'analyse typologique qu'ils ne maîtrisent pas. De l'autre côté, l'expert en analyse typologique essaye d'interpréter les critiques des experts en termes de regroupement et de réglages de l'outil de classification alors qu'il est étranger au domaine d'application (étape (7)). Pour tenir compte des connaissances supplémentaires, les actions les plus courantes sont :

- modifier la description des objets à classer : ajouter, supprimer ou modifier des attributs ;
- modifier la pondération de certains attributs dans le calcul de la similarité ;
- découper l'ensemble initial en sous-groupes plus homogènes ;
- modifier les paramètres de l'algorithme de formation des groupes ;
- choisir une autre mesure de ressemblance.

A notre connaissance, il n'existe pas dans la littérature de méthode appropriée pour vérifier la cohérence des règles de classification et les intégrer dans un processus d'Analyse Typologique.

Il s'avère donc que les outils d'analyse typologique pour l'implantation de la TG en conception et en fabrication permettent de classifier rapidement des quantités de données importantes et allègent le travail de classification du groupe d'experts au cours de l'étape (3) du cycle de classification. Cependant, du point de vue de l'ensemble du processus de classification, ces outils nécessitent une phase de « réglage » avant d'être exploitables. Une fois réglés, ils s'avèrent efficaces, rapides et sont parfaitement adaptés pour des traitements similaires et répétitifs. Ce point est important en pratique car le délai de mise au point de l'interface entre la méthode de classification et le problème à analyser peut être aussi long, voire plus long que le délai d'analyse avec d'autres méthodes.

I.3.2 Le cycle de classification en production

Le problème de la formation des îlots de production est sans doute l'un des plus étudié en TG. Il a suscité de nombreuses contributions utilisant une méthode d'analyse automatique des données pour définir les îlots de production (Offodile 1992; Singh 1993; Kandiller 1994; De Guio 1998 ; De Guio 1999). Ces recherches ont permis de définir précisément les attributs descriptifs et les différents critères à optimiser en fonction de l'objectif de la TG : maximiser la charge dans les îlots, minimiser les flux de manutention entre les îlots, etc. (étape (1) et (2)). Les méthodes de classification existantes permettent de traiter efficacement l'ensemble de ces problèmes et d'élaborer un consensus entre différents critères (dilemme flux-charge) De Guio (Barth 1993) (étape (3))

A la différence du cycle de classification en conception et en fabrication, il est souvent nécessaire pour la formation des îlots de prendre en compte des connaissances a priori (étape (5)) qui s'expriment sous forme de contraintes sur la partition recherchée. Certaines pièces ou machines doivent faire partie d'un même îlot ou inversement doivent appartenir à des îlots différents (Burbidge 1975; De Guio 1990; Plaquin 1998). Les origines de ces contraintes sont diverses et spécifiques à chaque entreprise. Ce peut être par exemple, pour des raisons de sécurité, pour l'utilisation de ressources communes (arrivée d'eau, alimentation), pour des raisons d'isolation phonique, afin de minimiser un trajet spécifique ou bien parce que tel groupe de machines nécessite la compétence du même opérateur. Se sont les contraintes de proximité et d'exclusion que nous dénommerons par le terme général de contraintes de proximités. Ces contraintes sont l'expression des différents points de vue métier qui contribuent à l'élaboration d'une solution. Le contremaître privilégiera par exemple les flux de manutention, l'ergonome l'isolation phonique. Elles peuvent s'avérer contradictoires. Par exemple, si l'isolation phonique nécessite de séparer des machines et ainsi de rallonger les distances de manutention. Il est courant de devoir tenir compte d'une dizaine de contraintes de ce type (Barth 1993; De Guio 1999). La démarche de résolution de ce problème consiste, dans un premier temps, à analyser la cohérence des connaissances supplémentaires (étape (6)) et à susciter la discussion pour favoriser l'élaboration d'un consensus; puis, dans un deuxième temps, on intégrera les connaissances supplémentaires cohérentes dans le processus de classification (étape (7)). Pour évaluer la pertinence des différentes solutions, il est souvent nécessaire de tester différentes contraintes (étapes (5),(6), (7), (3) et (4)). On retrouve pour la formation d'îlots les problèmes de dialogue entre les experts du domaine et l'analyste liés à la collaboration de deux domaines d'expertise différents. A notre connaissance, il existe deux approches qui prennent en compte ce type de connaissances supplémentaires : celle de développée par De Guio (De Guio 1990) (cf. II.2.1.2) et celle de Planquin (Plaquin 1998) (cf. II.2.4.2). Mais elles se limitent aux seules contraintes de proximités (et pas d'exclusion) et n'offrent aucun support particulier pour la tâche d'analyse des connaissances supplémentaires (étape (5)). Les méthodes existantes d'exploitation des résultats d'un processus d'Analyse Typologique, abordent essentiellement le problème de l'élaboration d'un consensus entre plusieurs critères à optimiser (De Guio, Barth 1993), mais peuvent éventuellement permettre d'analyser et d'intégrer des contraintes de proximité.

I.4 Limites de l'analyse typologique typologique en TG

Du point de vue de la classification, l'implantation de la TG pose deux catégories de problèmes. En conception et en fabrication, il s'agit de déterminer une classification représentative de l'activité d'un service qui capitalise et rationalise un savoir-faire métier. Pour la formation d'îlots, le problème est simplifié et ramené à la détermination d'une classification qui optimise un critère objectif en tenant compte de connaissances a priori sur la solution. La quantité de données mises en jeu rend laborieuse la définition manuelle des familles. Les outils d'Analyse Typologiques proposés dans la littérature imposent de fortes contraintes sur la représentation des données. Lorsque les données sont mises sous un format utilisable, les outils existants résolvent assez bien l'étape de classification. En revanche, les conditions industrielles d'exploitation des résultats sont peu étudiées (De Guio 1998). Pour certains problèmes de classification industrielle, le travail des chercheurs a permis de définir une partie des données pertinentes : code TG, critères d'opération pour la formation d'îlots, etc. Néanmoins, la variété des organisations ne permet pas de développer une méthode générale. Par exemple, la formation d'îlots nécessite de prendre un compte des contraintes propres à l'entreprise considérée, ne serait-ce que la forme des ateliers. De la conception à la production, chaque entreprise développe un savoir-faire qui la différencie de ses concurrents. Ce savoir-faire n'est pas utilisable tel quel : il est nécessaire de l'analyser et de l'intégrer dans le processus de classification, afin de permettre aux experts du domaine de le tester et en fin de compte, d'identifier les connaissances pertinentes par rapport aux objectifs de la TG. Les méthodes d'Analyse Typologiques proposées dans la littérature prennent rarement en charge l'analyse des connaissances supplémentaires et demande un effort de formalisation important pour les intégrer dans la définition des familles. Il n'existe pas, à notre connaissance, d'outil de classification spécialement conçu pour gérer le cycle de structuration des connaissances qui accompagne la structuration des données. Les outils d'analyse typologique ne sont pas utilisables sans une bonne maîtrise des techniques et de la théorie utilisée. C'est généralement le responsable du projet de classification ou bien un intervenant spécialisé qui possède la compétence nécessaire. Il analyse avec sa méthode personnelle et son expérience les critiques ou contraintes émises par les experts de l'entreprise et règle les paramètres du système de classification utilisé en fonction de son expérience. La résolution au coup par coup de ces problèmes sur le terrain et l'absence de guide méthodologique ralentit considérablement le processus de classification. Le problème se pose particulièrement en Conception et Fabrication, car le délai de mise au point de l'interface entre la méthode de classification et le problème à analyser peut être aussi long, voire plus long que le délai d'analyse avec d'autres méthodes que les méthodes d'Analyse Typologique. En formation d'îlots, les attributs descriptifs pertinents ont pu être identifiés une fois pour toutes. Le résultat dépend dans une moindre mesure du savoir-faire des experts. Cependant il est clair qu'un outil spécifique capable d'analyser et d'intégrer les contraintes de proximités dans le processus de classification permettrait de gagner en efficacité.

I.5 Affinement de connaissances par classification interactive

Il ressort de l'analyse précédente que les méthodes d'analyse typologique ne permettent pas de réduire significativement les délais d'implantation de la TG. Réduire ces délais revient à résoudre les deux problèmes de classification ci dessous.

Problème 1 : en conception et en fabrication	**Problème 2 : en production**
Etant donné :	**Etant donné :**
a Un ensemble important d'objets, sujets ou objets de l'activité d'un ou plusieurs experts.	**a** Un ensemble important d'objets, sujets ou objets de l'activité d'un ou plusieurs experts.
Que les experts en question :	Que les experts en question :
b ne peuvent pas formuler les classes a priori ;	**b** ne peuvent pas formuler les classes a priori ;
c peuvent valider ou critiquer une classification en fonction de connaissances du domaine implicites ;	**c** peuvent valider ou critiquer une classification en fonction de connaissances du domaine explicites ;
d lorsqu'ils critique une partition, ils expriment les connaissances du domaine sous la forme de règles de classification .	**d** émettent a priori, ou bien critiquent une partition en formulant les connaissances du domaine sous forme de contraintes de proximité
e La nécessité de tester la cohérence des règles de classification et de trouver un consensus en cas de contradictions	**e** La nécessité de tester la cohérence des contraintes de proximité et de trouver un consensus en cas de contradictions
f Une catégorie de critères de classification à partir de laquelle il faut définir des critères de classification.	**f** Un ensemble d'attributs descriptifs dont la pertinence est certaine.
	g Un critère explicite pour évaluer la qualité d'une classification.
Trouver :	**Trouver :**
h Une partition de l'ensemble des objets,	**h** Une partition de l'ensemble des objets,
i qui réduise la diversité en regroupant les objets similaires du point de vue des critères choisis ;	
j qui soit représentative de l'activité des experts ;	**j** qui optimise le critère d'évaluation de la pertinence d'une classification ;
k qui intègrent les connaissances du domaine (étape d) .	**k** qui respecte les contraintes de proximités.

Fig. I.4 Les problèmes de classification des données techniques pour l'implantation de la TG

L'objectif de cette thèse est de définir une méthode de classification automatique qui contribue à résoudre les deux problèmes précédents. Le problème n°2 est un sous-problème du problème n°1. Car les attributs pertinents et les critères d'évaluation de la pertinence d'une classification sont déjà définis. Les connaissances à prendre en compte sont un sous-ensemble du type de connaissances supplémentaires que nous allons traiter pour résoudre le problème n°1. Au long de ce mémoire, nous travaillerons donc uniquement sur le problème n°1 et validerons en conclusion la pertinence de notre approche pour le problème n°2.

Le problème n°1 ne fait pas explicitement référence à la TG. C'est un problème général de classification que nous baptiserons **Affinement de Connaissances par Classification Interactive** (ACCI). Par définition, ces problèmes de classification apparaissent dans les domaines complexes et peu formalisés (la classification joue un rôle de formalisation). C'est à dire, des domaines d'activité non triviaux pour lesquels il n'existe pas de modèle global. Les connaissances sur le domaine existent essentiellement sous forme d'expertise. La classification recherchée prend forme de manière itérative à partir de deux sources d'informations complémentaires : le savoir-faire des experts et les données.

Le problème qui consiste à trouver une partition représentative d'une activité à partir de savoir-faire est traité dans le cadre de l'Acquisition de Connaissances (cf. II.1.3.1) ; celui qui consiste à trouver une partition d'un ensemble de données est traité par une large gamme d'applications que nous appellerons les Outils de Classification Automatique (OCA). Ces outils proviennent essentiellement de quatre domaines différents : l'analyse Typologique, la Classification Conceptuelle, les Réseaux de Neurones et les Algorithmes Génétiques. Malgré les problèmes rencontrés dans le cadre de la TG avec les méthodes d'Analyse Typologique, il nous semble que les OCA peuvent contribuer à résoudre le problème n°1. Cette hypothèse sera le parti pris de notre démarche de recherche. L'analyse développée en I.3 et I.4 apporte des éléments de réponse quant à l'origine de l'inefficacité des méthodes d'Analyse Typologique. Cependant, la question reste entière de savoir si d'autres OCA seraient plus efficaces ou bien si les problèmes rencontrés sont inhérents aux principes même des Outils de Classification Automatique. Au quel cas, il est possible d'analyser ces problèmes et de proposer une démarche générale d'utilisation des OCA en Affinement de Connaissances par Classification Interactive.

Pour résoudre la problématique précédente, nous aborderons dans cette thèse, la démarche suivante :

1 définir le problème n°1 par rapports aux travaux d'Acquisition des Connaissances ;
2 faire un état de l'art des OCA et définir les caractéristiques de cette catégorie d'outils ;
3 analyser les limites des OCA par rapport au problème n°1 ;
4 proposer une approche originale de construction automatique de classification qui contribue à résoudre les problèmes 1 et 2 dans le cas où les experts s'expriment lors de l'étape (4) de validation des familles à l'aide de règles de classification.
5 montrer l'intérêt de cette approche pour réduire les délais d'implantation de la TG.

Nous avons mis cette approche en œuvre dans un système interactif de classification sous contraintes symboliques baptisé SYCLASCE. Ce système permet d'analyser et d'intégrer les règles de classification, dans un processus d'analyse typologique conventionnel basé sur un indice de similarité quelconque. Le protocole d'interaction guide l'analyste dans la démarche d'exploitation d'une base de règles de classification.

Notre contribution se situe à deux niveaux. D'une manière spécifique, SYCLASE contribue à résoudre les problèmes de classification pour l'implantation de la TG (problèmes 1 et 2 de la figure I.4). D'une manière générale notre contribution majeure est d'intégrer la connaissance supplémentaire de manière interactive et systématique dans le processus de classification. Le principe d'interactivité sur lequel repose le système définit un cadre pour résoudre les problèmes d'utilisation d'un Outil de Classification Automatique en Affinement de Connaissances par Classification Interactive. Notre champ d'expérimentation se limite essentiellement au périmètre de la productique. Cependant, en dehors du domaine de la classification industrielle, on trouvera des problèmes similaires dans les travaux de Bournaud (Bournaud 1996) en linguistique et didactique pour la construction de nouvelles classifications des caractères chinois, en systématique pour la construction de classifications d'éponges marines, en anthropologie pour l'étude de la classification des animaux dans la Grèce antique. Pour ces exemples, la construction d'une classification sur le domaine est le fruit d'une longue recherche. Elle présente un caractère itératif et l'utilisation directe d'un OCA (en l'occurrence des outils de Classification Conceptuelle) est difficile. Par rapport à ces problèmes, la particularité de notre travail réside principalement dans les connaissances supplémentaires prises en comptes : les règles de classification. Mais, comme nous le verrons par la suite, ce type de connaissances est lié, non au domaine d'application, mais à la structure de partition. Ce qui laisse penser que SYCLASCE présente un fort potentiel pour ces applications.

I.6 Guide de lecture

Dans le chapitre suivant nous précisons notre objectif et entamons un état de l'art. Tout d'abord, nous situons le problème n°1 par rapport aux travaux d'acquisition de connaissances, et précisons les objectifs de la catégorie de problèmes dont font partie les problèmes de classification pour l'implantation de la TG : l'Affinement de Connaissance par Classification Interactive. Dans un deuxième temps, nous présentons un état de l'art des outils susceptibles de contribuer à résoudre ce problème : les outils de classification automatique et définissons le principe de fonctionnement commun à ces outils.

Dans le troisième chapitre, nous analysons les limites que présentent ces outils lorsqu'ils sont employés pour résoudre un problème d'ACCI. Nous présentons ensuite le principe général des approches interactives susceptibles de contribuer à réduire ces limites.

Le quatrième chapitre est une présentation détaillée d'un système interactif de classification sous contraintes symboliques. Celui-ci adapte les principes de l'approche précédente afin d'apporter une solution aux problèmes de classification liés à l'implantation de la TG. La spécificité du système réside dans le choix d'un OCA particulier, en l'occurrence une méthode d'analyse typologique ainsi que dans le type de connaissances utilisées pour le dialogue utilisateurs - système : les règles de classification. Ces règles sont typiques de l'étape (4) de validation de la figure I.1. La systématisation du dialogue repose sur un modèle formel de représentation des règles de classification. Ce modèle est exploité par un ensemble de mécanismes qui permettent d'évaluer la base de règles du point de vue de l'utilité, la redondance et la cohérence. Nous définissons ensuite une méthode d'analyse systématique de l'information issue des règles de classification ainsi que le mécanisme d'intégration de cette information dans un outil d'Analyse Typologique.

Le cinquième chapitre débute par un bref aperçu d'un logiciel prototype du système interactif de classification sous contraintes symboliques développé au chapitre IV. Il expose ensuite un exemple d'application issu d'un problème d'implantation de la TG au bureau d'études d'une entreprise qui fabrique des pièces mécaniques.

Enfin, nous concluons sur ce travail dans le dernier chapitre et ouvrons différentes perspectives de recherches.

Chapitre II

Affinement de Connaissances par Classification Interactive : objectifs et méthodes

Introduction

Au cours du chapitre précédent, nous avons posé le problème de l'Affinement de Connaissances par Classification Interactive(ACCI). Ce chapitre, a pour objet de préciser les objectifs de l'Affinement de Connaissance par Classification Interactive (partie II.1) et ses méthodes (partie II.2).

L'objectif essentiel de l'ACCI est de définir une partition représentative de l'activité d'un groupe d'experts. Nous commencerons donc par poser les notions de base de la classification (II.1.1), avant d'analyser en quoi une classification peut représenter une activité (II.1.2). Cette façon de représenter une activité ainsi que des méthodes pour construire ce type de partitions sont étudiées dans le cadre de l'Acquisition des Connaissances. La partie (II.1.3) a pour objet de positionner l'ACCI par rapport à ce domaine de recherche.

La deuxième partie de ce chapitre (II.2) présente un état de l'art des outils susceptibles de contribuer à résoudre le problème de l'ACCI : les Outils de Classification Automatique (OCA). Nous passerons en revue : les méthodes d'Analyse Typologique (II.2.1), les méthodes de Classification Conceptuelles (II.2.2), les méthodes Neuronales (II.2.3) et les Algorithmes Génétiques (II.2.4). Notre objectif est d'une part d'évaluer la capacité de ces outils à résoudre l'étape (3) du processus de classification décrit par la figure I.3. Nous poserons dans ce chapitre les bases d'un principe commun aux OCA qui servira d'argument pour l'analyse du chapitre III.

II.1 Les objectifs de l'Affinement de Connaissances par Classification Interactive

Nous travaillons sur un problème particulier : trouver une classification représentative d'une activité menée par un ou plusieurs d'experts. Cette activité consiste essentiellement à résoudre le problème suivant : connaissant une situation, il faut lui associer une solution. Par exemple, dans le domaine de la productique, on rencontrera les associations ci-dessous :

Situation		**Solution**
Cahier des charges d'une pièce	$\Rightarrow$	Plan de la pièce
Plan d'une pièce	$\Rightarrow$	Gamme de fabrication

La modélisation de ce type d'activité est étudiée dans le domaine de l'Acquisition des Connaissances. Les modèles proposés reposent sur des notions de classes et de concepts que nous développons ci-dessous.

II.1.1 La classification : notions de base

Le terme classification est polysémique. Il désigne aussi bien l'activité de construction d'une classe ou celle de ranger un objet dans un groupe existant, que le résultat du processus de classification : un groupe de classes, généralement organisé. Il semble donc nécessaire de clarifier les concepts de bases liés à la notion de classification, et de poser quelques définitions.

Le concept de classification repose sur trois hypothèses (Encyclopédie de philosophie 1990). Premièrement, il existe des entités individuelles : les objets. Ce sont des entités déterminées et bien distinctes de notre entendement ou de notre perception. Deuxièmement, il existe des entités collectives qui font référence à une pluralité d'objets. La plupart des approches divergent sur la conception des entités collectives. Suivant les domaines, ce sont des groupes, des classes, des familles, des espèces, des concepts, des ensembles ou des catégories. Généralement, nous utiliserons le terme de classe, pour faire référence à une entité collective. Troisièmement, les entités individuelles ou collectives possèdent des **propriétés**. Les propriétés permettent de décrire et d'identifier les entités. Le concept de propriété est extrêmement vague. Ce peut être, la couleur, la taille, le nom du propriétaire, le fait d'être composé de plusieurs parties distinctes agencées d'une façon précise. Les objets, les classes et les propriétés donnent lieux à quatre activités : faire les classes, organiser les classes, placer un nouvel objet dans une classe et décrire une classe. Le terme classer recouvre les deux sens de regrouper les objets en classes et de ranger un objet dans une classe existante. Nous lui préférerons les termes **classifier** ou **catégoriser**, pour désigner l'activité de construction et d'organisation des classes en regroupant les objets. Le résultat de la catégorisation est une

classification, c'est à dire un groupe de classes organisées. L'organisation des classes définit une **structure** qui peut être une partition, un recouvrement, une hiérarchie, etc. Une fois les classes formées, se pose le problème de **classer** un objet ; c'est à dire, de placer un objet dans une classe préexistante. Une **méthode de classement** est une méthode pour classer des objets. Généralement, l'opération de classer, repose sur une comparaison de la description d'un objet et d'une classe. Pour pouvoir classer les objets, il est nécessaire de décrire la classe. **Décrire** une classe, c'est en donner une **description** qui permet de savoir si un objet est en relation ou pas avec la classe. La description d'une classe est construite à partir des propriétés définies sur le domaine d'investigation. Ces notions sont récapitulées ci-dessous.

1. **Les objets** sont des entités individuelles
2. **Les classes** sont des entités collectives qui font référence à une pluralité d'objets.
3. **Les propriétés** permettent de décrire et d'identifier les classes et les objets.
4. **Classifier** ou **catégoriser** désigne l'activité de construire des classes en regroupant des objets.
5. **Une classification** est le résultat d'une activité de catégorisation. C'est une **structure**, qui décrit les relations entre les classes.
6. **Classer un objet**, c'est placer un objet dans une classe particulière. **Une méthode de classement** permet, de classer un nouvel objet dans des classes préexistantes.
7. **Décrire une classe**, c'est en donner une **description** qui permet de savoir si la classe fait référence à cet objet ou pas et donc de classer l'objet. La description d'une classe est construite à partir des propriétés

Déf. II. 1 Quelques définitions relatives à la classification

II.1.2 Classification et représentation des connaissances : les concepts

La façon la plus simple de définir une classe, la définition en extension, consiste à donner la liste des objets qui la composent. La définition en extension s'avère extrêmement limitée. Dès que le nombre d'objets est important, il devient difficile pour la mémoire humaine de manipuler des classes sous la forme de listes exhaustives. C'est même impossible avec des listes infinies. Pourtant, la notion de classe joue un rôle primordial dans l'ensemble des activités humaines, ne serait-ce que par l'intermédiaire des mots et des idées. Nous possédons cette faculté d'user de généralité, c'est à dire de manipuler des entités collectives autrement qu'en énumérant la liste de leurs membres, ce sont les concepts des philosophes, les catégories des psychologues, les ensembles des mathématiciens, les objets des informaticiens, les espèces du naturaliste. Sans description ayant un caractère de généralité, les classes n'ont

pas de signification et ne présentent guère d'intérêt pour les activités humaines. C'est là un problème central en classification : comment décrire, définir les classes autrement que par une liste ?

Les philosophes, les logiciens et les mathématiciens, sont sans doute les premiers à avoir réfléchi sur la nature des regroupements et des collections que l'homme manipule par le biais des mots et des idées. Les classifications qu'ils utilisent sont souvent le fruit de réflexions qui tentent de dégager des théories explicatives du monde et d'approfondir la compréhension des objets de l'environnement (Foucault 1966). Platon utilisait les classifications dans un but définitoire. Aristote classait pour définir des îlots structurés de connaissances sans projet de classification systématique (Pellegrin 1982). C'est ainsi que les philosophes se sont intéressés aussi bien à la classification des être vivants qu'à celle des régimes politiques ou des formes de plaisir. Leurs approches s'articulent autour de la notion de concept. La théorisation qu'ils en ont faite a longtemps été étroitement liée aux exigences de la logique, c'est à dire la recherche des conditions dans lesquelles les concepts peuvent être utilisés pour porter des jugements dotés de vérité ou pour tirer des inférences conservant la vérité de leurs prémisses.

Pour ce qui est de sa nature, le **concept** est traditionnellement rattaché à la sphère de la **représentation** et suppose la distinction entre un sujet qui connaît et un objet à connaître (Encyclopédie de philosophie 1990). La **généralité** est en quelque sorte l'élément caractéristique de l'existence du concept, ce par quoi la représentation dépasse la donnée. L'expérience sensible présente les objets comme infiniment différents et donc singuliers. L'être humain accède à la généralité par le processus **d'abstraction**. Ce processus consiste à « isoler » certains « aspects » des objets et à en négliger les autres. L'aspect dégagé est commun à tous les objets en lesquels il se réalise ou pourrait être réalisé. Considéré comme un objet de pensée le résultat d'une opération d'abstraction est un concept.

Cette ambivalence du concept qui lie le général au particulier se retrouve dans ses deux composantes qui sont l'**intention** et l'**extension**. Ces notions, bien que sous-jacentes depuis l'Antiquité, ont été formulées par les logiciens de Port Royal, Antoine Arnaud et Pierre Nicole, dans leur traité « La logique ou l'Art de Penser » (connu aussi sous le nom de Logique de Port-Royal) parut en 1662. Le vocabulaire est légèrement différent du nôtre. Les concepts sont des **idées**, dont il faudra distinguer la compréhension (l'intention) et l'étendue (l'extension).« *J'appelle compréhension de l'idée, les attributs qu'elle enferme en soi, et qu'on ne lui peut ôter sans la détruire comme la compréhension de l'idée du triangle enferme extension, figure, trois lignes, trois angles, et l'égalité de ces trois angles à deux droits, etc.* »
« *J'appelle étendue de l'idée, les sujets à qui cette idée convient, ce qu'on appelle aussi inférieur d'un terme général, qui à leur égard est appelé supérieur, comme l'idée du triangle en général s'étend à toutes les diverses espèces de triangles.* »

Entre 1891 et 1892, Fredge écrit trois articles, « Fonction et concepts », « Sens et dénotation », « Concept et objets » dans lesquels il formalise la notion de concept, de façon à pouvoir l'intégrer dans un calcul (Blanché 1970). Un concept devient une fonction à un argument, dont la valeur est une valeur de vérité. Par exemple, le concept d'homme est la

fonction « ... est homme ». L'extension du concept est la **classe** des objets pour lesquels la fonction propositionnelle prend la valeur vrai, par exemple « Pierre, Paul, etc. », qui constituent la classe des hommes. L'intention du concept est une condition nécessaire et suffisante qui permet de déterminer si l'objet tombe ou pas sous le concept, par exemple « animal, bipède, sans plumes aux ongles plats et capable de penser». Fredge pose ainsi la définition moderne du concept telle qu'elle est utilisée en logique mathématique (logique des prédicats). Cette forme opérationnelle du concept s'éloigne des conceptions plus philosophiques comme celle de l'idée par exemple. L'interprétation que fait Fredge du concept, repose sur l'équivalence entre une propriété et une classe. C'est le principe d'abstraction : pour n'importe quelle propriété P il existe une classe constituée de tous les objets et seulement les objets qui vérifient P. Appliqué sans restriction, le principe d'abstraction engendre des paradoxes. Le plus célèbre, énoncé par Bertrand Russel consiste à prendre pour propriété P « ne pas être membre de soi même ». Il obtient ainsi la classe (appelons la C) de toutes les classes qui ne se contiennent pas elles mêmes. Dans ce cas précis, le principe d'abstraction engendre un énoncé contradictoire, à savoir : x est membre de C si et seulement si x n'est pas membre de C. Il y a eu de nombreuses tentatives pour formuler la notion de classe de façon à ce qu'elle n'engendre pas de paradoxes. La théorie dominante consiste à modifier le principe d'abstraction à l'aide du principe de séparation (Encyclopédie de philosophie 1990). L'équivalence entre une classe et une propriété ne sera valable que si les objets qui constituent la classe sont pris à l'intérieur d'un univers du discours donné. L'axiome de séparation est la pierre angulaire de la conception moderne des concepts. Il permet d'intégrer dans une théorie logiquement cohérente les différents éléments qui définissent un concept[1] (cf. Def. II.2)

1. **L'univers du discours U** : c'est le domaine d'investigation du sujet qui définit ou utilise un concept. Il peut s'agir des animaux, des figures géométriques, etc. Il suffit pour le définir d'utiliser une formule telle que U existe, et U comprend tous les objets et seulement les objets qui satisfont au prédicat P.
2. **Un mot** : pour référer au concept
3. **L'intention** : c'est une condition nécessaire et suffisante qui permet de décider si un objet de l'univers du discours exemplifie le concept ou pas.
4. **L'extension** : c'est la classe des objets de l'univers du discours qui vérifient l'intention du concept. Les membres de l'extension sont appelés des instances du concept.

Déf. II. 2 Spécification générale du rapport (objet, concept, mot) d'après (Sutcliffe 1995)

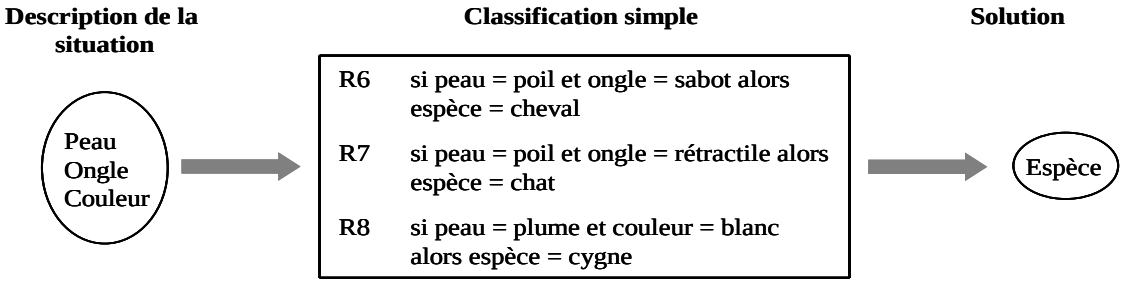
[1] Les ensembles sont une restriction de la notion de classe. Les ensembles se distinguent des concepts par le principe d'extensionalité : des ensembles ayant les même éléments sont identiques. Ce qui n'est pas forcément le

II.1.3 Classification et acquisition de connaissances : la classification simple

En intelligence artificielle, le terme Acquisition des Connaissances fait référence à une problématique précise : modéliser les connaissances d'un expert pour simuler son activité sur un ordinateur (Kodratoff 1991b). Ces connaissances correspondent à un savoir qui a été rendu opérationnel dans l'esprit de l'expert au fil des années d'expérience. Aussi, sait-il les utiliser, mais il se retrouve généralement incapable de les formaliser, surtout quand il agit dans des domaines qui mettent fortement en œuvre les facultés perceptives (Galloüin 1988) (Bisson 1993). Lorsqu'il verbalise ses connaissances, l'expert oublie souvent d'en préciser les conditions d'application (Bisdorff 1995). Son savoir est un savoir "privé", une somme d'enseignements tirés de l'expérience. Cet ensemble structuré d'expériences vécues dépend de la biographie de l'expert et lui appartient en propre (Vogel 1988). Les méthodes d'Acquisition de Connaissance ont pour objectif d'expliciter et de modéliser ces connaissances particulières que sont les savoir-faire. Le résultat est appelé un Système à base de Connaissance (David 1993). Ces systèmes reposent sur la distinction entre connaissances de raisonnement et connaissances du domaine. Les connaissances du domaine portent sur le domaine d'application, et des connaissances de raisonnement, portent sur la réalisation de la tâche elle-même (Alexis 1995). Les connaissances de raisonnement forment une méthode de résolution de problème qui décrit la structure du raisonnement utilisé, et donc la manière de résoudre un type de problèmes. Les connaissances du domaine jouent un rôle dans la méthode de résolution de problèmes. L'Acquisition des Connaissances[2] est vue comme la construction coopérative avec un ou plusieurs experts, d'une méthode de résolution de problème qui modélise l'activité de l'expert. Dans ce contexte, les classifications permettent de construire des systèmes experts effectuant des tâches d'identification. Partant de la description d'une situation, il s'agit de lui associer une solution. Ce peut être une maladie, une espèce, une gamme, etc. L'objectif n'est donc pas forcément de modéliser les mécanismes cognitifs de l'expert. Les grandes classes de méthodes de résolution de problèmes définissent une modélisation plus ou moins fine de l'activité des experts. Le modèle de la classification simple est la représentation la plus simple que l'on puisse faire de l'activité d'identification. Les exemples II.1 et II.2 ci-dessous présentent et comparent, pour la même activité d'identification, deux modèles de résolution de problèmes : la classification simple et une version élémentaire de la classification heuristique.

Exemple II. 1 La classification simple

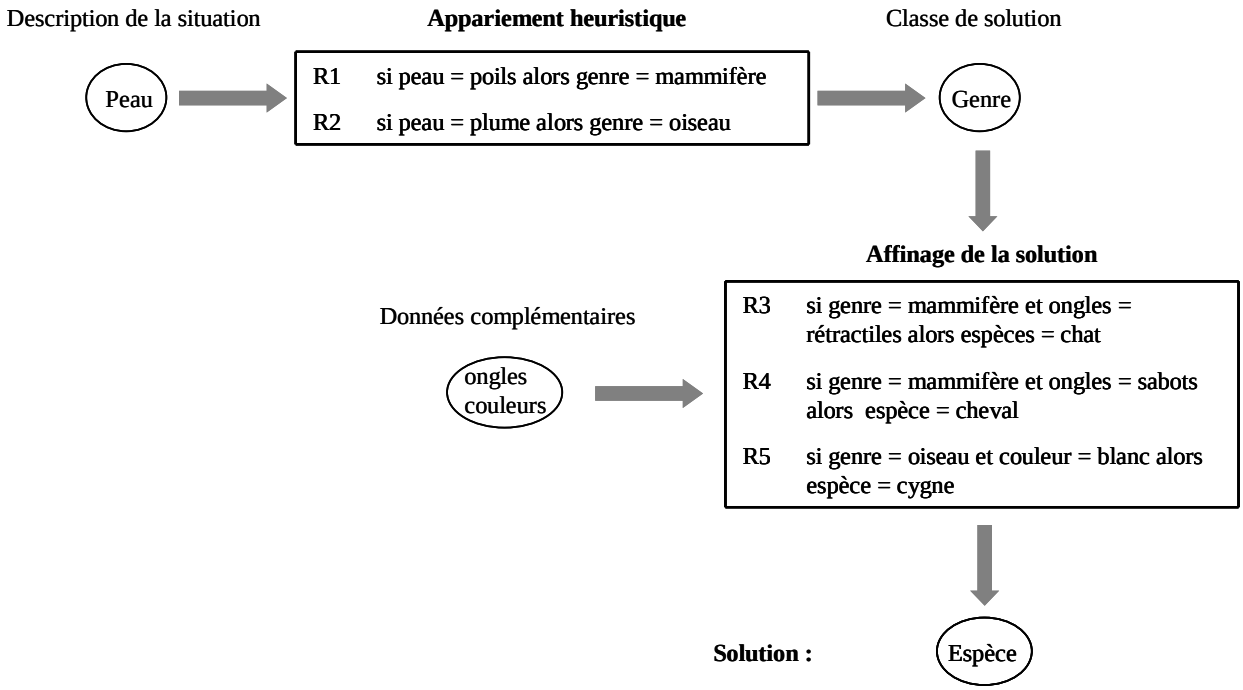
Nous présentons ci-dessous un exemple de système à base de connaissances qui détermine l'espèce de quelques animaux à partir de leur description par les attributs peau, ongle et couleur. Ce système à base de connaissances utilise une méthode de résolution de problème appelée classification simple. Le lien entre la solution et la description d'un cas est direct. Il n'existe aucun intermédiaire.



[2] En France l'expression ``Ingénierie des Connaissances" a été préférée à l'expression ``Acquisition des Connaissances" afin de marquer la rupture qu'il y a avec l'approche initiale et, à l'inverse, mettre l'accent sur l'activité de construction d'un modèle de connaissances (AFIA 1998).

Exemple II. 2 La classification heuristique

Nous présentons ci-dessous un système à base de connaissances qui réalise la même tâche d'identification que le système de l'exemple précédent. A la différence que celui-ci utilise une forme élémentaire d'une méthode de résolution de problèmes courante pour les problèmes d'identification : la classification heuristique (Clancey 1985). Le raisonnement effectué par le système comprend deux phases. Tout d'abord, une phase d'appariement heuristique qui permet d'associer une partie des données à une classe de solution : le genre. Ensuite, une phase de raffinement permet de préciser une solution particulière au sein de la classe de solution retenue. Cette méthode de résolution est beaucoup plus proche du raisonnement d'un expert en classification animale que le modèle de la classification simple.



En comparant les deux exemples précédents, on remarque que la classification simple ne rend absolument pas compte des mécanismes de raisonnement mis en jeu par l'expert. C'est une compilation brutale et pas vraiment explicite de ses connaissances. On retrouve à ce niveau les reproches faits aux méthodes d'Acquisition de Concepts en tant qu'outils d'Acquisition des Connaissances pour les systèmes à base de connaissances (Bisson 1993, Thomas 1996). Elles fournissent un ensemble de règles indépendantes qui court-circuitent le raisonnement de l'expert. En TG, par exemple, on retrouvera cette différence au service des méthodes, en comparant la méthode générative et la méthode des variantes. La méthode générative consiste à définir un système à base de connaissances en fabrication qui définit une gamme à partir d'un dessin technique. La méthode des variantes issues d'une implantation TG permet, connaissant le plan d'une pièce de lui associer une gamme en la rattachant à une famille de plans qui utilisent cette gamme.

II.1.4 Conclusion

Les classes n'ont d'intérêt et de sens que s'il est possible de les décrire. La description d'une classe est un type particulier de connaissances ayant un caractère de généralité. Les classes munies d'une description sous forme d'une condition nécessaire et suffisante sont des concepts. En terme de connaissance, l'objectif de la classification est de définir et d'organiser des concepts sur un domaine. La notion de concept est à la base de modèles plus complexes développés en Acquisition des Connaissances pour représenter l'activité d'un ou plusieurs experts sur un domaine.

Les travaux d'Acquisition de Connaissances fournissent un cadre solide pour définir le problème de l'Affinement de Connaissances par Classification Interactive. De ce point de vue, l'objectif de l'ACCI est de construire un modèle de classification simple d'une activité d'identification menée par un groupe d'experts. Bien que fortement apparenté à un problème d'Acquisition de Connaissances, l'ACCI s'en distingue par les points suivants :

En Acquisition des Connaissances, le problème principal est d'identifier la méthode de résolution de problème. Dans notre cas, la méthode de résolution de problèmes est définie d'emblée. Pour identifier les connaissances du domaine relatives à un modèle de classification simple, les outils développés en Acquisition de Connaissance (Galloüin 1988; Vogel 1988) s'apparentent aux méthodes traditionnellement utilisées en TG et ne sont donc pas efficaces.

L'expertise relative à l'activité d'identification considérée n'existe pas forcément a priori. Considérons par exemple le domaine de l'analyse des images satellites. Ces images sont composées d'un ensemble de pixels. Le travail d'analyse consiste à reconnaître des zones de pixels homogènes, puis à les identifier en terme d'éléments du paysage photographié : champs, forêts, rivières, habitations, etc. (Ketterlin 1995). Dans ce cas, un expert du domaine est capable de mener la tâche d'identification. Les travaux sur le domaine ont pour objectif de réaliser un système d'analyse automatique des données qui puisse retrouver les classes que ferait un expert. Dans notre cas, aucun des experts mis en jeu n'est capable a priori de mener complètement la tâche d'identification. Les classes sont à inventer. Le problème présente donc un aspect constructif. Les problèmes d'Acquisition de Connaissance présentent eux aussi un aspect constructif, mais dans ce cas, le terme fait référence au fait que l'on modélise le savoir-faire de l'expert par une méthode de résolution de problème qui n'est pas forcément un modèle de l'activité cognitive de l'expert. C'est généralement une façon parmi d'autre de conceptualiser les schémas de raisonnement de l'expert (Thomas 1996).

Le modèle de la classification simple repose sur trois éléments :

1. un ensemble de concepts d'extensions disjointes qui couvrent l'ensemble des situations ;
2. un ensemble de concepts disjointes qui couvrent l'ensemble des solutions ;
3. des associations entre les concepts cas et solutions (les règles de production).

L'objectif de l'ACCI consiste uniquement à construire une partition qui matérialise l'extension des concepts couvrant l'ensemble des solutions.

Les trois caractéristiques précédentes font du problème de l'ACCI un domaine d'investigation à part entière. A notre connaissance, il n'est pas étudié en tant que tel dans la littérature.

II.2 Les Outils de Classification Automatique

La quantité de données à traiter et l'insuffisance d'outils d'analyse élémentaires comme le tri, montrent l'utilité d'un outil capable de classifier les données au cours de l'étape (3) de catégorisation (fig. I .1).

Le problème qui consiste à trouver une partition d'un ensemble de données est traité par une large gamme d'applications que nous appellerons les Outils de Classification Automatique (OCA). Ces outils proviennent essentiellement de quatre domaines différents. Les méthodes d'Analyse Typologiques sont issues de l'Analyse de Données. Les méthodes de Classification Conceptuelles sont étudiées dans le cadre de l'Intelligence Artificielle en Apprentissage Symbolique Automatique. Les Algorithmes Génétiques et les Réseaux de Neurones forment chacun un domaine de recherche à part entière.

Nous avons observé dans l'introduction que l'emploi des méthodes d'Analyse Typologiques pour résoudre l'étape (3) de catégorisation liée à l'implantation de la TG (fig. I.1) n'est pas particulièrement probant. Le processus de classification prend la forme de la figure I.3. Cependant, la question reste entière de savoir si d'autres OCA seraient plus efficaces ou bien si les problèmes rencontrés sont inhérents aux principes même des Outils de Classification Automatique. Au quel cas, il est possible d'analyser ces problèmes et de proposer une démarche générale d'utilisation des OCA en Affinement de Connaissances par Classification Interactive.

L'objectif de ce chapitre est d'une part d'évaluer la capacité des OCA à résoudre l'étape (3) du processus de classification décri par la figure I.3 et d'autre part de donner une définition commune de ces outils en terme de principe de fonctionnement. Cette définition servira d'argument pour l'analyse du chapitre III.

Pour répondre à ces objectifs, nous présentons ci-dessous un état de l'art des OCA en détaillant les points suivants :

- Comment sont représentés les objets à classifier ?
- Quelle est la capacité de l'outil à prendre en compte des connaissances du domaine ?
- Comment sont élaborées les classifications à partir de l'information initiale ?
- Quels sont les limites de la méthode du point de vue de la quantité d'objets à classifier ?

II.2.1 Les méthodes d'Analyse Typologique

D'un point de vue formel, les méthodes d'Analyse Typologique permettent de décomposer une population d'individus ou d'objets décrits par un ensemble de caractéristiques, en un certain nombre de groupes homogènes (Chandon 1981). Plus précisément, la problématique de l'Analyse Typologique est donnée ci-dessous. La démarche standard a été présentée au chapitre I (cf. figure I.2). Nous focaliserons notre attention sur les méthodes de formation des groupes après avoir formalisé le concept de mesure de similarité.

Etant donné :

- *un ensemble d'objets homogènes ;*
- *un ensemble d'attributs monovalués et réguliers ;*
- *la donnée du type de structure recherché ;*
- *une mesure de similarité ;*
- *un critère d'évaluation F de la qualité d'une classification par rapport au tableau de similarité ;*

Trouver :

une classification qui optimise le critère F

Déf. II. 3 Problématique de l'Analyse Typologique

II.2.1.1 Choix de l'indice de similarité

Il existe de nombreux indices de similarité. Un indice de similarité sert à mesurer la ressemblance entre deux objets. Il prend sa valeur maximale lorsque l'on mesure la ressemblance d'un objet avec lui-même. Il est symétrique, car l'on suppose qu'un objet A ressemble autant à un objet B, que l'objet B ressemble à l'objet A. On trouvera en annexe II.1 des exemples d'indices de similarités couramment utilisés en Classification Automatique.

Un indice de similarité s est une application de ΩxΩ dans **R** telle que

$$\forall (w, w') \in \Omega x\Omega : s(w, w') = s(w', w) \text{ et } s(w, w) = s(w', w') \geq s(w, w')$$

La notation is(w, w') peut prêter à confusion. Le calcul de la mesure de similarité repose sur les variables descriptives. Pour être exact, il faudrait faire apparaître l'ensemble des variables descriptives EV = {Vi} dans l'écriture de s(w, w'). La forme exacte est :

$$s (w, w') = f (V1(w), ... , V2(w) ; V1(w'), ... , V2(w'))$$

II.2.1.2 La formation des groupes

L'étape de formation des groupes tente de résoudre le problème suivant : trouver une structure particulière Q qui respecte au mieux les conditions d'homogénéité et d'isolation.

Condition d'homogénéité : Les objets d'une même classe se ressemblent.
Condition d'isolation : Les objets de classes différentes ne se ressemblent pas.

La technique utilisée consiste à définir un critère qui mesure l'homogénéité et l'isolation d'une structure par rapport à un tableau de similarité. La structure recherchée est celle qui optimise le critère. La taille de l'espace de recherche interdit toute exploration exhaustive. La diversité des méthodes de Classification Automatique provient des multiples façons de formaliser les notions d'homogénéité et d'isolation, ainsi que des biais de classification utilisés pour explorer l'espace de recherche. On classifie généralement les méthodes en fonction de la structure recherchée : partition, recouvrement, hiérarchie, etc. Nous avons défini le problème de l'ACCI comme celui de la définition d'une partition. Notre investigation portera donc uniquement sur les méthodes qui génèrent une structure dont il est possible d'extraire simplement une partition : les méthodes hiérarchiques et les méthodes de partitionnement.

Etant donné :

- *Ω={wi} une population d'individus*
- *s : $\Omega x\Omega \rightarrow$ [0,1] une mesure de similarité*
- *Ts = (s(wi, wj)) : le tableau de similarité associé à s*
- *F : un critère qui interprète les règles d'homogénéité et d'isolation*

Trouver :

Sur un ensemble EQ de structure Q définie surΩ

une structure particulière Q, telle que* $\mathbf{Q^* = \max_{Q} F(Ts, Q)}$

Déf. II. 4 Problématique de l'étape de formation des groupes

II.2.1.2.1 La classification hiérarchique

Les méthodes hiérarchiques construisent une hiérarchie indicée sur l'ensemble des objets. Les différents niveaux de la hiérarchie sont obtenus en regroupant successivement les classes qui se ressemblent. L'indice d'agrégation permet d'ordonner les regroupements par ordre de création.

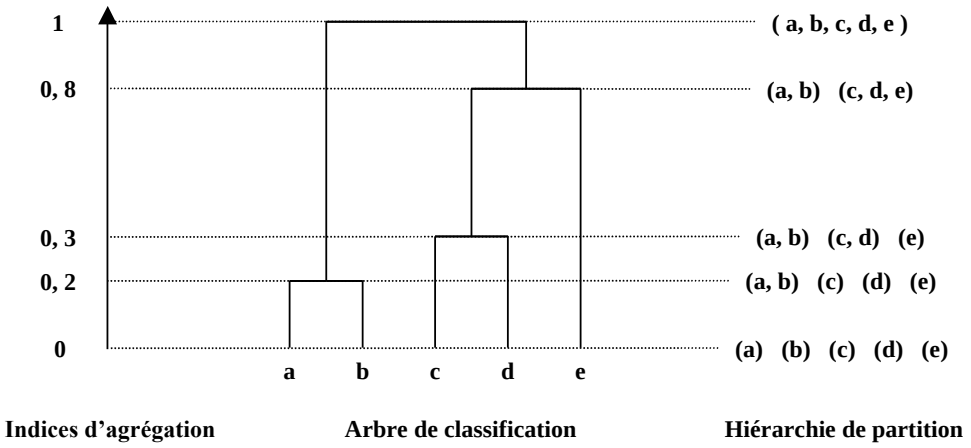


Fig. II.1 Hiérarchie indicée

Les méthodes hiérarchiques cherchent un ensemble de partitions emboîtées tel que chaque partition maximise la somme des similarités entre les objets d'une même classe (similarité intra classe) et minimise la somme des similarités entre des objets de classes différentes (similarité inter classe). Les méthodes basées sur un critère local utilisent l'algorithme de la Classification Ascendante Hiérarchique (CAH). Le critère est une distance entre groupes d'objets. Il est local, car il ne fait intervenir que les similarités entre objets des deux groupes à comparer. On passe d'un niveau de la hiérarchie au suivant en regroupant les deux classes les plus proches. La complexité de l'algorithme est de l'ordre de N^3 (N étant le nombre d'individus), réductible à N^2 par quelques astuces de programmation. C'est une méthode heuristique qui pose des problèmes d'optimum local. Selon la distance entre groupes d'objets utilisée, on privilégie l'homogénéité ou l'isolation. La plupart des programmes regroupent les deux premières classes rencontrées en cas d'ex æquo. Le résultat est donc sensible à l'ordre de lecture des données. On trouvera en annexe II.1 des exemples de critères locaux. Les méthodes utilisant un critère global reposent sur l'équivalence entre une hiérarchie indicée et un type de distance particulier, les ultramétriques. Le critère à optimiser est une mesure de « l'écart » entre le tableau de similarité initiale et une distance ultramétrique définie sur le même ensemble d'objets. Il n'existe pas d'algorithme exact pour des ensembles de plus de 20 objets (Chandon 1981) p129. Les méthodes utilisées reposent sur des heuristiques. L'influence du critère sur le résultat en terme d'isolation ou d'homogénéité des classes n'est pas étudiée à notre connaissance. On trouvera en annexe II.1 des exemples de critères globaux. Les méthodes hiérarchiques génèrent souvent un nombre important de partitions. S'il existe une partition particulièrement bien définie dans les données, elle est facilement repérable par une brusque variation de l'indice d'agrégation. Par exemple, la partition associée à l'indice d'agrégation 0,3 de la figure II.3 semble intéressante. Les classes sont bien séparées, il est nécessaire d'augmenter l'indice d'agrégation de façon importante, pour rassembler deux classes de cette partition. S'il n'existe pas de structure prédominante dans les données, les niveaux de la hiérarchie présentent les différents compromis possibles entre l'homogénéité et l'isolation. Par contre la contrainte d'emboîtement des partitions successives est un biais inutile lorsque l'on cherche uniquement une partition.

II.2.1.2.2 Les méthodes de partitionnement

Elles construisent directement une partition de l'ensemble des données. On les répartit généralement en fonction de deux principes dominants : la réallocation et l'agrégation de similarité. Les méthodes de réallocation utilisent une partition initiale, généralement choisie au hasard, et déplacent les objets d'un groupe à l'autre pour obtenir une meilleure partition au sens du critère utilisé. Elles reposent sur la représentation d'une classe par un ou plusieurs éléments centraux : les noyaux.

1. Sélectionner K noyaux (n1, ..., nk) parmi l'ensemble des objets O
2. Répéter
   2.1. Pour chaque noyau ni faire
   2.2. Définir la classe Ci, en lui affectant les objets plus proches de ni que des autres noyaux $Ci=\{w\in\Omega \mid \forall i\neq j\ d(w, ni) \leq d(w, nj)\}$
   2.3. Représenter la classe Ci par un nouveau noyau : ni = représenter(Ci)

   tant que les noyaux ne sont pas stables

Déf. II.5 Algorithme des méthodes de réallocation

La méthode des centroïdes (Forgy 1965) (Macqueen 1967) s'applique sur des variables métriques. Une classe est représentée par son centre de gravité. Lors de la phase de réallocation (2.1.1) les objets sont associés au centre de gravité le plus proche, les notions d'homogénéité et d'isolation sont assimilées respectivement à l'inertie intra classe et l'inertie inter classe. Cette méthode cherche la partition qui minimise l'inertie intra classe et maximise l'inertie inter classe. La méthode des nuées dynamiques proposée par Diday (Diday 1971) généralise la méthode des centroïdes. Les noyaux peuvent être un ensemble de points (les plus centraux), un axe principal, un plan principal, etc. La méthode est applicable sur tous les types de données, mais il convient de définir les étapes de réallocation (2.1) et de représentation (2.2) appropriées. A chaque itération, la méthode minimise un critère qui mesure l'adéquation de la partition avec l'ensemble des noyaux. Le critère s'interprète en fonction des distances utilisées au cours des étapes de réallocation (2.2) et de représentation (2.3). Les méthodes de réallocation sont des méthodes heuristiques particulièrement efficaces, leur complexité étant de l'ordre de N (N étant le nombre d'individus). Elles déterminent un optimum local du critère choisi et nécessitent de fixer le nombre de classes a priori. Pour rendre le résultat moins dépendant de la partition initiale, Diday propose d'utiliser plusieurs fois l'algorithme, avec des partitions initiales différentes, afin de dégager des « formes fortes ». C'est à dire les ensembles d'objets ayant toujours été classés ensembles.

L'agrégation de similarité (Marcotorchino 1982) est une méthode relativement récente, particulièrement bien adaptée aux données qualitatives. Lorsque les données se présentent sous la forme d'un tableau de similarité dont la valeur est comprise entre 0 et 1, le problème du partitionnement revient à chercher une relation binaire Y telle que (De Guio 1990) :

$$Y\alpha^* = \underset{Y}{Max} \sum_{ij} ((s(w_i, w_j) - (1-\alpha)) y_{ij}$$

Avec :
1. $y_{ij} = y_{ji}$
2. $y_{ij} + y_{jk} - y_{ik} \leq 1$
3. $y_{ij} = 0$ ou 1

$\Omega = \{w_i\}$: ensemble des objet à classifier
S : mesure de similarité définie sur $\Omega \times \Omega$
$Y = (y_{ij})$: relation d'équivalence définie sur $\Omega \times \Omega$
$y_{ij} = 1$ si wi et wj sont en relation, 0 sinon
$\alpha \in [0 ;1]$: paramètre de réglage de l'homogénéité et de l'isolation

Dans les méthodes d'agrégation de similarité, les notions d'homogénéité et d'isolation sont interprétées en termes de similarité intra classe (la somme des similarités à l'intérieur des classes) et de dissimilarité inter classe (la somme des disimilarités entre les objets de classes différentes). Le paramètre α compris entre 0 et 1 permet de privilégier l'un ou l'autre de ces deux critères.

II.2.1.3 Conclusion

En Classification Automatique, la formation des classes repose sur la notion de ressemblance. Les objets sont comparés sur un ensemble fixe d'attributs descriptifs et sont décrits dans un formalisme appelé attributs-valeurs. Les données utilisées sont régulières, homogènes et monovaluées. Les variables doivent toutes être du même type. Les mesures de ressemblances sont utilisées pour déterminer, à l'aide d'heuristiques, une partition qui optimise un critère variant selon les méthodes. Les critères utilisés sont une interprétation des notions d'homogénéité et d'isolation. Le processus de formation des classes repose sur un certain nombre d'hypothèses souvent implicites dont la plus importante est sans doute le critère utilisé. Il nécessite le réglage de paramètres généralement difficiles à interpréter pour un non spécialiste. Les méthodes numériques sont des outils efficaces pour classifier de grandes quantités de données. L'inconvénient est qu'elles produisent uniquement des classes et non des concepts. Il est souvent difficile d'interpréter les classes à partir des attributs descriptifs. Si les objets rassemblés présentent de fortes ressemblances pris deux à deux, il n'existe pas forcément d'attribut descriptif ayant la même modalité pour tous les objets de la classe. Ce type de classes est qualifié de polythétique par opposition aux classes monothétiques dont les objets présentent une ou plusieurs modalités communes sur l'ensemble des attributs considérés.

II.2.2 La classification conceptuelle

La problématique de la Classification Conceptuelle peut se résumer en ces termes : étant donné une grande quantité d'informations, comment regrouper ces informations en éléments moins nombreux ayant une signification. Par signification d'un regroupement, on entend une classe munie d'une intention, c'est à dire un concept (cf. I.1.2). Partant de ce principe, les méthodes de Classification Conceptuelle associent une description à chaque classe formée. Cette description est assimilée à intention d'un concept. Généralement le résultat se présente sous la forme d'une structure hiérarchique. Les classes et leur description sont associées aux nœuds de la hiérarchie. Les arcs représentent la relation d'inclusion entre les classes (cf. annexe II.2.1). Le langage utilisé pour la description des classes s'appelle le langage des hypothèses. En définissant des concepts à partir de données, les systèmes de Classification Conceptuels réalisent une forme particulière d'apprentissage inductif car ils infèrent une généralité à partir d'un ensemble fini de cas particuliers (Kodratoff 1986, Stepp 1986). En apprentissage inductif, la prise en compte de connaissances déclaratives spécifiques au domaine est très vite apparue comme un moyen efficace d'améliorer les résultats, en focalisant l'apprentissage vers des solutions plus pertinentes. Dans cet esprit, la plupart des méthodes de Classification Conceptuelle peuvent utiliser des connaissances du domaine, fournies sous une forme explicite par les utilisateurs (Bisson 1993). Les objets à classer sont appelés individus ou observations. C'est pourquoi cette forme d'apprentissage est aussi appelée Apprentissage à Partir d'Observations. La problématique de la Classification Conceptuelle est la suivante :

Etant donné :

- *un ensemble E d'observations*
- *un formalisme de description des observations : le langage des instances ;*
- *un formalisme de description des connaissances du domaine ;*
- *un formalisme de description des classes : le langage des hypothèses ;*
- *un critères d'évaluation de la qualité d'une classification.*

Trouver :

- *un ensemble de classes qui regroupent les observations ;*
- *une définition intensionnelle de chaque classe ;*
- *une organisation hiérarchique de ces classes*

Déf. II. 6 La problématique de la Classification Conceptuelle

Nous passons en revue ci-dessous, quelques réalisations parmi les plus représentatives. L'étude se limite aux systèmes qui génèrent des partitions, des recouvrements, des hiérarchies de parties ou de recouvrements. Ce qui exclut notamment les recherches basées sur la structure de treillis de Galois (Godin 1995).

II.2.2.1 Cluster

Le système Cluster/Paf (Michalski 1981) est sans doute le premier véritable programme de Classification Conceptuelle. Il a donné naissance à la famille des systèmes Cluster, dont l'implémentation la plus connue est Cluster/2 (Stepp 1983), suivie de Cluster/S (Stepp 1986).

Dans Cluster/2, les observations à classifier sont représentées dans le formalisme attributs-valeurs utilisé en Analyse de Données. L'auteur considère trois types de variables discrètes : les variables nominales, les variables linéaires et les variables structurées. Les variables linéaires sont des variables métriques discrètes, c'est typiquement une variable à valeur dans l'ensemble des entiers naturels. Les variables structurées[3] prennent leur valeur dans une hiérarchie de concepts ordonnés par une relation d'inclusion (cf. annexe II.2.2). Les données sont comme en Classification Automatique, régulières, homogènes et monovaluées. Par contre, il n'est pas nécessaire d'homogénéiser le type des variables. La description des classes est une condition nécessaire et suffisante. Elle prend la forme d'une conjonction de sélecteurs appelée complexe.

Exemple II. 3 Représentation des classes dans Cluster /2

Les objets de la classe désignée par le complexe ci-dessous présentent les caractéristiques suivantes : leur taille est inférieure à 3, leur forme est un carré ou une ellipse, leur poids est compris dans l'intervalle [2, 4].

$$[\ \text{taille} < 3\ ]\ [\ \text{forme} = \text{ellipse} \vee \text{carré}\ ]\ [\ \text{poids} = 2..4]$$

L'algorithme mis en œuvre dans CLUSTER/2 comprend un module de partitionnement et un module de construction de hiérarchie. Le module de partitionnement est une adaptation de l'algorithme des Nuées Dynamiques (cf. II.2.1.2.2.).

[3] La description des objets n'utilise que les feuilles de la hiérarchie. Les valeurs plus abstraites des niveaux supérieurs sont utilisées pour regrouper les objets et décrire les classes.

1. Sélectionner K noyaux **(n1, ..., nk)** parmi l'ensemble des objets O
2. Répéter
   2.1. Pour chaque noyau **ni** construire l'ensemble **Edi** des complexes qui couvrent **ni** et ne couvrent pas les autres noyaux. (Les ensembles Edi sont appelés étoiles).
   2.2. Définir k complexes **{co1, ..., cok}** qui forment une partition de O par sélection et modification des complexes des étoiles Edi.
   2.3. Sélectionner K nouveaux noyau à partir de { co1, ..., cok}

   tant que LEF ({ co1, ..., cok}) < α (liste de seuils numériques définis par l'utilisateur)

Déf. II. 7 L'algorithme de classification utilisé dans Cluster/2

L'algorithme commence par sélectionner K noyaux au hasard. Le système AQ (Michalski 1983) basé sur l'algorithme de l'étoile STAR, associe à chaque noyau un ensemble de complexes qui couvrent le noyau choisi et pas les autres. Cet ensemble s'appelle une étoile. En sélectionnant, ou bien le cas échéant, modifiant les complexes des étoiles, le système détermine un nouvel ensemble de complexes { co1, ..., cok} qui forment une partition de l'ensemble des observations et optimisent le critère LEF. Si cette partition améliore le critère d'évaluation LEF par rapport à la classification précédente, les nouveaux noyaux sont choisis comme les plus caractéristiques des classes (objets du centre). Sinon le système sélectionne des observations atypiques (objets du bord) afin de forcer le système à modifier sa partition. Le module de construction de hiérarchie utilise le module de partitionnement. Il effectue deux boucles qui constituent un cycle. Une boucle itérative effectue l'algorithme de classification en faisant varier le nombre K de classes et retient la partition qui optimise LEF. La boucle récursive effectue le processus itératif pour chaque classe obtenue. Le processus de classification est donc descendant. Les classes sont organisées en une hiérarchie de partitions indicée par le nombre de cycles de la boucle récursive.

La fonction LEF est composée de plusieurs critères qui sont : l'adéquation entre les descriptions (les complexes) et les classes, la simplicité de la description des classes, une distance inter-classe, et d'autres valeurs liées à la discrimination des classes. La fonction LEF est représentée par un liste de couples (critère, seuil). Une partition répond au critère si la mesure du critère est inférieure au seuil. Les critères sont hiérarchisés, en ce sens que les partitions sont d'abord évaluées sur le premier critère, celles qui respectent le seuil fixé, sont ensuite évaluées sur le deuxième critère et ainsi de suite.

Cluster est donc un outil de construction automatique de classification qui utilise la fonction LEF pour évaluer la qualité d'une partition. LEF présente l'intérêt d'être explicite et de tenir compte de la description des classes. Mais l'utilisation de l'algorithme nécessite le réglage de

nombreux paramètres. Ces réglages sont délicats et nécessitent la présence d'un spécialiste, comme le mentionne les auteurs : pour la fonction LEF, « le choix des critères élémentaires, leur ordre d'évaluation et les seuils de tolérance sont donnés par un analyste de données » (Stepp 1993). De plus la complexité de la fonction STAR rend l'algorithme inutilisable à grande échelle (Ketterlin 1995).

L'un des objectifs revendiqués de la Classification Conceptuelle est de déterminer des classes sensées. Concrètement, les classes sont considérées comme ayant un sens, s'il est possible de les décrire. Pour ce faire, il ne suffit pas de classifier à l'aide d'un algorithme de classification puis de décrire les classes en utilisant une méthode d'acquisition de concept par exemple. Car rien ne garanti que les classes auront une description simple. C'est par exemple le cas en Analyse Typologique. La stratégie utilisée dans Cluster consiste à ne pas séparer les étapes de classification et de description des classes. L'algorithme de l'étoile définit des complexes. Ces complexes définissent des classes. Les objets sont regroupés, non pas sur la base d'une distance, mais s'ils appartiennent à l'extension d'un même complexe. L'ensemble des complexes possibles est une donnée a priori du système. Les auteurs décrivent explicitement cette situation « Une collection d'objets forme une classe si cette classe peut être décrite par un concept, compte tenu d'un ensemble de concepts prédéfinis » (Stepp 1993). C'est le principe de la cohésion conceptuelle. Le langage des hypothèses, est une donnée a priori du système. C'est donc aussi un biais de classification qui limite la recherche à l'ensemble des classes que peut décrire le langage des hypothèses. Le regroupement conceptuel tel qu'il est développé dans Cluster consiste en fait à explorer à l'aide d'heuristiques, l'espace des descriptions possibles en optimisant un critère particulier. Pour évaluer la qualité d'une partition, ce critère considère entre autres, la simplicité des descriptions et leur adéquation avec les classes. Par rapport aux méthodes numériques, le biais du langage des hypothèses a l'avantage d'être explicite.

II.2.2.1.1 Cluster/S

Cluster/S est une extension de Cluster/2. Il permet d'utiliser des connaissances du domaine et autorise une description plus riche des objets et des classes. Dans Cluster/S, les connaissances, les classes et les observations sont représentées à l'aide du langage APC (calcul des prédicats annotés) (Michalski 1983). C'est une extension de la logique du premier ordre qui permet de représenter des objets structurés (cf. annexe II.2.3). Les objets sont décrits par une liste de prédicats. Cette liste varie d'un objet à l'autre. Ils ne sont donc pas réguliers. Les prédicats peuvent être de la forme [taille(objet) = 1 $\vee$ 2] par exemple et correspondent à des attributs multi-valués.

L'algorithme de Cluster/S reprend en fait celui de Cluster/2 avec une étape supplémentaire qui consiste à traduire les représentions relationnelles en représentation attribut-valeur. D'un point de vue algorithmique, Cluster/S présente donc les même inconvénients que son

prédécesseur. Cluster/2 utilise des représentations tabulaires. Le passage d'une représentation à l'autre repose sur des hypothèses simplificatrices qui déforment la sémantique des données.

Cluster/S prend en compte deux types de connaissances du domaine : des règles d'inférences et un réseau de dépendance des objectifs (RDO). Les règles d'inférences permettent de construire de nouveaux descripteurs et ainsi d'enrichir la description des objets. Le réseau de dépendance des objectifs construit par l'utilisateur, associe les buts de la classification aux descripteurs pertinents. Ce réseau permet de sélectionner les descripteurs pertinents quand ils existent et de choisir les règles d'inférence susceptibles de créer des descripteurs intéressants (cf. annexe II. 2.4).

II.2.2.2 Cobweb et la formation de concepts

Le système Cobweb (Fisher 1987) est à l'origine d'un nouveau domaine de la Classification Conceptuel : la Formation de Concepts. Cette approche reprend l'idée issue de la psychologie cognitive, d'une mémoire sémantique organisée hiérarchiquement qui se structure en fonction des expériences du sujet (Richard 1998). Les observations initiales sont organisées en une hiérarchie de concept. Chaque nouvelle observation s'intègre dans la structure existante et permet de la modifier, en affinant les concepts initiaux. La formation de concept est aussi appelée Classification Conceptuelle incrémentale.

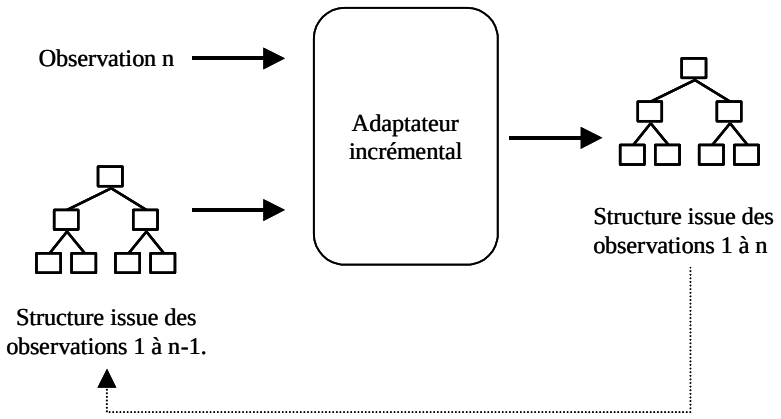


Fig. II.2 Principe de la Formation de Concept

Les objets sont décrits par une liste de couples attributs valeurs. Les attributs sont exclusivement nominaux. Les objets sont réguliers, homogènes et monovalués. Les classes sont décrites par un ensemble de distribution de probabilités. Ces distributions représentent la probabilité qu'un attribut prenne une valeur donnée, connaissant la classe d'un objet. Toutes les modalités de l'ensemble des attributs sont considérées pour chaque classe. Cette

représentation est inspirée de la notion de prototype. Elle se prête mal à la définition d'une fonction de reconnaissance simple, car elle rejoint la notion de classe polythétique (cf. II.2.1.4) (Decaestecker 1991) p42. Une description n'a pas forcément une caractéristique partagée par tous les membres de la classe, ni une caractéristique particulière qui le distingue des autres classes. L'ensemble classe et description est appelé un concept probabiliste. (cf. annexe II.2.5).Contrairement aux programmes de la famille Cluster, Cobweb n'utilise pas de connaissances du domaine. Le processus de classification est descendant. Il est dirigé par l'introduction d'une nouvelle observation dans le nœud courant (initialement la racine de la hiérarchie). Les probabilités du nœud courant sont mises à jour, afin de généraliser la nouvelle classe. Le système applique ensuite différents opérateurs (placer, créer, division, fusion) sur les fils du nœud courant (cf. annexe II.2.6), et sélectionne celui qui produit la meilleur partition par rapport au critère PU (partition utility). En fonction de l'opérateur, il sélectionne un nouveau nœud courant et réitère le processus récursivement. L'algorithme s'arrête lorsque la nouvelle observation est placée dans une feuille de la hiérarchie. Intuitivement, le programme fonctionne de la manière suivante : si une observation O correspond bien à un concept existant, alors elle y est placée. C'est l'opérateur placer. Si O est différente de tous les concepts, alors il faut la placer dans une nouvelle classe. C'est l'opérateur créer. Si O est en accord avec deux concepts, différents, alors le programme fusionne les concepts, afin d'éviter le développement de concepts trop similaires. C'est l'opérateur fusion. Si O est en accord avec l'un des sous concept d'un concept donné, tout en étant en désaccord avec les autres sous concepts, alors le programme élimine ce concept trop général et peu homogène. Il le remplace par ses fils. C'est l'opérateur division.

1. Procédure Ajout (observation, nœud-courant)
2. Intégrer objet dans nœud courant
3. Mettre à jour les probabilités du nœud courant
4. Pour tous les fils Fi du nœud courant

   Choisir l'opérateur qui conduit à la partition optimisant PU parmi :

   - Placer noeud-courant :=Fi ; Ajout(observation, nœud-courant)
   - Créer Créer un nouveau fils O correspondant à observation ; Fin
   - Fusion Créer un nouveau fils F par fusion de deux concepts Fi ; nœud courant := F ; Ajout(observation, nœud-courant)
   - Division Remplacer un concept Fi par ses fils (division) ; Ajout(observation, nœud-courant)

Déf. II. 8 Algorithme de classification employé dans Cobweb

L'exploration de l'espace de recherche (l'ensemble des hiérarchies) repose sur une méthode de gradient. Cette recherche est guidée par l'heuristique PU. Les opérateurs de restructuration placer, créer, fusion et division contribuent à réduire les problèmes de minima locaux propres

aux méthodes de gradients, ainsi que la sensibilité à l'ordre de présentation des observations. Le critère d'évaluation des partitions PU est basé sur la mesure d'utilité d'une catégorie : CU, développée par Gluk et Porter (Gluck 1985). L'utilité d'une catégorie dépend à la fois du degré de prédictivité des attributs et de leur typicité, en tenant compte de la fréquence de l'attribut.

$$CU(C) = \sum_{ij} P(Ai=Vij)\ P(Ai=Vij \mid C)\ P(C \mid Ai=Vij)$$

- La prédictivité correspond à la probabilité conditionnelle $P(C_k \mid A_i=V_{ij})$ qu'a un objet d'appartenir à une classe C_k sachant que l'attribut A_i prend la valuer V_{ij}.

- La typicité correspond à la probabilité conditionnelle $P(A_i=V_{ij} \mid C_k)$ qu'a un objet d'avoir V_{ij} pour valeur de l'attribut A_i sachant qu'il appartient à la classe C_k.

- La fréquence de la modalité j de l'attribut i se note $P(A_i=V_{ij})$

PU mesure le gain moyen de CU obtenu en divisant une classe C en sous classes C_k.

$$PU(C, C1, ..., CK) = \frac{1}{K}\left(\sum_{k} CU(Ck) - CU(C) \right)$$

D'un point de vue qualitatif, la mesure PU tend à maximiser la capacité de prédiction d'une partition (MacKusick 1991). A chaque itération, la partition optimise la capacité des descriptions des classes à prédire les valeurs manquantes d'une observation issue du concept père. Une fois construits, les concepts forment un tout. La hiérarchie permet en partant du haut vers le bas et en utilisant une opération d'appariement partiel de déterminer la classe qui prédit le mieux les valeurs manquantes d'une observation. Il est possible d'en extraire une partition utilisable pour elle-même en considérant le premier niveau de la hiérarchie. Cette partition représente un optimum local du critère PU.

Cobweb est une méthode de Classification Conceptuelle, en ce sens qu'il génère une description des classes construites. Par contre, les descriptions utilisées : les concepts probabilistes, se rapprochent de celles employées dans un contexte numérique. Le critère PU, sélectionne les classes qui matérialisent les associations d'attributs ayant le plus de chance de se produire dans les données. D'un point de vue statistique, les associations d'attributs les plus probables sont les plus courantes. Les concepts de prédictivité et de typicité, sont en fait une interprétation probabiliste des concepts de dissimilarité inter classe et de similarité intra classe (Fisher 1987).

II.2.2.2.1 Extensions

Suite à Cobweb, de nombreuses contributions ont été élaborées sur la base de la même structure de contrôle. **Classit** et **Labyrinth** considèrent respectivement des attributs à valeurs réelles et des objets structurés. **ADECLU** (Decaestecker 1991) permet de travailler sur des objets irréguliers avec des attributs structurés. Le programme utilise une heuristique différente de la PU : la mesure d'adéquation entre un concept et une observation. L'adéquation est assimilable à une mesure de similarité particulière. Un objet est placé dans un concept d'une partition, s'il est plus proche de ce concept que de tous les autres. Une particularité intéressante est que le programme apprend le degré de pertinence des attributs relativement à la classification qu'il est en train de construire. La pertinence d'un attribut par rapport à une classe est une estimation statistique de l'association entre la valeur de l'attribut et la fonction caractéristique de la classe. Ketterlin dans le domaine du traitement des images satellites, propose une adaptation de Coweb particulièrement efficace (Ketterlin 1995). Le système permet de travailler avec un type particulier d'objets structurés, que l'auteur qualifie de composites. Chaque objet est une hiérarchie de concepts ordonnés par une relation de subsomption. Les attributs peuvent être structurés et multivalués. Le système travaille avec des objets ayant tous la même structure. Ils sont donc homogènes et réguliers.

II.2.2.2.2 Le système KBG

KBG (Bisson 1993) est un algorithme de Formation de Concept un peu particulier. Il se distingue des systèmes de la famille Cobweb, car il utilise un indice de similarité associé à l'algorithme de la classification hiérarchique ascendante (cf. I.2.1.1). KBG présente cependant de nombreux avantages sur les systèmes de classification numériques : la mesure de similarité permet de classifier des objets structurés décrits dans un langage similaire au Calcul des Prédicats Annotés de Michalski (Michalski 1983). Le système permet d'exploiter des connaissances du domaine assimilables aux règles d'inférences utilisées dans Cluster. La stratégie consiste à saturer les exemples en appliquant toutes les règles à l'aide d'un moteur d'inférence en chaînage avant. Par contre, la complexité du calcul de la mesure de similarité limite les applications à quelques centaines d'objets (Kodratoff 1997).

II.2.2.3 Conclusion

Il y trois différences majeures entre les méthodes numériques et la Classification Conceptuelle :

1. La description des objets peut être structurée. Elle n'est pas forcément régulière et monovaluée.
2. La plupart des méthodes peuvent utiliser des connaissances fournies par l'utilisateur sous une forme explicite.
3. Les classes formées sont associées à une description qui en facilite l'interprétation.

La plupart des systèmes efficaces travaillent cependant sur des données non structurées ou alors utilisent des hypothèses restrictives. Par exemple, les objets ont tous la même structure. Les connaissances du domaine sont généralement des règles d'inférences qui permettent d'enrichir la description des objets. Pour déterminer des classes munies d'une intention, les systèmes de Classification Conceptuelle intègrent les processus de catégorisation et de description des classes. L'approche consiste à regrouper les objets non seulement parce qu'ils sont proches au sens d'une certaine distance (ce que font les méthodes d'Analyse Typologique), mais aussi parce que la classe formée, matérialise l'extension d'un certain concept (Hason 1986). C'est à dire qu'il est possible de décrire cette classe dans le formalisme des hypothèses. Les algorithmes reposent sur l'optimisation d'un critère qui tient compte de la qualité des descriptions recherchées. Le processus de formation des classes repose sur un certain nombre d'hypothèses, notamment le choix du critère à optimiser. Certaines de ces hypothèses sont explicites, par exemple le type intention recherchée. La Classification Conceptuelle offre des outils efficaces pour analyser des ensembles de données importants. Cependant, tout comme en Analyse Typologique, il est le plus souvent nécessaire de régler des paramètres dont la sémantique n'est pas toujours très claire pour un utilisateur non-spécialiste.

II.2.3 Les Réseaux de Neurones

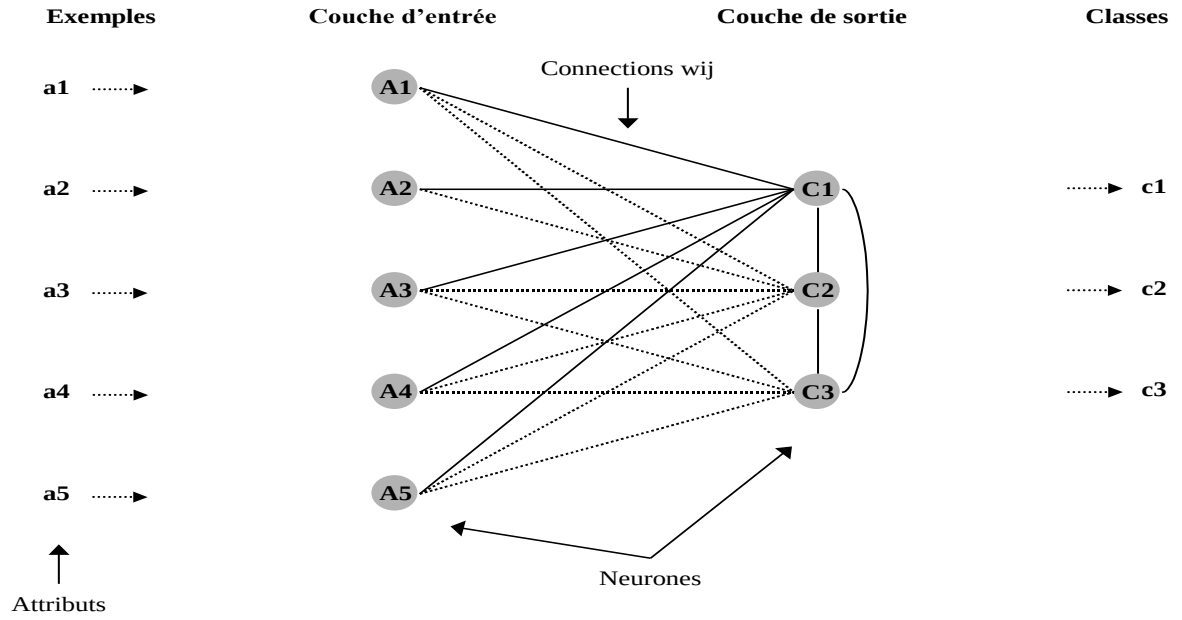
Les Réseaux de Neurones (RN) sont un modèle de traitement de l'information qui imite le fonctionnement du cerveau. Ils sont composés d'un ensemble de neurones formels connecté entre eux. Ce sont les chercheurs Mc Culooth et Pitts qui ont défini le premier neurone formel en 1943, donnant ainsi naissance à un nouveau domaine de recherche : le connexionisme. Les RN ont un domaine d'application très large : identification et contrôle, apprentissage, optimisation, reconnaissance des formes, etc. Dans le cadre de ce mémoire, notre objectif n'est pas de faire une analyse détaillée d'un domaine aussi vaste que les problèmes de classification par RN, mais plus simplement de passer en revue les méthodes les plus connues et ou les plus utilisées. Les tâches de classification (au sens classifier) sont essentiellement traitées par des réseaux à base de prototypes (Amy 1996). A partir d'exemples (les objets à classifier) ces modèles mémorisent des prototypes qui représentent une classe. Les réseaux à prototype les plus connus sont sans doute les cartes de Kohonen (Kohonen 1989).

II.2.3.1 Les cartes de Kohonen

Le modèle de Kohonen est un réseau à deux couches : la couche d'entrée et la couche de sortie. Les exemples à classifier sont présentés sur la couche d'entrée. Chaque neurone de la couche d'entrée représente un attribut descriptif des exemples. Le nombre d'attribut est fixé une fois pour toutes. Les objets sont donc réguliers. Chaque neurone prend une valeur et une seule. Les attributs sont monovalués. Les neurones de la couche de sortie représentent les classes.

Exemple II. 4 Carte de Kohonen à une dimension

Considérons un problème de classification en trois classes c1, c2 et c3 sur des objets décrits par 5 attributs : a1, a2, a3, a4 et a5. L'état des neurones Ai de la couche d'entrée est défini par la valeur des attributs ai. Les neurones Cj de la couche de sortie représentent les classes. Les connexions entre les neurones Ai et Cj sont pondérées par les poids wij.



En phase d'apprentissage, le réseau apprend en modifiant les poids des connections entre la couche d'entrée et la couche de sortie. A chaque neurone Cj, correspond un vecteur poids : (w1j, ... , wIj). Ce vecteur poids s'interprète comme un élément de l'espace des exemples. A la fin de l'apprentissage il représentera le prototype d'une classe d'exemple. En phase d'exploitation, tout exemple présenté en entrée du réseau activera le prototype auquel il ressemble le plus. Les connections entre les neurones de la couche de sortie interviennent uniquement en phase d'exploitation. Elles produisent un effet inhibiteur ou excitateur selon la distance au neurone considéré.

1. Affecter des valeurs aléatoires proches de zéro aux poids du réseau.
2. Initialiser le nombre d'époques : t=0
3. Initialiser le nombre maximal d'époques : T
4. Initialiser le coefficient de convergence $\gamma(0)$
5. Initialiser le rayon de voisinage d'un neurone Cj : VCj(0)
6. Tant que t < T (t : numéros de l'époque en cours, T nombre d'époques maximal) **faire**
   6.1. **Pour** chaque exemple E **faire**
      6.1.1. Présenter E à l'entrée du réseau
      6.1.2. Calculer la distance dj entre E et chaque neurone Cj de la couche de sortie
      6.1.3. Sélectionner le neurone Cj* de la couche de sortie le plus proche de E
      6.1.4. Modifier le vecteur poids du neurone Cj*, ainsi que ceux des neurones de la couche de sortie se trouvant dans son voisinage.
      6.1.5. $wij(t+1) = wij(t) + \gamma(t)(ei - wij(t))$ $\forall j \in VCj^*(t), \forall i \in [1, I]$
      6.1.6. $wij(t+1) = wij(t)$ $\forall j \notin VCj^*(t), \forall i \in [1, I]$
      Fait
   6.2. Incrémenter le nombre d'époque t = t+1
   6.3. Diminuer le rayon de voisinage
   Fait

Déf. II. 9 Algorithme des cartes de Kohonen

Dans un premier temps l'algorithme définit des prototypes au hasard. Chaque nouvel exemple est ensuite associé au prototype le plus proche au sens d'une distance d, fixée par l'utilisateur. Le prototype sélectionné ainsi que tous les prototypes compris dans son rayon de voisinage se « déplacent » en ligne droite d'une distance fixée par le rayon de convergence $\gamma(t)$, afin d'être plus représentatifs de leur classe. Approximativement l'algorithme a tendance à produire des prototypes équidistants des objets de leur classe. Le rayon de convergence est une fonction décroissante qui tend à stabiliser le réseau. La valeur initiale est importante. Elle joue sur le temps d'apprentissage. Si $\gamma(t)$ est faible les prototypes se rapprochent lentement de la solution. Si $\gamma(t)$ est trop importante, les prototypes oscillent autour de la solution. Une époque correspond au passage de tous les exemples. Il faut généralement plusieurs époques pour stabiliser le réseau. Le nombre d'époque est un critère d'arrêt courant. La méthode est sensible à l'ordre de présentation des exemples. Pour utiliser les cartes de Kohonen, l'utilisateur doit donc répondre aux suivantes questions :

1 Quelle distance utiliser ?
2 Quelle est la valeur du coefficient de convergence ?
3 Quelle est le nombre de neurones de la couche de sortie ?
4 Dans quel ordre faut-il présenter les exemples ?
5 Comment choisir le neurone vainqueur en cas d'égalité ?

Il existe de nombreuses variantes de cet algorithme. Les contributions ont pour objectif de corriger une faiblesse de la méthode ou bien d'en proposer un complément. De Sieno (DeSieno 1988), tente d'améliorer la convergence de l'algorithme. Fritzke (Fritzke 1994) propose un algorithme qui détermine la taille de la carte. Hammadi-Mesmoudi (Hammadi-Mesmoudi 1995) élabore une méthode qui répond aux questions 1 et 5.

II.2.3.2 Conclusion

Les objets traités sont homogènes, réguliers et monovalués. Les cartes de Kohonen et les méthodes dérivées ne prennent pas en compte de connaissances du domaine a priori, sous une forme explicite. A notre connaissance, il n'existe pas de méthode dérivée qui intègre les règles de classification ou des contraintes de proximité entre objets. L'algorithme utilisé est une variante des méthodes de réallocation utilisées en Classification Automatique (cf. II.1.1.1.2) très proches des méthodes de type centroïdes ou K-means. A la différence que la classification est incrémentale. Le critère d'évaluation de la qualité d'une partition n'est pas clair. L'algorithme est efficace, mais il détermine un optimum local dépendant du choix des prototypes initiaux. Cette méthode nécessite de régler des paramètres spécifiques difficiles à interpréter pour un utilisateur non-spécialiste. Les classes sont représentées par des prototypes. Il est difficile d'en déterminer l'intention

II.2.4 Les Algorithmes Evolutionnistes

Les Algorithmes Génétiques ont été introduits par John Holland en 1962 (Holland 1962). Il s'est inspiré de la théorie de l'évolution, notamment la notion de sélection naturelle de Darwin et les méthodes de recombinaison des gènes de Mendel. Les Algorithmes Génétiques génèrent une population qui évolue de génération en génération. Selon le principe de l'hérédité, les enfants héritent des caractéristiques de leurs parents. Entre deux générations, un mécanisme de sélection permet de déterminer les individus les mieux adaptés. Ces individus auront une descendance plus importante que les autres. L'enchaînement des générations permet donc d'obtenir des individus de mieux en mieux adaptés. A l'heure actuelle, le terme Algorithme Evolutionniste englobe différents paradigmes qui se sont développés à partir du principe initial (la programmation évolutionniste, les stratégies évolutives, la programmation génétique, ...). Spears et al. (Spears 1993) suggèrent un schéma général pour ces algorithmes que nous présentons ci-dessous (Déf. II.10). Les Algorithmes Evolutionnistes résolvent

essentiellement des problèmes d'optimisation. Les individus de la population représentent des solutions potentielles du problème considéré. Le critère à optimiser sert à mesurer l'adaptation des individus.

1. Initialiser le compteur de génération à 0 : t=0
2. Initialiser la population d'individus
3. Evaluer la qualité de tous les individus de la population initiale
4. **Tant que** le critère d'arrêt n'est pas vérifié (nombre de génération, valeur de la fonction objectif, etc. ..) **faire**

   4.1. Augmenter le compteur de génération t = t+1

   4.2. Sélection : Sélectionner une sous population pour produire des enfants

   4.3. Recombinaison : Recombiner les gènes des parents sélectionnés

   4.4. Mutation : Perturber aléatoirement la population recombinée

   4.5. Evaluation : Evaluer la population des enfants

   4.6. Survie : Sélectionner les survivants

   Fait

Déf. II. 10 Algorithme Evolutionniste

La population initiale peut être construite de plusieurs façons : tirage au hasard, solution heuristique, mélange de solutions heuristiques et aléatoires, tec. La sélection consiste à choisir les individus qui vont se reproduire. Elle est basée sur une fonction objectif ou fonction d'adaptation, qui mesure, pour chaque individu, le critère à optimiser. L'objectif est de reproduire les individus les mieux adaptés. La solution la plus connue consiste à assigner à chaque individu une probabilité d'être sélectionné proportionnelle à son adaptation. La recombinaison et la mutation son des opérateurs de reproduction. Ils sont généralement appliqués au hasard sur la population des individus sélectionnés selon deux probabilités appelées taux d'hybridation et taux de mutation. La recombinaison est une adaptation du principe d'hybridation (le terme hérédité est réservé aux être humains). A partir de deux individus différents, on obtient un hybride qui présente des caractéristiques de ses deux parents. Lors de la recombinaison, les deux parents sont divisés d'une manière aléatoire, en deux ou plusieurs parties. Les enfants sont formés en assemblant des parties qui proviennent des deux parents. La mutation consiste à changer de manière aléatoire des caractéristiques d'un individu. La survie est comparable à la sélection.

Comme pour les Réseaux de Neurones, nous ne prétendons pas présenter dans ce mémoire un panoramas exhaustif des problèmes de classifications traités par les Algorithmes

Evolutionnistes, mais limiterons notre investigation aux méthodes les plus connues et ou les plus utilisées notamment en TG. Dans le cadre des Algorithmes Evolutionnistes, les problèmes de classification ont essentiellement été traités avec des Algorithmes Génétiques et des Stratégies Evolutives (Kettaf 2000). Les individus représentent des partitions. L'algorithme permet de sélectionner la meilleure par rapport à un critère donné. Nous présentons ci-après une approche récente.

II.2.4.1 Classification par Algorithmes génétiques

L'approche développée par Kettaf et Asselin de Beauville (Kettaf 2000), présente l'intérêt de ne pas fixer le nombre de classe a priori et d'utiliser un critère original pour évaluer la classification. Les individus sont appelés chromosomes. Ils sont représentés par des chaînes de bits. Une sous chaîne de bits est appelée gène. Les auteurs utilisent une représentation dite « classée » pour représenter les partitions. Un gène code une classe. A l'intérieur de cette classe, la présence ou l'absence d'un objet correspond à un code binaire. Ce codage implique d'utiliser des chromosomes de longueurs variables, si l'on veut éviter de fixer le nombre de classe.

Exemple II. 5 Représentation classée d'une partition

Soit l'ensemble des objets Ω={1, 2, 3 ,4, 5, 6, 7}.

Partition	Représentation par un chromosome
P1 : (1, 3, 4, 6) (2, 5, 7)	(1011010) (0100101)
P2 : (1, 3) (4, 5) (2, 6, 7)	(1010000) (0001100) (0100011)

La fonction objectif mesure la qualité d'une partition. Il s'agit le plus souvent d'un critère issu de l'Analyse Typologique. Pour leur contribution, les auteurs développent un critère original. Il s'agit d'une mesure classique du rapport de l'inertie interclasse sur l'inertie totale. Mais celle-ci est pondérée, en fonction de deux paramètres réglables par l'utilisateur : le cardinal minimal d'une classe et l'éloignement minimal entre le centre de deux classes. Comme la plupart des critères de Classification Automatique, il repose sur une mesure de similarité.

Pour réaliser des tâches de catégorisation, les auteurs définissent des opérateurs de croisement et de mutation adaptés, ainsi que des opérateurs spécialisés que nous ne détaillerons pas. L'opération de croisement échange entre deux chromosomes des parties délimités par des points de coupure situés entre deux gènes. Ces points sont différents chez les deux parents. Cette méthode permet d'engendrer des chromosomes enfants de longueurs variées et donc des

partitions dont le nombre de classes varie. Le croisement consiste à échanger des classes entre deux partitions. Le résultat n'est pas forcément une partition. Les auteurs utilisent une procédure de correction, pour modifier les chromosomes incorrects. L'opération de mutation consiste à déplacer aléatoirement un objet de sa classe d'origine vers une autre classe. La mutation consiste à déplacer un objet de sa classe d'origine vers une autre classe.

II.2.4.2 Conclusion

Les Algorithmes Evolutionnistes sont des méthodes d'optimisation. En tant qu'outil de Classification Automatique, ils permettent de trouver une partition qui optimise un critère donné. La représentation des objets est directement fonction du critère utilisé. Les méthodes existantes utilisent généralement des critères issus de l'Analyse Typologique. Ces critères reposent sur un indice de similarité et imposent donc les même contraintes sur les données qu'en Classification Automatique. Une des caractéristiques des Algorithmes Evolutionnistes est d'utiliser très peu, voir même aucune connaissance du domaine pour déterminer le résultat (Goldberg 1995). Pour intégrer des connaissances spécifiques, il est nécessaire de développer des opérateurs de reproduction ou de définir une fonction de sélection adaptée. Du point de vue de la construction automatique de classification, le principal intérêt des algorithmes Evolutionnistes par rapport aux méthodes précédentes est d'être plus à même de déterminer l'optimum global de l'espace de recherche (Lefébure 1998). On retrouve cependant le problème du réglage de paramètres spécifiques à ces méthodes.

II.2.5 Conclusion

Après avoir passé en revue les principaux OCA, nous proposons de définir ces outils comme une méthode qui permet de résoudre le problème suivant :

Etant donné :

- *un ensemble d'objets*
- *un formalisme de description des objets*
- *éventuellement des connaissances sur le domaine*
- *éventuellement un formalisme de description des classes*
- *un critère d'évaluation de la qualité d'une classification*

Trouver :

- *une classification de l'ensemble des données*
- *qui optimise le critère d'évaluation de la qualité d'une classification*
- *qui soit cohérente avec les connaissances du domaine*
- *éventuellement une intention des classes*

Déf. II. 11 La problématique des Outils de Classification Automatique

Les langages de description des objets sont variés. Les données considérées vont des objets simples de l'Analyse Typologique aux objets structurés manipulés en Intelligence Artificielle. Les connaissances du domaine prises en compte par certaines méthodes sont, sous leur forme la plus générale, des règles d'inférences qui enrichissent la description des objets. Il n'existe pas à notre connaissance d'OCA capable d'intégrer les règles de classification présentées au chapitre I (chapitre I.3.1). La représentation des classes est très variée. A titre d'exemple, nous citerons les classes polythétiques de l'Analyse Typologique, les formes normale conjonctives utilisées par Cluster, les Concepts probabilistes de Cobweb.

Un OCA est d'autant moins contraignant pour des experts d'un domaine donné qu'il offre une représentation riche des objets à classifier. Les classes sont d'autant plus faciles à interpréter qu'elles sont accompagnées d'une intention De ce point de vue, les outils de classification automatiques issus de la Classification Conceptuelle sont sans doute les plus complets. Néanmoins, l'enrichissement du formalisme de description des objets s'accompagne le plus souvent d'une limitation en terme de performance. De plus, l'objectif de l'ACCI est de définir des classes en extension.

Quel que soit le domaine, le problème de la construction des classes d'une partition est traité comme un problème d'optimisation. La taille de l'espace de recherche interdit toute exploration exhaustive. Ce point détermine deux a priori fondamentaux qui sont à la base de tous les OCA présentés :

1. Le choix d'un critère à optimiser ;
2. Le choix d'une méthode d'optimisation.

Il n'existe pas de critère universel pour évaluer la pertinence d'une classification, comme l'atteste la diversité des méthodes passées en revue. Cependant, les critères utilisés ont essentiellement le même objectif : rassembler les objets qui ont des attributs communs. Ce sont les différentes interprétations du concept d'attributs communs qui engendrent la diversité des méthodes. En Analyse Typologique, la notion d'attributs communs est assez clairement traduite en terme de similarité. Les approches de la classification dans les domaines des Réseaux de Neurones et des Algorithmes Génétiques reposent sur les principes de l'Analyse Typologique et utilisent des mesures de similarité pour former des groupes d'objets ayant une forte ressemblance. C'est à dire une proportion importante d'attributs communs. D'une façon peut être moins évidente, il s'avère que l'idée de base commune aux travaux de classification conceptuelle est de comparer des exemples en terme de similarité (Lebowitz 1994), c'est à dire en terme d'attributs communs. La mesure d'utilité d'une partition utilisée dans Cobweb s'interprète comme la mesure d'une similarité particulière qui tient compte non seulement du nombre d'attributs communs, mais aussi de la fréquence des associations entre attributs. Le programme Cluster tient compte de la description des classes. Ces dernières auront une description d'autant plus simple et pertinente que les objets de la classe à caractériser auront d'attributs communs. Une fois le critère défini, les méthodes reposent généralement sur des heuristiques et déterminent un optimum local du critère choisi.

II.3 Conclusion

Pour mener à bien leur activité, les experts d'un domaine développent des connaissances et toutes sortes de stratégies personnelles. En terme d'Acquisition des Connaissances, l'objectif de l'ACCI est de définir une partition dont les classes sont l'extension de concepts du domaine. Plus précisément, cette partition est l'ensemble des classes solution d'un modèle dit de classification simple.

L'ensemble des outils utilisables pour classifier les objets sont très diversifiés, mais reposent sur un principe commun. Par opposition à la classification logique qui nécessite de hiérarchiser les variables descriptives, les outils de Classification Automatique sont des méthodes de classification empiriques. Conçus pour classifier un ensemble d'objet à partir d'un minimum d'informations, ils considèrent les attributs descriptifs comme d'égale importance par rapport à l'objectif de la classification, ou tout au moins d'une importance relative constante sur l'ensemble des objets. Le principe est de regrouper les objets qui ont le plus d'attributs en commun. En des termes plus familier, nous pourrions dire que les méthodes de classification empirique reposent sur le principe du « qui se ressemble s'assemble ». La recherche d'une partition qui respecte ce principe est généralement mis en oeuvre par l'intermédiaire d'une méthode d'optimisation et reposent sur de nombreuses hypothèses : choix de la représentation des objets, choix de la représentation des classes, définition d'un critère à optimiser et d'une stratégie d'exploration de l'espace de recherche.

A la lumière de l'état de l'art des Outils de Classification, il ne ressort pas d'argument discriminant en faveur d'une méthode plutôt qu'une autre. L'objectif du chapitre suivant est d'évaluer les difficultés posées par ces outils pour résoudre un problème d'ACCI. C'est à dire contribuer à l'ensemble du cycle de classification de la figure I.3.

Chapitre III

Les problèmes de validation des classifications

Introduction

Après avoir présenté les principes des OCA, nous allons aborder les problèmes d'utilisation. Ce chapitre a pour objet de répondre aux questions suivantes :

1. Dans quelle mesure les outils présentés au chapitre précédent, sont-ils capables de résoudre un problème d'ACCI ? C'est-à-dire, de construire sur un domaine peu formalisé, une classification des données jugées pertinente par les experts du domaine.

2. Comment peut-on résoudre les difficultés mises en évidence lorsque l'on essaye de répondre à la question précédente ?

L'observation des problèmes de validation rencontrés en Analyse Typologique et en Classification Conceptuelle (cf. III.1) servira de base pour la suite de notre travail. Nous mettrons en évidence l'existence d'un fonctionnement itératif du processus de classification que nous appelons le cycle de classification.

Au cours de la partie III.2 de ce chapitre, nous examinons d'une manière formelle les conditions nécessaires qu'il faut poser sur les données pour qu'un OCA puisse découvrir des concepts du domaine en regroupant les objets dont les descriptions se ressemblent.

Nous présentons ensuite une analyse des problèmes de validation en deux étapes (cf. III.3). Tout d'abord les chapitres III.3.1 et III.3.2 tentent de répondre à la question : pourquoi l'application directe d'un Outil de Classification Automatique donne-t-elle rarement des résultats validés par les experts ? Par la suite, nous nous attacherons à comprendre (cf. III.3.3) qu'elles sont la signification et les difficultés du cycle de classification.

A la lumière de l'analyse précédente, nous nous attacherons à présenter les approches interactives comme une solution pour accélérer le cycle de classification (cf. III.4).

III.1 Etat de l'art des problèmes de validation

Pour la suite de ce travail, nous limiterons notre investigation aux méthodes d'Analyse Typologiques et de Classification Conceptuelle. Cependant, les arguments développés sont valables pour l'ensemble des méthodes présentées aux chapitres II.2. (A l'exception de la régularité d'une mesure de ressemblance numérique qui concerne essentiellement les méthodes d'Analyse Typologique .) **Et non pas l'ensemble des méthodes de classification automatiques existantes**. Rappelons au lecteur que si nous avons pu dresser un panorama relativement complet des méthodes d'Analyse Typologique et de Classification Conceptuelle, nous avons restreint notre présentation à quelques méthodes « phares » parmi les plus connues des Réseaux de Neurones et des Algorithmes Génétiques. Nous annonçons que les arguments développés sont valables pour l'ensemble des méthodes présentées aux chapitres II.2, car d'une part ces arguments sont relatifs aux propriétés du domaine d'application (prédominance des savoir-faire) et aux principes généraux de fonctionnement d'un Outil de Classification Automatique définis en II.2.5 et II.3. Ils ne sont donc pas dépendants d'une famille de méthode. D'autre part, dans les quatre familles de méthodes présentées au chapitre II. Les Algorithmes Génétiques et les Réseaux de Neurones ne sont utilisés qu'en tant que méthodes d'optimisation particulières qui résolvent l'étape de formation des groupes d'un processus de catégorisation basé sur une mesure de ressemblance. A ce titre, elles se justifieraient d'un argumentaire particulier pour les biais de classification liés à l'algorithme de formation des groupes. Mais, à ce niveau, notre argument essentiel réside dans l'utilisation d'une méthode d'optimisation heuristique. Argument valable pour ces deux familles de méthodes.

III.1.1 Les problèmes de validation en Analyse Typologique

L'expérience des chercheurs du L.R.P.S nous permet de constater que dans le cadre de la TG, les familles obtenues à l'aide d'un logiciel d'Analyse Typologique sont rarement validées par les experts de l'entreprise (Frey 1990; Feltz 1993; Laget 1994; Guio 1998; Guio 1999). Ceci est en accord avec les expériences et écrits concernant l'Analyse Typologique appliquée à d'autres problèmes. Dans un domaine peu formalisé, lorsque les connaissances existantes sont essentiellement des savoir-faire, il est fréquent que les résultats d'une Analyse Typologique ne reflètent pas les concepts usuels attachés aux objets (Lagrange 1973, Napoli 1996, Friedman 1993). Pour obtenir un résultat significatif et compréhensible, la connaissance experte du domaine doit être intégrée dans le processus de catégorisation (Friedman 1993).

L'opération est délicate, car d'une part les experts ne sont pas experts de leurs connaissances. D'autre part, en tant qu'utilisateurs, ils ne maîtrisent pas suffisamment les techniques d'Analyse Typologique pour pouvoir en manipuler les méthodes. En effet, les méthodes d'Analyse de Données sont rarement exploitables directement. Elles nécessitent le plus souvent la présence d'un spécialiste que nous appellerons analyste, afin d'adapter la méthode au contexte et à l'objectif (Michalsky 1991; Friedman 1993; Patil 1993)

En pratique, l'interaction entre l'analyste et les experts du domaine prendra le plus souvent la forme d'un dialogue. On retrouvera le cycle de classification présenté au chapitre I dans la figure I.3. Cette conception cyclique du processus de catégorisation, lorsque l'on utilise un outil d'Analyse Typologique sur un domaine peu formalisé a été posée aux origines de l'Analyse Typologique (Chandon 1981). Sur la base d'une information de départ, l'analyste propose une classification initiale, que les experts valideront ou critiqueront. En tenant compte de ces remarques, l'analyste et les experts progresseront par essais erreurs, jusqu'à ce qu'émerge une solution satisfaisante. Nous appellerons ce processus le cycle de classification en analyse Typologique.

III.1.2 Les problèmes de validation en Classification Conceptuelle

Les problèmes de validation sont plus étudiés en Acquisition de Concepts qu'en Classification Conceptuelle. Ces termes recoupent la distinction qu'y est faite en Apprentissage Symbolique Automatique entre l'apprentissage supervisé (Acquisition de Concepts) et l'apprentissage non supervisé (Classification Conceptuelle) (Kodratoff 1986). La majeure partie des difficultés rencontrées dans un contexte supervisé se retrouvent dans un contexte non supervisé. C'est pourquoi, nous passerons en revue quelques problèmes de validation rencontrés en Acquisition de Concepts avant d'aborder la Classification Conceptuelle.

III.1.2.1 Problèmes de validation en Acquisition de Concepts

Lors de la construction d'une base de connaissance à l'aide d'un système d'Acquisition de Concept, les experts du domaine commencent par choisir des exemples qu'ils jugent représentatifs de leur activité. Les exemples sont ensuite décrits à l'aide du langage des instances et de la théorie du domaine. L'étape de classification des exemples est manuelle. Les experts regroupent les exemples qu'ils associent à un même concept. L'outil d'Acquisition de Concepts recherche alors une description de la classe à partir de la description des objets. Il s'ensuit une étape de validation. Si les descriptions sont validées par les experts, alors l'apprentissage s'arrête là ; sinon, il faut modifier une des entrées du système.

Il est rare d'obtenir des concepts satisfaisant à partir des données initiales (Bisson 1993). Les problèmes rencontrés sont principalement (Bisson 1993, Nedellec 1995) :

1. Les experts ont souvent du mal à énoncer les descripteurs initiaux.
2. Il y a des malentendus quant à la sémantique des descripteurs.
3. La classification des exemples n'est pas représentative de l'activité des experts.
4. La classification n'est pas cohérente avec les descripteurs.

L'apprentissage prend généralement la forme d'une série d'itérations (Bisson 1993). Comme en analyse Typologique, les outils d'Acquisition de Concepts ne sont pas utilisables sans de

bonnes connaissances des méthodes employées et nécessitent la présence d'un spécialiste. A chaque étape, les experts du domaine et le spécialiste dialoguent afin de raffiner les données initiales : choix des exemples, modification de la description, modification de la théorie du domaine, réglage des biais de l'outil d'Acquisition de Concepts [1]. Nous appellerons ce processus : le cycle de classification en Acquisition de Concepts. La figure III.2 ci-dessous, en donne les différentes étapes. Les numéros 1 à 8, indiquent les étapes communes avec le cycle de classification en Analyse Typologique (cf. figure I.3), à la différence que l'étape 3 de catégorisation est manuelle. L'étape 3.5 est spécifique à l'Acquisition de Concepts. Il s'agit de la définition de l'intention des classes par le système d'Acquisition de Concepts. Les flèches grises et plus épaisses représentent l'enchaînement théorique du processus. Les flèches noires et fines relient les étapes de bouclage. Les problèmes de validation rencontrés en Acquisition de Concepts ont donné lieu à des travaux de recherche, dont M. Gabriel présente une synthèse (Gabriel 1999), ainsi qu'à un certain nombre de contributions qui utilisent les compétences de l'expert pour guider l'apprentissage : Clint (Readt 1992), APT (Nedellec 1995), Mobal (Morik 1988).

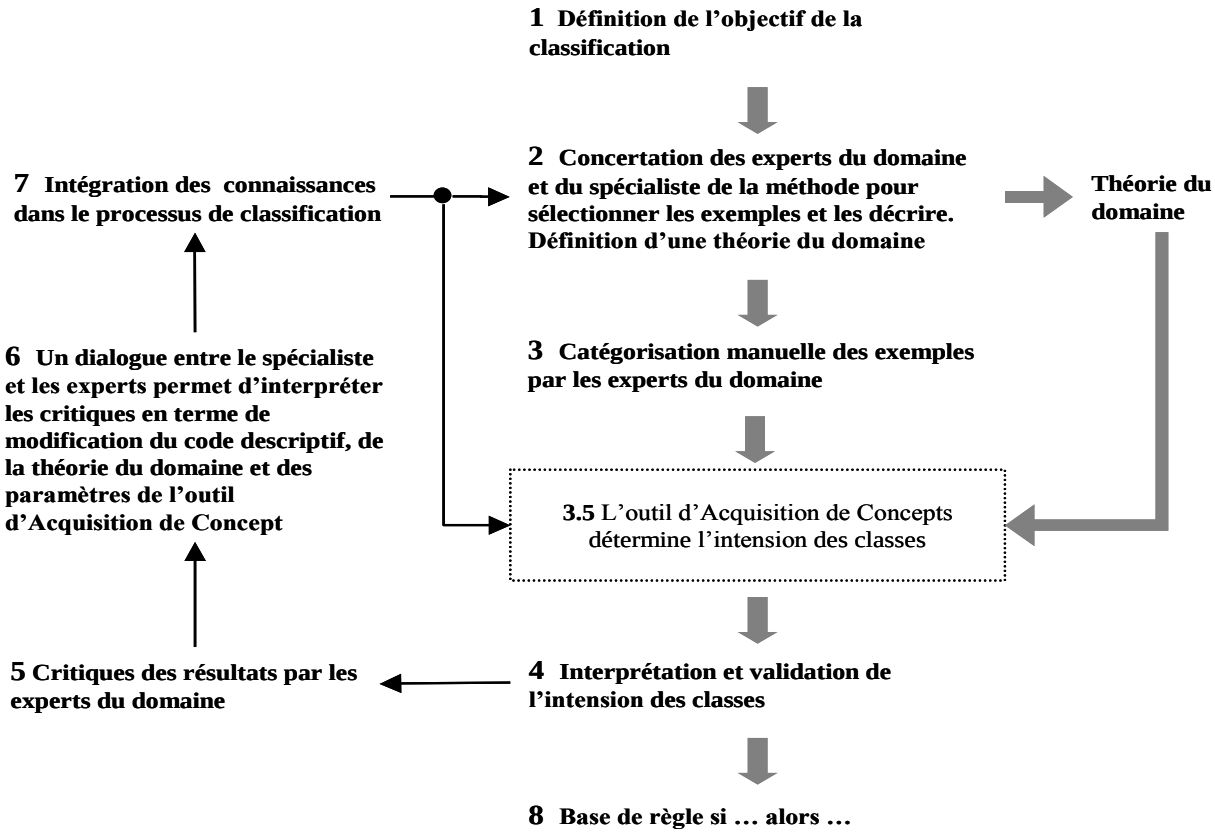


Fig. III.1 Cycle de validation en Acquisition de Concepts

[1] Dans un contexte supervisé, le problème est plus simple que dans un contexte non supervisé. Sous l'hypothèse que les regroupements sont corrects, les méthodes d'induction constructives permettent de chercher de nouveaux descripteurs qui simplifient la description des classes (Michalski 1991).

III.1.2.2 Problèmes de validation en Classification Conceptuelle

Les problèmes rencontrés en Acquisition de Concepts se retrouvent en Classification Conceptuelle. Notamment le problème numéros quatre (cf. III.1.2.1), lorsque la classification n'est pas cohérente avec les descripteurs. Cela signifie qu'il n'est pas possible de retrouver les classes à partir de la description des objets. Il est dès lors évident qu'un système de Classification Conceptuel ne serait pas capable en regroupant les objets sur la base de leur description de retrouver les classes formées manuellement par les experts. Sur un domaine peu formalisé les données initiales ne sont généralement insuffisantes et le processus de classification devient itératif. On retrouve un schéma similaire à celui de l'Acquisition de Concepts (cf. figure III.1), à la différence que les étapes 3 et 3.5 sont réalisées par l'outil de Classification Conceptuelle.

Pour contribuer à résoudre ces problèmes, G. Bisson (Bisson 1993; Bisson 1997) propose une approche itérative associée à des mécanismes d'explication. Dans le cadre de la Classification Conceptuelle et de l'apprentissage supervisé, l'enjeu lui semble être à la fois de construire des bases de connaissances et de systématiser le dialogue avec l'expert.

Selon les travaux de I. Bournaud (Bournaud 1996), la Classification Conceptuelle s'avère inadaptée pour une certaine classe de problèmes : les problèmes de classification pour l'organisation des connaissances. Par exemple : en linguistique et didactique, pour la construction de nouvelles classifications des caractères chinois, en systématique, pour la construction de classifications d'éponges marines, en anthropologie, pour l'étude de la classification des animaux dans la Grèce antique, en ingénierie pour l'organisation des bases de données. Ces problèmes sont tout à fait semblables aux problèmes de classification pour l'extraction de connaissances à partir de données sur un domaine complexe et peu formalisé. L'auteur constate que dans ce cadre, la construction d'une classification est souvent le fruit d'une longue recherche et précise qu'une méthode de catégorisation automatique qui voudrait contribuer à ce travail doit rendre compte de l'aspect itératif du processus. La classification doit pouvoir être modifiée de manière incrémentale jusqu'à ce que les experts du domaine soient satisfaits du résultat (Alberdi 1993). Or les systèmes de Classification Conceptuelle ne sont pas adaptés pour un fonctionnement itératif.

Les méthodes de Classification Conceptuelle sont employées dans le cadre de l'extraction de connaissances à partir des données. Pour illustrer notre propos, nous citerons un passage de l'article écrit par Y. Kodratoff : L'Extraction de Connaissances à partir de Données (Kodratoff 1997) (FD signifie Fouille de Données) : « *Les techniques existantes de FD, qu'elles soient issues des statistiques, des réseaux de neurones ou de l'ASA, comportent une sorte de cycle durant lequel le spécialiste de FD fait des recommandations au spécialiste du domaine. Dans la plupart des cas, le cycle est constitué comme suit : le spécialiste du domaine recueille des données, consulte le spécialiste de FD qui recommande certains changements dans la façon dont sont recueillies les données ou recommande de recueillir de nouvelles données, pour assurer que son outil de fouille travaille correctement. Les nouvelles données modifiées sont alors soumises à l'inspection des outils de FD, et leur résultat est analysé conjointement par le spécialiste de FD et un spécialiste du domaine, de façon à déterminer ce qui, dans les résultats obtenus, est issu des données, ce qui est issu des biais dans les données et ce qui vient de biais dans l'outil de FD. Les données et/ou l'outil de fouille sont alors révisés de façon à ce que les biais ne gênent pas le spécialiste du domaine*. »

III.2 L'hypothèse de similarité et la validation des classes

Cette partie a pour objet la question suivante : quelles sont les hypothèses suffisantes sur les données pour qu'un Outil de Classification Automatique puisse « découvrir » des concepts d'un domaine d'activité. Par définition, toute classe munie d'une intention définit un concept. Mais, notre objectif n'est pas de créer de nouveaux concepts arbitraires. Il s'agit de trouver des concepts qui retranscrivent une réalité du domaine. Sans rentrer dans des considérations métaphysiques, nous considérons que le domaine en question est muni d'une sémantique, c'est à dire un ensemble de concepts pré-existants, explicites ou non. Nous cherchons des concepts au moins cohérents avec cette sémantique, au mieux pertinents par rapport à un objectif donné.

III.2.1 L'hypothèse de similarité

D'une manière qualitative et très générale, les OCA rassemblent les objets dont les descriptions se ressemblent pour former des classes. Le terme de ressemblance est très vague. Pour illustrer notre propos, nous assimilerons la description d'un objet à un ensemble de propriétés. Dès lors, deux objets se ressemblent s'ils ont des propriétés communes. Dans ce contexte, conclure que les classes d'objets qui se ressemblent sont l'extension de concepts du domaine repose sur le raisonnement **iii** ci-dessous :

iii Si les descriptions de plusieurs objets ont des propriétés communes alors ces objets appartiennent à l'extension d'un même concept.

Ce raisonnement s'obtient en trois étapes. Pour la première étape, on part du principe que la définition en intention d'un concept monothétique est une conjonction de propriétés. Les objets d'un même concept vérifient donc tous la même conjonction de propriétés. Ceci, nous donne l'assertion **i**. Pour la deuxième étape, il faut considérer que les OCA n'ont accès qu'à la description des objets. On modifie donc l'assertion **i** pour trouver l'assertion **ii**. La troisième étape consiste à inverser le raisonnement **ii**, pour trouver **iii**.

i Si des objets appartiennent à un même concept du domaine alors ils ont des propriétés communes.

ii Si des objets appartiennent à un même concept du domaine alors leurs descriptions ont des propriétés communes.

Le passage de **i** à **ii** repose sur l'hypothèse qu'il existe une relation simple entre les propriétés qui définissent l'intension du concept et les propriétés utilisées pour décrire les objets.

Exemple III. 1 De la description des objets à l'intension du concept

Considérons un ensemble d'objet $\Omega = \{ wi \}_{1< i< n}$ inclus dans l'univers du discours U, et les variables descriptives à valeurs réelles V1, V2 et V3. La description d'un objet w est définie par l'ensemble des valeurs que prennent les variables descriptives pour cet objet. Une propriété est un prédicat défini par une expression du type Vi = x. Ce prédicat est vrai pour tous les objets de U qui prennent la valeur x pour la variable Vi.
Le cas de figure le plus simple consiste à utiliser les même propriétés pour décrire les objets et pour définir l'intension des concepts. Si on considère le concept C et les objets w1, w2, w3 ci-dessous, les propriétés V1 = 1 et V2 = 3 servent à définir l'intention du concept et à décrire les objets. Dans ce cas, l'assertion **ii** est évidente.

$C = \{ w \in U \text{ tels que } V1 = 1 \text{ et } V2 = 3 \}$

	V1	V2	V3
w1	1	3	5
w2	1	3	6
w3	1	3	2

Par contre, si la relation entre les propriétés qui décrivent les objets et les propriétés qui définissent l'intension des concepts du domaine devient plus complexe, l'assertion **ii** n'est plus valable. Considérons le concept D ci-dessous et les objets w4, w5, w6 ci-dessous. Les objets w4, w5 et w6 appartiennent à un même concept et pourtant leurs descriptions n'ont pas de propriétés communes.

$D = \{ w \in U \text{ tels que } V1^{\,2} + V2^{\,2} = 1 \}$

	V1	V2	V3
w4	0	1	2
w5	1	0	7
w6	-1	0	3

Le raisonnement **iii** repose sur une deuxième hypothèse. Pour trouver des objets ayant des caractéristiques communes, il faut que l'information soit redondante.

Exemple III. 2 Redondance de l'information et ressemblance

Considérons les deux tableaux de données Td1 et Td2 construits sur l'ensemble des objets { w1, w2, w3, w4 } et les variables binaires V1, V2, V3. Dans Td1, l'information n'est pas redondante. Si l'on compare deux à deux les objets de Td1, on constate qu'ils ne possèdent jamais plus d'une caractéristique commune sur trois. Il n'existe pas de groupe d'objets ayant une forte ressemblance entre eux et peu de ressemblance avec le reste de l'ensemble. Par contre, les objets du tableau Td2, se répartissent naturellement en deux classes. A l'intérieur de ces classes, les objets ont deux propriétés en commun sur trois. D'une classe à l'autre, les objets ne possèdent qu'une propriété en commun sur trois. Dans Td2 l'information est redondante.

Td1	V1	V2	V3
w1	0	0	0
w2	0	1	1
w3	1	0	1
w4	1	1	0

Td2	V1	V2	V3	
w1	0	1	0	**C1**
w2	0	1	1	
w3	1	0	1	**C2**
w4	0	0	1	

En résumé, les OCA sont capables de « trouver » des concepts d'un domaine s'il est possible d'appliquer le raisonnement ci-dessous :

Si : **1** Il existe une relation simple entre les attributs descriptifs et l'intension des concepts du domaine.
2 L'information est redondante.

Alors : **iii'** Il est possible de trouver des groupes d'objets ayant des propriétés communes.
iii'' Ces groupes d'objets appartiennent à l'extension d'un même concept du domaine.

Les assertions 1 et 2 définissent ce que nous appellerons l'hypothèse de similarité. Si les données respectent cette hypothèse, il suffit de rassembler les objets qui ont des caractéristiques communes ou d'une manière plus générale, qui se ressemblent pour trouver des concepts du domaine. Dans les exemples précédents, nous avons uniquement considéré des classes monotéthiques. La notion de ressemblance permet d'assouplir cette contrainte. S'il existe une relation très directe entre les attributs descriptifs et les intensions des concepts du domaine, alors il a de fortes chances pour qu'apparaissent des classes monothétiques. Mais, si cette relation est moins directe, les classes polythétiques composées d'objets qui se ressemblent restent les plus susceptibles de matérialiser l'extension des concepts du domaine.

Les Outils de Classification Automatique ont pour objectif de rassembler les objets qui se ressemblent (cf. II.1.3). Sous réserve de l'hypothèse de similarité, ces outils peuvent donc déterminer des concepts d'un domaine. Cependant, la notion de ressemblance est extrêmement vague. Chaque système en réalise une interprétation différente. C'est pourquoi, nous dirons que les OCA peuvent déterminer des concepts d'un domaine, si les données respectent l'hypothèse de similarité au sens de l'outil considéré.

III.2.2 La validation des classes

L'utilisation qui est faite des méthodes de classification empiriques pour déterminer les concepts d'un domaine reposent sur des mécanismes d'induction. Le raisonnement **iii** utilise deux formes d'induction (Kodratoff 1991). C'est tout d'abord une abduction, car il est construit en inversant la déduction **ii** qui est une forme de modus ponens. C'est ensuite une généralisation car il passe d'un cas particulier à une généralité. Sur un ensemble fini d'exemples on constate que plusieurs propriétés sont vraies en même temps. Par exemple V1 = 1 et V2 = 3. On pose donc l'hypothèse que cette conjonction est caractéristique d'un concept du domaine et par généralisation, que l'association entre ce concept et cette conjonction est toujours vraie.

Le raisonnement inductif n'a pas valeur de démonstration. Il est donc nécessaire de valider les concepts. Les classes prennent sens par rapport à l'ensemble des concepts liés à l'activité considérée : la sémantique du domaine. Dans les domaines peu formalisés, la sémantique du domaine reste la propriété des experts. En fonction de l'objectif et de l'activité qu'ils entretiennent avec les données, ces derniers développent des connaissances, des savoir-faire et des stratégies personnelles. Ce sont ces connaissances a priori qui leur permettent de valider ou tout au moins d'interpréter une classification. Valider une classification, ne signifie pas forcément que les experts reconnaissent des catégories qui leur sont propres ; il s'agit d'analyser les classes, afin de déterminer si les contingences observées sont le reflet d'un lien de causalité et si ces liens de causalité sont pertinents par rapports à l'objectif de la classification.

III.3 Analyse des problèmes de validation

Nous partons du principe que la solution existe, puisque les approches manuelles permettent d'élaborer des classifications pertinentes. Si les experts du domaine ne valident pas la classification élaborée par un Outil de Classification Automatique, c'est donc que :

1. les classes ne correspondent pas à l'extension de concepts du domaine

ou

2. les classes ne correspondent pas à l'extension de concepts pertinents par rapport aux objectifs de la classification.

Dans le premier cas, on peut conclure que les données ne respectent pas l'hypothèse de similarité au sens de l'outil utilisé. Plus précisément, nous identifions deux facteurs qui rendent difficile l'interprétation des résultats d'une Classification Automatique.

1. Les acteurs disposent d'une connaissance sur le domaine qui n'est pas accessible au système. Les données sont donc incomplètes et ou non pertinentes.
2. Le système interprète les données d'une façon qui lui est propre, on parle alors de biais de classification. Les données ne sont pas sous une forme qui permet à l'outil de déterminer des concepts intéressants.

Si l'information est généralement redondante, ces deux facteurs font qu'il n'existe pas forcément de relation simple entre les attributs descriptifs et l'intension des concepts recherchés.

Dans le deuxième cas, les données respectent l'hypothèse de similarité mais ne sont pas pertinentes. Pour la suite de ce travail, nous considérerons le premier cas, plus général.

III.3.1 L'incomplétude des données

En Analyse Typologique, le rôle des experts se limite à la seule définition des données, c'est à dire, le choix des objets et des attributs descriptifs afin de construire un tableau de données. Les méthodes d'Analyse Typologique permettent principalement de trouver des régularités dans un grand tableau de nombres en dégageant des nuages de points homogènes (Napoli 1996). La réussite de cette démarche suppose que l'ensemble de l'information pertinente par rapport au problème considéré soit contenu dans le tableau de données. En dehors du choix des objets à classifier, nous avons vu en III.1 que les attributs sont pertinents s'ils entretiennent une relation simple avec les propriétés caractéristiques des concepts recherchés. La nature des connaissances expertes et la complexité de certains domaines rend cette tâche non triviale.

- Pour certaines catégories de problèmes ou d'activité, il est possible de définir les attributs pertinents une fois pour toutes. Cela suppose une certaine régularité dans l'activité considérée. Les attributs ne seront valables que pour des problèmes tout à fait similaires. Pour les activités complexes et peu formalisées, il n'existe pas par définition d'attributs pertinents prédéfinis. Par exemple dans le cadre de l'organisation industrielle, il existait des codes TG prédéfinis. Mais la diversité des organisations nécessite de prendre en compte des particularités propres à chaque entreprise. Ces codes ne permettaient pas à eux seul, de définir une classification pertinente.

- De par la nature procédurale de leurs connaissances (cf. II.1.3), il est souvent difficile pour les experts de formuler leur savoir-faire en terme d'attributs descriptifs, et plus encore d'identifier les attributs pertinents pour la classification. Généralement sur un domaine complexe et peu formalisé, l'information initiale fournie par les experts n'est pas suffisante pour résoudre complètement le problème de classification considéré.

De la même façon, et comme l'ensemble des Outils de Classification Automatique, les méthodes de Classification Conceptuelle, ne font que reformuler les données. Il est donc supposé que l'information initiale soit complète par rapport au problème considéré. Les problèmes 1, 2 et 3 (cf. III.1.2.1) illustrent bien la difficulté à obtenir des données pertinentes. Le problème 4 est typique d'un non-respect de l'hypothèse de similarité. Dire que la classification n'est pas cohérente avec les descripteurs signifie qu'un Outil de Classification Automatique ne serait pas capable de retrouver les classes des experts en regroupant les objets qui se ressemblent. Les descripteurs permettent effectivement de décrire les exemples, mais pas forcément d'en déduire des regroupements significatifs. Une partie de la connaissance qui sert à regrouper les exemples n'est pas dans les données et celles-ci ne respectent pas l'hypothèse de similarité.

III.3.2 Les biais de classification

Du point de vu utilisateur un Outil de Classification Automatique se présente comme une boîte noire, avec en entrée des données constituées de la liste des descriptions des objets à classer, en sortie une partition dont les classes sont constituées d'objets similaires. Pour passer de l'un à l'autre, chaque OCA repose sur un certain nombre d'hypothèses, qui sont généralement implicites aux yeux de l'utilisateur. Elles influencent le résultat de l'analyse et peuvent gêner l'interprétation des classes. Les biais de classification sont l'ensemble des hypothèses propres à un Outil de Classification Automatique qui influence le résultat. Nous les regroupons dans quatre catégories (pas complètements indépendantes) :

- La représentation des objets,
- Le choix d'une mesure de ressemblance,
- Le concept de mesure de ressemblance comme une fonction régulière des attributs,
- Les méthodes de formation des groupes.

III.3.2.1 La représentation des objets.

Les systèmes d'Analyse Typologique imposent de fortes contraintes sur la représentation des objets à classifier. Si les être humains utilisent spontanément des attributs pour caractériser les objets qu'ils manipulent, ceux-ci définissent rarement des données simples. Les objets sont généralement homogènes. Par contre, il est courant que les attributs ne soient pas réguliers. Ainsi, pour décrire des pièces mécaniques, le diamètre s'appliquera uniquement pour des objets cylindriques. Il est fréquent qu'un objet prenne plusieurs modalités pour le même attribut, par exemple, un objet ayant plusieurs couleurs. Les attributs peuvent aussi être structurés, lorsque les modalités peuvent avoir différents niveaux de généralité (cf. II.2.2.1). L'attribut « forme » comprendra par exemple les modalités, rectangle, carré et polygone. La description d'un objet fait souvent appel à plusieurs type de variables. Des variables métriques pour les dimensions, des variables nominales pour la forme ou la matière. La représentation des objets utilisée en Analyse Typologique, ne rend pas compte de la richesse des descriptions employées par les experts. La phase de codification nécessite un effort d'adaptation important de la part des experts et a tendance à appauvrir la sémantique du domaine (Kodratoff 1991a)

Par rapport aux méthodes d'Analyse Typologique, les outils de Classification Conceptuelle présentent certaines améliorations. L'expressivité des langages de description des objets, ainsi que la capacité à prendre en compte le maximum d'informations sur le domaine, réduisent l'effort de codification des experts et contribuent à mieux respecter la sémantique du domaine. Cela permet de prendre en compte plus d'information, mais ne résout pas directement le problème d'explicitation des connaissances procédurales ainsi que celui de

déterminer les informations pertinentes par rapport au problème de classification considéré. Par exemple, le système Cluster/S (cf. II.2.2.1.1) prend en compte un Réseau de Dépendance des Objectifs (RDO) qui permet de tenir compte explicitement des buts de la classification. Il faut cependant connaître a priori l'ensemble des attributs pertinents par rapport à l'objectif. C'est avant tout un moyen pour tenir compte d'un point de vue dans le processus de classification.

III.3.2.2 Le choix d'une mesure de ressemblance

En Analyse Typologique, le mécanisme de classification repose explicitement sur la donnée d'un indice de similarité. Il n'existe pas de mesure de similarité universelle. Il est bien spécifié dans la littérature qu'il y a « une distance par problème » (Chandon 1981) et que « l'utilisateur seul peut la définir » (Saporta 1990), mais les utilisateurs ne maîtrisent généralement pas suffisamment les techniques d'Analyse Typologique pour définir l'indice de similarité. Celui-ci est fixé par le logiciel utilisé ou bien par un spécialiste de la méthode. Les hypothèses implicitement faites sur les données et inhérentes au choix d'une mesure de similarité influencent directement le résultat. Un choix inapproprié peut rendre difficile l'interprétation des classes.

Exemple III. 3 Le problème du choix de l'indice de similarité

Considérons les quatre objets w1, w2, w3 et w4 décrits par les variables binaires de type présence absence V1 à V5 dans le tableau de données Td, ci à droite. Les indices de similarité adaptés à ces variables, sont construits en combinant les quatre quantités P, A, N et T.

Td	V1	V2	V3	V4	V5
w1	1	1	0	0	0
w2	1	1	0	1	1
w3	1	0	0	0	0
w4	0	1	1	0	1

- P : le nombre de co-présences
- A : le nombre de co-absences
- N : le nombre de non-coïncidences
- T : le nombre total d'attributs

	w1			w2			w3			w4		
	P	A	N	P	A	N	P	A	N	P	A	N
w1	2	3	0	2	1	2	1	3	1	1	1	3
w2				4	1	0	1	1	3	2	0	3
w3							1	4	0	0	1	4
w4										3	2	0

Selon que les quantités prises en compte, les indices de similarité donnent des résultats différents voir contradictoires. Par exemple, en appliquant l'indices de Russel et Rao sur les données précédentes, on trouve le tableau de similarité Ts1 ; en appliquant l'indice de Sokal et Michener, on obtient le tableau de similarité Ts2.

Indice de Russel et Rao : $is1 = \frac{P}{T}$

Indice de Sokal et Michener : $is2 = \frac{P + A}{T}$

Ts1	w1	w2	w3	w4
w1		40	20	20
w2			20	40
w3				0
w4				

Ts2	w1	w2	w3	w4
w1		60	80	40
w2			40	40
w3				20
w4				

En exploitant ces tableaux de similarité, un outil de Classification Automatique élémentaire qui regroupe les objets dont la similarité est supérieure à la similarité moyenne, donnera la partition P1 = { {w3}, {w1, w2, w4} } pour Ts1 et P2 = { {w4}, {w1, w2, w3} } pour Ts2. Ces résultats sont pertinents selon qu'il est nécessaire ou pas de tenir compte des co-absences pour évaluer la ressemblance de deux objets.

De nombreuses méthodes de Classification Conceptuelles reposent sur une mesure de ressemblance, mais celle-ci n'est pas forcément explicite. C'est notamment le cas de Cobweb (et de ses descendants) qui optimise une mesure de similarité statistique (Fisher 1987) (cf. II.2.2.2). Ou du système KBG, basé sur une mesure de similarité numérique. Cluster, quant à lui, regroupe les objets qui appartiennent à une même intention Ce qui revient à comparer les objets sur l'ensemble des intensions qui les admettent comme membre de leur extension. Cette façon de faire est assimilable à une mesure de ressemblance complexe.

III.3.2.3 La mesure de ressemblance est une fonction régulière des attributs

L'essentiel du problème de la classification, revient souvent à formuler l'objectif en termes d'attributs pertinents. Encore faut-il que ces derniers soient exploitables par les outils de Classification Automatique. Les systèmes de classification numérique exploitent les données en appliquant le principe fondamental de l'Analyse Typologique : comparer les objets sur un ensemble de critères d'égale importance ou tout au moins d'une importance relative constante. C'est à dire que tous les objets sont comparés sur tous attributs, même s'il est possible de définir une pondération, il est nécessaire qu'elle soit constante sur l'ensemble des objets. C'est sur la base de cette hypothèse qu'il est possible d'utiliser des indices de similarité qui soient une fonction régulière des attributs.

A contrario, les mécanismes de catégorisation de l'expert sont plutôt basés sur l'utilisation d'un petit nombre de critères variant selon les situations (Wang 1995). Généralement, les attributs donnés par les experts ne sont pas d'importance égale, mais implicitement hiérarchisés en fonction du contexte. Le code conçu par les experts admet un « mode d'emploi » différent de celui qui est présupposé par toutes les méthodes d'Analyse Typologique.

Exemple III. 4 Utilisation des attributs par les experts

Cet exemple a été conçu uniquement pour illustrer les limites de la modélisation d'une mesure de ressemblance à l'aide d'une fonction régulière des attributs. Il est cependant représentatif de cas réels. Notamment celui développé au chapitre V.

Le tableau de données Td ci dessous, présentent la description de 21 objets décris par les variables L/D, FE, FI, US et FO. A l'issue d'une classification par une méthode d'Analyse Typologique, les objets sont répartis dans six classes qui forment une partition P. La dernière colonne du tableau indique la classe des objets. Par exemple, les 2 et 3 sont rangés dans la classe n°1.

Td	L/D	FE	FI	US	FO	P
1	2	2	1	0	1	0
2	0	0	4	4	2	1
3	0	4	4	6	2	1
4	1	1	1	6	0	2
5	1	0	1	6	0	2
6	0	7	1	0	0	4
7	0	7	1	1	3	4
8	0	7	1	1	4	4
9	0	7	1	1	0	4
10	0	7	1	1	9	4
11	2	4	5	1	4	5
12	2	4	5	1	2	5
13	2	4	5	4	4	5
14	2	5	5	4	4	5
15	0	7	2	1	6	4
16	0	4	1	1	6	4
17	0	0	1	1	6	4
18	1	1	7	0	0	3
19	1	1	1	0	0	3
20	2	2	1	3	1	0
21	1	5	4	6	0	2

Considérons que lorsque l'on présente aux experts du domaine, la partition P obtenue par Analyse Typologique du tableau de données Td, ils énoncent les règles de classification répertoriées dans la base B ci-dessous.

Base de règle B :

R0 : Les pièces ayant un L/D=0 ne doivent pas cohabiter avec les pièces ayant L/D=2.

R1 : Les produits ayant L/D=0 et FE=7 et FO différent de 6 ne doivent pas cohabiter avec les autres produits.

R2 : Les pièces ayant L/D=0 et FO=6 définissent une famille.

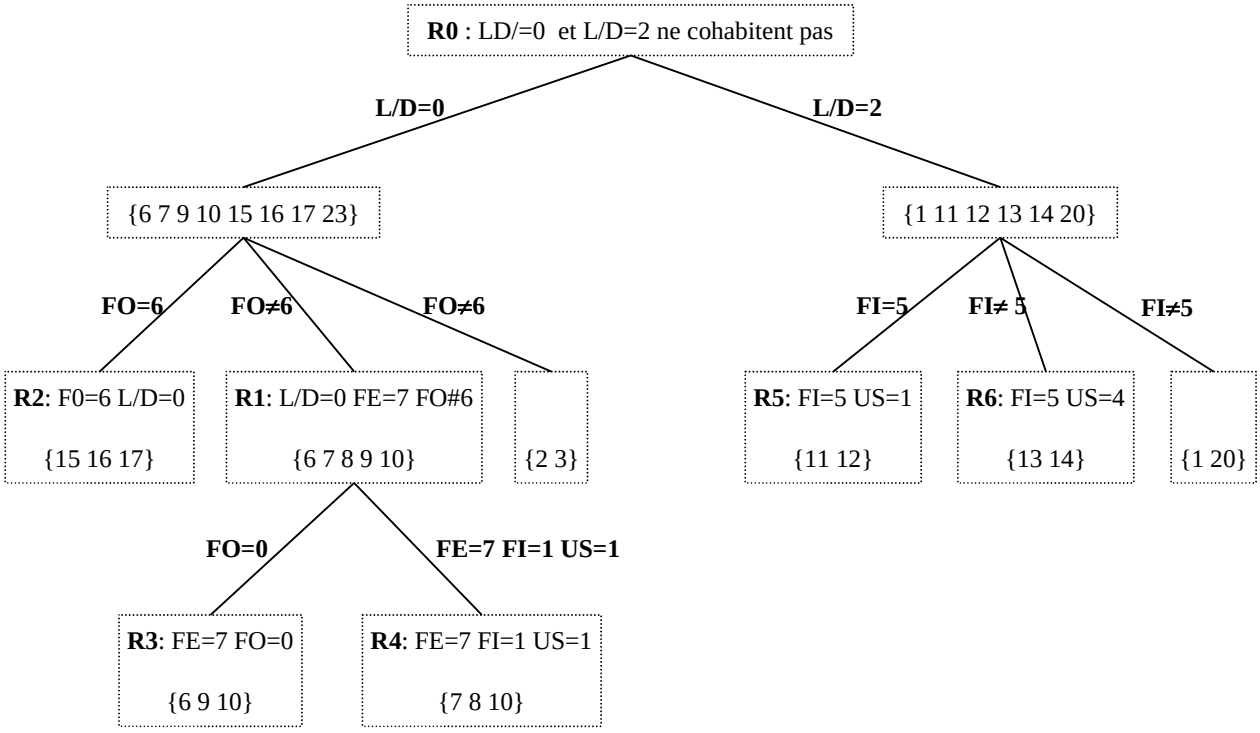
R3 : Les pièces ayant FE=7 et FO=0 appartiennent à une même famille.

R4 : Les pièces ayant FE=7, FI=1 etUS=1 appartiennent à une même famille.

R5 : Les pièces ayant FI=5 et US=1 appartiennent à la même famille.

R6 : Les pièces ayant FI=5 et US=4 appartiennent à la même famille.

Afin de mettre en évidence l'aspect contextuel du rôle des modalités par rapport au regroupement des objets, nous représentons cette base de règles par un arbre, dont les nœuds correspondent aux groupes d'objets désignés par les classes. Cet arbre ne définit pas une hiérarchie de l'ensemble des objets. Par exemple, la règle R0 divise les objets en deux groupes qui ne doivent pas cohabiter : les objets tels que L/D=0 et L/D=2. Cette règle ne donne aucune information sur les objets tels que L/D=1.



Cette base de règles illustre la façon dont les experts peuvent utiliser les attributs. En tant que critères de classification, ils varient en fonction de la valeur des autres variables. Ces connaissances supplémentaires montrent qu'une partie de la connaissance du domaine nécessaire à l'élaboration des classes n'était pas dans les données initiales. C'est à dire qu'il est très difficile voire impossible de déterminer des regroupements pertinents sur la base des régularités. En d'autres termes, les données ne respectent pas l'hypothèse de similarité au sens de la méthode d'Analyse Typologique utilisée.

III.3.2.4 Hypothèses de formation des groupes

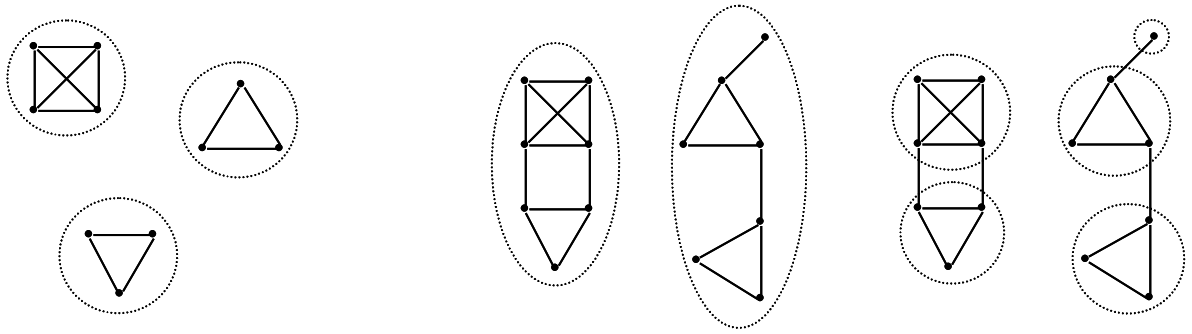
L'étape de formation des groupes peut être vue comme un problème d'optimisation. Chaque méthode d'Analyse Typologique utilise un critère particulier qui est une interprétation des notions d'homogénéité et d'isolation. La taille de l'espace de recherche interdit toute exploration exhaustive. Les heuristiques utilisées nécessitent souvent l'utilisation d'hypothèses supplémentaires (nombre de classes, définition d'une distance entre deux groupes). Elles peuvent favoriser l'homogénéité ou l'isolation. Elles peuvent encore déterminer la forme des groupes que la méthode peut reconnaître dans l'espace de

représentation des données. Par exemple, l'utilisation du concept de distance à un centre de classe, fait l'hypothèse de nuages hyper-sphériques ou hyper-elliptiques. La recherche de groupes isolés et homogènes n'admet pas forcément de solution satisfaisante. Si les groupes ne sont pas clairement définis par la notion de ressemblance, il ne sera pas possible de satisfaire ces deux conditions. Les méthodes d'Analyse Typologique, sont capables de trouver une structure dans les données alors qu'il n'existe pas réellement de classes « naturelles ». La structure trouvée est la meilleure au sens du critère employé et les classes n'admettent pas forcément d'interprétation. Les hypothèses utilisées pour former les groupes biaisent le résultat. Elles ont tendance à modifier la sémantique du domaine par l'utilisation de facteurs symboliques implicites (Kodratoff 1986).

Exemple III. 5 Dualité homogénéité et isolation

Dans l'exemple ci-dessous, les objets sont représentés par des points. Deux objets sont considérés comme similaires si et seulement si ils sont reliés par un arc. Les frontières des classes sont en traits pointillés.

Cas idéal **Cas difficiles**



Partition P1
classe homogènes
et isolées

Partition P2.1
classe isolées
mais pas homogènes

Partition P2.2
classe homogènes
mais pas isolées

Dans le cas idéal, les groupes sont clairement définis. La partition P1 remplit les conditions d'homogénéité et d'isolation. Dans les cas difficiles, la ressemblance entre objet ne permet pas forcément de trouver une partition satisfaisant ces deux conditions. Dans la partition P2.1, les classes sont isolées, car les objets de classes différentes ne se ressemblent pas. En revanche, les classes ne sont pas homogènes, car elles contiennent des objets qui ne se ressemblent pas. inversement, les classes de la partition P2.2, sont homogènes mais pas isolées.

En Classification Conceptuelle, le processus de catégorisation est aussi traité comme un problème d'optimisation. On retrouve donc les biais liés au choix d'un critère à optimiser et aux heuristiques employés pour explorer l'espace de recherche.

Par exemple, les critères qui optimisent la capacité de prédiction, peuvent conduire à des hiérarchies qui ne reflètent pas la structure sous-jacente des données (MacKusick 1991). A ce propos, Fischer insiste sur le fait que la majorité des systèmes de Regroupement Conceptuel « *ne répondent finalement pas à la problématique générale du Regroupement Conceptuel qui est de trouver des regroupements pour structurer les données* » (Fischer 1996).

Les heuristiques utilisées posent le problème des minima locaux ainsi que celui de leur interprétation en termes de conséquence sur le résultat de la classification. Ce problème est résolu de façon élégante dans le système KBG (Bisson 1993). Le programme utilise un paramètre de contrôle de l'apprentissage qui présente la double qualité d'être explicite et d'avoir une sémantique claire. Il s'agit de la distance maximale entre deux exemples regroupés. Plus elle est courte, plus les concepts sont spécifiques.

En théorie, « les méthodes de Classification Conceptuelle regroupent des objets, non seulement parce qu'ils sont proches au sens d'une certaine distance, mais aussi parce que considérés en groupe ils matérialisent l'extension d'un concept » (Haton 1986). Plus concrètement, il apparaît que la description recherchée pour les classes influe sur le processus de catégorisation. Fixer a priori la description des classes recherchées est un biais de classification. De plus, il ne suffit pas de rechercher des classes que l'on puisse décrire pour trouver des classes significatives. Encore faut-il que les propriétés utilisées soient suffisamment pertinentes pour permettre de décrire des classes intéressantes (cf. problème n°4 chapitre II.2.2.1) .

.

III.3.2.5 Conclusion

Sur un domaine complexe et peu formalisé, les données initiales respectent rarement l'hypothèse de similarité au sens de l'outil considéré. Plus précisément, les problèmes de validation des classes ont donc deux origines :

1 Une partie importante des connaissances qui servent à définir les classes n'est pas dans les données. Les experts ne sont pas toujours capables de formaliser a priori leurs connaissances en termes d'attributs pertinents par rapport à l'objectif de la classification.

2 Les classes obtenues sont le reflet de l'information contenue dans les données et des biais de classification propres à chaque OCA.

D'une manière générale, les outils d'Analyse Typologiques sont des systèmes formels à sémantique numérique. On code les objets par des vecteurs et on définit les opérations de comparaison par une formule mathématique globale. La méthode est efficace, mais en contrepartie il apparaît d'une part que la codification des données a tendance à appauvrir la sémantique du domaine (Kodratoff 1991) et à la modifier par l'utilisation de facteurs symboliques implicites (Kodratoff 1986) liés aux biais de classification. D'autre part, le système de classification et les experts n'utilisent pas les attributs de la même façon.

En Classification Conceptuelle, la volonté de rester proche du langage de l'utilisateur tant dans la description des données que dans celle des classes, réduit les efforts de formalisation de la part des utilisateurs, diminue les risques de déformation de la sémantique du domaine et contribue à résoudre les problèmes de validation. Cependant, les biais sont rarement explicites et décrire les biais de chaque programme d'apprentissage est une partie intégrante du programme de recherche en Apprentissage Symbolique Automatique (Kodratoff 1997). C'est une première étape car il nous semble que l'objectif à terme sera d'interpréter les biais en terme de conséquence sur la classification obtenue, et de développer une expertise qui permette de diagnostiquer et de corriger les problèmes d'interprétation des classes en fonction des biais utilisés

III.3.3 Le cycle de classification

Les problèmes de validation semblent admettre une réponse simple : pour obtenir des classes significatives, il suffit d'expliciter les connaissances expertes pertinentes par rapport à l'objectif, puis, de mettre ces connaissances sous une forme qui tienne compte des contraintes propres aux systèmes de Classification Automatique Utilisé (le format des informations et les biais de classification).

III.3.3.1 Le dialogue entre les experts du domaine et le spécialiste

Les OCA ne sont pas utilisables sans une bonne connaissance de leur fonctionnement et nécessitent la présence d'un spécialiste (appelé Analyste en Analyse de Données). En pratique, l'interaction entre les experts et le spécialiste prend le plus souvent la forme d'un dialogue (cf. III.1). Sur la base d'une information de départ, le spécialiste propose une classification initiale, que les experts valideront ou critiqueront. En tenant compte de ces remarques, le spécialiste et les experts progresseront par essais erreurs, jusqu'à ce qu'émerge une solution satisfaisante. Nous appellerons ce processus le cycle de classification en Classification Automatique (fig. III.2).

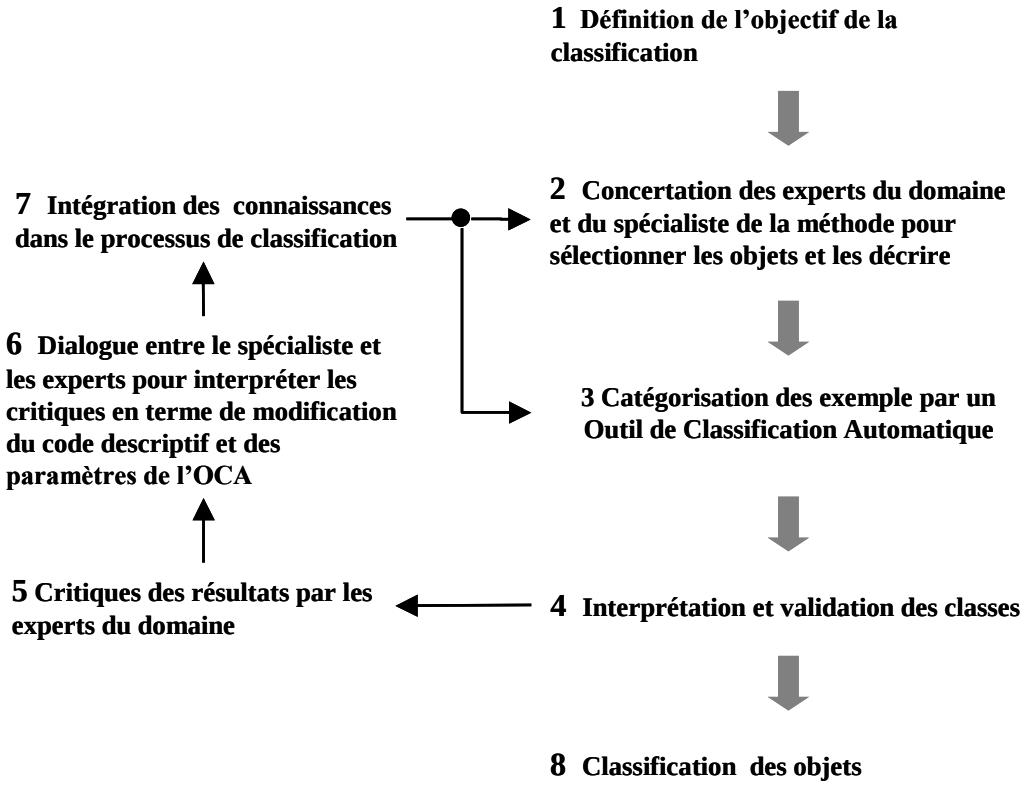


Fig. III.2 Le cycle de classification en Classification Automatique

III.3.3.2 Les deux processus du cycle de classification

Le cycle de création de classes significatives repose sur deux processus distincts : d'une part, les experts du domaine explicitent et structurent leurs connaissances ; d'autre part, ils intègrent les biais du système tandis que le spécialiste en Classification Automatique apprend une partie des connaissances du domaine.

Lors de l'étape (5) du cycle de classification, les experts du domaine formulent des connaissances supplémentaires. Cette formalisation tient plus d'une tentative que d'une certitude inébranlable. Ces connaissances sont verbalisées, puis confrontées les unes aux autres, lors de l'étape (6). C'est le processus d'explicitation et de mise en cohérence des connaissances sous-jacentes aux classes qui est difficile. Chaque expert est amené à formuler ses connaissances sous forme de critères de classification. Tout le problème réside dans l'adéquation des critères explicites avec les critères implicites de l'expert. C'est presque le même travail que de rédiger une idée confuse. Il faut la structurer pour l'écrire et il faut l'écrire pour la structurer. En observant le processus de classification, on constate que les experts de l'entreprise affinent leurs connaissances par essais erreurs jusqu'à obtenir un résultat satisfaisant. L'ambivalence du processus de structuration implique l'incrémentalité. Pour " tester " les connaissances, il est nécessaire de faire des classifications successives et de répéter le cycle (2, 3, 4, 5, 6,7).

Le processus d'apprentissage respectif tient d'un équilibre qui s'établit entre l'expert, l'outil et le spécialiste. D'un côté, l'expert s'adapte à l'outil, ses contraintes et ses biais. Il faut du temps aux experts pour expliciter le mode d'emploi de leurs attributs et pour comprendre comment le système les utilise. De l'autre côté, le spécialiste s'adapte au domaine. Il lui faut prendre un minimum contact avec le domaine, ne serait-ce que pour comprendre les explications de l'expert et les traduire en terme de modification dans les données ou les paramètres du système.

Au cours du cycle de classification, on modifie généralement peu l'algorithme de formation des groupes, l'essentiel du travail revient à modifier les données pour que les groupes puissent apparaître. Il s'agit de :

1 Compléter les données, c'est à dire, trouver les objets et les attributs pertinents par rapport au problème de classification considéré.

2 Formuler les données de façon à ce qu'elles respectent l'hypothèse de similarité au sens de l'outil utilisé.

III.3.3.3 Difficultés du cycle de classification

Le cycle de classification est un processus complexe. Le spécialiste essaie de définir une mesure de similarité et un code descriptif, afin de classer les objets d'un domaine qu'il ne connaît pas. Pour cela, il va devoir utiliser les informations données par les experts. Le dialogue entre le spécialiste et les experts présente donc une double difficulté.

D'une part, le spécialiste, étranger au domaine, travaille avec des informations partielles, parfois même incohérentes et contradictoires. L'ajustement du processus d'Analyse Typologique à partir des remarques expertes est uniquement fonction de l'expérience de

l'analyste. Il n'existe pas à notre connaissance de méthodologie d'intégration des connaissances expertes dans un processus d'Analyse Typologique. En Acquisition de Concepts et en Classification Conceptuelle, la richesse des langages de représentation des connaissances utilisés facilitent le déroulement de ce cycle et simplifient le dialogue entre les experts et le spécialiste. Néanmoins, l'intégration des connaissances supplémentaires dans le processus de classification reste délicate. Ce problème a donné lieux à un certains de travaux (cf. III.2.2.1 et II.5.2 (les approches de type boîte noire)). Cependant, il reste difficile de modifier localement une classification. Le travail consiste principalement à changer la représentation des objets. Les règles d'inférence sont à cette fin d'une grande utilité, mais restent d'un emploi limité. Par exemple, elles ne permettent pas de représenter des Règles de Classification (cf. I.3.1), car ces dernières ne modifient pas la représentation des objets, mais la façon de les comparer.

D'autre part, si les experts de l'entreprise ne sont souvent pas à même d'énoncer leurs connaissances, il leur est encore plus difficile de le faire en tenant compte des contraintes propres aux systèmes de Classification Automatique. Ces contraintes imposent de modifier le code descriptif initial d'une façon qui peut sembler artificielle aux yeux des experts de l'entreprise. Si l'on considère les bases de règles des exemples I.3 et III.4, il est impossible d'en rendre compte à l'aide d'une pondération des variables. Il faudra donc reformuler les attributs descriptifs de façon à ce que l'outil de classification puisse trouver des groupes qui respectent les contraintes énoncées dans la base de règle. Ce travail est particulièrement laborieux. A priori, il existe deux façons de résoudre ce problème : en aidant les experts à comprendre l'OCA (c'est l'approche de type boîte en verre cf.III.5.2) ou bien en aidant le système à comprendre les experts (c'est notre approche).

Une solution pour réduire la durée du cycle de classification consiste à prendre en charge l'aspect cyclique du processus de classification et à élaborer des outils qui améliorent le dialogue entre les experts et le spécialiste. C'est le principe des approches interactives.

III.4 Les approches interactives

Les termes « interactif » et « interactivité » sont apparus dans les années 80 (Dictionnaire 1986). Ces concepts ont principalement été utilisés dans le contexte de la communication homme-machine. De ce point de vue et dans le domaine de l'aide à la décision, Lévine et Pomerol (Lévine 1989) en proposent la définition suivante : Considérant « un système de résolution de problème pratiquant la recherche heuristique » au sein d'un espace d'états, le système « est interactif si tout ou une partie du contrôle de la recherche est laissé à l'opérateur ». Le recours à l'interactivité dans une procédure peut se justifier par une déficience des procédures classiques existantes ou un apport spécifique lié à l'intervention humaine. Une procédure interactive se présente comme une procédure itérative fondée sur l'alternance (Vanderpooten 1990) d'une phase de calcul ayant pour but de construire des propositions avec une phase de dialogue visant à faire réagir le décideur relativement à la proposition courante et à obtenir ainsi de l'information pour orienter la construction d'une nouvelle proposition. Pour réaliser une procédure interactive, il ne suffit pas de mettre des outils d'investigation à la disposition du décideur. Il faut également aider l'utilisateur à s'en servir en organisant et guidant l'interaction. Il s'agit de définir un **protocole d'interaction** qui régit les phases de dialogue et la manière dont elles s'enchaînent avec les phases de calcul.

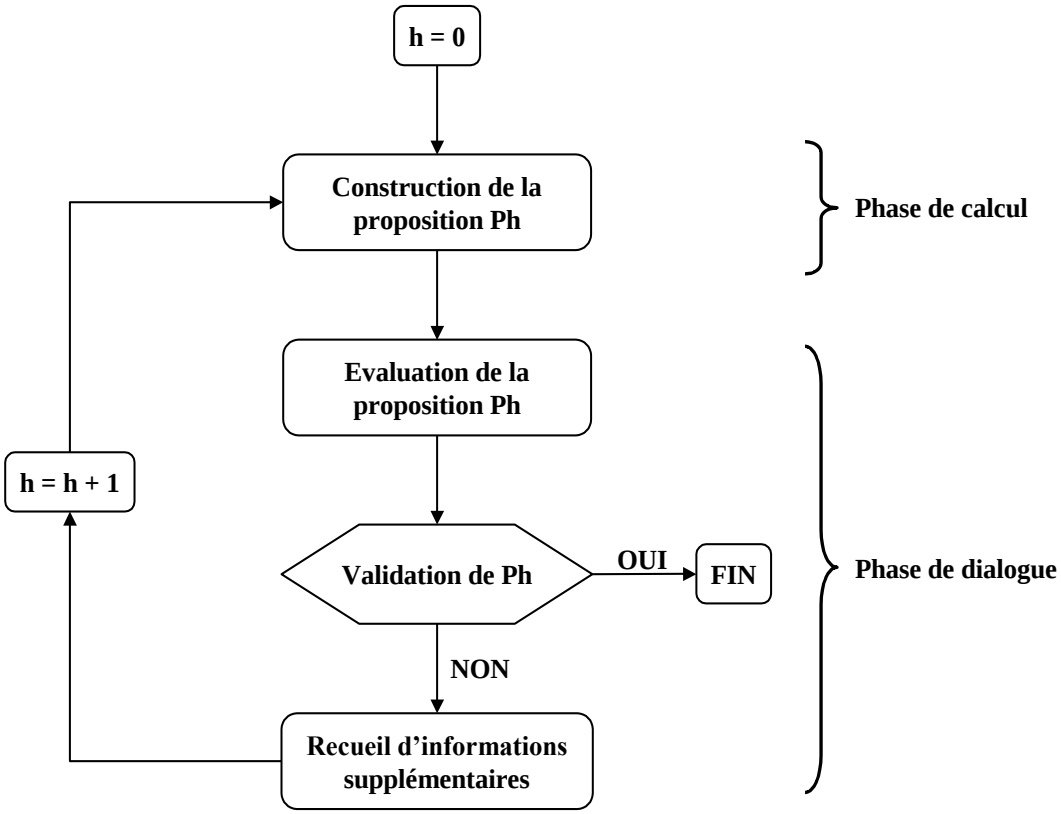


Fig. III.3 Schéma d'une procédure interactive

Dans le contexte de la classification, la phase de calcul correspond à la construction d'une classification à partir des informations recueillies. La phase de dialogue comprend la validation et la critique de la classification obtenue par les experts du domaine. Définir un protocole d'interaction, c'est proposer un environnement qui enchaîne les phases de calcul et de dialogue et guide les experts du domaine pour corriger la classification obtenue. Nous distinguerons les méthodes selon la stratégie employée pour aider l'utilisateur à corriger la classification.

Corriger directement le résultat. C'est la méthode la plus simple. Il faut d'abord classifier les éléments, corriger la classification jusqu'à ce les classes soient validées, puis déterminer une description des classes. Cette méthode devient extrêmement laborieuse voir impossible à appliquer dès que le nombre d'objets dépasse quelques centaines. Elle n'offre aucun cadre de référence aux experts pour structurer leurs connaissances, si ce n'est la structure recherchée.

Les méthodes de type boîte noire, consistent à disposer d'une expérience qui permet en fonction des erreurs constatées dans la classification d'en diagnostiquer les origines au niveau des paramètres de l'Outil de Construction Automatique de classification ou bien au niveau des données. Des tentatives dans le sens d'une meilleure formalisation des « recettes » de mise au point ont été faites dans le cadre du projet Machine Learning Toolbox (Craw 1992). Le système MLT possède un système expert en apprentissage qui conseille des modifications possibles des paramètres d'apprentissage en fonction des erreurs observées par les utilisateurs. Ce système fonctionne dans un contexte supervisé et ne traite pas la sélection et la représentation des données.

Les méthodes de type boîte en verre (Bisson 1993; Nedellec 1995 ; Bisson 1997; Gabriel 1999) vont guider la démarche de correction des erreurs constatées par les experts. Le système doit pouvoir expliquer ce qui justifie dans les données et le fonctionnement du système le résultat obtenu. Il s'agit de montrer comment le système a trouvé un résultat, de façon à ce que les liens entre les entrées et les sorties du système soient les plus claires possibles. Si l'expert comprend comment le système utilise les données initiales, il prendra plus facilement conscience de l'absence de descripteurs importants ou de la présence de descripteurs non pertinents. On obtient ainsi la base d'un dialogue qui va aider à la définition des attributs descriptifs des objets.

III.5 Conclusion

Dans le but de résoudre un problème d'Affinement de Connaissance par Classification Interactive, nous essayons d'employer un Outil de Classification Automatique.

Ces outils structurent les données sur la base de l'hypothèse de similarité. En l'absence d'informations complémentaires cette hypothèse reste somme toute la plus raisonnable. Si dans les domaines simples il est concevable que les données initialement fournies à l'outil de catégorisation soient suffisamment complètes et justes pour que celui-ci découvre par ses « propre facultés d'analyse » une solution satisfaisante, ce présupposé ne tient plus dès lors que l'on travaille dans des domaines peu formalisés où l'essentiel de la connaissance disponible se présente sous la forme de savoir-faire. Après l'analyse des problèmes de validation rencontrés en Analyse Typologique et en Classification Conceptuelle, pour ce type de contexte, il apparaît que les données initiales respectent rarement l'hypothèse de similarité. Les causes majeures de ce constat sont :

1 l'incomplétude des données initiales due à la nature procédurale des savoir-faire ;
2 les biais de classification qui déforment la sémantique du domaine.

Les données sont incomplètes, car par définition les experts ont du mal à énoncer les connaissances relatives à leur domaine d'activité. Les attributs ne reflètent pas nécessairement d'une manière directe les concepts manipulés par les experts ou bien des concepts pertinents par rapport à l'objectif de la classification.

Les Outils de Classification Automatique traitent le processus de catégorisation comme un problème d'optimisation. Ils reposent sur un ou plusieurs langages de représentation des connaissances disponibles sur le domaine, un critère d'évaluation de la qualité d'une classification et des heuristiques d'exploration de l'espace de recherche. Les langages de représentation des connaissances nécessitent un effort de codification plus ou moins important de la part des utilisateurs et ne lui permettent pas forcément d'exprimer ses connaissances. Le critère mesure l'adéquation entre les classes et les ressemblances observées dans la description des objets. Le concept de ressemblance est extrêmement vague et difficile à opérationaliser. Il n'existe pas de critère universel d'évaluation de la qualité d'une classification et les critères varient d'une méthode à l'autre. D'un point de vue algorithme, la taille des espaces de recherche nécessite d'utiliser des heuristiques et de se contenter d'un optimum local. Le résultat est purement syntaxique. La classification obtenue est le reflet de l'information initiale et des biais de classifications employés.

Généralement, la classification obtenue avec les données initiales sert de point de départ à un cycle de classification (cf. III.3). Si les experts du domaine ne valident pas la classification lors de l'étape (4), l'interaction entre le spécialiste et les experts du domaine prend la forme d'un dialogue. Le spécialiste de la méthode est nécessaire, car les Outils de Classification

Automatique sont généralement complexes et nécessitent une expertise particulière pour être utilisés. Lors de l'étape (5) les experts formulent des connaissances supplémentaires destinées à corriger le résultat obtenu. En tenant compte de ces remarques, le spécialiste et les experts progressent par essais erreurs, jusqu'à ce qu'émerge une solution satisfaisante. Le cycle d'élaboration de classes significatives repose sur deux processus distincts :

1. La structuration des connaissances du domaine pour les experts, au niveau individuel et collectif ;
2. L'apprentissage ou l'intégration des biais du système par les experts du domaine et l'apprentissage par le spécialiste d'une partie de connaissances du domaine.

Ces deux processus permettent de raffiner les données afin qu'elles respectent l'hypothèse de similarité au sens de l'outil de classification utilisé. La durée du cycle de classification est directement dépendante de ces deux processus. La structuration des connaissances est un processus inhérent à la nature experte des connaissances. A chaque étape, les modifications proposées sont plus une tentative que l'expression d'une certitude. C'est une démarche d'exploration nécessairement itérative.

Les Outils de Classification Automatique sont par définition indépendants d'un domaine d'application particulier. Mais leur caractère général fait qu'il est souvent difficile de les exploiter directement et ce particulièrement sur un domaine complexe et peu formalisé. Pour obtenir des résultats, il est nécessaire de déterminer les attributs pertinents et les paramètres de l'OCA. C'est en quelque sorte une étape de réglage. Une fois « réglés », les Outils de Classification Automatique s'avèrent rapides et particulièrement efficace pour des traitements répétitifs et une application précise. Par contre l'étape de « réglage » s'avère souvent difficile. Dans le cadre de la TG elle est aussi longue voir plus longue que les méthodes de classification « manuelles ».

Le processus de « réglage » est par nature itératif et définit le problème d'Affinement de Connaissances par Classification Interactive. Au cours du cycle de classification, ce sont les étapes (6) et (7) d'analyse et d'interprétation des connaissances supplémentaires qui sont les plus délicates. Le dialogue est difficile car il met en jeu des experts de domaine différents. D'un côté, les experts du domaine essayent de formuler leur savoir-faire pour résoudre un problème de classification déjà difficile, mais, de plus, ils doivent tenir compte des contraintes liées à un outil de classification qu'ils ne maîtrisent pas. De l'autre côté, le spécialiste de l'outil, doit exploiter des connaissances sur un domaine qui lui est étranger, pour adapter son outil au problème considéré. Ce travail est généralement long et entièrement à la charge du spécialiste de l'outil utilisé. Une solution pour réduire la durée de la phase de « réglage » consiste à prendre en charge l'aspect cyclique du processus de classification et à élaborer des outils qui améliorent le dialogue entre les experts et le spécialiste. C'est le principe des approches interactives.

Les approches interactives prennent en charge le cycle de classification et proposent des protocoles d'interaction pour faciliter le dialogue entre les experts du domaine et le système. La présence du spécialiste de l'outil de catégorisation est toujours nécessaire, mais cette voie de recherche contribue à rendre les experts autonomes avec leurs données. On peut imaginer qu'à long terme, il ne faudra plus être un spécialiste de la classification pour classifier.

Il n'existe pas, à notre connaissance, de contribution spécifique dans la catégorie des méthodes qui corrigent le résultat. Les contributions existantes font partie des méthodes boîte noire ou boîte en verre. Ces méthodes consistent à permettre à l'expert de reformuler ses connaissances et ses données en fonction de l'outil de classification. D'une manière générale, il s'agit de formuler les données de façon à ce qu'elles respectent l'hypothèse de similarité au sens du système utilisé. L'approche oblige les experts à reformuler leurs connaissances sur le domaine essentiellement sous formes d'attributs descriptifs des objets tels qu'il devient possible d'en déduire les classes par des opérations de comparaison simple. Ces méthodes présentent l'intérêt de fournir un cadre cohérent pour aider l'expert à structurer ses connaissances. Les approches de type boîtes en verre tentent de minimiser la charge de travail liée à l'acquisition des principes de l'outil par les experts. Cependant, l'approche reste orienté processus. Ce sont les hommes qui doivent s'adapter à l'outil. Nous présentons au chapitre suivant les bases théoriques d'une approche qui a pour vocation de s'adapter aux experts en intégrant dans le processus de classification les connaissances supplémentaires qu'ils sont à même de formuler pour critiquer une partition.

Chapitre IV

Définition d'un système de classification automatique interactif

Introduction

Dans le chapitre précédent nous avons défini le cycle de classification inhérent à l'utilisation d'un Outil de Classification Automatique dans le cadre d'un problème d'Affinement de Connaissances par Classification Interactive (cf. figure III.2). Pour accélérer le processus de classification, nous proposons d'utiliser une approche interactive. L'efficacité d'une méthode de classification interactive dépend des outils utilisés pour réaliser les étapes (6) et (7) d'analyse et d'intégration des connaissances supplémentaires dans le cycle de classification. Dans ce chapitre, nous présentons une méthode de classification interactive destinée à accélérer le cycle de classification en TG. Cette approche présente les particularités suivantes :

- l'outil de classification automatique est un outil traditionnel d'Analyse Typologique ;
- les connaissances supplémentaires émises au cours de l'étape 4 de validation sont des règles de classification (cf. chapitre I.3).

Dans la partie IV.1, nous présentons le principe de fonctionnement de notre approche. Dans la partie IV.2, sur la base d'un corpus de connaissances supplémentaires recueillies lors de l'étape de validation et thésaurisées par les chercheurs du L.R.P.S, nous développons un langage de représentation des connaissances expertes. Les connaissances ainsi formalisées constituent la base de règles de l'étape 5 du cycle de classification. Les parties IV.3 et IV.4 présentent les concepts et les méthodes développés pour réaliser l'étape (6) d'analyse et l'étape (7) d'intégration dans un logiciel d'Analyse Typologique, des connaissances supplémentaires formalisées.

IV.1 Un système interactif de classification automatique sous contraintes symboliques

Comme nous l'avons montré dans le chapitre précédent, l'utilisation des outils de construction automatique de classification pour l'extraction de connaissances à partir de données sur les domaines complexes et peu formalisés, implique l'existence d'un cycle de classification. Ce cycle permet d'adapter la méthode au contexte et à l'objectif de la classification. Une solution pour réduire la durée de cette phase de « réglage » consiste à prendre en charge l'aspect cyclique du processus et à élaborer des outils qui améliorent le dialogue entre les experts et le spécialiste. C'est le principe des approches interactives.

Le cycle de classification est la conséquence de deux processus distincts : la structuration des connaissances des experts et l'intégration des biais de classification de l'outil utilisé. Les approches interactives existantes de type boîte noire et boîte en verre, améliorent essentiellement le processus d'intégration des biais en aidant l'utilisateur à comprendre comment le système utilise les données. Pour modifier une classification, l'utilisateur devra modifier les données ou les paramètres de l'outil.

Exemple IV. 1 Les approches interactives qui corrigent les données

Imaginons que pour un projet de formation d'îlots, les experts du domaine évaluent que deux machines X et Y séparées par le système de classification utilisé, devraient appartenir à la même classe. Les approches de type boîte noire ou boîte en verre vont permettre d'expliquer à l'expert pourquoi les objets sont rassemblés et ce qu'il faut faire pour les séparer. Dans l'hypothèse d'un système très simple qui rassemble les objets s'ils possèdent un nombre minimum S de ressemblances, le dialogue entre les utilisateurs et le système devrait ressembler à ceci :

utilisateur :	pourquoi les système rassemble-t-il les objets X et Y
système :	parce que l'objet X et l'objet Y présente plus de S ressemblances
utilisateur :	comment faire pour séparer l'objet X et l'objet Y
système :	il faut soit : - modifier les modalités des attributs de l'objet X ou de l'objet Y - définir de nouveaux attributs qui permettent de mieux différencier X de Y - éliminer des attributs qui prennent la même valeur pour X et Y - augmenter le seuil S d'agrégation des objets à la valeur S'

Dans le cadre des problèmes d'implantation de la TG, la majeure partie des connaissances supplémentaires est difficile à formuler en terme d'attributs descriptifs ou en terme de modification des paramètres de l'outil de classification (cf. chapitre I.3). Par contre, nous constatons une certaine régularité dans la structure de ces connaissances (cf. chapitre I.3.1). C'est pourquoi, afin d'accélérer le cycle de classification en TG, nous proposons non pas d'apprendre à l'expert comment modifier ses données en fonction du système, mais de systématiser l'étape n°6 d'interprétation des connaissances supplémentaires. Sur la base d'un

corpus de règles de classification représentatif des connaissances supplémentaires formulées par les experts du domaine dans le cadre de la TG, nous avons développé un langage formel de représentation des connaissances. Ce langage très proche de la formulation spontanément utilisée par les experts nécessite un effort d'apprentissage minimum. Il est à l'origine du protocole d'interaction ci-après (fig. IV.1). Tout d'abord, une phase d'initialisation détermine une partition initiale de l'ensemble des données (1), (2), (2.5), (3), (4) et (8). Les étapes sont numérotés de façon à pouvoir faire le rapprochement avec les étapes du cycle de classification présenté dans la figure III.2. Les experts examinent les familles (4) et formulent un ensemble de règles de classification sur la façon de regrouper les objets. Ils les transcrivent dans le langage proposé, définissant ainsi une base de règle (5). La base de règles est analysée par un programme spécialisé (6). Les éventuelles incohérences sont levées par recours direct aux experts. Les règles sont interprétées (7) en terme de modification du tableau de données (étape 2.5) et / ou des paramètres du logiciel d'Analyse Typologique (étape 3). L'algorithme de classification construit de nouvelles familles, en cohérence avec les connaissances expertes de la base (5). Les familles sont à nouveau soumises aux experts pour en apprécier le sens. Ils peuvent modifier les règles existantes ou bien en ajouter de nouvelles. Le cycle (5), (6), (7), (2), (2.5), (3) et (4) se poursuit jusqu'à ce que les experts soient satisfaits des familles obtenues. Il faut généralement plusieurs cycles pour obtenir une partition pertinente. La base de connaissance s'enrichit et l'étape de validation garantit la cohérence de l'ensemble des connaissances supplémentaires recueillies. En pratique, le processus converge généralement vers une partition pertinente au bout de quelques cycles.

Par rapports aux méthodes existantes, cette approche présente les avantages suivants :

- La majeure partie des règles de classification s'interprètent non pas en terme de modifications des attributs descriptifs, mais décrivent comment utiliser les attributs pour comparer les objets (cf. chapitre III.2.1.2.3). Notre approche respecte la façon dont les experts utilisent spontanément les attributs initiaux pour regrouper ou séparer les objets, elle permet à l'expert de formuler ses connaissances dans un cadre qui lui est familier, sans devoir modifier les attributs descriptifs et intégrer l'hypothèse de similarité.

- Les experts comprennent de mieux en mieux la façon dont le système classe les objets puisque le système fait en partie ce que les experts lui disent. Les classes obtenues ont plus de chance d'être pertinentes car elles sont construites selon les principes utilisés par les experts.

- Au fur et à mesure des cycles de classification, la base de règles permet de capitaliser les connaissances supplémentaires qui s'enrichissent tout en restant cohérentes. On obtient, au final, une partie du mode d'emploi des attributs, complémentaire de l'hypothèse de similarité. Nous verrons par la suite que la base de règle finale définit en complément de l'outil de classification une mesure de similarité adaptée au problème considéré.

- Le spécialiste en Analyse Typologique est libéré d'une grande partie de travail d'interprétation et d'analyse des connaissances supplémentaires. Il peut se consacrer à superviser le cycle de classification. A terme, il est même envisageable de rendre les utilisateurs complètements autonomes. L'étape d'analyse et d'interprétation est sensiblement accélérée. Les experts peuvent tester et corriger rapidement leurs connaissances.

Une fois les classes formées et validées par les experts, il est possible d'entamer un travail d'analyse pour déterminer l'intension des classes. En collaboration avec les experts, il s'agit d'expliciter les connaissances sous-jacentes aux classes, en terme d'attributs communs aux objets qui les constituent.

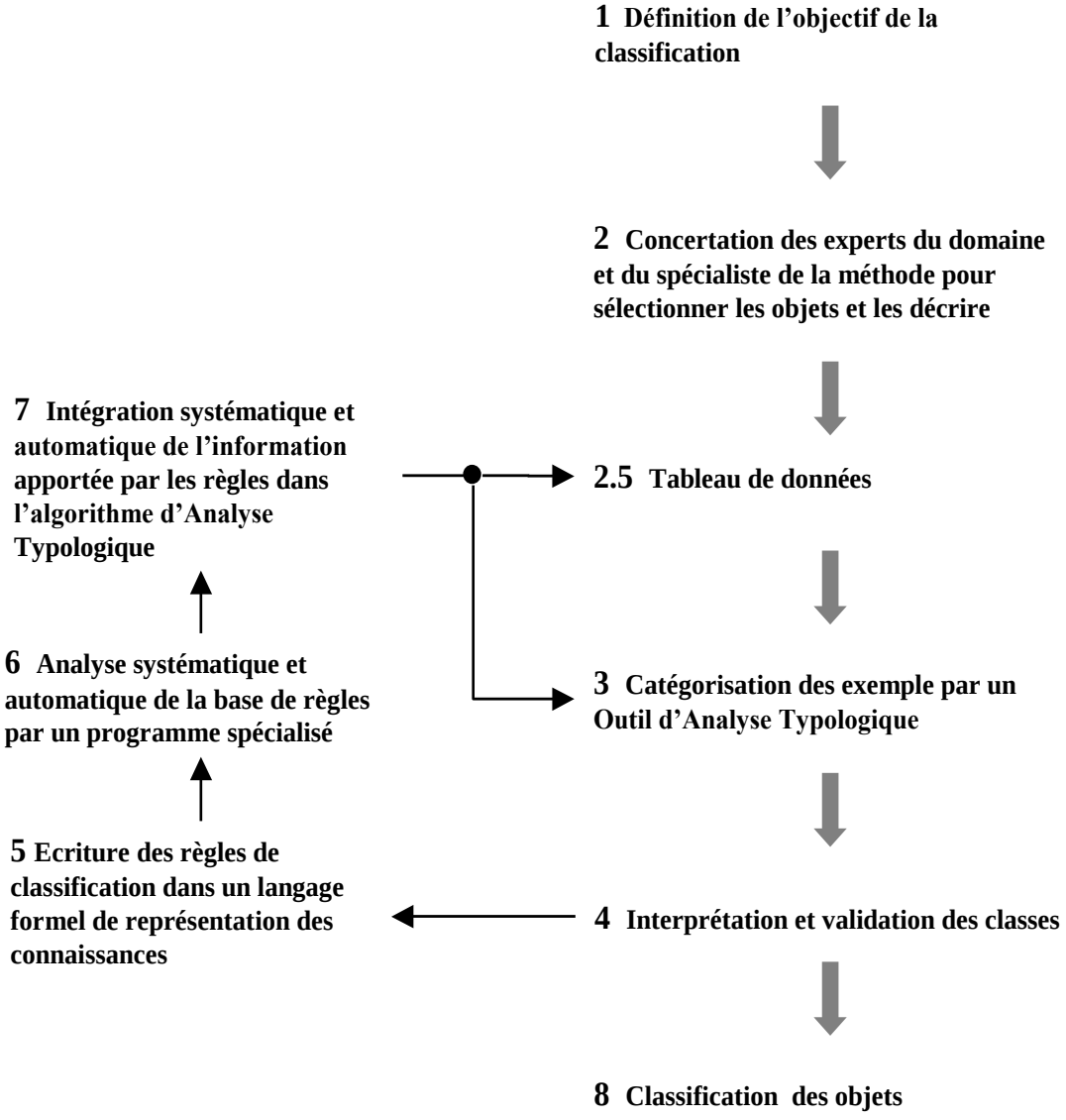


Fig. IV.1 Principe du fonctionnement de SYCLASCE

IV.2 Représentation des règles de classification

Dans cette partie, nous définissons le langage de représentation des connaissances qui permet de formaliser certaines connaissances supplémentaires émises par les experts du domaine lorsqu'ils évaluent la pertinence d'une classification automatique pour un problème d'organisation industrielle. Les connaissances supplémentaires formalisées définissent une base de règle. C'est l'étape (5) du cycle de classification de la figure IV.1.

Nous appellerons règles de classification le corpus de connaissances supplémentaires capitalisé qui servira de base pour analyser les remarques émises par des experts du domaine. Les règles de classification se regroupent autour de formulations types. Cette étude porte sur les deux catégories de règles les plus courantes : les règles de regroupement et les règles de codification (cf. chapitre I.3). Les règles de regroupement indiquent comment regrouper ou séparer les objets en fonction de leur code pour former une classification. Les règles de codification spécifient comment utiliser les attributs du code des objets pour les comparer. Pour chaque catégorie de règle, nous définissons une représentation formelle qui permettra aux experts de formuler leurs observations et d'en systématiser l'interprétation. On trouvera en annexe IV.1 le corpus de règles de classification qui est à l'origine de cette étude.

IV.2.1 Etude des règles de regroupement

Les règles de regroupement sont définies par un couple (description, lien). La description permet d'identifier les objets auxquels s'applique la règle. Le lien définit le type de relation qui doit s'établir entre les objets. La description est une liste de descriptions élémentaires. Chaque description élémentaire est un concept dont l'extension fait référence à une classe d'objets. Selon que les descriptions élémentaires sont utilisées en intension ou en extension, nous distinguerons pour une même règle de regroupement, une représentation intensionnelle et une représentation extensionnelle. Les deux liens fondamentaux sont la cohabitation et l'exclusion. La cohabitation signifie que les objets appartenant à l'extension des différentes descriptions élémentaires doivent appartenir à la même famille. L'exclusion spécifie que les objets appartenant à l'extension de descriptions élémentaires différentes doivent appartenir à des familles différentes.

Exemple IV. 2 Les différentes représentations d'une règle de regroupement

La représentation générique d'une règle de regroupement est un couple (description, lien) noté (D, l). Partant de cette structure, nous détaillons ci-dessous les différentes représentations d'une règle d'exclusion. La description D de cette règle est composée de trois descriptions élémentaires d1, d2 et d3. Le lien L est de type NA : il spécifie une relation d'exclusion entre les descriptions élémentaires. Pour la forme intensionnelle la description est remplacée par la liste des descriptions élémentaires. Considérant le tableau de données Td ci-dessous, chaque description élémentaire admet une extension sur Td notée respectivement P1, P2 et P3. En calculant ces extensions, il est possible de construire la représentation extensionnelle de la règle. Pour celle-ci la description est remplacée par la liste des extension des descriptions élémentaires.

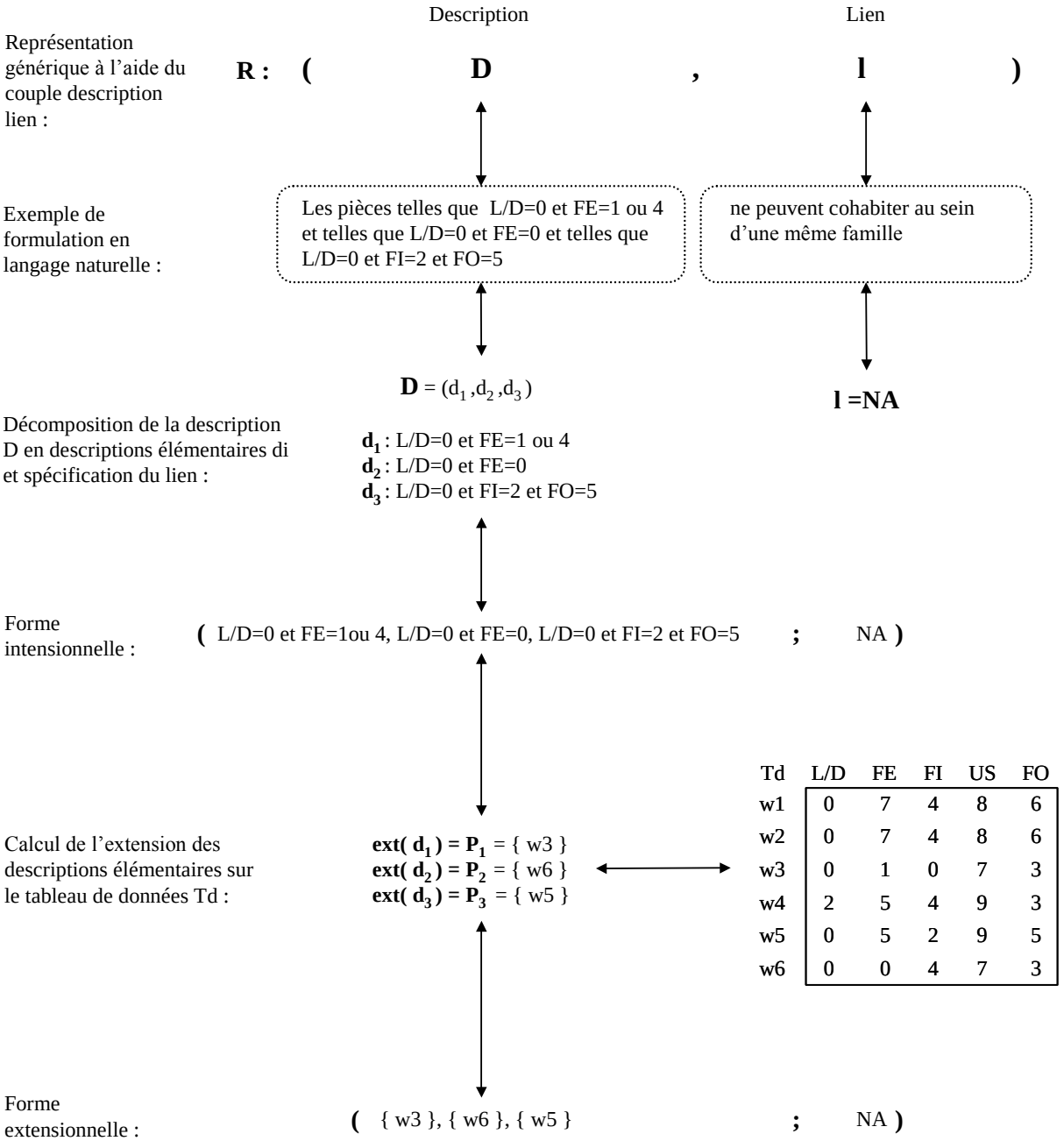


fig. IV. 1 Les différentes représentations d'une règle de regroupement

IV.2.1.1 Structure et représentations des règles de regroupement

IV.2.1.1.1 Représentation intensionnelle

Une base de règles **B** (ou base de connaissances) est un ensemble de règles de classification $\mathbf{R_i}$. Nous représentons une règle de regroupement Ri par un couple (description, lien) noté $\mathbf{(D_i, l_i)}$. La description D_i d'une règle R_i permet d'identifier les objets sur lesquels porte la règle. Di est une liste de descriptions élémentaires $\mathbf{d_{ij}}$.

Base de règle	**:**	$\mathbf{B = \{R_i\}_{i \in I}}$	I=[1, card(B)]
Forme intentionnelle d'une règle	**:**	$\mathbf{Ri = (D_i ; l_i)}$ avec $\mathbf{D_i = (d_{ij})_{j \in J}}$	J=[1, card(D_i)]

Chaque **description élémentaire** $\mathbf{d_{ij}}$ fait référence à une partie $\mathbf{p_{ij}}$ de l'ensemble des objets. Le couple (d_{ij}, p_{ij}) est un concept d'intention d_{ij} et d'extension p_{ij}.

Description élémentaire	**:**	$\mathbf{d_{ij}}$
Extension de d_{ij} sur Ω	**:**	$\mathbf{p_{\Omega ij} = ext_{\Omega}\,(d_{ij}) = \{\, w \in \Omega \mid d_{ij}(w) = vrai \,\}}$

Le **lien l** d'une règle Ri = (D_i ; $\mathbf{l_i}$) spécifie la relation qui doit s'établir entre les objets membres des extensions des descriptions élémentaires. Nous recensons quatre liens différents et appelons **L** l'ensemble des liens. Pour les règles de regroupement, la structure de la description est la même quelle que soit la conclusion. Ces règles sont donc essentiellement caractérisées par leur lien. Pour analyser une base de règle, il est commode de distinguer les règles en fonction des liens qu'elles définissent. L'ensemble des règles de même lien définira un type de règle. Le nom du type s'obtient en ajoutant un R devant le nom du lien visé.

Type de conclusion	Lien	Type de règles	Règles de B1
appartenir à la même famille	A	RA : règle de cohabitation	R9
ne pas appartenir à la même famille	NA	RNA : règle d'exclusion	R1 et R5
être une fonction discriminante	D	RD : règle discriminantes	R7
définir une famille	F	RF : règle familiale	R6

Fig. IV. 2 Conclusions types, liens et type de règles de regroupement

Ensemble des liens de regroupement	**:**	$\mathbf{L = \{ A, NA, D, F \}}$
Lien de regroupement	**:**	$\mathbf{l \in L}$

IV.2.1.1.2 Représentation extensionnelle

Pour définir les liens d'une façon formelle, c'est à dire comme une fonction sur l'ensemble des objets (cf. IV.2.1.3), il est commode de considérer la description comme la liste des extensions de ses descriptions élémentaires. Nous utiliserons le terme de forme extensionnelle, par opposition à la forme intensionnelle pour qualifier cette représentation. Une même représentation intensionnelle admet plusieurs représentations extensionnelles, selon l'ensemble considéré pour calculer l'extension des descriptions élémentaires. Généralement nous calculerons les extensions sur l'ensemble des objets Ω. Afin d'alléger la notation, nous conviendrons d'omettre l'indice précisant l'ensemble de référence lorsqu'il s'agit de Ω

Forme extensionnelle d'une règle : $\mathbf{R_{i\Omega} = (P_{i\Omega} ; l_i)}$ avec $\mathbf{P_{i\Omega} = (p_{\Omega ij})_{j \in J}}$

IV.2.1.2 Représentation des descriptions élémentaires

Une description élémentaire est une conjonction de termes. Les termes peuvent être conditionnels ou relationnels.

IV.2.1.2.1 Les termes conditionnels u

Les termes conditionnels servent à spécifier les valeurs des modalités d'une variable descriptive. Ce sont des prédicats définis sur l'ensemble des objets. Un terme conditionnel est de la forme : **[V # Q]**

Avec : $V : \Omega \rightarrow O$: variable définie sur l'ensemble des objets Ω à valeur sur l'ensemble O
$\#$: un symbole qui représente l'opérateur = ou ≠
$Q \subseteq O$: le référent

Le prédicat [V = Q](w) prend la valeur vraie si et seulement si $V(w) \in Q$
Le prédicat [V ≠ Q](w) prend la valeur vraie si et seulement si $V(w) \notin Q$

Exemple IV. 3 Termes conditionnels

Soit le terme conditionnel **[Forme ≠ {5, 4}]** qui signifie : « les objets dont la forme est différente de 5 et de 4 » et le terme conditionnel **[Couleur = {1, 2}]** qui signifie : « les objets de couleur 1 ou de couleur 2 »., Nous présentons, ci-dessous, l'extension de ces deux termes conditionnels sur le tableau de données Td.

Td	Couleur	Forme	[Couleur = {1, 2}] (w)	[Forme ≠ {5, 4}] (w)
w1	0	1	faux	**vrai**
w2	1	1	**vrai**	**vrai**
w3	1	1	**vrai**	**vrai**
w4	0	5	faux	faux
w5	2	4	**vrai**	faux

IV.2.1.2.2 Les termes relationnels r

Les termes relationnels sont des relations définies sur l'ensemble des objets. Un terme relationnel est de la forme : **[V = {#}]**

Avec : $V : \Omega \rightarrow O$: variable définie sur l'ensemble des objets Ω à valeur sur l'ensemble O

\# : un symbole qui représente l'opérateur = ou ≠

La relation [V = {=}](w1, w2) prend la valeur vraie si et seulement si V(w1) = V(w2)

La relation [V ={≠}](w1, w2) prend la valeur vraie si et seulement si V(w1) ≠ V(w2)

Exemple IV. 4 Termes relationnels

Soient le terme relationnel **[Couleur = {=}]** qui signifie : « les objets de même couleur » et le terme relationnel **[Forme = {≠}]** qui signifie : les objets de formes différentes. Sur le tableau de données Td ci-dessous, chacun de ces termes définit respectivement les relations r1 et r2.

Td	Couleur	Forme
w1	0	1
w2	1	1
w3	1	1
w4	0	5
w5	2	4

R1	w1	w2	w3	w4	w5
W1	1	0	0	1	0
W2	0	1	1	0	0
W3	0	1	1	0	0
W4	1	0	0	1	0
W5	0	0	0	0	1

Relation associée au terme **[Couleur = {=}] (w, w')**

R2	w1	w2	w3	w4	w5
w1	0	0	0	1	1
w2	0	0	0	1	1
w3	0	0	0	1	1
w4	1	1	1	0	1
w5	1	1	1	1	0

Relation associée au terme **[Forme = {≠}] (w, w')**

IV.2.1.2.3 Notation des descriptions élémentaires

Les descriptions élémentaires sont des conjonctions de termes. Afin d'alléger la notation, nous omettrons les symboles de conjonction, les crochets, ainsi que les accolades lorsqu'un terme conditionnel ne fait référence qu'à une seule modalité.

Exemple IV. 5 Notations des descriptions élémentaires

Soit la description élémentaire d1 définie par la conjonction du terme 1 et du terme 2. Nous présentons ci-dessous, la notation formelle, la notation « allégée » et son interprétation en langage naturel.

		terme 1		terme 2
d1 :	Représentation en langage naturelle	Forme vaut 5 ou 4	et	Couleur différente de 1
	Représentation formelle	**[Forme = {5, 4}]**	∧	**[Couleur ≠ { 1 }]**
	Notation allégée	**Forme = { 5, 4 }**		**Couleur ≠ 1**

IV.2.1.2.4 Description élémentaire conditionnelle

Une description élémentaire conditionnelle est une conjonction de termes conditionnels. C'est un prédicat dont l'extension définit une partie de l'ensemble des objets.

Soit : $\{ u_1, ... , u_i, ... , u_n \}$: un ensemble de termes conditionnels
d : une description élémentaire conditionnelle construite à partir des termes u_i

d admet la forme générale suivante : $\mathbf{d(w) = u_1(w) \wedge ... \wedge u_n(w)}$

Exemple IV. 6 Description élémentaire conditionnelle

Soit la description élémentaire conditionnelle d, nous calculons son extension sur le tableau Td ci-dessous.

Notation : d = Couleur={1, 2} Taille=0 Poids={2, 5}

Extension de d : ext(d) = p = { w2 }

Td	Couleur	Forme	Taille	Poids	[Couleur={1, 2}](w)	[Taille=0](w)	[Poids={2, 5}](w)	**d(w)**
w1	0	1	0	2	faux	vrai	vrai	faux
w2	**1**	**1**	**0**	**2**	**vrai**	**vrai**	**vrai**	**vrai**
w3	1	1	0	4	vrai	vrai	faux	faux
w4	0	5	1	5	faux	faux	vrai	faux
w5	2	4	1	5	faux	faux	vrai	faux

IV.2.1.2.5 Description élémentaire relationnelle

Une description élémentaire relationnelle est une conjonction de termes relationnels et conditionnels qui comprend au moins un terme relationnel. C'est une relation dont l'extension définit un ensemble de couples d'objets. Pour représenter les descriptions élémentaires relationnelles, nous ajouterons un exposant au symbole qui désigne la description élémentaire. Ainsi la notation $\mathbf{d^2}$ précise que d est l'intention d'une relation. L'extension $\mathbf{d^2}$ est notée $\mathbf{p^2}$, p^2 est un ensemble de couples d'objets, et non pas un ensemble d'objets.

Soit : $\{ u_1, ... , u_i, ... , u_n \}$: un ensemble de termes conditionnels
$\{r_1, ... , r_j, ... ,r_m\}$: un ensemble de termes relationnels
d^2 : une description élémentaire relationnelle construite à partir des termes ui et rj.

d^2 admet la forme générale suivante : $\mathbf{d^2(w1, w2) = \wedge_i (u_i(w1) \wedge u_i(w2)) \ \wedge_j r_j(w1, w2)}$

Une description élémentaire relationnelle fait référence à l'ensemble des couples qui vérifient r_1 et ... et r_n, tels que chacun des membres du couple vérifie u_1 et ... et u_n. L'écriture des descriptions élémentaires relationnelles admet certaines restrictions que nous préciserons en IV.2.1.3.6, car elles n'ont de sens qu'une fois définie l'interprétation des règles (cf. IV.2.1.3).

Exemple IV. 7 Description élémentaire relationnelle

Soit la description élémentaire relationnelle d^2 et le tableau Td ci-dessous.

Notation de d^2 : Couleur = { = } Forme = { 1, 5 }

Notation détaillée : $d^2(w1, w2) = [\text{ Couleur= } \{ = \}](w1, w2) \wedge [\text{ Forme= } \{1, 5\}](w1) \wedge [\text{ Forme= } \{1, 5\}](w2)$

Extension de d^2 : $ext(d^2) = p^2 = \{ (w2, w3), (w1, w4) \}$

Td	Couleur	Forme	Taille	Poids
w1	0	1	0	2
w2	1	1	0	2
w3	1	1	0	4
w4	0	5	1	5
w5	2	4	1	5

[Forme={1, 5}](w)
vrai
vrai
vrai
vrai
faux

d^2	w1	w2	w3	w4
w1	**1**	0	0	**1**
w2	0	**1**	**1**	0
w3	0	**1**	**1**	0
w4	**1**	0	0	**1**

La partie conditionnelle d'une description élémentaire relationnelle permet de restreindre le domaine d'application de la relation sur le sous-ensemble des objets qui vérifient la condition. Ci-dessus, la condition Forme = {1, 5} restreint la relation Couleur = { = } à l'ensemble des objets { w1, w2, w3 ,w4 }. Sur cet ensemble, la relation « avoir la même couleur » est vrai pour les couples (w2, w3) et (w1, w4). D'où, l'extension de d^2 : $p^2 = \{ (w2, w3), (w1, w4) \}$

IV.2.1.3 Interprétation des règles de regroupement

Les règles de regroupement décrivent partiellement la partition recherchée. Elles indiquent qu'au sein d'une « bonne » partition, certains objets doivent être rassemblés ou séparés. L'objectif de ce chapitre est d'interpréter et de représenter l'information qu'apporte une règle sur l'ensemble des objets. Pour cela, nous utiliserons le concept de fonction d'appartenance. Cette fonction associe à chaque couple d'objet l'information spécifiée par une règle donnée. Au lieu d'aborder directement une présentation formelle, nous commencerons par un exemple introductif. Celui-ci a pour objectif d'aider le lecteur à comprendre le sens de notre démarche et à aborder la formalisation des règles de regroupement avec un minimum de recul.

Exemple IV. 8 Tableau et fonction d'appartenance d'une règle de regroupement

Considérons la règle R8, l'ensemble des objets Ω et le tableau de données Td ci-dessous. Pour cet exemple, nous disposons de la partition recherchée P.

Partition : P = { {w1, w2, w3} ; {w4, w5, w6} }

R8 : (L/D=0 et FO=6 et FE=7 ; A)

R8 : ({w1, w2} ; A)

Td	L/D	FE	FI	US	FO
1	0	7	4	8	6
2	0	7	4	8	6
3	0	1	0	7	3
4	2	5	4	9	3
5	0	5	2	9	5
6	0	0	4	7	3

Tableau de données

Ar8	1	2	3	4	5	6
1	**1**	**1**	-	-	-	-
2	**1**	**1**	-	-	-	-
3	-	-	-	-	-	-
4	-	-	-	-	-	-
5	-	-	-	-	-	-
6	-	-	-	-	-	-

Fonction d'appartenance de R8

P	1	2	3	4	5	6
1	**1**	**1**	1	0	0	0
2	**1**	**1**	1	0	0	0
3	1	1	1	0	0	0
4	0	0	0	1	1	1
5	0	0	0	1	1	1
6	0	0	0	1	1	1

Relation d'équivalence de P

Le lien A de R8 signifie que les objets désignés par les descriptions élémentaires doivent être rassemblés. Appliqué à l'ensemble Ω, on trouve que les objets 1 et 2 doivent appartenir à la même famille. Donc, dans la partition recherchée, les objets 1 et 2 doivent être rassemblés. Pour représenter cette information, la fonction d'appartenance prendra la valeur 1 pour tous les couples composés des objets 1 et 2. Par contre, R8 ne donne pas d'information pour les autres couples de ΩxΩ. La fonction d'appartenance prendra la valeur « - » qui représente l'absence d'information. En représentant la fonction d'appartenance Ar8 à l'aide d'un tableau (objet x objet), on obtient le tableau Ar8 ci-dessus. On remarquera qu'il est alors facile de comparer directement l'information que fournit la règle par rapport à la partition recherchée P, en utilisant le tableau cartésien de la relation d'équivalence associées à P.

Plus généralement, une règle de regroupement appliquée sur un couple d'objets quelconques (wi, wj) aboutit à une des quatre interprétations suivantes :

- wi et wj appartiennent la même famille,
- wi et wj n'appartiennent pas à la même famille
- la règle ne donne pas d'information pour le couple (wi, wj),
- les informations sont contradictoires, les wi et wj doivent à la fois être et ne pas être rangés dans la même famille.

IV.2.1.3.1 Fonction d'appartenance

Il est donc possible de modéliser une règle de classification à l'aide d'une fonction nommée **fonction d'appartenance** qui à tout couple d'objets associe :

- la valeur **1**, si les partenaires doivent cohabiter,
- la valeur **0** si les partenaires ne doivent pas cohabiter,
- la valeur - si la règle ne donne pas d'information pour ce couple
- la valeur 💣 si la règle donne des informations contradictoires pour ce couple.

Toute règle R définit une fonction d'appartenance Ar. Cette fonction représente toute l'information que la règle fournit sur la partition recherchée. Une fonction d'appartenance Ar est une fonction d'un ensemble ΩxΩ fini dénombrable sur l'ensemble des valeurs {0,1,-,💣}. Typiquement Ar est définie sur ΩxΩ. Par défaut, nous considérons ΩxΩ comme ensemble de définition.

Ar :	Ω x Ω	→	{0, 1, -, 💣 }
	(w1, w2)	→	Ar(w1, w2)

Par la suite, nous utiliserons souvent un tableau croisant les objets de Ω pour représenter les fonctions d'appartenance en extension. Nous appellerons cette représentation un tableau d'appartenance. La fonction d'appartenance dépend de l'ensemble des objets considérés. Une règle sous sa forme intentionnelle admet donc autant de fonctions d'appartenance que d'ensembles d'objets sur lesquels on l'applique. Nous détaillons ci-dessous l'élaboration des fonctions d'appartenance à partir de la description extensionnelle d'une règle.

IV.2.1.3.2 Règles de type RA : règles de cohabitation

Les règles de cohabitation servent à regrouper des objets. Elles précisent que les groupes d'objets cités dans la description doivent appartenir à la même famille. Pour un couple d'objets dont chaque élément vérifie au moins une des descriptions élémentaires de la règle, la fonction d'appartenance doit prendre la valeur 1.

Par exemple, la règle R1 (**(p1)**, A) signifie que tous les éléments de p1 doivent appartenir à la même famille. Donc, pour tout couple composé d'éléments de p1 la fonction d'appartenance associée prend la valeur 1.

La règle R2 (**(p1, p2, p3)**, A) signifie d'une part que les éléments de p1 appartiennent à la même famille, ainsi que les éléments de p2 et de p3. D'autre part, cette règle signifie que les éléments de p1 cohabitent avec les éléments de p2 et avec les éléments de p3. Dans ce cas, la multitude des descriptions revient à désigner un seul ensemble formé par l'union des pi, dont

les éléments doivent cohabiter. Ainsi, pour tout couple composé d'éléments issus de p1 ou de p2 ou de p3, la fonction d'appartenance associée prend la valeur 1.

Exemple IV. 9 Fonctions d'appartenance des règles de cohabitation

Considérons l'ensemble Ω composé de 10 objets désignés par leur numéro. Nous donnons ci-dessous les règles R1 et R2 sous leur forme extensionnelle et les fonctions d'appartenance Ar1 et Ar2 associées.

R1 (**p1**, A) avec p1= {4, 5, 6}
R2 (**(p1, p2, p3)**, A) avec p1={2, 3}, p2={6,7}, p3={9 }

Ar1	1	2	3	4	5	6	7	8	9	10
1	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-
4	-	-	-	1	1	1	-	-	-	-
5	-	-	-	1	1	1	-	-	-	-
6	-	-	-	1	1	1	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-

Ar2	1	2	3	4	5	6	7	8	9	10
1	-	-	-	-	-	-	-	-	-	-
2	-	1	1	-	-	1	1	-	1	-
3	-	1	1	-	-	1	1	-	1	-
4	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-
6	-	1	1	-	-	1	1	-	1	-
7	-	1	1	-	-	1	1	-	1	-
8	-	-	-	-	-	-	-	-	-	-
9	-	1	1	-	-	1	1		1	-
10	-	-	-	-	-	-	-	-	-	-

Fig IV. 2 Fonctions d'appartenance des règles de cohabitation

Plus généralement, une règle de cohabitation R= **((p_j)$_{j\in J}$, A)** s'interprète comme une fonction d'appartenance Ar telle que :

$$\mathbf{Ar}: \quad \Omega \times \Omega \rightarrow \{0, 1, -, \text{💣}\}$$

$$(w, w') \rightarrow \mathbf{1} \quad \textbf{si } (w, w') \in \bigcup_{i\in J} pi \times \bigcup_{j\in J} pj$$

$$\rightarrow \mathbf{-} \quad \textbf{sinon}$$

IV.2.1.3.3 Règles de type RNA : règles d'exclusion

Les règles d'exclusion servent à séparer des groupes d'objets. Elles précisent que les différents groupes d'objets cités dans la description ne doivent pas appartenir à la même famille. Pour un couple dont les membres vérifient chacun une description élémentaire différente, la fonction d'appartenance associée prend la valeur 0.

La règle R2 (**(p1, p2)**, NA) signifie que les objets de p1 ne cohabitent pas avec les objets de p2. Donc, pour tous les couples composés d'un objet qui appartient à p1 et d'un objet qui appartient à p2 la fonction d'appartenance de R2 prend la valeur 0. Par contre, la règle ne donne pas d'information pour les couples composés uniquement d'éléments de p1 ou uniquement d'éléments de p2.

La règle R3 (**(p1, p2, p3)**, NA) signifie que les objets de p1, p2 et p3 ne cohabitent pas. Donc, pour les couples dont les membres appartiennent à deux ensembles pi (i=1, 2, 3) différents la fonction d'appartenance prend la valeur 0.

En revanche, la règle R1(**(p1)**, NA) n'a pas de sens, car elle ne précise pas avec quels éléments les objets de p1 ne doivent pas cohabiter (il est difficile de concevoir qu'un objet ne puisse cohabiter avec lui même).

Exemple IV. 10 Fonctions d'appartenance de règles d'exclusion

Considérons l'ensemble Ω composé de 10 objets désignés par leur numéro. Nous donnons ci-dessous les règles R2 et R3 sous leur forme extensionnelle et les fonctions d'appartenance Ar2 et Ar3 associées.

R2 (**(p1, p2)**, NA) avec p1={2, 3, 4}, p2={7, 8, 9}
R3 (**(p1, p2, p3)**, NA) avec p1={2, 3}, p2={6, 7}, p3={9}

Ar2	1	2	3	4	5	6	7	8	9	10
1	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	0	0	0	-
3	-	-	-	-	-	-	0	0	0	-
4	-	-	-	-	-	-	0	0	0	-
5	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-
7	-	0	0	0	-	-	-	-	-	-
8	-	0	0	0	-	-	-	-	-	-
9	-	0	0	0	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-

Ar3	1	2	3	4	5	6	7	8	9	10
1	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	0	0	-	0	-
3	-	-	-	-	-	0	0	-	0	-
4	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-
6	-	0	0	-	-	-	-	-	0	-
7	-	0	0	-	-	-	-	-	0	-
8	-	-	-	-	-	-	-	-	-	-
9	-	0	0	-	-	0	0	-	-	-
10	-	-	-	-	-	-	-	-	-	-

Fig IV. 3 Fonctions d'appartenance des règles d'exclusion

Plus généralement, une règle d'exclusion **R=({ p_j }$_{j\in J}$, NA)** s'interprète comme une fonction d'appartenance Ar telle que

Ar : $\Omega \times \Omega \rightarrow \{0, 1, -, \text{💣}\}$

$(w, w') \rightarrow$ **0** **si** $(w, w') \in \bigcup_{i\in J} pi \times \bigcup_{j\in J} pj - \bigcup_{i\in J} (pi \times pi)$

$\rightarrow$ **-** **sinon**

IV.2.1.3.4 Règles de type RD : règles discriminantes

Les règles discriminantes isolent un groupe d'objets. Elles précisent que les éléments de la description et les éléments du complément de la description ne doivent pas cohabiter. Si pour un couple d'objets, l'un vérifie au moins une description élémentaire et pas l'autre, alors la fonction d'appartenance doit prendre la valeur 0.

Par exemple, la règle R1 (**(p1)**, D) signifie que les objets de p1 ne doivent pas cohabiter avec les objets du complément de p1 sur Ω : ¬p1. Ainsi pour tout couple composé d'un élément de p1 et d'un élément de ¬p1, la fonction d'appartenance associée prend la valeur 0.

La règle R2 ((**p1, p2, p3)**, D) signifie que les objets de p1, p2 et p3 ne cohabitent pas avec les codes qui ne sont ni de p1, ni de p2, ni de p3. Donc, pour tout couple composé d'un élément de p1∪p2∪p3 et d'un élément de ¬ (p1∪p2∪p3), la fonction d'appartenance associée prend la valeur 0.

Exemple IV. 11 Fonctions d'appartenances de règles discriminantes

Considérons l'ensemble Ω composé de 10 objets désignés par leur numéro. Nous donnons ci-dessous les règles R1 et R2 sous leur forme extensionnelle et les fonctions d'appartenance Ar1 et Ar2 associées.

R1 (**(p1)**, D) avec p1={4, 5, 6}
R2 (**(p1, p2, p3)**, D) avec p1={2, 3}, p2={6,7}, p3={9}

Ar1	1	2	3	4	5	6	7	8	9	10
1	-	-	-	0	0	0	-	-	-	-
2	-	-	-	0	0	0	-	-	-	-
3	-	-	-	0	0	0	-	-	-	-
4	0	0	0	-	-	-	0	0	0	0
5	0	0	0	-	-	-	0	0	0	0
6	0	0	0	-	-	-	0	0	0	0
7	-	-	-	0	0	0	-	-	-	-
8	-	-	-	0	0	0	-	-	-	-
9	-	-	-	0	0	0	-	-	-	-
10	-	-	-	0	0	0	-	-	-	-

Ar2	1	2	3	4	5	6	7	8	9	10
1	-	0	0	-	-	0	0	-	0	-
2	0	-	-	0	0	-	-	0	-	0
3	0	-	-	0	0	-	-	0	-	0
4	-	0	0	-	-	0	0	-	0	-
5	-	0	0	-	-	0	0	-	0	-
6	0	-	-	0	0	-	-	0	-	0
7	0	-	-	0	0	-	-	0	-	0
8	-	0	0	-	-	0	0	-	0	-
9	0	-	-	0	0	-	-	0	-	0
10	-	0	0	-	-	0	0	-	0	-

Fig IV. 4 Fonction d'appartenance des règles discriminante

Plus généralement, une règle discriminante **R=((p_j)$_{j\in J}$, D)** s'interprète comme une fonction d'appartenance Ar telle que

Ar : $\Omega \times \Omega \rightarrow \{0, 1, -, \text{💣}\}$

$(w, w') \rightarrow$ **0** **si (w, w')** $\in \bigcup_{i\in J} pi \times \neg(\bigcup_{j\in J} pj) \cup \neg(\bigcup_{i\in J} pi) \times \bigcup_{j\in J} pj$

$\rightarrow$ - sinon

IV.2.1.3.5 Règles de type RF : règles familiales

Les règles RF définissent directement une famille de la partition recherchée. Elles précisent que les groupes d'objets cités dans la description définissent une famille. Ils doivent être regroupés, et ne pas cohabiter avec des objets n'appartenant pas à un des groupes de la description. Une règle de type RF s'interprète simultanément comme une règle d'appartenance et une règle discriminante ayant la même description.

Par exemple, la règle R1 (**(p1)**, F) signifie que les objets de p1 définissent une famille. Donc que ces objets doivent être regroupés, et qu'ils ne doivent pas cohabiter avec les autres objets. Ce qui revient à dire d'une part que ((p1), A) : les objets de p1 doivent être regroupés et d'autre part que ((p1), F) : les objets de p1 n'appartiennent pas à une autre famille.

De la même façon, R2 (**(p1, p2, p3)** , F) s'interprète comme une règle d'appartenance et une règle discriminante de description P = (p1, p2, p3).

Exemple IV. 12 Fonctions d'appartenances des règles familiales

Considérons l'ensemble Ω composé de 10 objets désignés par leur numéro. Nous donnons ci-dessous les règles R1 et R2 sous leur forme extensionnelle et les fonctions d'appartenance Ar1 et Ar2 associées.

R1 (**(p1)**, F) avec p1={4, 5, 6}
R2 (**(p1, p2, p3)**, F) avec p1={2, 3}, p2={6,7}, p3={9}

Ar1	1	2	3	4	5	6	7	8	9	10
1	-	-	-	0	0	0	-	-	-	-
2	-	-	-	0	0	0	-	-	-	-
3	-	-	-	0	0	0	-	-	-	-
4	0	0	0	1	1	1	0	0	0	0
5	0	0	0	1	1	1	0	0	0	0
6	0	0	0	1	1	1	0	0	0	0
7	-	-	-	0	0	0	-	-	-	-
8	-	-	-	0	0	0	-	-	-	-
9	-	-	-	0	0	0	-	-	-	-
10	-	-	-	0	0	0	-	-	-	-

Ar2	1	2	3	4	5	6	7	8	9	10
1	-	0	0	-	-	0	0	-	0	-
2	0	1	1	0	0	1	1	0	1	0
3	0	1	1	0	0	1	1	0	1	0
4	-	0	0	-	-	0	0	-	0	-
5	-	0	0	-	-	0	0	-	0	-
6	0	1	1	0	0	1	1	0	1	0
7	0	1	1	0	0	1	1	0	1	0
8	-	0	0	-	-	0	0	-	0	-
9	0	1	1	0	0	1	1	0	1	0
10	-	0	0	-	-	0	0	-	0	-

Fig IV. 5 Fonctions d'appartenance des règles familiales

Plus généralement, une règle familiale **R=((p_j)$_{j\in J}$, F)** s'interprète comme une règle de cohabitation et une règle discriminante de description (p_j)$_{j\in J}$. La fonction d'appartenance associée se définit comme il suit :

$$\mathbf{Ar}: \quad \Omega \times \Omega \rightarrow \{0, 1, -, \text{💣}\}$$

$$(w, w') \rightarrow \mathbf{1} \quad \text{si } (w, w') \in \bigcup_{i\in J} pi \times \bigcup_{j\in J} pj$$

$$\mathbf{0} \quad \text{si } (w, w') \in \bigcup_{i\in J} pi \times \neg\left(\bigcup_{j\in J} pj\right) \cup \neg\left(\bigcup_{i\in J} pi\right) \times \bigcup_{j\in J} pj$$

$$- \quad \text{sinon}$$

IV.2.1.3.6 Cas particuliers liés aux descriptions élémentaires relationnelles

Les méthodes de construction de la fonction d'appartenance données dans la section précédente, ne sont valables que si la description ne contient que des descriptions élémentaires conditionnelles. Nous définissons ci-après la construction de la fonction d'appartenance lorsque la description d'une règle comprend une description élémentaire relationnelle.

Les descriptions élémentaires relationnelles définissent directement un ensemble de couples. Nous considérons que le lien de la règle s'applique sur les couples de cet ensemble. Ainsi, une règle de cohabitation rassemble les objets des couples de la relation définie par la description élémentaire relationnelle. Une règle d'exclusion sépare les objets des couples de la relation définie par la description élémentaire relationnelle.

- **Si la description d'une règle ne comprend qu'une seule description élémentaire relationnelle**, alors la fonction d'appartenance définie par les liens de cohabitation (A) et d'exclusion (NA) est définie comme suit.

Règle de cohabitation extensionnelle R **((p^2) ; A)**	**Ar** :	$\Omega \times \Omega \rightarrow$ (w1, w2)	{0, 1, -, ☼} $\rightarrow$ -	**1** sinon	si (w1, w2) $\in p^2$
Règle d'exclusion extensionnelle **((p^2) ; NA)**	**Ar** :	$\Omega \times \Omega \rightarrow$ (w1, w2)	{0, 1, -, ☼} $\rightarrow$ -	**0** sinon	si (w1, w2) $\in p^2$

- **Si la description d'une règle comprend** :

  - plus d'une description élémentaire relationnelle, alors la descriptions fait référence à plusieurs ensembles de couples ;
  - au moins une description élémentaire relationnelle et au moins une description élémentaire conditionnelle alors la description fait référence à au moins un ensemble de couples et au moins un ensemble d'objets ;

  alors il n'existe pas d'interprétation simple de la règle qui permette d'en déduire une relation entre objets.

Exemple IV. 13 Règles dont la partie description définit une relation

Considérons les règles de regroupement R1, R2, R3 et le tableau de données Td ci-dessous.

R1 (couleur = { ≠ } ; NA)
R2 (couleur = { = } ; forme = { = } ; A)
R3 (couleur = { = } ; forme = { 1 } ; A)

Td	**Couleur**	**Forme**
w1	0	1
w2	1	1
w3	1	1
w4	2	5
w5	2	4

La description de la règle R1 comprend une seule description élémentaire relationnelle. Cette description élémentaire définit un ensemble de couple, c'est la relation r1. Le lien d'exclusion spécifie que les objets de ces couples ne doivent pas appartenir à la même famille. Cette information est représentée par la fonction d'appartenance Ar1.

r1	w1	w2	w3	w4	w5
w1	0	1	1	1	1
w2	1	0	0	1	1
w3	1	0	0	1	1
w4	1	1	1	0	0
w5	1	1	1	0	0

Relation associée à la description élémentaire de R1

Ar1	w1	w2	w3	w4	w5
w1	-	0	0	0	0
w2	0	-	-	0	0
w3	0	-	-	0	0
w4	0	0	0	-	-
w5	0	0	0	-	-

Fonction d'appartenance de R1

La description de la règle R2 comprend deux descriptions élémentaires relationnelles : couleur = { = } et forme = { = }. Ces descriptions élémentaires définissent deux ensembles de couples. Ce sont respectivement les relations r21 et r22. Il n'existe pas d'interprétation du lien de la règle qui s'impose d'une façon claire. En langage naturel, on remarque aisément l'ambiguïté de la phrase : « les objets de même couleur et les objets de même forme doivent cohabiter ».

r21	w1	w2	w3	w4	w5
w1	1	0	0	0	0
w2	0	1	1	0	0
w3	0	1	1	0	0
w4	0	0	0	1	1
w5	0	0	0	1	1

Relation associée à couleur = { = }

r22	w1	w2	w3	w4	w5
w1	1	1	1	0	0
w2	1	1	1	0	0
w3	1	1	1	0	0
w4	0	0	0	1	0
w5	0	0	0	0	1

Relation associée à forme = { = }

La description de la règle R3 comprend une description élémentaire relationnelle : couleur = { = } et une description élémentaire conditionnelle : forme = { 1 }. Ces descriptions élémentaires définissent respectivement un ensemble de couples : la relation r21 et un ensemble d'objets : la partie p3= { w1, w2, w3 }. Comme pour la règle R2, il n'existe pas d'interprétation simple du lien de la règle sur ces deux ensembles. En langage naturel, on remarque l'ambiguïté de la phrase : « les objets de même couleur et les objets de forme égale à 1 doivent cohabiter ».

Afin d'éviter l'ambiguïté des cas de figure précédents nous restreignons l'utilisation des descriptions élémentaires relationnelles à l'aide des quatre règles ci-dessous.

1 Une description ne peut contenir à la fois des descriptions élémentaires relationnelles et des descriptions élémentaires conditionnelles.

2 Une description ne peut contenir plus d'une seule description élémentaire relationnelle.

3 Les règles de type RD ne peuvent utiliser de descriptions élémentaires relationnelles. En effet, le lien D établit une relation entre l'ensemble des objets de la description et son complément. Si la description fait référence à un ensemble de couples, alors le complément de la description fait aussi référence à un ensemble de couples. Nous retrouvons le cas de la règle R2 de l'exemple IV.13.

4 Une règle de type RF ne peut pas utiliser de descriptions élémentaires relationnelles, car elle s'interprète simultanément comme une règle de type RA et une règle de type RD.

IV.2.2 Etude des règles de codification

D'une manière générale, les règles de codification indiquent comment utiliser les attributs pour comparer les objets. Elles reposent sur une structure en quatre composantes. Nous les représenterons par le quadruplet ci-dessous.

Soit : c : l'identification du contexte.
m1 : la description d'un premier groupe de modalités.
m2 : la description d'un deuxième groupe de modalités.
u : la définition d'un lien entre les deux groupes de modalités.

Règle de comparaison : R = (c, m1, u, m2)

Exemple IV. 14 Structure d'une règle de codification

	Contexte	1er groupe de modalités	Lien u	2ème groupe de modalités	Fin de la phrase
R4	Si L/D = 0 alors	les modalités FI=1 et FI=4	peuvent être groupées en	une seule FI=1ou4	pour comparer deux pièces

Le contexte d'une règle de codification désigne les objets pour lesquels la règle s'applique. C'est une description élémentaire conditionnelle.

Contexte : **c**
Extension de c sur Ω : $\mathbf{p_\Omega c = ext_\Omega (c) = \{ w \in \Omega \mid c(w) = vrai \}}$

Les groupes de modalités sont des listes de modalités d'une variable descriptive assimilables à des descriptions élémentaires conditionnelles ne comprenant qu'un seul terme.

Groupe de modalité : **mX** $X \in \{1, 2\}$
Extension de ci sur Ω : $\mathbf{p_\Omega mX = ext_\Omega (mX) = \{ w \in \Omega \mid mX(w) = vrai \}}$

Les règles de codification utilisent un lien unique, noté E. Il existe d'autres règles de codification basées sur la même structure et utilisant un lien différent (cf. annexe IV.1), mais leur étude est encore en projet.

Exemple IV. 15 Représentation intensionnelle d'une règle de codification

Soit la règle de codification R ci-dessous.

		C		m1		u		m2	
R4	**(**	L/D = 0	**,**	FI={1, 4}	**,**	E	**,**	FI={N}	**)**

- Le contexte est défini par la description élémentaire conditionnelle L/D = 0.
- Le groupe de modalité m1 est défini par le terme FI= { 1, 4}.
- Le groupe de modalité m2 est défini par le terme FI = { N }.

Comme pour les règles de regroupement, nous distinguerons pour une même règle, une représentation intentionnelle et une représentation extensionnelle.

Représentation : intensionnelle $\mathbf{R_i = (c_i, m1_i, u_i, m2_i)}$

Représentation : extensionnelle $\mathbf{R_i = (p_\Omega c_i, p_\Omega m1_i, u_i, p_\Omega m2_i)}$

IV.2.2.1 Interprétation des règles de codification

La règle de codification R= (c, m1, E, m2) signifie que pour l'ensemble des objets c, il faut remplacer les modalités du groupe m1 par la modalité du groupe m2. Il n'est donc pas nécessaire de distinguer les modalités du groupe m1 pour classer les objets. Celles-ci peuvent être considérées comme une seule et même modalité. Ces règles sont comparables aux règles d'inférence utilisées en Classification Conceptuelle (cf. II.1.1.1.3). Elles modifient la représentation des objets.

Exemple IV. 16 Règle de regroupement des modalités

Considérons la règle R de l'exemple précédent et le tableau de données Td ci-dessous.

Td	L/D	FE	FI
1	2	2	1
2	**0**	0	**4**
3	1	0	1
4	**0**	7	**1**
4	**0**	7	2
6	2	7	1

L/D = 0	FI={1, 4}
vrai	vrai
vrai	vrai
vrai	

Fc4	L/D	FE	FI
1	-	-	-
2	-	-	N
3	-	-	-
4	-	-	N
5	-	-	-
6	-	-	-

Td'	L/D	FE	FI
1	2	2	1
2	0	0	**N**
5	1	0	1
6	0	7	**N**
7	0	7	2
10	0	7	1

Pour tous les objets tels que L/D=0 la règle R remplace les modalités 1 et 4 de la variable FI par une seule nouvelle modalité N. Le tableau Fc4 représente l'ensemble de l'information apportée par la règle. La valeur « - » signifie que la règle n'apporte pas d'information, La valeur N signifie que la règle a permis de remplacer une modalité de FI par la modalité N. Après application de la règle, le nouveau tableau de données Td' ne distingue plus les modalités FI=1 et FI=4 lorsque L/D=0.

IV.2.2.1.1 La fonction de codification

La fonction de codification Fc associe à chaque couple (objet, variable) du tableau de données, l'information apportée par une règle de codification.

Soit : EV ={ Vi} : l'ensemble des variables descriptives.
V : $\Omega \rightarrow$ O : une variable de EV.
Fc : une fonction de codification.
N : une modalité de la variable V

Fc : Ω x EV $\rightarrow$ { -, 💣 } $\cup$ { N }
(w, V) $\rightarrow$ **« - »** signifie que la règle n'apporte pas d'information
« N » signifie que la variable V prend la valeur N pour l'objet w
« 💣 » signifie que la règle est incohérente pour ce couple d'objets

Pour une règle de codification donnée, la fonction de codification associée est la suivante :

Soit	R	: une règle de regroupement de modalités.
	(c, V=Q, E, V=N)	: la forme intensionnelle de R.
	EV	: l'ensemble des variables descriptives
	Fc	: la fonction de codification associée à R.

$$\mathbf{Fc}: \quad \Omega \times EV \rightarrow \{ -, N \}$$
$$(w, v) \rightarrow \mathbf{N} \quad \mathbf{si\ c(w) \wedge [V=Q](w)}$$
$$\mathbf{« - »} \quad \mathbf{sinon}$$

On remarquera que la fonction de codification associée à une règle de codification ne génère pas d'incohérences. Celles-ci sont susceptibles d'apparaître au cours de l'opération d'union des fonctions de codification (cf. IV.3.4). Comme pour les règles de regroupement, nous utiliserons souvent pour manipuler les fonctions de codification une représentation en extension sur le tableau (objet x variables).

IV.3 Evaluation de la base de règles

La base de règles est le reflet d'une connaissance experte qui n'est pas formalisée. Sa verbalisation relève d'une démarche du type essais erreurs. C'est pourquoi il est nécessaire d'évaluer cette base de règles avant de l'exploiter. C'est l'étape 6 du cycle de classification de la figure IV.1. L'évaluation porte sur trois critères : la cohérence, l'utilité et la redondance. Nous formalisons ci-après ces concepts pour les règles de classification et développons un cadre méthodologique pour les utiliser.

IV.3.1 Utilité d'une règle

Intuitivement, une règle de classification est d'autant plus utile qu'elle donne de l'information sur les objets à classifier. Pour évaluer l'utilité d'une règle, il faut définir une mesure de la quantité d'information que fournit une règle sur l'ensemble des objets. La quantité d'information fournie par une règle est fonction de son objectif. Les règles de regroupement rassemblent ou séparent des couples d'objets. Elles n'ont pas le même objectif que les règles de codification qui modifient le code des objets. Nous sommes donc amenés à définir deux mesures d'utilité : une pour les règles de regroupement et une autre pour les règles de codification. Avant d'étudier l'utilité des règles de regroupement, nous définissons le concept de composantes d'une fonction d'appartenance qui sera utilisé par la suite.

IV.3.1.1 Composantes d'une fonction d'appartenance

Les composantes d'une fonction d'appartenance sont les ensembles de couples d'objets pour lesquels cette fonction apporte la même information. Une fonction d'appartenance Ar admet quatre composantes :

La composante 1 : $\underline{\mathbf{1}}(Ar) = \{ (w1, w2) \in \Omega x \Omega$, tels que $Ar(w1, w2)=1 \}$

La composante 0 : $\underline{\mathbf{0}}(Ar) = \{ (w1, w2) \in \Omega x \Omega$, tels que $Ar(w1, w2)=0 \}$

La composante nulle : $\underline{-}(Ar) = \{ (w1, w2) \in \Omega x \Omega$, tels que $Ar(w1, w2)= - \}$

La composante incohérente : $\underline{\text{💣}}(Ar) = \{ (w1, w2) \in \Omega x \Omega$, tels que $Ar(w1, w2)=$ 💣 $\}$

Pour manipuler les composantes, nous utilisons quatre fonctions : $\underline{\mathbf{1}}(x)$, $\underline{\mathbf{0}}(x)$, $\underline{-}(x)$, $\underline{\text{💣}}(x)$ appelées fonctions composantes qui à une fonction d'appartenance associent une de ses composantes.

IV.3.1.2 Utilité des règles de regroupement

L'utilité d'une règle de regroupement est une fonction croissante du nombre de couples que la règle permet de rassembler ou de séparer. Ces couples sont définis par la fonction d'appartenance associée à la règle. Nous utilisons la mesure suivante :

Soit : ERR : l'ensemble des règles de regroupement d'une base de règle B.
U : la fonction d'utilité des règles de regroupement.
R : une règle de regroupement
Ar : la fonction d'appartenance associée
| | : la mesure du cardinal d'un ensemble

$$\mathbf{U} : ERR \rightarrow [0, 1]$$

$$R \rightarrow \frac{|1(Ar)| + |0(Ar)|}{|\Omega \times \Omega|}$$

Exemple IV. 17 Utilité des règles de regroupement

Considérons la règle R, l'ensemble des objets Ω ={wi} et le tableau de données Td ci-dessous. Pour représenter la fonction d'appartenance Ar de la règle R, nous omettrons le symbole « - » afin d'alléger la notation.

R : (L/D=0 FE=0 FI=1 US=0 FO=0 ; F)

Td	L/D	FE	FI	US	FO
w1	0	7	4	8	6
w2	**0**	**0**	**1**	**0**	**0**
w3	2	5	4	9	3
w4	0	7	0	7	6
w5	**0**	**0**	**1**	**0**	**0**

Ar	w1	w2	w3	w4	w5
w1		0			
w2	0	1	0	0	1
w3		0			0
w4		0			0
w5	0	1	0	0	1

| 1(Ar) | = 4

| 0(Ar) | = 11

U(R) = (4 + 11) / 5x5 = 60%

Lorsqu'une règle comprend plusieurs descriptions élémentaires conditionnelles (cf. IV.2.1.3.6), l'utilité ne permet pas forcément de déterminer si une des descriptions n'est pas inutile, en ce sens qu'elle a une extension vide sur l'ensemble des objets. Il est donc nécessaire d'évaluer l'utilité de chaque description élémentaire en mesurant le rapport du cardinal de son extension sur l'ensemble des objets.

Soit : Ud : la fonction d'utilité des descriptions élémentaires
Ed : l'ensemble des descriptions élémentaires
d : une description élémentaire.

$$\mathbf{Ud} : Ed \rightarrow [0, 1]$$

$$d \rightarrow \frac{|ext_{\Omega}(d)|}{|\Omega|}$$

IV.3.1.3 Utilité d'une règle de codification

L'utilité d'une règle de codification est une fonction croissante du nombre d'objets dont la règle modifie le code descriptif. Nous utilisons la mesure suivante :

Soit : ERM : l'ensemble des règles de codification d'une base B.
Uc : la fonction utilité des règles de codification
R : une règle de codification.
(c, V=Q, E, V=N) : la forme intensionnelle de R.

$$\mathbf{Uc} : ERM \rightarrow [0, 1]$$

$$R \rightarrow \frac{|ext(c(w) \wedge [V=Q](w))|}{|\Omega|}$$

En plus du calcul de l'utilité, il est nécessaire de vérifier que chaque modalité de la variable V désignée par le référent Q est utile. Il est fort possible que l'une des modalités ne permette pas d'identifier d'objets.

Soit : Ucd : la fonction utilité des modalités citées par une règle de codification
R : une règle de codification.
(c, V=Q, E, V=N) : la forme intensionnelle de R.
$v \in Q$: une modalité de V utilisée par R.
V=v : la description élémentaire qui porte uniquement sur la modalité v.
Ed : l'ensemble des descriptions élémentaires

$$\textbf{Ucd} : \quad Ed \rightarrow [0, 1]$$

$$V=v \rightarrow \frac{|ext(c(w) \wedge [V=v](w))|}{|\Omega|}$$

Exemple IV. 18 Utilité d'une règle de codification

Considérons la règle R et le tableau de données Td ci-dessous.

R (L/D=0 , FI={1, 4} , E , FI=N)

Td	L/D	FE	FI
w1	2	2	0
w2	**0**	**0**	**4**
w3	1	1	2
w4	**0**	**7**	**4**
w5	2	7	1

L/D=0	FI={1, 4}
vrai	**vrai**
vrai	**vrai**
	vrai

FI=1
vrai

FI=4
vrai
vrai

Tc	L/D	FE	FI
w1	-	-	-
w2	**-**	**-**	**N**
w3	-	-	-
w4	**-**	**-**	**N**
w5	-	-	-

ext(L/D=0 FI={1, 4 }) = { w2, w4} ⇒ **Uc(R) = 2/5 = 40%**
ext(L/D=0 FI=1) = ∅ ⇒ **Ucd(FI=1) = 0**
ext(L/D=0 FI=4) = 2 ⇒ **Ucd(FI= 4) = 2/5 = 40%**

La modalité FI=1 citée dans la règle n'est d'aucune utilité sur l'ensemble des données considéré, car elle ne permet pas de générer d'informations supplémentaires.

IV.3.2 Redondance d'une règle

Des règles sont d'autant plus redondantes qu'elles apportent la même information. Pour mesurer la redondance de deux règles, il faut définir une mesure du taux de recouvrement de l'information apportée par les règles. Nous distinguerons le cas des règles de regroupement et celui des règles de codification.

IV.3.2.1 Redondance des règles de regroupement

La redondance des règles de regroupement se situe à deux niveaux : au niveau des composantes de deux règles différentes : c'est la redondance inter règles, au niveau des descriptions élémentaires d'une même règle, c'est la redondance intra règle.

IV.3.2.1.1 Redondance inter règles

Deux règles sont redondantes, si elles apportent la même information. A savoir, si elles séparent ou rassemblent les mêmes couples d'objets. Cela se traduit par : un recouvrement des composantes en **1** si les règles rassemblent les mêmes couples d'objets ; un recouvrement des composantes en **0** si les règles séparent les même couples d'objets. Pour mesurer la redondance d'une base de règles, nous utiliserons un tableau d'inclusion des composantes **1**, noté **Tic1** et un tableau d'inclusion des composantes **0**, noté **Tic0**.

Soit : $B=\{R_k\}$: une base de règle de regroupement.
Tic1 : le tableau d'inclusion des composantes 1.
Tic0 : le tableau d'inclusion des composantes **0**.

$$\mathbf{Tic1} = (\ ic1_{kl}) = \frac{|1(R_k) \cap 1(R_l)|}{|1(R_k)|}$$

$$\mathbf{Tic0} = (\ ic0_{kl}) = \frac{|0(R_k) \cap 0(R_l)|}{|0(R_k)|}$$

Les éléments icX_{kl} mesurent l'inclusion de la composante $X(R_k)$ dans la composante $X(R_l)$. Lorsque icX_{kl} vaut 100%, $X(R_k)$ est incluse dans $X(R_l)$. La redondance des composantes en **1** peut être utile. Elle permet par transitivité inter règles d'inférer de nouvelles informations (cf. IV.2.3.2.3)

IV.3.2.1.2 Redondance intra règle

Deux descriptions élémentaires d'une même règle sont d'autant plus redondantes que leurs extensions se recouvrent. Pour mesurer la redondance des descriptions élémentaires d'une règle de regroupement, nous utiliserons un tableau d'inclusion des descriptions élémentaires, noté Tide.

Soit : R : une règle de regroupement
$((p_l) ; l)$: la forme extensionnelle de R.
Tide : le tableau d'inclusion des descriptions élémentaires associé à R.

$$\mathbf{Tide} = (\ ide_{kl}) = \frac{|p_k \cap p_l|}{|p_k|}$$

Les éléments ide_{kl} mesurent l'inclusion de l'extension élémentaire d_k dans l'extension élémentaire d_l. Lorsque ide_{kl} vaut 100%, d_k est incluse dans d_l, le recouvrement des descriptions élémentaires d'une règle de type RA ou RDF n'est pas forcément inutile. Par transitivité (intra-règle) ces recouvrements permettent de générer des informations (cf. IV.3.3.2.3). Ce type de redondance ne s'applique pas pour les règles de type RNA, car leurs descriptions élémentaires ne peuvent se chevaucher sans générer d'incohérence (cf. IV.3.3.1.1).

Exemple IV. 19 Redondance des règles de regroupement

Soit les règles R0, R1 et le tableau de données Td ci-dessous.

R0 ((FE=7 **,** US=8 **,** FI={1, 4}) **;** A **)**

R1 (L/D=0 FO=6 **;** A **)**

Td	L/D	FE	FI	US	FO
w1	0	7	3	9	6
w2	2	7	3	8	0
w3	0	5	3	8	6
w4	0	0	1	8	5
w5	1	0	5	0	0

Tableau de données

Ar0	w1	w2	w3	w4	w5
w1	**1**	1			
w2	1	1	1	1	
w3		1	**1**	1	
w4		1	1	1	
w5					

Fonction d'appartenance de R0

Ar1	w1	w2	w3	w4	w5
w1	**1**		1		
w2					
w3	1		**1**		
w4					
w5					

Fonction d'appartenance de R1

R1 n'a qu'une seule description élémentaire il ne peut donc y avoir de redondance intra règle. R0 admet trois descriptions élémentaires p01, p02 et p03. On voit sur le tableau d'intersection des descriptions élémentaires (Tide) de R0 que p01 et p02 se recouvrent partiellement et que p03 est incluse dans p02. La description élémentaire p03 est donc inutile sur l'ensemble Ω. Il faudra en référer à l'expert du domaine. Les règles R0 et R1 sont faiblement redondantes, comme le montre le tableau d'intersection des composantes **1** : Tic1. Leurs composantes **1** se recouvrent pour les couples (w1, w1) et (w2, w2)

p01 = ext(FE=7) = { w1, w2 }
p02 = ext(US=8) = { w2, w3, w4 }
p03 = ext(FI={1, 4}) = { w4 }

R0	p01	p02	p03
p01	-	50	0
p02	33	-	33
p03	0	**100**	-

Tide de R0

Tic1	R0	R1
R0	-	17%
R1	50%	-

IV.3.2.2 Redondance des règles de codification

Deux règles de codification sont redondantes si elles impliquent les mêmes modifications du code pour des objets. La redondance des règles de codification n'existe que pour les règles qui portent sur la même variable V et si les groupes de modalités à modifier présentent des modalités communes. Pour mesurer cette redondance, nous utiliserons un tableau de d'inclusion des règles de codification noté Tic.

Soit : B={Ri} : une base de règles de codification
(ci, V=Qi, E, V=Ni) : la forme intensionnelle des règles Ri.
Tic : le tableau d'inclusion des règles de codification
Avec $Qi \cap Qj \neq \varnothing$

$$\mathbf{Tic} = (\ ic_{ij}) = \frac{\left| ext(\ ci \wedge V=Qi\ ) \cap ext(\ cj \wedge V=Qj) \right|}{\left| ext(\ ci \wedge V=Qi\ ) \right|}$$

Exemple IV. 20 Redondance des règles de codification

Considérons les règles R1, R2 et le tableau de données Td ci-dessous. Pour cet exemple, les règles portent sur la même variable FI et n'apportent aucune information pour le reste du code descriptif. Nous représentons les fonctions de codification Fc1 et Fc2 des règles R1 et R2, uniquement sur la variable FI.

R1 **(** L/D=0 **,** FI={1, 4} **,** E **,** FI=N **)**

R2 **(** FE={5, 6} US=2 **,** FI={2, 4} **,** E **,** FI=N **)**

Sur le tableau Tic, on peut lire que R2 est incluse dans R1. Cette règle est donc inutile sur l'ensemble des objets considérés. Il faudra en référer à l'expert ; soit la règle est erronée ; soit l'ensemble des objets n'est pas représentatif des objets manipulés par l'expert.

Td	L/D	FE	FI	US
w1	2	2	0	4
w2	0	6	4	2
w3	0	1	1	3
w4	0	5	4	2
w5	2	7	1	5

Fc1	FI
w1	-
w2	N
w3	N
w4	N
w5	-

Fc2	FI
w1	-
w2	N
w3	-
w4	N
w5	-

Tic	R1	R2
R1	-	67%
R2	100%	-

IV.3.3 Synthèse et cohérence des règles de regroupement

Chaque règle de regroupement représente une parcelle des connaissances nécessaires pour élaborer une classification pertinente à partir du code descriptif des objets. Le concept de fonction d'appartenance fournit une représentation homogène de ces différents points de vue et permet de les intégrer. En " unissant " les fonctions d'appartenances associées aux règles de la base de connaissances, il est possible de construire une **fonction d'appartenance experte Ae**. Cette fonction représente " tout " ce que les experts sont capables de dire sur une classification pertinente des objets de leur domaine de compétence.

La fonction d'appartenance experte est construite en deux étapes. La première consiste à intégrer l'information des différentes règles sous la forme d'une seule fonction : **la fonction d'appartenance union** notée **U** (pour union). L'objectif de la deuxième étape est de compléter la fonction U en utilisant les propriétés de transitivité de la relation d'appartenance afin d'obtenir **Ae**. La formulation des règles est souvent entachée d'erreurs. S'il existe des règles contradictoires, la construction de U et de Ae fait apparaître ces incohérences. Elles seront corrigées avec l'aide des experts.

IV.3.3.1 Construction de l'union des fonctions d'appartenances

La fonction U est construite en appliquant l'opération d'union sur une base de règles de regroupement. D'un point de vue syntaxique, l'opération d'union est une loi de composition interne sur l'ensemble des fonctions d'appartenance, notée : $\oplus$.

Soit :	B={Ri}	: une base de règles de regroupement
	EAr	: l'ensemble des fonctions d'appartenance définies sur Ω
	Ari	: les fonctions d'appartenance respectives des règles Ri sur Ω
	$\oplus$	: l'opération d'union entre fonctions d'appartenance.
	U	: la fonction d'appartenance union des règles de B

$$\mathbf{U} = \oplus_i \mathbf{Ari}$$

Avec

$$\oplus : \quad EAr \times EAr \rightarrow EAr$$
$$(Ari, Arj) \rightarrow Ari \oplus Arj$$

Avec

$$\forall(w_1, w2) \in \Omega x\Omega \quad : Ari \oplus Arj\ (w1, w2) = Ari(w1, w2)\ \oplus'\ Arj(w1, w2)$$

$\oplus'$: est une opération de composition interne sur l'ensemble { 0, 1, -, 💣}, nous l'appellerons disjonction étendue. C'est une opération associative et commutative.

$\oplus$	0	1	-	💣
0	0	💣	0	💣
1	💣	1	1	💣
-	0	1	-	💣
💣	💣	💣	💣	💣

Pour saisir le sens de la disjonction étendue, il est commode de considérer les cas de figures qui apparaissent lorsque l'on essaye de synthétiser l'information apportée par deux règles pour un couple d'objets donné.

Soit : B={R1, R2} : une base de règles
Ar1 : la fonction d'appartenance de la règle R1
Ar2 : la fonction d'appartenance de la règle R2
(w, w') : un couple d'objets

Ar1(w, w')	**Ar2(w, w')**	**Ari(w1, w2) $\oplus$ Arj(w1, w2)**	**Information fournie par la base B**
" - "	" - "	" - "	Pas d'information
1	1	1	w1 et w2 doivent cohabiter
0	0	0	w1 et w2 doivent être séparés
1	0	💣	la base est contradictoire
0	1	💣	la base est contradictoire

Par abus de langage, il est pratique d'écrire $\oplus_i$ **Ri** au lieu de $\oplus_i$ **Ari.** Il faut néanmoins garder à l'esprit que la notation $\oplus_i$ **Ri** est une manière de désigner la fonction d'appartenance union des règles Ri, et non pas une nouvelle règle. En effet, $\oplus$ est une opération définie sur les fonctions d'appartenance et une règle implique une fonction d'appartenance mais pas l'inverse.

Exemple IV. 21 Union des fonctions d'appartenance

Considérons les règles R1, R2 et le tableau de données Td ci-dessous. Soit U la fonction d'appartenance union de R1 et R2. Elle représente l'information apportée par les règles R1 et R2 sur l'ensemble des objets considérés.

Td	L/D	FE	FI	US	FO
w1	0	7	4	8	6
w2	1	7	0	7	3
w3	2	5	4	9	3
w4	0	7	0	7	6
w5	2	6	4	7	3

R1 : (L/D={#} ; NA)

R2 : (L/D=0 FO=6 FE=7 ; A)

U = Ar1 ⊕ Ar2 = « R1 ⊕ R2 »

Ar1	w1	w2	w3	w4	w5
w1		0	0		0
w2	0		0	0	0
w3	0	0		0	
w4		0	0		0
w5	0	0		0	

Fonction d'appartenance de R1

⊕

Ar2	w1	w2	w3	w4	w5
w1	1			1	
w2					
w3					
w4	1			1	
w5					

Fonction d'appartenance de R2

=

U	w1	w2	w3	w4	w5
w1	1	0	0	1	0
w2	0		0	0	0
w3	0	0		0	
w4	1	0	0	1	0
w5	0	0		0	

Fonction d'appartenance union

IV.3.3.1.1 Incohérence directe

Les règles ne sont pas toujours complémentaires. De par le caractère empirique de la connaissance experte, il est fréquent que deux règles soient contradictoires. Pour être précis : deux règles sont en contradiction directe pour un couple d'objets, quand l'une précise que les partenaires doivent être rassemblés, et l'autre qu'ils doivent être séparés.

La construction de la fonction U met en évidence un type d'incohérence particulier que nous appelons les incohérences directes. Les contradictions observées sont corrigées en modifiant les règles à l'aide des experts. Dans un tableau d'appartenance, la contradiction est signalée par le symbole 💣 qui présente l'avantage d'être facilement repérable.

Exemple IV. 22 Incohérences directes des règles de regroupement

Considérons les règles R1 et R2 de tableaux d'appartenance Ar1 et Ar2. Lorsque l'on calcule l'union des fonctions d'appartenance, il apparaît une incohérence : pour le couple (w1, w2), la règle R1 vaut 0, alors que la règle R2 vaut 1. Il est à la charge des experts du domaine de modifier la règle pour lever l'incohérence.

Ar1	w1	w2	w3	W4	w5
w1		0	0		0
w2	0		0	0	0
w3	0	0		0	
w4		0	0		0
w5	0	0		0	

⊕

Ar2	w1	w2	w3	w4	w5
w1	1	1			
w2	1	1			
w3					
w4					
w5					

=

U	w1	w2	w3	w4	w5
w1	1	💣	0		0
w2	💣	1	0	0	0
w3	0	0		0	
w4		0	0		0
w5	0	0		0	

IV.3.3.2 Construction de la fonction d'appartenance experte

Avant d'utiliser les règles de regroupement pour construire une partition, il est nécessaire de vérifier si l'information fournie par la fonction d'appartenance union est compatible avec la structure de partition. Une fonction d'appartenance est compatible avec une partition s'il est possible de remplacer les " - " du tableau d'appartenance par des " 1 " ou des " 0 " de façon à trouver une relation d'équivalence. Cela n'est possible que si la fonction d'appartenance vérifie certaines conditions liées aux propriétés de réflexivité, symétrie et transitivité d'une relation d'équivalence. Pour vérifier la compatibilité d'une fonction d'appartenance avec une partition, nous appliquons sur celle-ci trois opérateurs de fermeture : fermeture réflexive, fermeture symétrique et fermeture transitive qui sont une extension des opérateurs classiques définis sur les relations binaires, aux fonctions d'appartenance. On démontre (cf. annexe IV.2) qu'une fonction d'appartenance est compatible avec une relation d'équivalence si et seulement si, sa fermeture réflexive et sa fermeture symétrique et sa fermeture transitive ne génèrent pas d'incohérences.

Soit : Ar : une fonction d'appartenance
FR : l'opérateur de fermeture réflexive d'une fonction d'appartenance
FS : l'opérateur de fermeture symétrique d'une fonction d'appartenance
FT : l'opérateur de fermeture transitive d'une fonction d'appartenance

$$\textbf{Ar est compatible avec une partition} \Leftrightarrow \begin{cases} \underline{\partial}(\mathrm{FR}(\mathrm{Ar})) = \varnothing \\ \underline{\partial}(\mathrm{FS}(\mathrm{Ar})) = \varnothing \\ \underline{\partial}(\mathrm{FT}(\mathrm{Ar})) = \varnothing \end{cases}$$

Exemple IV. 23 Compatibilité d'une fonction d'appartenance avec la structure de partition

La relation d'appartenance U1, ci-dessous est notamment compatible avec les partitions P1 et P2 . En remplaçant les " - " du tableau d'appartenance de U1 il est possible de construire les relations d'équivalence EP1 et EP2 associées respectivement à P1 et P2.

P1={ {w1, w2, w3} ; {w4}}
P2={ {w1, w2} ; {w3, w4}}

U1	w1	w2	w3	W4
w1	1	1	-	0
w2	1	1	-	0
w3	-	-	-	-
w4	0	0	-	1

EP1	w1	w2	w3	w4
w1	1	1	1	0
w2	1	1	1	0
w3	1	1	1	0
w4	0	0	0	1

EP2	w1	w2	w3	w4
w1	1	1	0	0
w2	1	1	0	0
w3	0	0	1	1
w4	0	0	1	1

La relation d'appartenance U2 ci-dessous n'est pas compatible avec une partition, car il est impossible de la compléter pour former une relation d'équivalence. Une relation d'équivalence est réflexive, or U2 prend la valeur 0 pour le couple (w3, w3). Une relation d'équivalence est symétrique, or le couple (w1,w2) prend une valeur différent du couple (w2, w1). Une relation d'équivalence est transitive. Par conséquent : U2(w1, w4) = 1 et U2(w4, w3) = 1 impliquent U2(w1, w3) = 1. Ce qui n'est pas vérifié dans le tableau de U2.

U2	w1	w2	w3	w4
w1	1	0	-	1
w2	1	-	-	-
w3	0	-	0	1
w4	0	-	-	1

IV.3.3.2.1 Fermeture réflexive d'une fonction d'appartenance

Comme pour les relations binaires, c'est l'union avec la " diagonale " **D** de l'ensemble ΩxΩ qui permet de construire la fermeture réflexive d'une fonction d'appartenance. A cette différence, que la diagonale de l'ensemble Ω est la fonction d'appartenance composée de 1 sur la diagonale et de tirets ailleurs.

Soit : **EAr** : l'ensemble des fonctions d'appartenance
FR : l'opérateur de fermeture réflexive d'une fonction d'appartenance

$$\mathbf{FR}: \quad EAr \rightarrow EAr$$
$$Ar \rightarrow \mathbf{Ar \oplus D}$$

La fermeture réflexive de U permet d'une part de faire des inférences. D'autre part, elle met en évidence les incohérences des règles de type RNA, liées à l'intersection de leurs descriptions élémentaires. Les objets qui appartiennent à plus d'une description élémentaire d'une règle RNA se trouvent dans l'obligation de ne pas cohabiter avec eux même. La règle engendre donc des 0 sur la diagonale du tableau d'appartenance. Ces 0, sont en contradiction avec les 1 de la diagonale.

Exemple IV. 24 Fermeture réflexive d'une fonction d'appartenance

Considérons la fonction d'appartenance U ci-dessous. U' est la fermeture réflexive de U.

U	w1	w2	w3	W4	w5
w1	1	1			
w2	1	1	0	0	0
w3		0		0	
w4		0	0		0
w5		0		0	

Fonction d'appartenance U

⊕

D	w1	w2	w3	w4	w5
w1	1				
w2		1			
w3			1		
w4				1	
w5					1

Diagonale de l'ensemble C

=

U'	W1	w2	w3	w4	w5
w1	1	1			
w2	1	1	0	0	0
w3		0	1	0	
w4		0	0	1	0
w5		0		0	1

Fermeture réflexive de U

IV.3.3.2.2 Fermeture symétrique d'une fonction d'appartenance

Une fonction d'appartenance Ar est symétrique si ses composantes le sont. Les composantes sont des relations binaires. Nous appellerons fermeture symétrique de Ar, la fonction d'appartenance obtenue en remplaçant les composantes de Ar par leur fermeture symétrique. La fermeture symétrique ne permet pas d'enrichir une relation d'appartenance en général et U en particulier, car les fonctions d'appartenance sont " symétriques " par construction.

∀ R une règle de regroupement de fonction d'appartenance Ar définie sur ΩxΩ :

Ar(w1, w2) = Ar(w2, w1)

La fonction d'appartenance union est aussi symétrique car :

U (w1 , w2) = $\oplus_i$ Ari (w1, w2) = $\oplus_i$ Ari (w2, w1) = U(w2, w1)

IV.3.3.2.3 Fermeture transitive d'une fonction d'appartenance

Si l'on considère une fonction d'appartenance Ar, la notion de transitivité se décline en deux nouvelles propriétés distinctes mais complémentaires : la 1-transitivité et la 0-transitivité.

1-transitivité : ∀w1, w2, w3 **si** Ar(w1, w2)=1 et Ar(w2, w3)=1 **alors** Ar(w1, w3)=1

0-transitivité : ∀w1, w2, w3 **si** { Ar(w1, w2)=1 et Ar(w2, w3)=0) ou (Ar(w1, w2)=0 et Ar(w2, w3)=1) } **alors** Ar(w1, w3)=0

Dans le cas d'une relation, ces deux propriétés sont confondues car la valeur par défaut vaut 0 (principe du tiers exclu). Dans le cas d'une fonction d'appartenance, la valeur par défaut n'est ni 1, ni 0, mais un symbole désignant l'absence d'information : " - ". En appliquant la X-transitivité sur une fonction d'appartenance, il est possible de " remplacer " des " - " par des 1 ou des 0, et ainsi de mettre en évidence toute l'information " contenue " dans la fonction étudiée.

Pour calculer les fermetures 1-transitives et 0-transitives d'une fonction d'appartenance, nous utiliserons deux fonctions : la fonction d'inférence de la 1-transitivité notée F1 et la fonction d'inférence de la 0-transitivité notée F0. F1 et F0 associent à une fonction d'appartenance donnée, une fonction d'appartenance qui contient toute l'information qu'il est possible d'en inférer par respectivement 1-transitivité et 0-transitivité.

Soit : Ar : une fonction d'appartenance.
F1 : la fonction d'inférence de la 1-transitivité
F0 : la fonction d'inférence de la 0-transitivité

F1(Ar): $\Omega \times \Omega \rightarrow \{0, 1, -, \bullet\}$

$(w1, w3) \rightarrow$ **1** si $\exists$ w2 tel que Ar(w1, w2)=1 et Ar(w2, w3)=1

" - " Sinon

F0(Ar): $\Omega \times \Omega \rightarrow \{0, 1, -, \bullet\}$

$(w1, w3) \rightarrow$ **0** si $\exists$w2 tel que { (Ar(w1, w2)=1 et Ar(w2, w3)=0) ou (Ar(w1, w2)=0 et Ar(w2, w3)=1) }

" - " Sinon

La fermeture X-transitive d'une fonction d'appartenance est égale à l'union de cette fonction avec son image par la fonction d'inférence de la X-transitivité.

Fermeture 1-transitive de Ar : **Ar ⊕ F1(Ar)**
Fermeture 0-transitive de Ar : **Ar ⊕ F0(Ar)**

La 1-transitivité met en évidence certaines " conséquences " des règles expertes qui n'étaient pas " visibles " à partir de la liste des règles. La fermeture 1-transitive de U peut mettre en évidence de nouvelles incohérences dans la base de connaissances. Nous les appellerons des incohérences indirectes. Par contre, la fermeture 0-transitive appliquée après la fermeture 1-transitive ne mettra pas d'incohérence en évidence. Car toutes les incohérences possibles ont été révélées avec la fermeture 1-transitive (cf. annexe IV.3). Nous définissons la fermeture transitive d'une fonction d'appartenance comme la fermeture 0-transitive de sa fermeture 1-transitive.

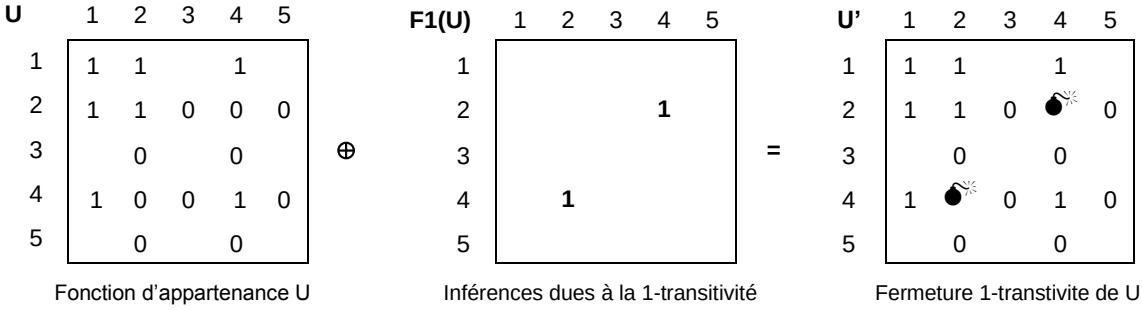
Soit : EAr : l'ensemble des fonctions d'appartenance
FT : l'opérateur de fermeture transitive d'une fonction d'appartenance

FT : EAr $\rightarrow$ EAr

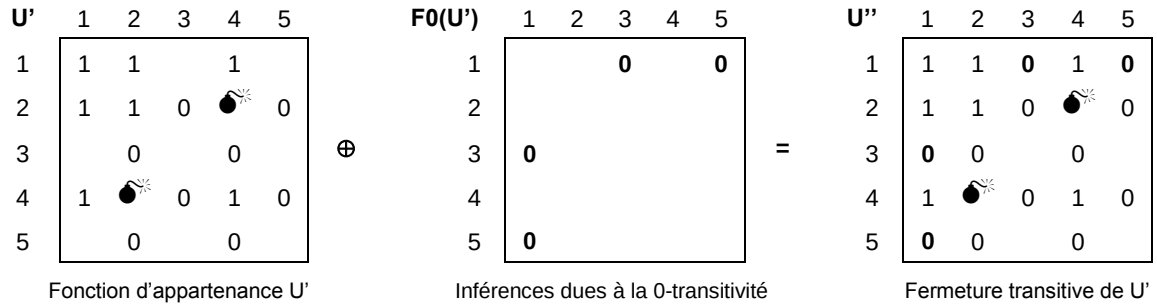
Ar $\rightarrow$ **Ar ⊕ F1(Ar) ⊕ F0(Ar ⊕ F1(Ar))**

Exemple IV. 25 Fermeture 1-transitive et 0-transitive d'une fonction d'appartenance

Considérons la fonction d'appartenance U ci-dessous. En calculant sa fermeture 1-transitive, il est possible de faire des inférences et dans le cas ci-dessous de mettre des incohérences en évidence.



D'après la fonction U, (w1, w2) appartiennent à la même famille et (w4, w1) appartiennent à la même famille. Par transitivité, il en résulte : (w2, w4) appartiennent à la même famille. Ce résultat est en contradiction avec l'information initiale. Les erreurs mises en évidence, sont soumises aux experts du domaine afin qu'ils modifient les règles en conséquence. Le tableau TaU' peut encore être enrichi grâce à la propriété de 0 transitivité. Par exemple, U' spécifie que (w1, w4) appartiennent à la même famille et que (w4, w3) n'appartiennent pas à la même famille. Par 0-transitivité, il en résulte que (w1, w3) n'appartiennent pas à la même famille. Il revient aux experts de modifier soit les données, soit les règles de regroupement afin de corriger la fonction d'appartenance union.



IV.3.3.2.4 Fonction d'appartenance experte

En appliquant les opérateurs de fermeture sur la fonction U union d'une base de règles de regroupement, il est possible de vérifier si les connaissances supplémentaires fournies par les experts sont compatibles avec la structure de partition. Au cours du processus de fermeture, les experts sont amenés à corriger la base de règles ou les données afin de lever les incohérences qui ont pu apparaître.

Les opérateurs de fermeture permettent aussi de générer de nouvelles informations contenues implicitement dans la base de règles. Nous appellerons **Ae : fonction d'appartenance experte**, la fonction d'appartenance construite à partir de l'union des règles enrichie par les

opérateurs de fermeture. Ae représente explicitement toute l'information contenue dans la base de règles de fonction d'appartenance U.

Soit : Ae : la fonction d'appartenance experte d'une base de règle B
U : la fonction union d'une base de règle B

$$\mathbf{Ae = FR(U) \oplus FT(U)}$$

Une fois corrigée, Ae est compatible avec une structure de partition (cf. IV.3.3.3.2)).

IV.3.3.3 Analyse de la fonction d'appartenance experte

La fonction d'appartenance experte Ae est généralement compatible avec un très grand nombre de partitions différentes. Elle suffit rarement à définir directement la partition recherchée. Il est néanmoins possible d'en extraire des informations qui simplifient le problème de classification. Considérons les parties Cx de l'ensemble des objets, définies par la propriété suivante :

$$\mathbf{\forall Cx \neq Cy \quad \forall (w, w') \in Cx \times Cy \Rightarrow \quad Ar(w, w') = 0}$$

Les parties Cx sont des familles d'objets totalement isolées les unes des autres. Dans la mesure où elles existent et sont validées par les experts, le problème de classification initial se décompose en autant de sous problèmes de classification qu'il y a de parties Cx. Pour identifier ces parties indépendantes, il suffit de rechercher les composantes connexes du graphe obtenu en remplaçant les "-" du tableau d'appartenance expert par des " 1 ".

A l'intérieur de chaque partie indépendante Cx, il est possible d'identifier des groupes d'objets indissociables Cxy. Les parties Cxy sont définies par la propriété ci-dessous. Cette propriété est liée à la " transitivité " de la fonction d'appartenance experte. Pour déterminer les groupes Cxy, il suffit de rechercher les composantes connexes du graphe obtenu en remplaçant les " - " par des " 0 ".

$$\mathbf{\forall Cxx \neq Cxy \quad \forall (w, w') \in Cxx \times Cxy \quad \Rightarrow \quad Ar(w, w') = 0 \text{ ou " - "}}$$

Si la fonction d'appartenance experte est relativement complète, il est possible d'interroger directement les experts sur la nature des relations entre les sous groupes Cxy, avec des questions du type : " faut-il regrouper ou séparer les groupes Cxx et Cxy ? ". Lorsque le nombre de questions à poser est important, il est nécessaire de définir des stratégies de questionnement pour organiser la liste des questions (De Guio 1999). Le cas précédent est

vrelativement rare ou bien se produit après plusieurs itérations du cycle de classification. Généralement, l'information de la base de règles de regroupement est intégrée dans un système de classification automatique (cf. IV.3).

Exemple IV. 26 Analyse de la fonction d'appartenance experte

Considérons la fonction d'appartenance expert Ae ci-dessous. Son analyse met en évidence trois parties indépendantes : C1={w1, w2}, C2={w3, w5, w6} et C3 = { w7, w8, w9, w10}.

		C1		C2			C3				
	Ae	1	2	3	4	5	6	7	8	9	10
C1	1	**1**	**1**	0	0	0	0	0	0	0	0
	2	**1**	**1**	0	0	0	0	0	0	0	0
C2	3	0	0	**1**	-	-	0	0	0	0	0
	4	0	0	-	**1**	-	0	0	0	0	0
	5	0	0	-	-	**1**	0	0	0	0	0
C3	6	0	0	0	0	0	**1**	**1**	-	-	**0**
	7	0	0	0	0	0	**1**	**1**	-	-	**0**
	8	0	0	0	0	0	-	-	**1**	**1**	-
	9	0	0	0	0	0	-	-	**1**	**1**	-
	10	0	0	0	0	0	**0**	**0**	-	-	**1**

Les parties C1, C2 et C3 sont isolées les unes des autres. Le problème de classification se décompose en trois sous problèmes indépendants. La partie C1 définit complètement une famille d'objets. La partie C3 comprend des groupes d'objets indissociables : C31={w6, w7 }, C32={ w8, w9 }, C33={ w10 }. Sur cet exemple, le tableau d'appartenance expert est suffisamment complet pour que l'on puisse continuer l'analyse en questionnant les experts sur la nature des relations entre, d'une part les objets de C2 et d'autre part les groupes d'objets de C3.

Sous problèmes de classification sur C2

C2	3	4	5
3	**1**	-	-
4	-	**1**	-
5	-	-	**1**

Sous problème de classification sur C3

		C31		C32		C33
	C3	6	7	8	9	10
C31	6	**1**	**1**	-	-	**0**
	7	**1**	**1**	-	-	**0**
C32	8	-	-	**1**	**1**	-
	9	-	-	**1**	**1**	-
C33	10	**0**	**0**	-	-	**1**

IV.3.4 Synthèse et cohérence des règles de codification

De la même façon que pour les règles de regroupement, nous synthétisons l'ensemble de l'information apportée par les règles de codification dans une fonction de codification experte. Cette synthèse peut mettre en évidence des incohérences qui seront soumises aux experts pour qu'ils les corrigent.

IV.3.4.1 Construction de la fonction de codification experte

La fonction de codification experte Fce est construite en appliquant une opération d'union sur la base de règles de codification. Cette opération est notée **⊗.**

Soit : B={Ri} : une base de règles de codification.
Fci : les fonctions de codification respectives des règles Ri.
Fce : la fonction de codification experte.

$$\mathbf{Fce = \otimes_i Fci}$$

L'opération d'union entre fonctions de codification est une loi de composition interne sur l'ensemble des fonctions de codification.

Soit : EFc : l'ensemble des fonctions de codification.
Fei : Ω x EV $\rightarrow$ { -, 💣 } $\cup$ { Ni } : une fonction de EFc.
Fej : Ω x EV $\rightarrow$ { -, 💣 } $\cup$ { Nj } : une fonction de EFc.
⊗ : l'union entre fonctions de codification.

⊗:	EFc x EFc	→	EFc
	(Fci, Fcj)	→	**Fci ⊗ Fcj**

Avec :

Fci ⊗ Fcj (w, V)		Fcj (w, V)			
		-	**Nj**	💣	
	-	**-**	**Nj**	💣	Xij = Ni si Ni = Nj
Fci(w, V)	**Ni**	**Ni**	Xij	💣	Xij = 💣 si Ni ≠ Nj
	💣	💣	💣	💣	

IV.3.4.2 Incohérence des règles de codification

Deux règles de codification sont incohérentes, si elles apportent des informations contradictoires pour une même couple (objet, variable). C'est à dire si les règles spécifient deux valeurs différentes pour la variable considérée. Les incohérences apparaissent lors de la construction de la fonction de codification experte. Elles sont soumises aux experts du domaine pour qu'ils corrigent les règles. Les incohérences sont signalées par la valeur " 💣 " facilement repérable dans le tableau de codification de la fonction Fce.

Exemple IV. 27 Union et cohérence des règles de codification.

Considérons les deux fonctions de codification Fc1 et Fc2 ci-dessous. Lors de l'opération d'union, il apparaît que ces deux fonctions de codification sont incohérentes pour le couple (w3, FE). Pour ce couple, la fonction Fc1 prend la valeur N1, alors que la fonction Fc2 prend la valeur N2.

Fc1	L/D	FE	FI
w1	-	N1	-
w2	-	-	-
w3	-	N1	-
w4	-	N1	-
w5	-	-	-

⊗

Fc2	L/D	FE	FI
w1	-	-	-
w2	-	N2	-
w3	-	N2	-
w4	-	-	-
w5	-	-	-

=

Fce	L/D	FE	FI
w1	-	N1	-
w2	-	N2	-
w3	-	💣	-
w4	-	N1	-
w5	-	-	-

IV.3.5 Cohérence des règles de regroupement et de codification

La cohérence des règles de regroupement et de codification se détermine en comparant l'effet des règles par rapport à l'objectif global des connaissances supplémentaires, à savoir : rassembler ou séparer les objets. Pour les règles de regroupement cette information est directement représentée par la fonction d'appartenance experte. L'effet des règles de codification est moins direct. Rappelons que les règles de classification corrigent le résultat d'une classification automatique obtenue à l'aide d'un outil d'analyse typologique. Les outils d'analyse typologique utilisent des indices de similarité. En modifiant le code descriptif, les règles de codification modifient la ressemblance entre les objets. Le principe de la classification empirique consiste à regrouper les objets qui se ressemblent et inversement de séparer les objets différents. Donc si règles de codification augmentent la similarité entre des objets, alors elles ont pour objectif de rassembler ces objets. Inversement, si les règles de codification diminuent la similarité entre des objets alors elles ont pour objectif de séparer ces objets. Sur la base de ces observations, il est possible de caractériser les différents cas de figures comparant l'action des règles regroupement et de codification en terme de cohérence, incohérence et redondance. La redondance entre les règles de regroupements et de codification n'est pas considérée comme un défaut de la base de règles. Bien au contraire, c'est le signe d'une convergence des différents types de règles.

		Effet des règles de codification		
		Pas d'effet	**Rapprocher**	**Eloigner**
Effet des règles de regroupement	**Pas d'effet**	Cohérence	Cohérence	Cohérence
	Rassembler	Cohérence	Redondance	**Incohérence**
	Séparer	Cohérence	**Incohérence**	Redondance

IV.3.5.1 Evaluation de l'effet des règles de codification

L'évaluation de l'effet des règles de codification en terme de regroupement comprend les étapes suivantes :

1. Construire le tableau de données expert Tde.
2. Construire le tableau de similarité Tse associé à Tde.
3. Construire le tableau des variations de la similarité : ΔTs
4. Construire la fonction d'appartenance Ac associée aux règles de codification.

1ère étape : construction de tableau de données expert Tde

Les règles de codification modifient le code descriptif des objets et changent le tableau de données initial Td. Pour modifier Td, nous lui associons une fonction de codification Fd et utilisons une opération de superposition entre Fd et la fonction de codification experte : Fce.

Soit : Td : le tableau de données initial.
Fd : la fonction de codification associée à Td.
EV = {Vi} : l'ensemble des variables descriptives
Vi : $\Omega \rightarrow$ Oi : une variable de EV

Fd : Ω x Ev $\rightarrow$ $\cup_i$ Oi
(w, V) $\rightarrow$ **V(w)**

Soit : B : une base de règles de codification
Fce : la fonction de codification experte associée a la base B.
Fde : la fonction du tableau de données expert.

$$\mathbf{Fde = Fd \underline{\otimes} Fce}$$

Avec

Fd (w, V)		Fce(w, V)		**Fdc (w, V)**
V(w)	$\underline{\otimes}$	« - »	=	V(w)
V(w)	$\underline{\otimes}$	N	=	N

2ème étape : construction du tableau de similarité expert Tse

En appliquant l'indice de similarité utilisé par le logiciel de classification automatique sur le tableau de données on construit le tableau de similarité expert : Tse. La construction de Tse nécessite cependant de préciser les notations traditionnellement utilisées en Classification Automatique. Généralement, l'indice de similarité est défini comme une fonction de l'ensemble des couples d'objets sur l'intervalle [0, 1]. Cette définition fait abstraction de la fonction de codage. L'indice de similarité, ne compare pas des objets, mais des données. La notation exacte est :

Soit : is : un indice de similarité
Ω : l'ensemble des objets à classifier.
EV = {Vi } : l'ensemble des variables descriptives.
Vi : Ω → Oi : une variable de EV.
c : la fonction de codage

$\mathbf{is_c}$: Ω x Ω → [0, 1]
(c(w), c(w')) → **is (c(w), c(w'))**

c : Ω → O1 × ... × Om
w → **c(w) = (V1(w), ... , Vm(w))**

La fonction de codage s'écrit en fonction de la fonction de codification comme suit :

$$\mathbf{c(w) = (V1(w), ... , Vm(w)) = (Fd(w, V1), ... , Fd(w, Vm))}$$

La fonction de codification experte définit une fonction de codage experte, notée ce :

$$\mathbf{ce(w) = (Fce(w, V1), ... , Fce(w, Vm))}$$

L'indice de similarité calculé sur le tableau de données expert, appelé indice de similarité expert s'écrit donc :

$\mathbf{is_{ce}}$: Ω x Ω → [0, 1]
(ce(w), ce(w')) → is (ce(w), ce(w'))

Dès lors, le tableau de similarité expert calculé en appliquant l'indice de similarité sur le tableau de données expert s'écrit :

$$\mathbf{Tse = (is_{ce} (wi, wj))}$$

3ème étape : construction du tableau de variation de la similarité : ΔTs

Pour évaluer si les règles ont tendance à séparer ou rassembler les objets, il faut calculer les variations de la mesure de similarité entre l'indice de similarité expert et l'indice de similarité initiale. La différence de ces indices pour chaque couple d'objets et représentée dans le tableau de variation de la similarité.

Soit : **Ts** = (is_c (wi, wj))
Tse = (is_{ce} (wi, wj))

$$\mathbf{\Delta Ts = (is_{ce} (wi, wj) - is_c (wi, wj))}$$

4ème étape construction de la fonction d'appartenance des règles de codification : Ac

Si les règles de codification augmentent la similarité entre deux objets, alors la variation de similarité est positive et les règles contribuent à rassembler ces deux objets. Inversement, si les règles de codification diminuent la similarité entre deux objets, alors la variation de similarité est négative et les règles contribuent à séparer ces deux objets

Ac	: Ω x Ω	→	{ 0, 1, « - » }	
	(w, w')	→	**1**	**si $\Delta ts_{ij} > 0$**
			0	**si $\Delta ts_{ij} < 0$**
			« - »	**si $\Delta ts_{ij} = 0$**

IV.3.5.2 Cohérence des règles de codification et de regroupement

Une base de règles de codification et une base de règles de regroupement sont incohérentes s'il existe des couples d'objets pour lesquels les deux bases apportent des informations contradictoires. C'est-à-dire que l'une des bases rassemble des objets, alors que l'autre les sépare. Dans le cadre formel défini précédemment, cette définition prend la forme suivante :

Soit : **B1** : une base de règles de codification.
B2 : une base de règles de regroupement.
Ae : la fonction d'appartenance experte des règles de regroupement de B1.
Ac : la fonction d'appartenance des règles de codification de B2.

$$\textbf{B1 et B2 sont cohérentes} \Leftrightarrow \underline{\bullet}\textbf{(Ae} \oplus \textbf{ac)} = \varnothing$$

IV.3.5.3 Redondance des règles de codification et de recouvrement

L'effet des règles de codification et des règles de regroupement s'exprimant en termes de relation d'appartenance, il suffit d'utiliser les mesures de redondances entre fonction d'appartenance **Tic1** et **Tic0**.

IV.3.6 Méthodologie d'évaluation de la base de règles

Nous précisons ci-dessous, la méthodologie d'évaluation d'une base de règles selon les trois critères : d'utilité, de redondance et de cohérence.

1ére étape : Saisie de la base de règles de classification

Les experts du domaine examinent une partition établie par un programme de classification automatique. S'ils valident la classification alors le processus s'arrête. Dans le cas contraire, ils émettent un certain nombre de critiques. L'analyste aide alors les experts à formuler ces connaissances supplémentaires en termes de règles de classification.

2ème étape : Evaluation de la base de règles de codification

Les règles de codification sont étudiées en premier car elles modifient le tableau de données que les règles de regroupement utilisent. Il est préférable de mesurer d'abord l'utilité des règles avant d'en examiner la cohérence et la redondance, afin d'identifier directement les règles dont la contribution est nulle. La base de règles validée permet de construire le tableau de données expertes. Après quoi, il faut vérifier la cohérence de la syntaxe des règles de regroupement avec le nouveau tableau de données. En effet, une règle de regroupement devient inapplicable si elle fait référence à une modalité qui a été modifiée par les règles de codification.

3ème étape : Evaluation de la base de règles de regroupement

Il est commode pour travailler sur les règles de regroupement d'utiliser le code projeté. C'est-à-dire la restriction de l'ensemble des variables aux seuls attributs qui sont cités dans les descriptions des règles. La première étape consiste à mesurer l'utilité des descriptions élémentaires, principalement pour détecter celles d'extension nulle. La construction et le calcul des fonctions d'appartenance permettent alors de vérifier l'utilité des règles, la redondance intra puis inter règles. Il est utile de mesurer la redondance inter règles sur la fermeture transitive de chaque règle. Cela révèle certains cas d'inclusion qui n'apparaissent pas lorsque l'on considère les fonctions d'appartenance initiales. Les incohérences directes apparaissent lors de la construction de l'union des règles. La fermeture réflexive met en évidence l'incohérence des règles d'exclusion. La fermeture transitive révèle les incohérences indirectes. La fonction d'appartenance experte ainsi construite est validée par rapport aux critères d'utilité, redondance et cohérence. L'analyse de cette fonction permet de vérifier s'il est possible de résoudre le problème de classification en complétant directement la fonction d'appartenance experte ou bien de réduire le problème en sous problèmes de classification indépendants.

4ème étape : Cohérence des règles de classification et de regroupement

La dernière étape du processus de validation consiste à vérifier la cohérence et la redondance des règles de regroupement et des règles de comparaison. L'ordre de ces deux opérations est indifférent.

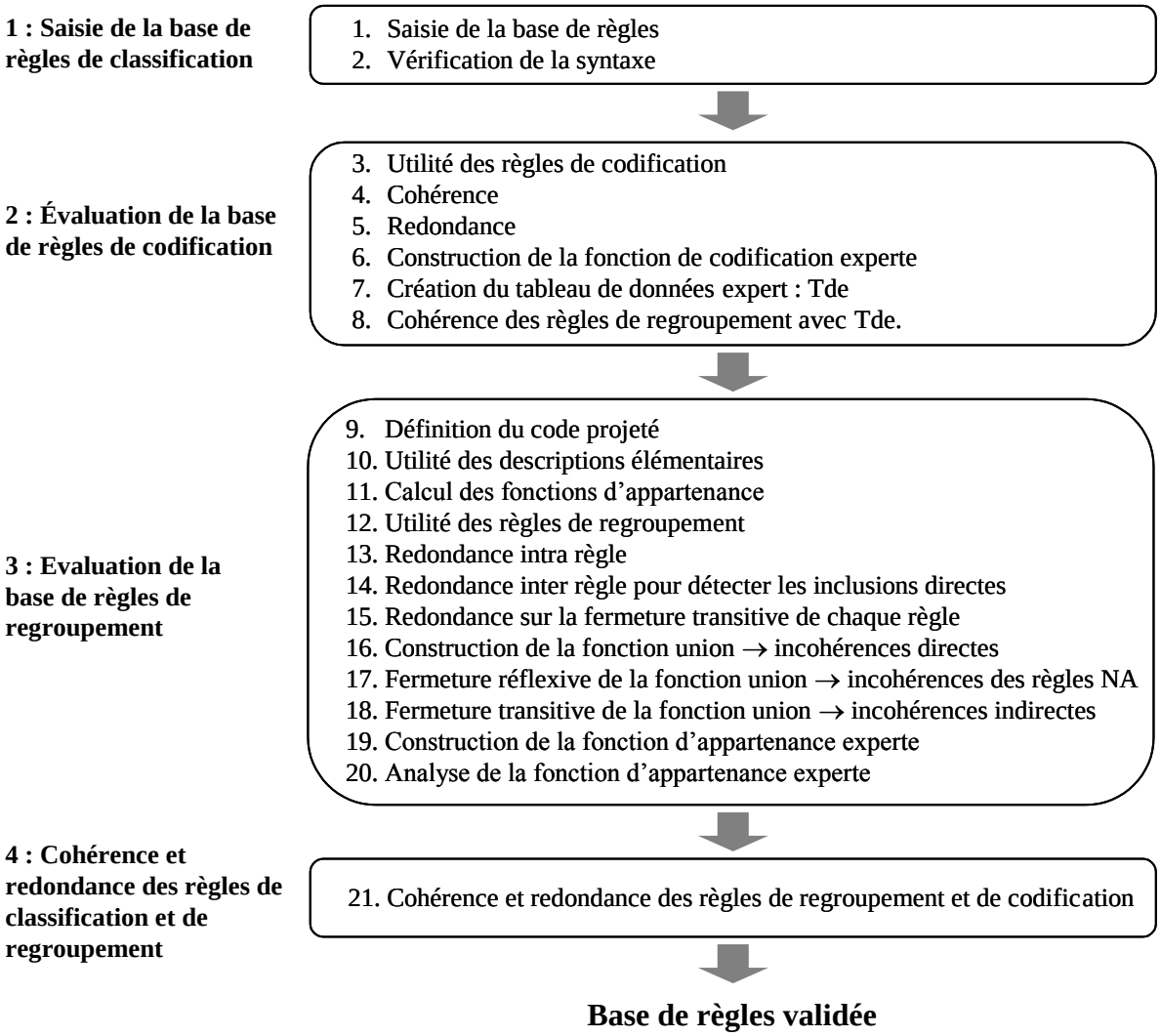


Fig. IV. 3 Méthodologie d'évaluation de la base de règles

IV.4 Intégration de l'information issue de la base de règle dans un processus de Classification Automatique

Les règles de classification ne permettent généralement pas de définir à elles seules une partition. Elles sont complémentaires des données initiales et de l'outil de classification. Pour déterminer une partition qui tienne compte des règles de classification, nous intégrons l'information issue de ces connaissances supplémentaires dans le processus de Classification Automatique. C'est l'étape 7 de la figure IV.1

L'information issue des règles de codification est intégrée en modifiant le tableau de données. La construction du tableau de données expert est détaillée en IV.2.5.1. Elle repose sur l'opération de superposition entre deux fonctions de codification.

L'ensemble de l'information issue des règles de regroupement réside dans le tableau d'appartenance expert. Cette représentation n'est pas directement compatible avec le processus de Classification Automatique. Ce dernier transforme un tableau de similarités en une relation d'équivalence. De manière qualitative, les outils d'Analyse Typologique, rassemblent les objets qui se ressemblent et séparent les objets qui sont différents. En inversant ce raisonnement, il est possible d'interpréter la fonction d'appartenance experte en termes de similarité. Si les objets doivent être rassemblés alors ils se ressemblent et s'ils doivent être séparés alors ils sont différents. Une façon d'intégrer l'information des règles de regroupement dans le processus de Classification Automatique consiste donc à associer une similarité maximale aux objets rassemblés par la fonction d'appartenance experte et à associer une similarité minimale aux objets qu'elle sépare. Nous formalisons ce raisonnement à l'aide d'une opération de superposition entre l'indice de similarité expert et la fonction d'appartenance experte. Le nouvel indice de similarité ainsi constitué s'appelle l'indice de similarité corrigé.

Soit : isc_{ce} : l'indice de similarité corrigé.
is_{ce} : l'indice de similarité expert.
Ae : la fonction d'appartenance.

$$\mathbf{isc_{ce} = is_{ce} \underline{\oplus} Ae}$$

avec

$\mathbf{is_{ce}(w, w')}$		Ae(w, w')		$\mathbf{isc_{ce}(w, w')}$
s	$\underline{\oplus}$	0	=	Smin
s	$\underline{\oplus}$	1	=	Smax
s	$\underline{\oplus}$	-	=	s

Soit : $Tsc = (isc_{ce}(wi, wj))$: le tableau de similarité corrigé.
$Tse = (is_{ce}(wi, wj))$: le tableau de similarité expert
Ae : la fonction d'appartenance.

Par abus de langage, nous écrirons aussi : **Tsc = Tse $\underline{\oplus}$ Ae**

L'opération de superposition, que ce soit pour les règles codifications ou les règles de regroupement, repose sur le principe de la priorité des connaissances supplémentaires sur l'information initiale. Il s'agit de corriger l'information initiale ou tout au moins la « partie » de l'information initiale qui n'est pas validée par les experts. Le code descriptif sert de référence aux experts. Il existe souvent pour des raisons historiques ou autres et n'est pas forcément adapté à l'objectif de la classification. Compte tenu de cet objectif, il y a peu de chances que les données respectent l'hypothèse de similarité. C'est à dire qu'il n'existe pas de relation simple entre le code descriptif et les classes recherchées. Il ne suffit donc pas de rassembler les objets qui se ressemblent pour trouver des classes pertinentes. Cependant, l'information initiale n'est souvent pas dénuée de sens par rapport à l'objectif de la classification. Le code utilisé n'est pas arbitraire. Il entretient une relation avec l'activité des experts. Une « partie » de l'information initiale est donc pertinente. Les opérations de superposition permettent d'en tenir compte en corrigeant uniquement ce qui est considéré comme non pertinent par les experts. C'est en ce sens que les règles de classification sont complémentaires de l'outil de classification. Elles sont valables et utilisables pour elles-mêmes, mais tant que la base n'est pas complète, elles ne définissent qu'une partie de la classification. C'est approximativement la partie correspondant aux données qui ne respectent pas l'hypothèse de similarité.

Le tableau de similarités corrigé contient toute l'information apportée par les règles de classification. Sous cette forme, cette information est utilisable par n'importe quel logiciel de Classification Automatique. Les contraintes de regroupement sont traduites en terme de similarités maximales et minimales, compatibles avec une structure de partition. En vertu de l'hypothèse de similarité, n'importe quel système de Classification Automatique doit respecter les contraintes de regroupement et fournir une partition qui intègre ces connaissances supplémentaires. C'est ce qui se passe dans la plus grande majorité des cas. Cependant, il n'est pas impossible que dans certains cas l'optimisation du critère utilisé par le système de Classification Automatique l'amène à ne pas respecter toutes les contraintes des règles de regroupement. Pour être certain du résultat il suffit de réaliser l'union de la fonction d'appartenance experte avec la relation d'équivalence de la partition élaborée par le logiciel. Si des incohérences apparaissent, c'est que l'algorithme de formation des groupes ne respecte pas les contraintes du tableau d'appartenance expert. Pour pallier cet inconvénient, nous avons défini une méthode de classification qui garantit le respect des règles de regroupement (De Guio 1999).

IV.5 Conclusion

Nous avons présenté en détail notre approche de la classification pour l'extraction de connaissances à partir de données sur un domaine complexe et peu formalisé, dans le cas de l'implantation de la TG. Elle se déroule selon les étapes du cycle de classification de la figure IV.1. Dans le cadre des problèmes de classification pour l'organisation industrielle que nous traitons, les connaissances supplémentaires présentent une certaine stabilité. Un langage de représentation des connaissances élaboré sur la base des connaissances supplémentaires les plus courantes permet de formaliser les critiques émises par les experts sous la forme de règles de classification. A partir de ce cadre formel, nous vérifions les qualités de cohérence d'utilité et de redondance de la base de règles. L'analyse de la fonction d'appartenance experte permet d'établir s'il est possible de diviser le problème initial en sous problèmes de classification. L'information issue de cette base est alors intégrée dans le processus de classification automatique, de façon à élaborer une partition qui rende compte des connaissances supplémentaires. Nous présentons dans le chapitre suivant une application dans le domaine de l'industrie mécanique. Cette application donne une première validation de l'intérêt de notre approche.

Chapitre V

Mise en œuvre et expérimentation de SYCLASCE

Introduction

Nous avons présenté dans le chapitre précédent la méthode développée pour contribuer à résoudre les problèmes de classification rencontrés dans le contexte de l'organisation industrielle. Cette méthode a donné lieu à la réalisation d'un prototype implanté en Pascal. Nous avons baptisé ce logiciel du nom de SYCLASCE pour Système de Classification Sous Contraintes Expertes. Nous présentons dans la première partie de ce chapitre (V.1) l'architecture du système SYCLASCE et ses performances. La deuxième partie (V.2) relate une application à la classification des pièces mécaniques d'une entreprise manufacturière dans le cadre d'une implantation de la TG au bureau d'étude.

V.1 Présentation de SYCLASCE

SYSCLASE est un prototype de système de classification automatique interactif sous contraintes symboliques. Implanté en Pascal, le système comprend trois modules : un module d'analyse de la base de règles, un module d'interface et un module de classification automatique.

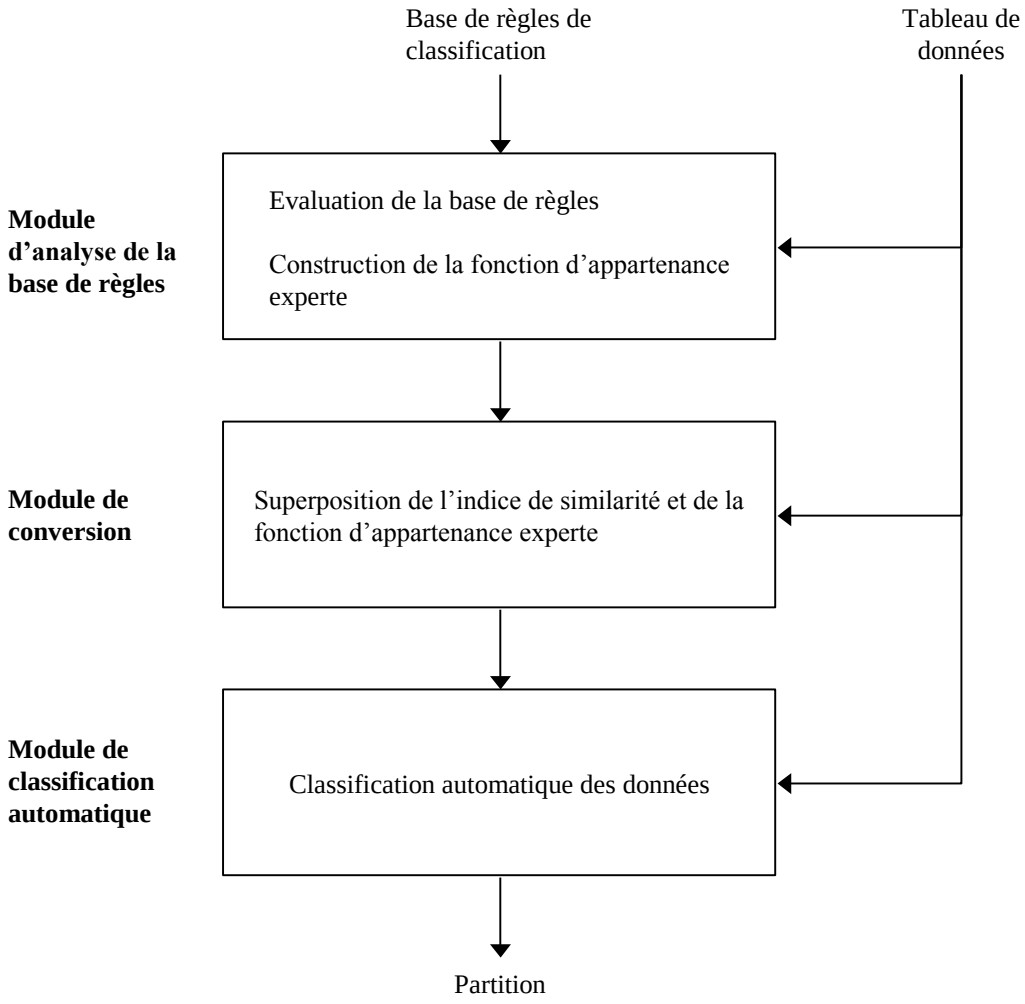
Le module d'analyse de la base de connaissances permet essentiellement de construire et d'analyser la fonction d'appartenance experte (cf. IV.3). Il détecte les incohérences dues à l'union des règles et aux fermetures réflexives et transitives.

Le module de conversion utilise l'opération de superposition entre l'indice de similarité et la fonction d'appartenance experte (cf. IV.4). Il construit le tableau de similarité corrigé. Celui-ci contient l'ensemble de l'information issue de la base de règles sous une forme exploitable par un système de classification automatique.

Le module de classification automatique est construit autour du logiciel SERMIX (Frey 1990; De Guio 1990) basé sur une méthode d'agrégation de similarité. Ce logiciel classifie les données en optimisant un critère d'homogénéité réglable par l'utilisateur à l'aide du paramètre α (cf. II.2.1.2.2)

Cette architecture permet d'utiliser indépendamment les différents modules, notamment pour des tâches exclusivement de classification automatique ou pour un travail d'analyse sur une base de règle indépendamment de la classification automatique.

La complexité de l'algorithme du module d'analyse de la base de connaissances est de l'ordre de (N^2.K.C). N étant le nombre d'objets, K le nombre total de descriptions élémentaires des règles de classifications et C le nombre total de modalités des variables descriptives. A titre d'exemple, il faut compter quelques secondes pour construire la fonction d'appartenance experte d'une base de règles comprenant en tout une vingtaine de descriptions élémentaires, sur un échantillon d'environ trois cents objets d'écrits par une cinquantaine de modalités. Le logiciel de classification automatique SERMIX repose sur une heuristique qui ramène la complexité du problème d'agrégation de similarité à l'ordre de (N^2.C). Ces deux algorithmes fonctionnent aisément sur un PC standard (200 MHz et 64 M de RAM)



Fig, V.1 Architecture de SYCLASCE

V.2 Une application à la classification des pièces mécaniques

L'entreprise étudiée fabrique des réducteurs de puissance. Les réducteurs sont composés de trois types de produits : les bâtis, l'habillage de la machine (couvercles, cales, visserie) et les produits mobiles (arbres, pignons, etc.). Le bâti et l'habillage sont spécifiques à chaque commande, par contre, la diversité des produits mobiles est susceptible d'être rationalisée. Les produits mobiles se répartissent en deux groupes : les pièces élémentaires et les rotors. Les rotors sont les arbres qui transmettent le mouvement et le couple dans la machine. Les pièces mécaniques élémentaires sont les pièces assemblées sur un rotor (coussinet, collet, déflecteur, anneau, manchons, frette, etc.). Une première implantation de la TG au bureau d'étude (Feltz 1993) a permis de rationaliser la diversité des rotors.

Suite à ce projet, il est apparu intéressant d'implanter la TG pour les pièces mécaniques élémentaires. Afin de réduire la charge de travail qu'implique la définition d'un code dédié, l'étude de faisabilité d'implantation de la TG pour les pièces élémentaires a été réalisée à l'aide d'un code prédéfini : le code OPITZ. Cette implantation a débuté en suivant les quatre étapes de la méthode présentée au chapitre I (cf. figure I.1). Le groupe de travail a utilisé le logiciel SERMIX (cf. V.1) pour l'étape (3) de classification des données. L'étape (4) de validation a donné lieu à la définition d'une base de règle de classification. Les itérations suivantes du cycle de classification ont été gérées manuellement. L'étude qui suit présente une application de SYCLASCE sur la base de règle issue de l'étape de validation initiale et montre comment les concepts développés au chapitre précédent contribuent à faciliter l'analyse de cette base de règles.

V.2.1 Construction du tableau de données (étape (2))

Le groupe de travail a sélectionné pour l'étude un échantillon de 297 pièces mécaniques élémentaires. Elles ont été codées à l'aide des variables du code OPTIZ qui comprend 11 variables qualitatives réparties en trois groupes : le code principal (cinq variables), le code additif (trois variables) et le code supplémentaire (trois variables). On trouvera le tableau de données en annexe V.1. Les variables V2 et V5 ne sont pas définies sur l'ensemble des objets. Nous ne pourrons pas en tenir compte lors de l'étape de classification automatique car les outils d'analyse typologique nécessitent des données régulières (cf. I.2). Nous utiliserons le code composé de toutes les autres variables.

Code		Variables
Principal :	**L/D**	: rapport de la longueur sur le diamètre de la pièce (3 modalités)
	FI	: caractéristiques de la forme intérieure (7 modalités)
	FE	: caractéristiques de la forme extérieure (7 modalités)
	US	: nature de l'usinage (9 modalités)
	FO	: type de forage auxiliaire (7 modalités)
Additif :	**V1**	**:** matériaux
	V2	**:** tolérance
	V3	**:** longueur
Supplémentaire :	**V4**	**:** poids
	V5	**:** complexité
	V6	**:** série de fabrication

Fig. V.2 Variables descriptives

Exemple V. 1 Extrait du tableau de données

Pi*	Pf*	Désignation	Plan	L/D	FE	FI	US	FO	V1	V2	V3	V4	V5	V6
0	0	Bouchon M16x1	3U061	0	2	0	3	0	0		0	0	0	0
0	0	Couvercle sup. GV	3U297	0	4	4	3	4	5	0	5	1		5
0	0	Bague de centrage	3U298	0	1	1	1	5	5		5	3		3
6	0	Anneau	3U438	0	4	1	0	0	4	6	0	0	0	3
9	0	Rampe de graissage	3U538	2	2	1	0	1	0		0	3		2

(la signification des ces deux colonnes est expliquée par la suite)

V.2.2 Classification automatique des données (étape (3))

Dans un premier temps, le groupe de travail a réalisé une classification automatique des données à l'aide du logiciel SERMIX. Au cours de l'implantation manuelle, aucunes des partitions obtenues en faisant varier systématiquement le paramètre d'homogénéité α (cf. II.2.1.2.2) du logiciel SERMIX n'a été validé par les experts. Les données ne respectent donc pas l'hypothèse de similarité au sens de l'outil de classification. Le groupe de travail a choisi comme classification initiale, celle qui est jugée comme la plus acceptable par les experts : la partition Pi. Chaque classe de cette partition est désignée par un numéro. La colonne Pi du tableau de données indique le numéro de la classe d'un objet. Par exemple, l'objet Bouchon M16x1 de la ligne 1 de l'exemple précédent appartient à la classe n°0 de la partition Pi.

V.2.3 Etape de validation (étape (4))

A partir de la classification Pi les experts du domaine ont formulé les bases de règles de classification Bc1 et Br1 ci-dessous.

Base de règles de codification : Bc1

	Contexte	m1	u	m2
R4	**(** L/D = 0 **,**	FI={1, 4} **,**	E **,**	FI={N} **)**

Base de règles de regroupement : Br1

	Descriptions D		Lien l	
R1	**(** L/D = { ≠}	**;**	NA	**)**
R5	**(** (L/D=0 FE={1, 4} ; L/D=0 FE=0 ; L/D=0 FI= 2 FO=5)	**;**	NA	**)**
R6	**(** L/D=0 FE=0 FI=1 US=0 FO=0	**;**	A	**)**
R7	**(** L/D=0 FE=0 FI=1 US=0 FO=0	**;**	F	**)**
R8	**(** L/D=0 FE=7	**;**	D	**)**
R9	**(** L/D=0 FO=6	**;**	A	**)**
R10	**(** L/D=0 FE=7 FO=6	**;**	A	**)**
R11	**(** L/D=0 FE#7 FO=6	**;**	A	**)**
R12	**(** L/D=0 US=6	**;**	A	**)**
R13	**(** L/D=0 FE=1 FI=1 US=0 FO=0	**;**	F	**)**
R14	**(** L/D=0 FE={1, 4} FI=1 US=0 FO=2	**;**	F	**)**
R15	**(** L/D=0 FE=4 FI=4 US=0 FO=0	**;**	D	**)**
R16	**(** L/D=0 FE=0 FI=0 US=0	**;**	A	**)**

V.2.4 Gestion manuelle du cycle de classification

A partir de cette base de règle, le groupe de travail a géré manuellement les itérations du cycle de classification. Au bout d'un travail d'environ deux mois, le projet a abouti à la définition d'une partition notée Pf (pour Partition Finale) de l'ensemble des pièces sélectionnées. Chaque classe de cette partition est désignée par un numéro que l'on trouvera dans la colonne Pf du tableau de données (cf. exemple V.1).

L'étude qui suit présente une application de SYCLASCE sur les bases de règles Bc1 et Br1 issues de l'étape de validation initiale. Nous suivrons la méthodologie présentée au chapitre IV.3.6. et montrerons comment les concepts développés au chapitre précédent facilitent l'analyse de cette base de règle. A l'occasion nous ferons référence à la partition Pf pour valider la pertinence de certains résultats que SYCLASCE va permettre de trouver très rapidement.

V.2.5 Etude des règles de codification

La règle de codification remplace les modalités 1 et 4 de la variable FI, lorsque L/D=0 par une seule modalité notée N. R4 présente une utilité de 15% c'est-à-dire qu'elle modifie le code de 44 objets sur 297. La règle est donc conservée.

Les règles R6, R7, R13, R14 et R15 utilisent la modalité FI=1 ou la modalité FI=4, il faut donc modifier ces règles en remplaçant dans leurs descriptions élémentaires les modalités 1 et 4 de FI par la modalité N.

V.2.6 Etude des règles de regroupement

V.2.6.1 Représentation des données sur le code projeté

Pour évaluer la base de règles, nous utilisons le code projeté composé des seules variables citées dans les règles de classification. En l'occurrence, il s'agit de LD, FE, FI, US et FO. A partir de ce code, nous travaillons sur un tableau de données projeté qui contient la liste des codes discriminés par ces variables. Ce tableau (cf. annexe V.2) ne fait donc pas référence à des objets, mais à des groupes d'objets ayant le même code projeté. Cette représentation permet de ne représenter que l'information utilisée par les règles. Elle allège la manipulation des données et des règles.

Exemple V. 2 Code projeté

On trouve ci-dessous la liste des objets désignés par le code projeté 01102. Les objets sont référencés par leur numéro de ligne dans le tableau de données complet.

N° de ligne	L/D	FE	FI	US	FO	V1	V3	V4	V6
35	0	1	1	0	2	6	5	3	3
37	0	1	1	0	2	5	9	2	2
38	0	1	1	0	2	6	5	1	2
39	0	1	1	0	2	5	1	2	2
40	0	1	1	0	2	5	1	1	2
43	0	1	1	0	2	5	1	0	2
44	0	1	1	0	2	4	1	1	2
48	0	1	1	0	2	6	9	1	3

Code projeté

Code complet

V.2.6.2 Mesure de l'utilité des règles de regroupement

En mesurant l'utilité des règles sur les données, nous constatons qu'il n'existe pas d'objets répondant aux descriptions des règles suivantes.

R5 : description élémentaire d53 : L/D=0 FI=2 FO=5
R15 : description élémentaire : L/D=0 FE=4 FI=4 US=0 FO=0

(R5 admet trois descriptions élémentaires notées selon la convention définie en IV.2.1.1.1 : d51 : L/D=0 FE={1, 4} ; d52 : L/D=0 FE=0 ; d53 : L/D=0 FI= 2 FO=5)

On conclut donc que soit l'échantillon d'objets sélectionné est incomplet, soit que les règles sont erronées. Après explication, il apparaît que la règle R15 fait référence à des pièces très spécifiques qui ont été exclues de l'échantillon représentatif de la production, défini au cours de l'étape 2 du processus de classification (cf. figure I.1). R15 est donc correcte, mais la connaissance qu'elle représente a déjà été prise en compte lors de la sélection des objets à classifier. Par contre, la description élémentaire d53 de R5 fait référence à des pièces qui n'existent plus. Ces observations prouvent que la classification agit comme un « révélateur » de la connaissance des experts. Les règles énoncées ne se réduisent pas à de simples corrections de la partition examinée, mais sont l'expression de la connaissance qu'ont les experts de leur activité.

La règle R5 est donc utile et correspond à l'expression du savoir-faire des experts. Mais, dans la mesure où elle n'apporte pas d'information sur l'échantillon étudié, nous n'en tiendrons pas compte pour la suite. Quant à la règle R5, nous la remplacerons par la règle R5' qui comprend uniquement les descriptions élémentaires d51 et d52.

V.2.6.3 Mesure de la redondance des règles

L'étude de la redondance des règles comprend deux étapes. Tout d'abord, la redondance intra règle mesure le recouvrement des descriptions élémentaires d'une même règle. Ensuite, la redondance inter règles mesure le recouvrement des fonctions d'appartenance des différentes règles.

Redondance intra règle

Seules les règles qui comprennent plusieurs descriptions élémentaires sont susceptibles de présenter une redondance intra règle. L'unique règle qui comprend plusieurs descriptions élémentaires est la règle R5' avec d51 et d52 (d53 ayant été éliminée). La seule lecture de d51 et d52 révèle que ces deux descriptions élémentaires sont disjointes, car la formule (FE=1 ou FE=4) et FE=0 est incohérente.

d51 : L/D=0 FE={1, 4}
d52 : L/D=0 FE=0

ext(d51) ∩ ext(d52) = ext(L/D=0 FE={1,4} FE=0) = ∅

Redondance inter règle

La redondance inter règle des composantes en 0 n'apporte pas d'informations significatives. Nous présenterons donc uniquement l'analyse de la redondance inter règle des composantes en 1. Les résultats sont donnés en pourcentages.

Tic1	R6	R7	R9	R10	R11	R12	R13	R14	R16
R6		**100**	0	0	0	0	0	0	0
R7	**100**		0	0	0	0	0	0	0
R9	0	0		**25**	**75**	0	0	0	0
R10	0	0	**100**		0	0	0	0	0
R11	0	0	**100**	0		0	0	0	0
R12	0	0	0	0	0		0	0	0
R13	0	0	0	0	0	0		0	0
R14	0	0	0	0	0	0	0		0
R16	0	0	0	0	0	0	0	0	

Fig V. 1 Tableau de redondance inter règle des composantes en 1

En observant le tableau d'inclusion des composantes en 1, nous constatons les faits suivants :

1. Les composantes en 1 de R6 et R7 sont identiques.
   Car Tic1(R7, R6) = Tic1(R6, R7) = 100%. Cela était visible à la seule lecture des règles. Mais ce n'est pas toujours le cas.

2. Les composantes en 1 de R10 et de R11 sont incluses dans celles de R9.
   Car Tic(R10, R9)=100% donc R10 est incluse dans R9, par contre Tic(R9, R10)=25%, donc R9 n'est pas incluse dans R10. Le raisonnement est le même pour R11.

R6 et R7 sont donc redondantes, de même, R10 et R11 sont redondantes par rapport à R9. Ces règles doivent donc être examinées et modifiées par les experts. Ces informations auraient été difficiles à déterminer sans une analyse systématique de la redondance inter règle. Les problèmes de redondances proviennent tout d'abord du recouvrement des points de vue entre les différents experts. C'est somme toute une bonne chose. Ils mettent cependant en évidence que les experts ne formulent pas, au moins dans un premier temps, leurs connaissances de la

même manière[1]. Si l'on examine la partition finale (cf. V.2.4) en tenant compte des observations précédentes, on constate les faits suivants :

- Les règles R6 et R7 correspondent une même famille. Une des deux règles R6 et R7 est donc inutile.
- La règle R9 regroupe deux familles de la partition finale. Elle est donc erronée.

La méthode utilisée a permis de cerner rapidement ces défauts de la base de règle. R7 étant plus informative que R6, nous ne prendrons pas R6 en compte pour la suite de cette analyse, de même que nous n'utiliserons pas la règle R9.

V.2.6.4 Définition d'une base de règles utile et non redondante

Une fois les règles inutiles et redondantes identifiées, il est possible de centrer l'analyse sur l'ensemble des règles utiles et non redondantes. Nous les recensons dans la base Br2 ci-dessous.

Base de règles de regroupement : Br2

R1	**(**	L/D = { ≠}	**;**	NA	**)**
R5'	**(**	(L/D=0 FE={1, 4} ; L/D=0 FE=0)	**;**	NA	**)**
R7	**(**	L/D=0 FE=0 FI=N US=0 FO=0	**;**	F	**)**
R8	**(**	L/D=0 FE=7	**;**	D	**)**
R10	**(**	L/D=0 FE=7 FO=6	**;**	A	**)**
R11	**(**	L/D=0 FE#7 FO=6	**;**	A	**)**
R12	**(**	L/D=0 US=6	**;**	A	**)**
R13	**(**	L/D=0 FE=1 FI=N US=0 FO=0	**;**	F	**)**
R14	**(**	L/D=0 FE={1, 4} FI=N US=0 FO=2	**;**	F	**)**
R16	**(**	L/D=0 FE=0 FI=0 US=0	**;**	A	**)**

[1] On remarquera en examinant la forme intensionnelle des règles, qu'il était possible d'anticiper les redondances observées en comparant la seule écriture des règles. Cette observation a une autre conséquence. Lors de l'exposé de la méthode au chapitre IV, nous avons uniquement utilisé l'extension des fonctions pour définir les concepts de cohérence, redondance et utilité. La mesure de l'utilité nécessite de travailler sur le tableau de données, par contre les concepts de redondance et d'incohérence reposent essentiellement sur des calculs d'intersection entre fonctions d'appartenance. Celles-ci sont définies par des formules propositionnelles. Il est donc tout fait envisageable d'utiliser un système de calcul formel pour vérifier la cohérence et la redondance des règles sans avoir recours systématiquement à leur extension. Nous en reparlerons au chapitre VI.

V.2.6.5 Cohérence des règles de regroupement

En calculant la fonction d'appartenance experte associée à Br2 sur le code projeté, le système relève les incohérences directes suivantes :

Couples de codes projetés	L/D	FE	FI	US	FO	Incohérences
code 23	0	0	1	6	0	R5' donne 0 et R12 donne 1
code 11	0	4	4	6	2	
code 52	0	0	1	1	6	R5' donne 0 et R11 donne 1
code 38	0	1	6	1	6	
code 52	0	0	1	1	6	R5' donne 0 et R11 donne 1
code 39	0	1	1	1	6	
code 52	0	0	1	1	6	R5' donne 0 et R11 donne 1
code 51	0	4	1	1	6	

Ces incohérences prouvent que les règles R5', R12 et R11 doivent être corrigées par les experts du domaine, car elles sont contradictoires pour certains objets. Ce type d'erreur admet principalement deux origines :

- Il peut s'agir d'une mauvaise définition des descriptions élémentaires. Le plus souvent, les descriptions élémentaires initiales sont trop générales. Par exemple, les règles R5' et R12 deviennent cohérentes, si l'on ajoute la condition FE=0 à la description élémentaire de R12. La nouvelle règle s'écrirait : R12' : (L/D=0 US=6 FE=0 ; A).

- Les règles incohérentes font référence à des points de vue contradictoires qu'il faudra arbitrer en fonction des objectifs de la classification.

L'analyse manuelle des règles de regroupement se fait généralement à l'aide d'une base de données. Des requêtes élaborées à partir des descriptions des règles permettent de définir des sous-groupes de l'ensemble des objets. Cette méthode ne permet pas de détecter systématiquement les incohérences de la base de règle. En examinant la partition finale, nous constatons que les règles R5' et R12 séparent des couples d'objets qui sont regroupés dans la partition validée par les experts. Elles ont donc été modifiées et / ou supprimées au cours du processus de classification manuel. Ceci accrédite la pertinence d'une analyse systématique des incohérences qui permet d'identifier rapidement des problèmes qui peuvent nécessiter plusieurs itérations dans le cas d'une gestion manuelle du cycle du classification.

V.2.6.6 Analyse de la fonction d'appartenance experte

La fonction d'appartenance experte se construit sur une base de règle cohérente. Pour illustrer l'analyse de cette fonction, nous considérons la base de règle Br3 ci-dessous, composée d'un sous-ensemble cohérent des règles de Br2.

Base de règles de regroupement : Br3

R1	**(** L/D = { ≠}	**;**	NA	**)**
R7	**(** L/D=0 FE=0 FI=N US=0 FO=0	**;**	F	**)**
R8	**(** L/D=0 FE=7	**;**	D	**)**
R10	**(** L/D=0 FE=7 FO=6	**;**	A	**)**
R11	**(** L/D=0 FE#7 FO=6	**;**	A	**)**
R13	**(** L/D=0 FE=1 FI=N US=0 FO=0	**;**	F	**)**
R14	**(** L/D=0 FE={1, 4} FI=N US=0 FO=2	**;**	F	**)**
R16	**(** L/D=0 FE=0 FI=0 US=0	**;**	A	**)**

L'analyse de la fonction d'appartenance experte comprend deux étapes. Tout d'abord l'identification des ensembles d'objets indépendants (cf. IV.3.3.3), puis l'analyse individuelle de ces ensembles.

Identification des ensembles d'objets indépendants

Cette analyse met en évidence 7 ensembles d'objets indépendants, notés C1 à C7 et appelés composantes. Nous donnons ci-dessous la structure du tableau d'appartenance expert obtenu.

Ae	C1	C2	C3	C4	C5	C6	C7
C1	1 -	0	0	0	0	0	0
C2	0	1 -	0	0	0	0	0
C3	0	0	-	0	0	0	0
C4	0	0	0	-	0	0	0
C5	0	0	0	0	1	0	0
C6	0	0	0	0	0	1	0
C7	0	0	0	0	0	0	1

Fig V. 2 Identification des composantes indépendantes du tableau d'appartenance

Les symboles placés dans les cases (Ci, Cj) s'interprètent comme suit :

- « **0** » signifie que la fonction d'appartenance Ae ne prend que la valeur 0 sur l'ensemble des couples formés à partir des objets de Ci et de Cj.
- « **1** » signifie que la fonction d'appartenance Ae ne prend que la valeur 1 sur l'ensemble des couples formés à partir des objets de Ci et de Cj.
- « **-** » signifie que la fonction d'appartenance Ae ne prend que la valeur - sur l'ensemble des couples formés à partir des objets de Ci et de Cj.
- « **1 -** » signifie que la fonction d'appartenance Ae prend au moins une fois la valeur 1 et la valeur « - » sur l'ensemble des couples formés à partir des objets de Ci et de Cj.

Analyses individuelles des ensembles d'objets indépendants

Dans un premier temps, notre attention se portera sur les composantes C5, C6 et C7. Chacune de ces parties définit une famille. Considérons par exemple C5. La case (C5, C5) ne comprend que des « 1 », donc les objets de C5 doivent tous être rassemblés. C5 étant indépendante, elle définit une famille. La base de règle permet donc de définir directement trois familles d'objets : C5, C6 et C7. Si les experts valident ces familles, le problème de classification est résolu pour les objets qui les composent. Si l'on compare ce résultat avec la partition finale (cf. V.2.4), il apparaît que C5 est une des familles recherchées. Par contre, les composantes C6 et C7 séparent une des familles recherchées.

En examinant l'utilité des règles sur le tableau précédent, il est facile de déterminer la contribution d'une règle à la définition d'une composante. Par exemple, le tableau ci-dessous représente l'utilité de la règle R7 sur les différents ensembles de couples d'objets définis par les composantes C1 à C7. La case (i,j) représente en pourcentage l'utilité de R7 sur l'ensemble des couples appartenant à Ci x Cj.

U(R7)	C1	C2	C3	C4	**C5**	C6	C7
C1	0	0	0	0	**100**	0	0
C2	0	0	0	0	**100**	0	0
C3	0	0	0	0	**100**	0	0
C4	0	0	0	0	**100**	0	0
C5	**100**	**100**	**100**	**100**	**100**	**100**	**100**
C6	0	0	0	0	**100**	0	0
C7	0	0	0	0	**100**	0	0

La composante C5 est donc entièrement définie par la règle R7. De la même façon, il apparaît que les composantes C6 et C7 sont respectivement définies par les règles R13 et R14. Les objets de ces deux composantes sont rassemblés au sein d'une même famille dans la partition validée par les experts (cf V.2.4) Les règles R13 et R14 sont donc erronées.

Dans un deuxième temps, nous examinons les composantes C1 à C4. Elles ne définissent pas directement des familles, car pour certains couples composés à partir de ces parties, la fonction d'appartenance prend la valeur « - ». La base de règle ne donne donc pas suffisamment d'information pour classifier ces objets. Deux cas de figure sont envisageables :

1 Soit la classification effectuée par le logiciel de classification automatique était satisfaisante, dans ce cas, les experts n'ont pas voulus modifier cette classification. L'information contenues dans le code descriptif est suffisante et correctement interprétée par la logiciel de classification automatique.

2 Soit les experts n'ont pas encore formulé les connaissances relatives à ces objets.

Dans tous les cas, l'identification des composantes synthétise l'information apportée par la base de règle. Elles donnent une représentation de l'information fournie par les experts directement en terme de regroupement d'objets. Cela permet de focaliser l'attention des experts sur les regroupements d'objets que cette base permet de définir, ainsi que sur ses lacunes.

Détermination des parties indépendantes

Le problème de classification étant résolu pour les familles définies par R7, R13 et R14, nous retirons ces objets de l'ensemble à classifier et continuons l'analyse sur la base de règle Br4 ci-dessous.

Base de règles de regroupement : Br4

R1	**(** L/D = { ≠}	**;**	NA	**)**
R8	**(** L/D=0 FE=7	**;**	D	**)**
R10	**(** L/D=0 FE=7 FO=6	**;**	A	**)**
R11	**(** L/D=0 FE#7 FO=6	**;**	A	**)**
R16	**(** L/D=0 FE=0 FI=0 US=0	**;**	A	**)**

L'analyse de la fonction d'appartenance experte met en évidence quatre sous-ensembles indépendants C1, C2, C3 et C4. En examinant l'utilité des règles sur les ensembles de couples (Ci, Cj), il apparaît que :

- la règle R1 donne des informations sur les couples composés des objets issus de C1, C2, C3 et C4 ;
- les règles R8, R10, R11 et R16 donnent des informations sur les couples composés des objets issus des composantes C1 et C2.

Cette situation s'interprète comme il suit :

- C1 : correspond à l'ensemble des objets tels que L/D=0
- C2 : correspond à l'ensemble des objets tels que L/D=0 et FE=7
- C3 : correspond à l'ensemble des objets tels que L/D=1
- C4 : correspond à l'ensemble des objets tels que L/D=2

La distinction C1UC2, C3, C4 est due à R1 qui sépare les objets en fonction de la valeur de la variable L/D. La règle R8 définit les deux composantes C1 et C2, car dans l'ensemble des objets tels que L/D=0, elle isole les objets tels que L/D=0 et FE=7. Les règles R10, R11 et R16 apportent des informations sur des sous-ensembles d'objets tels que L/D=0 et portent donc uniquement sur les objets des composantes C1 et C2. Pour étudier l'information apportée par les règles sur les objets de C1 et C2, nous représentons ci-dessous la fonction d'appartenance experte sur le code projeté, restreinte à l'ensemble C1∪C2. Ce tableau ne représente pas des couples d'objets, mais des couples de codes projeté qui font eux-même référence à plusieurs objets. C'est pourquoi la diagonale de ce tableau ne comprend pas systématiquement des 1. Pour plus de clarté, nous omettrons le symbole « - ».

		C1					**C2**											
	Tap	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	1						0	0	0	0	0	0	0	0	0	0	0	0	0
	2						0	0	0	0	0	0	0	0	0	0	0	0	0
C1	3						0	0	0	0	0	0	0	0	0	0	0	0	0
	4						0	0	0	0	0	0	0	0	0	0	0	0	0
	5				C11	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	**1**	C21											
	7	0	0	0	0	0		**1**	**1**	C22									
	8	0	0	0	0	0		**1**	**1**										
	9	0	0	0	0	0				**1**	**1**	**1**							
	10	0	0	0	0	0				**1**	**1**	**1**	C23						
C2	11	0	0	0	0	0				**1**	**1**	**1**							
	12	0	0	0	0	0													
	13	0	0	0	0	0													
	14	0	0	0	0	0													
	15	0	0	0	0	0													
	16	0	0	0	0	0													
	17	0	0	0	0	0													
	18	0	0	0	0	0													

Fig. V.3 Fonction d'appartenance experte sur l'ensemble des codes projetée de C1∪C2

Sur ce tableau, on distingue les composantes C1 et C2 qui sont indépendantes, ainsi que quatre groupes d'objets indissociables : C11, C21, C22 et C23. A ce stade de l'analyse il est possible d'interroger de façon systématique les experts sur la pertinence des regroupements observés afin de compléter le tableau d'appartenance expert (cf . IV.3.3.3 et (De Guio 1999)). A titre d'exemple, nous présentons ci-dessous l'organisation d'un questionnaire d'analyse de la composante C1.

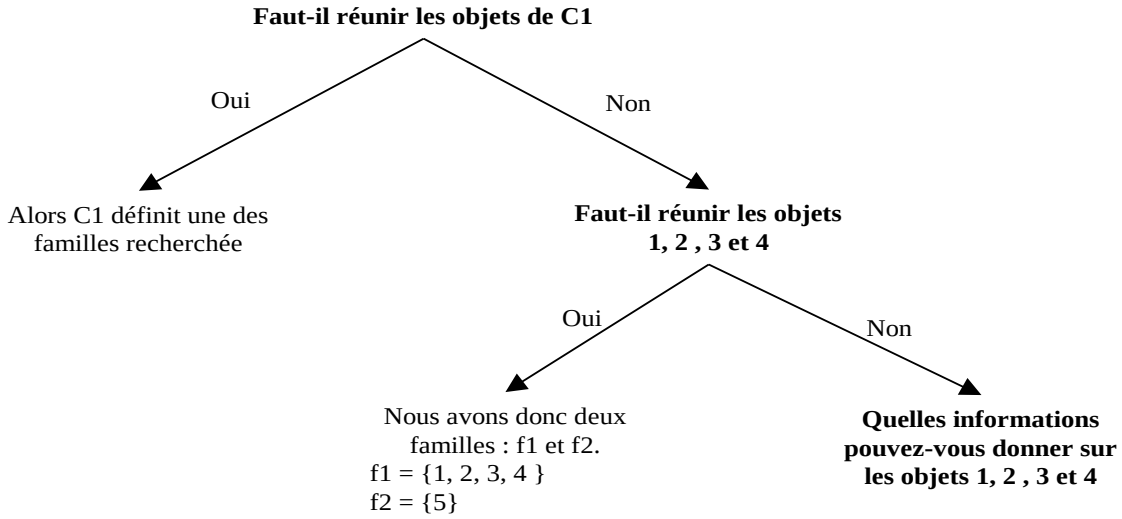


Fig V. 3 Questionnaire d'analyse de la composante C1

La méthode proposée pour analyser la fonction d'appartenance experte permet donc dans un premier temps d'identifier les familles complètement définies par la base de règle. Si les experts valident ces regroupements, alors le problème de classification est résolu pour les objets considérés. Sur l'application que nous traitons, il apparaît en examinant la partition finale que les sous composantes C1, C22 et C23 correspondent chacune à une famille de la partition finale. Dans un deuxième temps, cette méthode permet de concentrer l'analyse sur les lacunes de la base de règle. Si l'information est suffisante, il est possible de guider les experts pour compléter la base de règle. Le cas échéant, il faudra procéder à une nouvelle itération du cycle de classification.

V.2.7 Cohérence des règles de regroupement et de codification

En comparant la fonction d'appartenance experte avec la fonction d'appartenance des règles de codification, le système révèle que R4 est incohérente avec les règles R7, R13 et R14. La mesure de la redondance indique que les composantes en 1 de ces règles sont incluses dans la composante en 1 de R4.

Cette situation s'interprète comme il suit : la règle R4 a tendance à ressembler les objets tels que L/D= 0 et FI=1 ou FI=4. Les règles R7, R13 et R14 définissent des familles incluses dans cet ensemble d'objets. Elles apportent donc une information plus précise, complémentaire de R4.

V.2.8 Conclusion de l'évaluation de la base de règle

A l'issue de la première itération, le traitement à l'aide de SYCLASCE des connaissances supplémentaires émises par les experts lors de l'étape de validation a permis :

- d'identifier les règles inutiles et redondantes ;
- de détecter les règles incohérentes et d'identifier précisément les objets pour lesquels ces règles sont incohérentes ;
- d'identifier les familles totalement définies par la base de règle et de les valider directement ;
- de guider la démarche d'investigation des groupes d'objets insuffisamment spécifiés.

V.3 Conclusion

Nous avons présenté dans ce chapitre l'architecture du système SYCLASCE et une application dans le domaine de la classification des pièces mécaniques.

Suite à cette application, nous constatons que les méthodes et les concepts développés dans SYCLASCE permettent d'utiliser efficacement les connaissances supplémentaires fournies par les experts lors de l'étape de validation. L'évaluation selon les critères d'utilité, de redondance et de cohérence fait rapidement apparaître les défauts de la base de règles. L'analyse de la base de règles met en évidence les familles potentielles et permet de simplifier le problème de classification en sous problèmes indépendants. Le travail de construction des familles sur les sous-ensembles indépendants réduit considérablement la complexité du processus de classification pour les experts du domaine. Le travail d'évaluation de la base de règles aide les experts à structurer leurs connaissances. Les modifications apportées consistent généralement à restreindre la portée des règles initiales. Ce processus est caractéristique de la formulation d'un savoir-faire. Les experts oublient souvent de préciser les conditions d'application de leurs connaissances. Ils ont tendance à énoncer des généralités en omettant les cas particuliers. Lors de la correction, le système met en évidence les incohérences. Il oriente l'attention vers les « faiblesses » de la première formulation des règles et contribue à structurer la connaissance experte. Il apparaît donc que les concepts et méthodes proposés au chapitre IV contribuent à résoudre la principale difficulté rencontré pour réaliser les itérations du cycle de classification : utiliser rigoureusement les connaissances supplémentaires des experts pour classifier les objets.

Les connaissances supplémentaires émises lors de l'étape de validation ne sont pas fortement dépendantes de l'échantillon utilisé. Celui-ci sert de « révélateur » de la connaissance experte. Ces derniers formulent les règles en utilisant un savoir-faire construit sur l'ensemble des objets qu'ils manipulent.

Le processus de classification ne permet pas forcément de déterminer l'intention de toutes les familles. L'objectif n'est pas de compléter le tableau d'appartenance, mais bien de corriger les « faiblesses » de l'outil de classification à l'aide des connaissances des experts, notamment, lorsque les données ne respectent pas l'hypothèse de similarité. Une fois les familles validées, le problème consiste à classer de nouveaux objets dans les familles existantes. Une première solution est d'utiliser l'indice de similarité corrigé pour un fonctionnement de type base de cas, en classant un nouvel objet dans la famille la plus proche. Une autre solution consiste à déterminer avec l'aide des experts l'intension des familles qui ne sont pas expliquées par la base de règles. Ce travail doit être mené avec les experts du domaine, car les familles n'admettent pas forcément d'intension simple sous la forme d'une conjonction d'attributs du code descriptif. Par exemple, certaines catégories de pièces ont pu être regroupées car elles sont fabriquées pour le même client. Il n'existe alors pas forcément de relation apparemment cohérente entre le code des pièces et la famille constituée. Il faudra sans doute définir un

nouvel attribut pour ces pièces. Mais cet attribut n'est pas nécessairement pertinent pour classifier d'autres pièces qui sont distribuées à plusieurs clients. On remarquera que le processus classique de classification automatique aurait obligé les experts à identifier directement cet attribut et celui-ci aurait faussé la mesure de similarité entre les autres objets. Les règles de classification permettent d'éviter ce travail de reconstruction des attributs en autorisant l'expert à formuler dans les termes qui lui sont le plus familier les associations d'objets qu'il effectue couramment.

Chapitre VI

Conclusion

La TG est un concept organisationnel qui a fait l'objet de nombreuses recherches cette dernière décennie. Malgré de nombreuses applications potentielles, peu d'entreprises en France s'appuient rigoureusement sur les principes de la TG. Ils peuvent être assez « lourds » à mettre en œuvre et leur exploitation n'est pas immédiate. L'essentiel de l'effort de recherche s'est orienté vers l'utilisation d'outils d'Analyse Typologique pour former les familles et réduire les délais d'implantation. Les outils d'Analyse Typologique résolvent assez bien l'étape de formation des familles à partir d'un code adapté. Mais leur utilisation pour l'implantation de la TG en conception ou en fabrication implique un délai de « mise au point ». Une fois réglés, ils s'avèrent efficaces et sont parfaitement adaptés pour des traitements similaires et répétitifs. En pratique le délai de mise au point de l'interface, entre la méthode de classification et le problème à analyser, peut être aussi long, voir plus long que le délai d'analyse avec d'autres méthodes que les méthodes de classification automatique.

Dans cette thèse nous avons défini un outil de Classification Automatique adapté à la problématique de la TG afin de réduire les délais de l'étude d'implantation. Nos travaux s'inscrivent dans le cadre de Classification Automatique, mais nous abordons une problématique qui a été assez peu explorée que nous avons baptisée Affinement de Connaissances par Classification Interactive.

Contributions

Notre première contribution a été de définir un système interactif de classification sous contraintes symboliques qui contribue à résoudre le problème de classification n°1 défini au chapitre I, figure I.4. Ce système prend en charge le cycle de classification caractéristique de l'utilisation des outils d'Analyse Typologique pour l'implantation de la TG en conception et en fabrication. Il permet d'analyser et d'exploiter les connaissances supplémentaires fournies par les experts lors de l'étape de validation. La systématisation du dialogue entre les utilisateurs et le système de classification accélère sensiblement le cycle de classification.
La méthode employée résout en partie un obstacle majeur à l'exploitation de la TG lié à l'utilisation de codes TG très généraux comme le code OPTIZ ou Multi M. Ces codes capitalisaient les critères de classification pertinents pour la TG, mais leur utilisation demandait une grande expérience et nécessitait une laborieuse étape de codage des pièces. L'utilisation de ces codes traduit une conception classique de la classification, selon laquelle l'essentiel de l'information réside dans la description des objets. Il est donc nécessaire de trouver une description adéquate qui permette de réaliser simplement les familles. La contribution proposée permet de s'affranchir en partie de la pertinence du code descriptif pour élaborer une classification intéressante. Il est, dès lors, possible d'utiliser directement les informations manipulées par les experts au cours de leur activité pour classifier les pièces. Par définition, ces informations sont généralement facilement disponibles.

Pour déterminer l'intension des familles, nous proposons une approche manuelle. Une fois les familles validées par les experts, il est généralement facile d'en déterminer l'intension. De plus, le code n'entretient pas forcément de relation directe avec l'intension des familles. Il est donc nécessaire de passer par les experts pour les déterminer et formuler les propriétés communes sous-jacentes aux familles.

Ce système contribue aussi à résoudre le problème de classification n°2 défini au chapitre I, figure I.4, caractéristique de la formation d'îlots. Si les îlots sont établis sur la base d'une mesure de similarité entre postes de travail, il suffit de définir les contraintes de proximité à l'aide des règles de classification, avant d'effectuer une classification automatique des données. Si les îlots sont construits en utilisant une technique de sériation sur la matrice articles × postes (De Guio 1999) les techniques de quadri décomposition (De Guio 1990) permettent de tenir compte de la matrice postes × postes dans le processus de classification. Il suffit donc comme précédemment d'utiliser les règles de classification pour définir les contraintes de proximités sur la matrice postes × postes, avant d'effectuer une Classification Automatique des données.

Notre deuxième contribution est d'avoir analysé et défini un cadre général pour résoudre les problèmes de validation liés à l'utilisation d'un Outil de Construction Automatique de Classification en Classification pour l'Extraction de Connaissances à partir de Données sur un domaine complexe et peu formalisé. Bien que couramment cité dans la littérature, ces problèmes semblent peu étudiés. D'une manière générale, la classification sert à définir et organiser des concepts sur un domaine de connaissances. Dans la tradition de Platonicienne, la classification repose sur le principe de la division logique. Il s'agit de construire une hiérarchie de propriétés qui définissent des concepts de plus en plus précis. C'est uniquement le choix des propriétés discriminantes et leur place dans la hiérarchie qui définit les groupes. A l'opposée, les outils de construction automatique de classification utilisent une méthode de

regroupement. Ils forment des groupes d'objets dont les descriptions se ressemblent et reposent sur un raisonnement inductif. L'ordre des propriétés qui permettent de décrire les objets n'a plus d'importance, mais les attributs utilisés pour décrire les objets doivent exprimer les propriétés communes aux instances d'un même concept pour que des groupes significatifs puissent apparaître. Dans un domaine complexe et peu formalisé, les experts sont généralement incapables de formuler a priori les attributs descriptifs pertinents capables de mettre les concepts en évidence. Le processus de classification prend une forme cyclique que nous avons appelé cycle de classification. La stratégie consiste à définir des propriétés, puis à tester à l'aide de l'outil, si elles font apparaître des groupes censés d'objets ayant des propriétés semblables. C'est en fait une méthode de recherche des attributs pertinents. Le cycle de classification est un processus de structuration des connaissances expertes et d'intégration des biais du système de classification automatique. De par la nature du savoir-faire des experts, le processus de structuration des connaissances est indispensable et nécessairement itératif. L'étape de « réglage » des outils de construction automatique de classification semble incontournable, par contre, il est possible de la simplifier. Les approches interactives existantes, prennent en charge le cycle de classification en aidant les utilisateurs à intégrer les biais de classification de l'outil utilisé.

Nous proposons de simplifier cette approche en épargnant aux experts du domaine l'intégration des principes de fonctionnement de l'outil de classification, afin qu'ils se consacrent essentiellement à la structuration de leurs connaissances. Le principe de cette approche est de mettre à la disposition des experts, un langage qui leur permette de corriger directement le fonctionnement du système de construction automatique de classification. Ce langage doit être adapté au type de connaissances supplémentaires que peuvent fournir les experts lors de l'étape de validation. En exploitant ces connaissances, il est possible « d'apprendre » au système comment les experts catégorisent les objets et d'aider les experts à tester la cohérence de leurs connaissances supplémentaires.

Dans le cas de la TG, nous avons développé un tel langage ainsi que des mécanismes d'exploitation. Les connaissances sont modélisées par des contraintes sur la classification solution. Nous les avons appelées des contraintes symboliques car les règles de classification sont des définitions en intention d'ensembles de couples d'objets ou de groupes d'objets. Cette approche généralise les méthodes de classification sous contrainte de contiguïté existantes. L'interprétation et l'exploitation de ces connaissances supplémentaires sont organisées dans un protocole d'interactivité qui guide les experts du domaine et améliore l'efficacité du cycle de classification. La base de règles s'enrichit au fur et à mesure des cycles. Elle permet de tester la cohérence de toute nouvelle connaissance et de capitaliser la façon dont les experts utilisent les attributs existants pour définir des familles.

Limites

Le système réalisé est un prototype qui admet un certain nombre de limites. Il repose sur un système de classification automatique conventionnel. Les objets sont représentés dans le format « attribut valeurs » de l'Analyse de Données. Le langage de représentation des connaissances supplémentaires est défini pour des variables nominales. Les mécanismes de vérification de la cohérence et d'exploitation de l'information sont conçus pour la seule structure de partition. Nous n'avons pas réussi à formaliser d'une manière satisfaisante l'ensemble des connaissances supplémentaires ; et l'expressivité du langage utilisé demande à être améliorée. L'étape d'analyse de l'information issue des règles de classification repose sur la manipulation en extension des fonctions d'appartenance. Même en utilisant le code projeté, la quantité de données mises en jeu peut rendre cette étape laborieuse. Dans l'état actuel des choses, le système n'est pas capable d'identifier les règles incriminées dans les incohérences qui apparaissent lors de la fermeture transitive de la fonction d'appartenance union. Cette recherche délicate doit être menée manuellement.

Perspectives

Pour améliorer l'efficacité du cycle de classification, il faut prendre en compte l'ensemble des connaissances disponibles sur le domaine et limiter les contraintes de codification liées au système de classification automatique. Un système complet doit prendre en compte des attributs ordinaux, numériques et multivalués et permettre de travailler sur des objets irréguliers. En utilisant les techniques de saturation développées en Classification Conceptuelle, il est possible d'utiliser des connaissances du domaine pour enrichir la description des objets en créant de nouveaux attributs à partir des attributs existants. L'idéal serait de mettre à la disposition des utilisateurs un formalisme pour définir les opérations de combinaisons entre attributs, comme les opérations mathématiques simples pour les attributs numériques et par exemple des règles d'inférences semblables à celles développées dans Cluster (Michalski 1983). En plus des règles d'inférences, le cycle de classification s'accompagne souvent de modifications manuelles de la description des objets : changement de certaines valeurs, suppression d'un attribut ou ajout d'un nouvel attribut indépendant. Pour faciliter ces opérations, l'utilisateur devrait disposer d'un environnement adapté basé sur les fonctionnalités d'un tableur comme dans l'approche développée par Thomas (Thomas 1996). Ces deux catégories d'opérations de modification des attributs doivent être prises en charge par un protocole d'interactivité. En utilisant une technique similaire à celle utilisée pour vérifier la cohérence des règles de codification et de regroupement, il est possible de tester la cohérence des règles d'inférences et des modifications manuelles de la description des objets, l'une par rapport à l'autre et aussi par rapport aux règles de classification. Les règles de classifications sont le résultat d'une première étude de la formalisation des connaissances supplémentaires que les experts sont capables de fournir au cours de l'étape de validation. L'expressivité du formalisme demande à être améliorée. Cela nécessitera de nombreuses

recherches, tant pour capitaliser les différentes catégories de connaissances supplémentaires que pour les formaliser.

La méthode développée consiste à construire l'extension des règles de classification pour les analyser. Nous avons déjà posé les bases d'une approche en intention pour la vérification de la cohérence des règles. Les incohérences proviennent d'intersection entre les descriptions élémentaires de certaines règles. Il est relativement simple de calculer l'intension de l'intersection entre deux descriptions élémentaires. Il suffit alors de demander aux experts s'il existe des objets qui vérifient cette intension. Cette méthode ouvre une perspective de recherche qui consiste à travailler uniquement sur des intensions, sans devoir construire un tableau de données. Le code permet de construire l'ensemble des descriptions. Au fur et mesure que des erreurs sont détectées et corrigées, il est possible de capitaliser de la connaissance sur les codes réels. En fin de processus, l'ensemble des codes réels peut être défini par l'ensemble des attributs et une base de règles assimilable à des contraintes sur l'espace des descriptions qui permet de savoir si un code correspond ou non à un objet réel. Ce type d'approche a déjà été partiellement développé par Ziani dans le cadre de la sélection des variables sur un ensemble d'objets symboliques (Ziani 1996) et a donné lieu à la thèse de Vautrain (Vautrain 2000) qui développe une approche en intension des règles de classification.

D'un point de vue méthodologique SYCLASCE a été élaboré sur la base d'un corpus de connaissances capitalisées au cours d'applications TG. Il est donc spécifiquement adapté à ce type d'application. Cependant les règles de classification ne sont pas définies par rapport à un élément caractéristique de la TG. Elles sont définies par rapport à l'objectif structurel d'une classification : regrouper ou séparer des objets. La méthode semble donc utilisable d'une manière générale en classification pour l'extraction de connaissances à partir de données sur un domaine complexe et peu formalisé. Bien entendu cet argument ne prendra du poids qu'appuyé de nombreuses expérimentations dans des domaines différents. Si les connaissances supplémentaires utilisées se recoupent, il est possible d'envisager un système interactif de classification sous contraintes symboliques généraliste utilisable pour une large gamme d'applications.

Un axe de recherche complémentaire au précédent consiste à étudier, s'il est possible d'étendre la définition interactive d'une relation à l'aide de contraintes symboliques à d'autres types de relations que les relations d'équivalences, notamment les relations d'ordres, les recouvrements, ou bien des structures plus complexes comme les hiérarchies.

Bibliographie

(AFIA 1998) AFIA 1998. Ingénierie des connaissances. Bulletin de l'AFIA. juillet 98.

(Alberdi 1993) E. Alberdi and D.H. Sleeman, 1993. Theory Refinement and Scientific Classification : A Study in the Domain of Plant Taxonomy. Workings Notes of MLNet Workshop on Machine Discovery. P. Edwards : p51-55.

(Alexis 1995) K. Alexis, F. Bouali, et al. 1995. EVA : modélisation et représentation de connaissances hétérogènes, guidées par les besoins en Explication Validation et Acquisition. In 2ème Conférence sur l'Evolution Artificielle (EA'95), Brest, France, p263-282, 1995.

(Amy 1996) B. Amy, 1996. Recherches et perspectives dans le domaine des réseaux connexionnistes. Rapport technique de l'équipe Réseaux de Neurones et d'Automates du laboratoire Leibniz, octobre 1996.

(Askin 1993) R.G. Askin and C.R. Standridge, 1993. Modeling and Analysis of Manufacturinf system. New York, John Wiley and Sons, 1993.

(Batagelj 1998) V. Batagelj and A. Ferligoj, 1998. Constrained Clustering Problem. In Advances in Data Science and Classification, p137-144. Springer-Verlag

(Bisdorff 1995) R. Bisdorff, S. Laurent, et al. 1995. Knowledge Engineering with CHIP Application to a Production Scheduling Problem in the Wire-Drawing Industry. Luxembourg, Cellule Statistique et Décision du Centre de Recherche Public Centre Universitaire, 1995.

(Bisson 1993) G. Bisson, 1993. KBG Induction de Bases de Connaissances en Logique des Prédicats. Paris, thèse de l'Université de Paris-Sud, 1993.

(Bisson 1997) G. Bisson and J.M. Gabriel, 1997. Construction de bases de données par catégorisation interactive. INRIA Rhône-Alpes, rapport d'activité du projet SHERPA, 1997.

(Blanché 1970) R. Blanché, 1970. La logique et son histoire d'Aristote à Russell. Paris, A. Colin, 1984.

(Bouchon-Meunier 1990) B. Bouchon-Meunier, S. Després, et al. 1990. Aspects de l'interface entre symbolique et numérique. 3e journées nationales PRC- GDR Intelligence artificielle. Paris-la-Défense, 1990.

(Bournaud 1996) I. Bournaud, 1996. Regroupement conceptuel pour l'organisation de connaissances. Paris, thèse de l'Université Pierre et Marie Curie, 1996.

(Burbidge 1975) J.L. Burbidge, 1975. Production Flow Analysis. Production Ingeneering (1975) : p742-752.

(Cadet 1998) B. Cadet, 1998. Psychologie Cognitive. Paris, Press Editions, 1998.

(Chandon 1981) J.L. Chandon and S. Pinson, 1981. Analyse Typologique Théorie et Applications. Paris, Masson, 1981.

(Choi 1991) M.J. Choi and W.E. Riggs, 1991. GT Coding and Classification Systems for Manufacturing Cell Design. Production and Inventory Management Journal. vol32 : p28-32.

(Clancey 1985) W.J. Clancey, 1985. Heuristic Classification. Artificial Intelligence, vol27 n°3 : p289-350.

(Cornuéjols 1996) A. Cornuéjols, 1996. Analogie, principe d'économie et complexité algorithmique. In 11èmes Journées Françaises d'Apprentissage (JFA'96), Sète, France, p233-247, May 1996.

(Craw 1992) S. Craw, D. Graner, et al. 1992. CONSULTANT: Providing advice for The Machine Learning Toolbox, in M. Bramer (ed.), Proceedings of the BCS Expert Systems '92 Conference, Cambridge University Press, Cambridge, UK, p5-23.

(David 1993) J.M. David, 1993. Second Generation Expert systems. Spinger Verlag, 1993.

(David 1977) R. David, B. Buchanan, et al. 1977. Production Rules as a Representation for a Knowledge Based Consultation Program. Artificial Intelligence, vol8 n°1 : p15-45.

(Davies 1987) Davies and Russell 1987. A Logical Approach to Reasoning by Analogy. 10th International Joint Conference on Artificial Intelligence, Milano, 1987.

(Decaestecker 1991) C. Decaestecker, 1991. Apprentissage en Classification Conceptuelle Incrémentale. Bruxelles, thèse de l'Université libre de Bruxelles, 1991.

(De Guio 1990) R. De Guio, 1990. Contribution à l'organisation d'ateliers en îlots de fabrication. Strasbourg, thèse de l'Université Louis Pasteur, 1990.

(De Guio 93) R. De Guio, M. Barth, et al. 1993. An Help for Solving Dilemmas Encountered in Flow Analysis. IEEE computer society press : p 45-61, 1993.

(De Guio 1998) R. De Guio, 1998. Mémoire d'habilitation à diriger des recherches. Strasbourg, Université Louis Pasteur, 1998.

(De Guio 1999) R. De Guio and M. Barth, 1999. Cell Formatio Using Production Flow Analysis. Handbook on Cellular Manufacturing. J. Wiley, 1999.

(De Guio 1999) R. De Guio, V. Laget, et al. 1999. Approche intégrée d'exploitation de similarités en conception. APII - JESA. vol33 : p1251-1287.

(De Guio 1999) R. De Guio and M. Barth, 1999. Gestion Industrielle Polycopié de cours et support d'étude de cas, ENSAIS, LRPS, 1999.

(DeSieno 1988) DeSieno, 1988. Adding a Conscience to Competitive Learning. 2nd annual IEEE International Conference on Neural Networks. IEEE press : p117-124, 1988.

(Dictionnaire 1986) Dictionnaire 1986. Dictionnaire de la langue française. Paris Larousse, 1986.

(Dictionnaire-de-psychologie 1993) Dictionnaire-de-psychologie. Grand dictionnaire de la psychologie. Paris, Larousse, 1993.

(Diday 1971) E. Diday, 1971. Une nouvelle méthode en classification automatique et reconnaisance des formes. Revue de Statistiques Appliquées. vol19 n°2.

(Dieng 1990) R. Dieng, 1990. Méthodes et outils d'acquisition des connaissances. INRIA-SOPHIA ANTIPOLIS rapport de recherche n°1319, novembre 1990

(Encyclopédie-de-philosophie 1990) Encyclopédie-de-philosophie, 1990. Encyclopédie philosophique universelle : Les notions philosophiques. Paris : Presses universitaires de France, 1990.

(Erbeja 1995) T. Erbeja, R. De Guio, V. Laget, 1995. Classification sous contraintes symboliques . Troisièmes rencontres de la Société Francophone de Classification, SFC'95, p41-42, Namur 28-29 Septembre 1995 :.

(Erbeja 1996) T. Erbeja, R. De Guio, V. Laget, 1996. Classification Under Symbolic Constraints . Vingtième conférence annuelle de la Société Allemande de Classification, JFKL'96, p27-28, Freiburg 6-8 Mars 1996.

(Erbeja 1997) T. Erbeja, R. De Guio, V. Laget, 1997. A clustering approach for GT family formation problems. EDA' 97, First International Conference on Engineering Design and Automation, Bangkok, Thaïland March 18-21, 1997.

(Erbeja 1998) R. De Guio, T. Erbeja, V. Laget. Approche intégrée d'exploitation de similarité en conception, Journal Européen des Systèmes Automatisés, vol33 n°10 décembre 1998.

(Feltz 1993) J. Feltz, 1993. Etude de faisabilité technique d'une intégration de la Technologie de Groupe au Bureau d'Etudes. Strasbourg, mémoire de diplôme d'Ingénieur CNAM en Mécanique, présenté à l'ENSAIS au LRPS.

(Ferligoj 1983) A. Ferligoj and V. Batagelj, 1983. Some Types of Clustering with Relational Constraints. Psychometrika. vol 48 : p541-552.

(Fischer 1996) D. Fischer, 1996. Iterative Optimisation and Simplification of Hierarchical Clustering. Journal of Artificial Intelligence Research n°4 : p147-179.

(Fisher 1987) D.H. Fisher, 1987. Knowledge Acquisition via Incremental Conceptual Clustering. Machine Learning journal n°2 : p139-172.

(Forgy 1965) E.W. Forgy, 1965. Cluster Analysis of Multivariate Data : Efficiency vs Interpretability of Classifications". Biometrics, vol 21 : p768-769.

(Fortin 1989) C. Fortin and R. Rousseau, 1989. Psychologie Cognitive Une approche de traitement de l'information. Sainte-Foy (Québec) Canada, Télé Université, 1989.

(Foucault 1966) M. Foucault, 1966. Les mots et les choses. Paris, Gallimard, 1966.

(Frey 1990) G. Frey, 1990. Contribution au développement de modules de classification automatique pour la TG. Strasbourg, mémoire de DEA du Laboratoire de Recherche en Productique de Strasbourg (LRPS).

(Friedman 1993) H.P. Friedman, 1993. Cluster analysis in and out of context. Summary of invited talk, IFCS-93, Paris.

(Fritzke 1994) B. Fritzke, 1994. Growing Cell Structure - a Self Organizing Network for Unsupervised and Supervised Learning. Neural Networks, vol7 n°9 : p1441-1460.

(Gabriel 1999)J.M. Gabriel, 1999. Un modèle collaboratif pour le dialogue entre un système d' apprentissage empirique inductif et son utilisateur. Grenoble, thèse de l'Institut National Polytechnique de Grenoble, Grenoble.

(Galloüin 1988) J.F. Galloüin, 1988. Transfert de connaissances : systèmes experts techniques et méthodes. Paris, Eyrolles, 1988.

(Gluck 1985) M. Gluck and J. Corter, 1985. Information Uncertainly and the Utility of Categories. Seventh annual Conference of the Cognitive Science Society.

(Godin 1995) R. Godin, G. Mineau, et al. 1995. Méthodes de classification conceptuelle basées sur les treillis de Galois et applications. Revue d'Intelligence Artificielle, vol9: p105-137.

(Goldberg 1994) D.E. Goldberg, 1994. Algorithmes génétiques, exploration, optimisation et apprentissage automatique. Paris: Addison-Wesley, 1994.

(Gordon 1996) A.D. Gordon, 1996. A Survey of Constrained Classification. Computational Statistics & Data Analysis. vol21 : p17-29.

(Hammadi-Mesmoudi 1995) F. Hammadi-Mesmoudi, 1995. Classifieur neuronal d'images de télédétection. Strasbourg, thèse de l'Université Louis Pasteur.

(Han 1986) C. Han and I. Ham, 1986. Multiobjective Cluster Analysis for Part Family Formations. Journal of Manufacturing system, vol 5 n° 4 : p223-230.

(Hanson 1986) S.J. Hanson, 1986. Regroupement conceptuel et catégorisation : combler la lacune entre l'induction et les modèles de causalité. Apprentissage Symbolique une approche de l'Intelligence Artificielle, tome II, Toulouse, Cépaduès-Editions, 1994.

(Holland 1962) J.H. Holland, 1962. Outline for a Logical Therory of Adaptative Systems. Journal of the Association for Computing Machinery, vol3: p297-314.

(Kandiller 1994) L. Kandiller, 1994. A Comparative Study of Cell Formation in Cellular Manufacturing Systems. International Journal of Production Research, vol34: p919-946.

(Kettaf 2000) F.-Z. Kettaf and J.-P.A.d. Beauville, 2000. Des Algorithmes évolutionnaires pour la classification automatique. La revue de Modulab, juin 2001 : p23-46.

(Ketterlin 1995) M.A. Ketterlin, 1995. Découverte de Concepts Structurés dans les Bases de Données. Strasbourg, thèse de l'Université Louis Pasteur.

(Kodratoff 1986) Y. Kodratoff, 1986. Leçons d'Apprentissage Symbolique Automatique. Toulouse, Cépaduès-Editions, 1986.

(Kodratoff 1991a) Y. Kodratoff and E. Diday, 1991. Induction symbolique et numérique à partir de données. Toulouse, Cépaduès-Editions, 1991.

(Kodratoff 1991b) Y. Kodratoff, 1991. Bases terminologiques de l'Intelligence Artificielle, Techniques et documentations, Lavoisier, 1991.

(Kodratoff 1992) Y. Kodratoff, D. Sleeman, et al. 1992. Building a machine learning toolbox, in B. L. Pape and L. Steels (eds), Enhancing the Knowledge Engineering Process - Contributions from Esprit, Elsevier, p81-108.

(Kodratoff 1997) Y. Kodratoff, 1997. L'extraction de connaissances à partir de données : un nouveau sujet pour la recherche scientifique. Revue Electronique sur l'Apprentissage par les Données, vol1 n°1, Juin 1997, p1-28.

(Kohonen 1989) T. Kohonen, 1989. Self-Organization and associative memory. Berlin, SpringerVerlag, 1989.

(Kusiak 1987) A. Kusiak, 1987. "The Generalized Group Technology." Int. J. Prod. Res. vol 25 n°4 : p561-569.

(Laget 1994) V. Laget, 1994. Modèle d'un processus de classification intégrant la connaissance experte. Strasbourg, mémoire de DEA du Laboratoire de Recherche en Productique de Strasbourg.

(Lagrange 1973) M.-S. Lagrange 1973. Analyse sémiologique et histoire de l'art : examen critique d'une classification. Editions Klincksieck, 1973.

(Lebart 1978) L. Lebart 1978. Programme d'agrégation avec contraintes. Les cahiers de l'Analyse de Données : p275-287.

(Lebowitz 1994) M. Lebowitz L'utilité de l'apprentissage fondé sur des similarités dans un monde nécessitant des explications. Apprentissage Symbolique une approche de l'Intelligence Artificielle, tome II, Toulouse, Cépaduès-Editions, 1994.

(Lechevalier 1980) Y. Lechevalier, 1980. Classification sous contraintes. Optimisation en Classification Automatique. INRIA, Paris : p677-693.

(Lefébure 1998) R. Lefébure and G. Venturi, 1998. Le Data Mining. Paris, Eyrolles, 1998.

(Lévine 1989) P. Lévine and J.C. Pomerol, 1989. Systèmes interactifs d'aide à la décision et systèmes experts. Paris, Hermés, 1989.

(MacKusick 1991) K.B. McKusick and P. Langley, 1991. Constraints on Tree Structure in Concept Formation. Proceedings of the Twelfth International Conference on Artificial Intelligence, Sydney, Australia. Morgan Kaufmann : p810-816.

(Macqueen 1967) J. B. MacQueen, 1967. Some methods of classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, p281-297, 1967.

(Marcotorchino 1982) F. Marcotorchino and P.Michau, 1982. Agrégation de similarité en Classification Automatique. Revue de Statistique Appliquée, volXXX : p22-44.

(Michalski 1991) R.S. Michalski and E. Bloedorn, 1991. Constructive Induction from Data in AQ17-DCI, Artificial Intelligence Center George Mason University. Reports of the Machine Learning and Inference Laboratory, MLI91-12, George Mason University.

(Michalski 1983) R.S. Michalski and E. Stepp, 1983. Learning from Observation : Conceptual Clustering. Machine Learning: An Artificial Intelligence Approach, Berlin, Springer-Verlag", 1984 : p 331-363.

(Michalski 1981) R.S. Michalski and R.E. Stepp, 1981. An application of IA techniques to Structuring Objetcs Into a Conceptual Hierarchy. Proceedings of the Seventh International Joint Conference on Artificial intelligence (IJCAI), Vancouver, Canada, August 24-28, 1981.

(Michalski 1983) S.R. Michalski, 1983. A Theorie and Methodolgy of Inductive Learning. Machine Learning an Artificial Intelligence Approach, Berlin, Springer-Verlag, 1984 : p83-129.

(Michalsky 1991) R.S. Michalsky, 1991. Searching for Knowledge in a Word Flooded with Facts. Applied Stochastic Models and Data Analysis, vol7 : p153-166.

(Murtagh 1985) F. Murtagh, 1985. A Survey for Algorithms for Contiguity-Constrained Clustering and Related Problems. The Computer Journal, vol28 : p82-88.

(Mutel 1984) B. Mutel, 1984. Reconnaissance de groupements technologiques par des méthodes d'analyse de données. International Congress on Productic and Robotic, Bordeaux, 1984.

(Mutel 1993) B. Mutel, 1993. La Technologie de Groupe. Support de cours. Strasbourg, ENSAIS.

(Mutel 1992) B. Mutel, L. Bouzid, et al. 1992. Application of Conceptual Learning Techniques to Generalized Group Technologie. Applied Artificial Intelligence, vol6 : p443-458.

(Nadif 1987) A. Nadif, 1987. Contribution à la classification automatique des données de production en Technologie de Groupe. Metz, thèse de l'Université de Metz.

(Nadif 1985) A. Nadif, M. Constantini, et al. 1985. Mesures de ressemblance de gammes de fabrication. RAIRO APII, vol19 n°5 : p455-470.

(Napoli 1996) A. Napoli, J.F. Mari, et al. 1996. Aspects de la classification. Rapport de recherche de l'INRIA n°2909, juin 1996..

(Nedellec 1995) C. Nedellec, 1995. APT, un système d'apprentissage coopératif. Acquisition et Ingénierie des Connaissances : tendances actuelles. Coordinateurs: N. Aussenac-Gilles, P. Laublet, C. Reynaud. Toulouse, Editions Cépaduès. 1996. p 307-328

(Offodile 1992) F.O. Offodile, M. Abraham, et al. 1992. Cellular Manufacturing : a Taxonomic Review Framework. Journal of Manufacturing Systems. vol13 : p193-200.

(Offodile 1991) O.F. Offodile, 1991. Application of Similarity Coefficient Method to Parts Coding and Classification Analysis in GT. Journal of Manufacturing Systems. vol10 : p442-448.

(Patil 1993) P. Patil, D.C. Brown, et al. 1993. Intelligent data analysis systems. Journal of Intelligent Manufacturing. vol.4, sep 1993 : p121-137.

(Pellegrin 1982) P. Pellegrin, 1982. La classification des animaux chez Aristote. Statut de la biologie et unité de l'aristotélisme. Paris, Les Belles Lettres, 1982.

(Plaquin 1998) M.F. Plaquin, 1998. Contribution des algorithmes évolutionnistes à la constitution d'îlots de fabrication. Productique. Clermont Ferrand, thèse de l'Université Blaise Pascal.

(Richard 1998) J.-F. Richard 1998. Les activités mentales Comprendre, raisonner, trouver des solutions. Paris, Armand Colin, 1998.

(Roux 1996) B.L. Roux and J. Thomas 1996. Raffinement d'une méthode de résolution de problèmes et engagement ontologique. Java 96 : journées acquisition, apprentissage 96), Sète 1996.

(Saporta 1990) G. Saporta, 1990. Probabilités Analyse de données et statistiques. Technip, 1990.

(Sedqui 1995) A. Sedqui, 1995. Nouvelles approches pour la Classification des Gammes de Production. Lyon, thèse de l'Institut National des Sciences Appliquées de Lyon.

(Seifoddini 1986) H. Seifoddini and P. Wolfe, 1986. Application of the Similarity Coefficient Methode in Group Technology. IIE transactions, September 1986 : p271-277.

(Sharker 1996) B.R. Sharker, 1996. The Resemblance Coefficient in Group Technology A Survey and Comparative Study of Relational Metrics. Computer & Industrial Engineering. vol30 : p103-116.

(Singh 1993) N. Singh, 1993. Design of Cellular Manufacturing Systems : An Invited Review. European Journal of Operational Research. vol69 : p284-291.

(Spears 1993) W.M. Spears, K.A. Dejong, et al. 1993. An Overview of Evolutionary Computation. Proceedings of European Conference on Machine Learning (ECML-93), Vienna, Austria, April 1993. Lecture Notes in Artificial Intelligence 667, Berlin Springer Verlag, 1993 : p442-459.

(Stepp 1986) R.E. Stepp and R.S. Micalsky, 1986. Classification Conceptuelle : Apprentissage à partir d'observations et classification d'objets structurés avec objectif. Apprentissage Symbolique une approche de l'Intelligence Artificielle, tome I, Toulouse, Cépaduès-Editions, 1994.

(Sutcliffe 1995) J.P. Sutcliffe, 1995. Mécanisme logique pour décider de ce qui relève ou non de la classification. Bulletin de la Société Française de Classification, n°4, mars 1995.

(Thomas 1996) J. Thomas, 1996. Vers l'intégration de l'apprentissage symbolique et de l'acquisition des connaissances basée sur les modèles : le système ENIGME. Paris, thèse de l'Université Pierre et Marie Curie.

(Vakharia 1994) A.J. Vakharia and H.M. Selim, 1994. Group Technology. Handbook of Design, Manufacturing and Automation. R.C. Dorf and A. Kusiak, John Wiley : p435-460.

(Vanderpooten 1990) D. Vanderpooten, 1990. L'approche interactive dans l'aide multicritère à la décision. Paris, thèse de l'Université Dauphine.

(Vautrain 2000) F. Vautrain, 2000. Analyse des contraintes expertes en Classification Automatique. Paris, thèse de l'Université Dauphine.

(Vogel 1988) C. Vogel, 1988. Génie Cognitif. Paris, Masson, 1988.

(Wang 1995) J.W. Wang, 1995. Système cognitif d'extraction automatique de règles de décision. Paris, thèse de l'Ecole Nationale Supérieure des Télécommunications.

(Ziani 1996) D. Ziani 1996. Sélection de variables sur un ensemble d 'objets symboliques. Paris, thèse de l'Université Dauphine.

Annexe II.1 : L'Analyse Typologique

AII.1.1 La représentation des objets en Analyse Typologique

La données initiale est celle d'un ensemble d'objets.

Ensemble des (noms) des objets : $\Omega = \{ \omega i \mid i \in I \}$

Les objets sont caractérisés par des attributs. Chaque attribut est représenté par une variable, c'est à dire une fonction de l'ensemble des objets dans l'ensemble des valeurs possibles pour l'attribut considéré. Les valeurs d'un attribut s'appellent des modalités.

Les variables descriptives yj : $yj : \Omega \rightarrow Oj$
$w \rightarrow yj(w)$
Ensemble des variables : $Y = \{ yj \mid j \in J \}$

L'espace des descriptions Δ est l'ensemble de toutes les descriptions qu'il est possible de faire en combinant systématiquement les modalités des variables descriptives. Il est construit en faisant le produit des ensembles de valeurs possibles pour chaque attribut (Silva, 191 #129). Pour rester cohérent avec le vocabulaire utilisé en TG, nous appellerons codes, les éléments de Δ.

Espace des descriptions : $\Delta = O1 \times ... \times Om = \{ \delta k \mid k \in K \}$
Code : $\delta = (\delta 1, ..., \delta m)$

La fonction de codage est une application de l'ensemble des objets dans l'espace des descriptions. Elle associe à chaque objet le code composé des modalités que les variables descriptives associent à l'objet considéré.

Fonction de codage : $c : \Omega \rightarrow \Delta$
$w \rightarrow c(w) = (y1(w), ... , ym(w))$

Certains codes de l'espace des descriptions sont effectivement l'image par la fonction de codage d'un ou plusieurs objets de Ω. Ces codes sont dis réels, par opposition aux codes virtuels auxquels ne correspond aucun objet de Ω. Un code réel est aussi appelé la description d'un objet.

Ensemble des codes réels : $C = c(\Omega) = \{ cl \mid l \in L \}$
Ensemble des codes virtuels : $V = \Delta - C = \{ vm \mid m \in M \}$

Plusieurs objets peuvent avoir le même code. Pour ne pas perdre cette information, il est d'usage de travailler non pas sur l'ensemble des codes réels, mais sur un tableau de données. Le tableau de données admet les objets en ligne et les variables descriptives en colonne. Une donnée est donc un couple (objet, code) et le tableau de données l'ensemble de ces couples. Souvent le tableau de données est constitué de couples (poids, code). Dans ce cas, le poids représente la proportion d'objets ayant la description c.

Une donnée : **li = (wi, c(wi))**
Le tableau de données : **Td = { li = (wi, c(wi)) | i∈I }**

ExempleAII.1.1 La codification des objets en classification automatique.

Soit Ω ={w1; w2; w3; w4} un ensemble de quatre objets caractérisés par les variables binaires y1 et y2. On dispose des tableaux de données Td et Td'. Td est un tableau du type objet – code. Td' est un tableau du type poids - code

Td	y1	y2
w1	0	1
w2	0	1
w3	1	0
w4	1	1
w5	1	1

Tableau de données Td

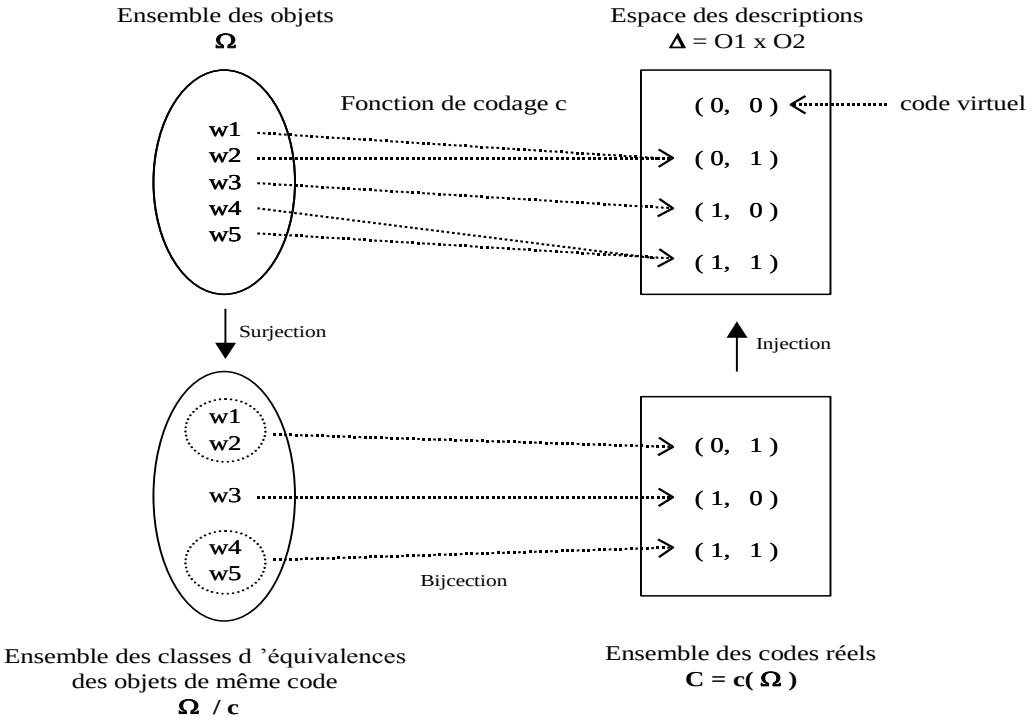
y1 : Ω → O1={ 0;1}
y2 : Ω → O2={ 0;1}

Variables descriptives

Td'	y1	y2
2/5	0	1
1/5	1	0
2/5	1	1

Tableau de données Td'

Les différents éléments liés à la codification des objets de l'ensemble Ω, s'organisent conformément au schéma de décomposition canonique de l'application c.



AII.1.2 Les indices de similarités utilisés en Classification Automatique

Un indice de similarité est une mesure de la ressemblance entre deux objets. Le caractère subjectif de la notion de ressemblance explique la variété des indices utilisés en Analyse Typologique. Certains systèmes utilisent aussi une distance ou une mesure de dissimilarité. La dissimilarité est une mesure de la différence entre deux objets. Une distance est un type de dissimilarité particulier.

Un indice de dissimilarité prend sa valeur minimale lorsque l'on mesure la différence d'un objet avec lui-même. Il est symétrique, car l'on suppose qu'un objet A diffère autant à un objet B, que l'objet A diffère de l'objet B.

Un indice de dissimilarité est une application de ΩxΩ dans **R** telle que

$\forall (w, w') \in \Omega x\Omega$ $\quad d(w, w') \geq 0$

$d(w, w') = d(w', w)$

$d(w, w) = 0$

Il existe plusieurs façons de transformer un indice de similarité en dissimilarité. La plus simple est de complémenter la similarité à sa valeur maximale :

$$d(w, w') = \max_{w1, w2 \in \Omega x\Omega} d(w1, w2) - is(w, w')$$

Distance Euclidienne	$d(w1,w2) = \sqrt{\sum_k (V_k(w1) - V_k(w2))^2}$	variables métriques
Distance rectangulaire	$d(w1,w2) = \sum_k \lvert V_k(w1) - V_k(w2) \rvert$	variables métriques
Distance de Minkowski	$d(w1,w2) = \left[\sum_k (V_k(w1) - V_k(w2))^r \right]^{1/r}$	variables métriques
Indice de Jacquard	$s(w1,w2) = \frac{a}{a+b+c}$	variables binaires
Indice de Ochiaï	$s(w1,w2) = \frac{a}{\sqrt{a+b+c}}$	variables binaires
Indice de Russel et Rao	$s(w1,w2) = \frac{a}{a+b+c+d}$	variables binaires

Tableau **AII.1.1** Quelques mesures de similarité et de dissimilarité

Les variables binaires sont des variables nominales à valeur dans {0, 1}. 1 indique la présence d'une caractéristique, 0 en indique l'absence. Les mesures de similarité définies sur les variables binaires sont construites en combinant quatre nombres associés à tout couple d'objets (w1, w2).

a : le nombre de caractéristiques communes ou concordance de 1.
b : le nombre de caractéristiques possédées par w1 et pas par w2.
c : le nombre de caractéristiques possédées par w2 et pas par w1.
d : le nombre de caractéristiques possédées ni par w1, ni par w2 ou concordance de 0.

Il n'y a pas a proprement parler de similarité spécifique aux variables nominale. On utilise généralement une similarité sur des variables binaires, après avoir transformé les attributs initiaux en attributs de type présence absence.

Exemple AII.1.2 Tableau de données, tableau disjonctif complet et tableau de similarité

Considérons l'ensemble Ω des 6 objets décris par le tableau de données Td ci-dessous. V1, V2 et V3 sont des variables nominales à valeur dans l'ensemble {0, 1, 2}. Le tableau disjonctif complet Tc est obtenu en considérant chaque modalité des attributs comme des variables binaires de type présence absence. L'indice de similarité comptabilise les caractéristiques communes (concordance de 1).

Td	v1	v2	v3
1	0	0	2
2	0	0	0
3	0	1	1
4	1	2	1
5	1	1	1
6	1	2	2

	v1			**v2**			**v3**		
Tc	0	1	2	0	1	2	0	1	2
1	1	0	0	1	0	0	0	0	1
2	1	0	0	1	0	0	1	0	0
3	1	0	0	0	1	0	0	1	0
4	0	1	0	0	0	1	0	1	0
5	0	1	0	0	1	0	0	1	0
6	0	1	0	0	0	1	0	0	1

Ts	1	2	3	4	5	6
1	3	2	1	0	0	1
2	2	3	1	0	0	0
3	1	1	3	1	2	0
4	0	0	1	3	2	1
5	0	0	2	2	3	1
6	1	0	0	1	1	3

Pour interpréter les mesures a, b, c et d, il faut considérer que les modalités des variables V1, V2 et V3 sont des caractéristiques que les objets sont susceptibles d'avoir ou pas. Par exemple, si l'on considère le couple d'objets (1, 2) les mesures a, b, c, d prennent les valeurs suivantes :

a(1, 2) = 2 : les objets 1 et 2 possèdent deux caractéristiques communes : V1= 0 et V2=0.
b(1, 2) = 1 : l'objet 1 possède la caractéristique V3=2 que ne possède pas l'objet 2.
c(1, 2) = 1 : l'objet 2 possède la caractéristique V3=0 que ne possède pas l'objet 1.
d(1, 2) = 5 : les objets 1 et 2 ne possèdent pas les caractéristiques suivantes : V1=1, V1=2, V2=1, V2=2 et V3=1.

AII.1.3 Exemples de critères locaux utilisés pour l'algorithme de la C.A.H

Les méthodes basées sur l'algorithme de Classification Ascendante Hiérarchique (C.A.H.) optimisent un critère local. Il s'agit d'une distance entre groupes d'objets. Il est local, car il ne fait intervenir que les similarités entre les objets des deux groupes à comparer. On passe d'un niveau de la hiérarchie au suivant en regroupant les deux classes les plus proches. Les critères utilisés biaisent le résultat. Le critère du plus proche voisin privilégie l'isolation au détriment de l'homogénéité. Inversement, le critère du voisin le plus éloigné crée des classes homogènes, mais mal isolées.

Critère du plus proche voisin	$D(C1,C2) = \underset{\substack{w1 \in C1 \\ w2 \in C2}}{\mathrm{Min}}\ d(w1, w2)$
Critère du voisin le plus éloigné	$D(C1,C2) = \underset{\substack{w1 \in C1 \\ w2 \in C2}}{\mathrm{Max}}\ d(w1, w2)$
Critère de Ward	$D(C1,C2) = \frac{\lvert C1\rvert \lvert C2\rvert}{\lvert C1\rvert + \lvert C2\rvert}\, d^2(\, g(C1),\, g(C2)\,)$

g(C) : centre de gravité de la classe C

Tableau AII.1.2 Quelques distance entre classes utilisées en Classification Hiérarchique Ascendante

AII.1.4 Exemples de critères globaux utilisés en Classification Hiérarchique

Les méthodes utilisant un critère global reposent sur l'équivalence entre une hiérarchie indicée et un type de distance particulier : les ultramétriques. Le critère à optimiser est une mesure de « l'écart » entre le tableau de similarité initial et une distance ultramétrique définie sur le même ensemble d'objets.

Critère de l'écart absolu	$C(Ts, Tu) = \sum_{w1,\, w2 \in \Omega} \lvert\, s(w1, w2) - u(w1,w2)\,\rvert$
Critère des moindres carrés	$D(Ts, Tu) = \left[\sum_{k} (\, s(w1, w2) - u(w1, w2)\,)^2 \right]^{1/2}$

u : distance ultramétrique
Tu : tableau récapitulatif des distances ultramétrique u entre toutes les paires d'objets

Tableau AII.1.3 Quelques critères globaux utilisés en Classification Hiérarchique

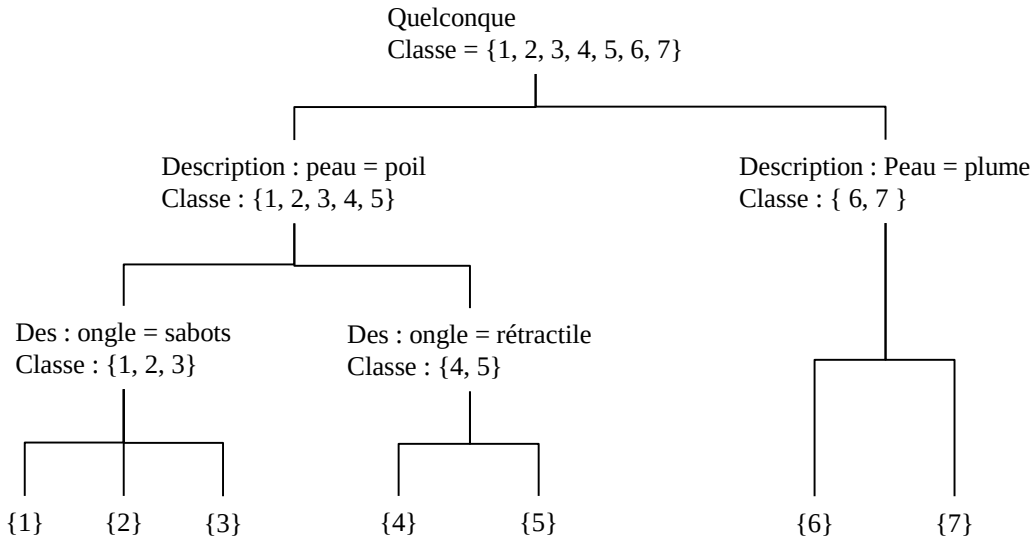
Annexe II.2 : La classification Conceptuelle

AII.2.1 Exemple de classification conceptuelle

La plupart des méthodes de Classification Conceptuelle construisent une structure hiérarchique à partir d'un ensemble d'observations. Les classes et leur description sont associées aux nœuds de la hiérarchie. Les arcs représentent la relation d'inclusion entre les classes.

Observation	Peau	Couleur	Poids	Ongle
1	poil	brun	450	sabot
2	poil	cendré	500	sabot
3	poil	noir	300	sabot
4	poil	blanc	10	rétractile
5	poil	gris	11	rétractile
6	plume	blanc	15	-
7	plume	blanc	19	-

Algorithme de Classification Conceptuel



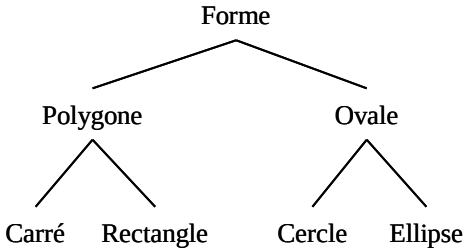
Sur l'exemple précédent, un système de Classification Conceptuelle a pu retrouver des concepts pertinents du domaine et les organiser. Au deuxième niveau de la hiérarchie en partant de la racine, on trouve deux concepts qui en première approximation recoupent la distinction entre le genre des oiseaux et celui des mammifères. Au troisième niveau, le système a subdivisé la classe des mammifères en deux sous classes qui recoupent la distinction entre les félidés et les plantigrades.

AII.2.2 La représentation des objets dans Cluster/2

Considérons l'ensemble des objets {e1, e2, e3} décris par les variables taille, forme et couleur :

taille : est une variable linéaire à valeur dans {1, 2, 3} ;
poids : est une variable linéaire à valeur dans {1, 2, 3, 4, 5};
couleur : est une variable nominale à valeur dans { rouge, vert, bleu} ;
forme : est une variable structurée à valeur { carré, rectangle, cercle, ellipse, polygone, ovale}.

Les valeurs de la variable forme définissent la hiérarchie ci dessous :



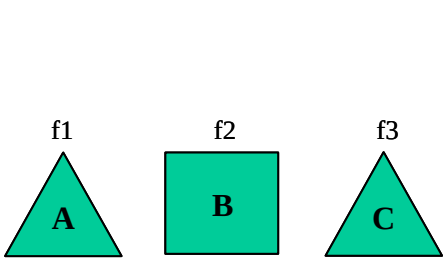
Comme en Classification Automatique les objets sont décris dans un tableau de données :

Objets	taille	poids	couleur	forme
e1	1	5	rouge	cercle
e2	2	2	vert	ellipse
e3	1	4	vert	carré

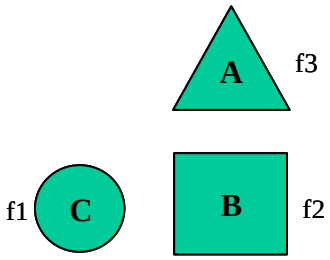
Il existe de nombreuses façons de tenir compte des attributs structurés pour regrouper des objets (Kodratoff 1986). D'une manière qualitative, Cluster regroupe les objets sur la base du nombre de descriptions qui incluent les objets dans leur extension. Deux objets qui possèdent des attributs communs dans la hiérarchie des modalités d'un attribut structuré admettent plus de descriptions en commun et ont plus de chances d'être regroupés.

AII.2.3 Représentation relationnelle et représentation tabulaire d'un objet structuré

Le langage APC utilisé dans Cluster/S est une extension de la logique du premier ordre qui permet de représenter des objets structurés. Un objet est dit structuré lorsqu'il est composé de plusieurs parties et qu'il existe des relations entre ces parties. Les représentations de type attribut valeur s'avèrent insuffisantes pour manipuler ce type d'objets. Car il est nécessaire de définir un attribut pour chacun des couples des relations considérées. Le nombre d'attribut devient vite très important. Par exemple, considérons des objets composés de formes élémentaires : carré, rond, triangle, tels que ces formes élémentaires soient alignées ou bien superposées. Pour décrire et pouvoir différencier ces objets, il faut préciser d'une part, les formes élémentaires qui composent chaque exemple, d'autre part, la position relative des composants.



Objet 1



Objet 2

Représentation dans un langage de type attribut valeur :

Objet	f1	f2	f3	f1 avant f2	f1 avant f3	f2 avant f3	f1 dessus f2	f2 dessus f1	f1 dessus f3	f3 dessus f1	f2 dessus f3	f3 dessus f2
1	triangle	carré	triangle	oui	oui	oui	non	non	non	non	non	non
2	rond	carré	triangle	oui	non	non	non	non	non	non	non	oui

Représentation dans un langage relationnel :

objet 1 : triangle(A) carré(B) triangle(C) avant(A,B) avant(B,C)
objet 2 : triangle(A) carré(B) rond (C) avant(C,B) dessus(A,B)

Les langages relationnels sont beaucoup plus riches que les langages de type attribut valeurs, mais ils sont aussi beaucoup plus exigeant d'un point de vue algorithmique. A première vue, le langage APC permet de décrire des objets homogènes, irréguliers et multivalués. Cela n'est pas dû directement à l'aspect relationnel du langage. Généralement ces langages décrivent les objets par une liste de prédicats. Cette liste varie d'un objet à l'autre. Les objets ne sont pas réguliers. APC admet des sélecteurs comme [taille = $1 \vee 2$] qui correspondent à des attributs multivalués.

AII.2.4 Les connaissances du domaine dans Cluster/S

Cluster/S prend en compte deux types de connaissances du domaine : des règles d'inférences et un réseau de dépendance des objectifs (RDO).

Les règles d'inférences permettent de construire de nouveaux descripteurs et ainsi d'enrichir la description des objets. Considérons par exemple, les objets e1 et e2 initialement décris par les attributs forme, volume et poids.

	Forme	**Volume**	**Poids**
e1	cube	5	2
e2	sphère	3	1

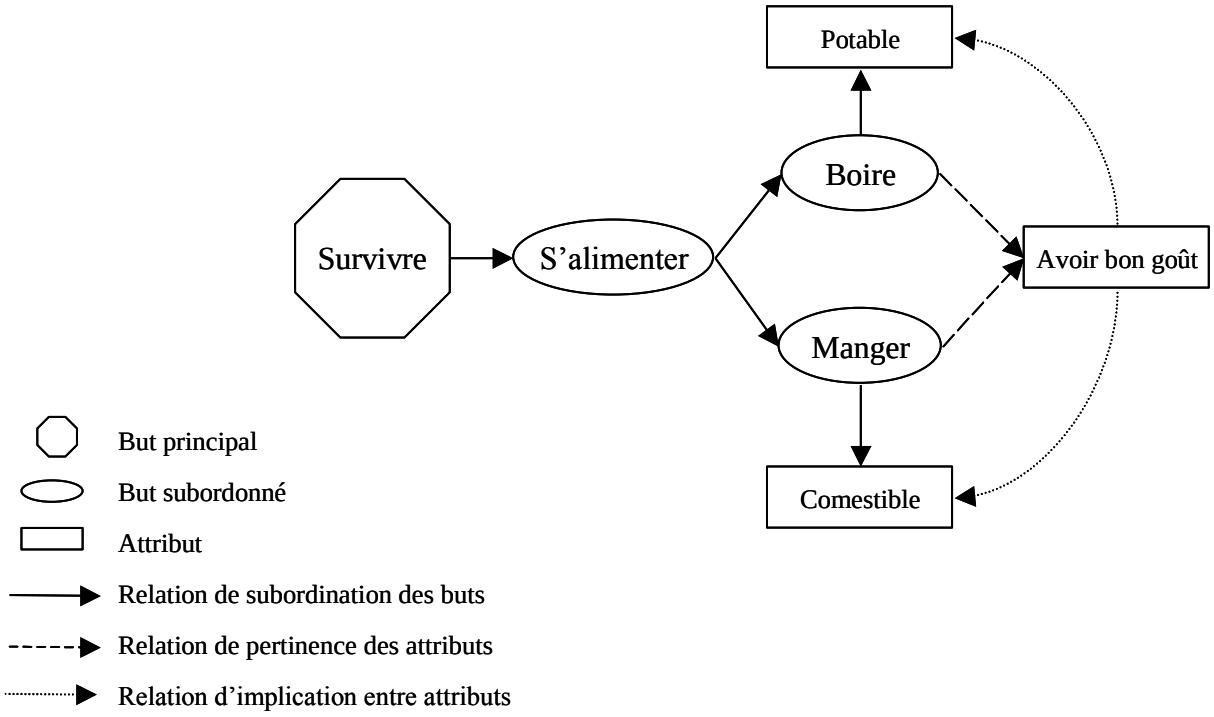
Si l'utilisateur fournit les règles d'inférence suivantes :

∀ objet, masse volumique (objet) = (volume(objet)/ poids(objet))
∀ o1, o2, o3, [au dessus(p1, p2)] [au dessus(p2, p3)] ⇒ [au dessus(p1, p3)]
∀ objet [forme(objet) = cube] ⇒ [forme(objet) = polyèdre]

Alors le système peut compléter les descriptions de e1 et e2, en utilisant les règle 1 et 3. La règle 2 ne s'appliquent pas sur ces deux exemples.

	Forme	**Volume**	**Poids**	**Masse volumique**	**Forme 2**
e1	cube	5	2	5/2	polyèdre
e2	sphère	3	1	3	-

Le réseau de dépendance des objectifs associe les buts de la classification aux descripteurs pertinents. Ce réseau permet de sélectionner les descripteurs pertinents quand ils existent et de choisir les règles d'inférence susceptibles de créer des descripteurs intéressants. Par exemple, si le but d'un agent est de survivre, il lui fait dans un premier temps s'alimenter. S'alimenter inclut deux sous buts : celui de boire et celui de manger. Il faut donc que l'eau soit potable et les aliments comestibles. Une classification des éléments de l'environnement élaborée dans le but de survivre distinguera dans les objets sur la base de deux attributs qui sont potable et comestibles. Les auteurs de Cluster/S représentent ce type d'enchaînement qui va de l'objectif de la classification à la définition des attributs par un graphe qui s'appelle le réseau de dépendance entre objectifs (RDO). Les nœuds correspondent aux buts, sous buts et attributs. Les arcs représentent les relations de subordination entre les buts, les sous buts et les attributs, ainsi que les relations d'implication entre les attributs.



Considérons que si les objets ont bon goût alors ils sont comestibles. Si l'utilisateur fourni au programme Cluster/S, le RDO précédent, alors le système va classer les objets en utilisant les attributs potables et comestibles. Si ces attributs n'existent pas, le programme essayera de les créer en utilisant l'attribut avoir bon goût. Les liens d'implication entre attributs correspondent en fait à des règles d'inférence.

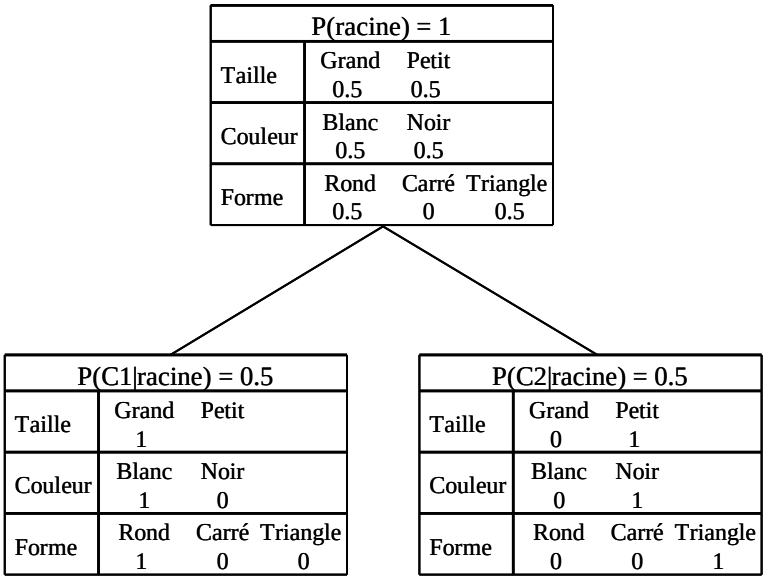
$\forall$ objet, [avoir bon goût (objet)] [liquide(objet)] $\Rightarrow$ [potable(objet)]
$\forall$ objet, [avoir bon goût (objet)] [solide(objet)] $\Rightarrow$ [comestible(objet)]

Le concept de RDO permet de tenir compte explicitement des buts de la classification. Il faut cependant connaître à priori, l'ensemble des attributs pertinents. Avec un RDO, le problème de classification se ramène presque à un problème de tri. Cette approche est particulièrement intéressante si l'on connaît les attributs pertinents en fonction de l'objectif. Elle permet de tenir compte d'un point de vue dans le processus de classification, mais elle n'apporte pas d'éléments de réponse au problème fondamentale qui est de définir les attributs pertinents. Elle ne permet pas non plus de représenter des règles de classification. Ce type de connaissances supplémentaires correspondrait à un réseau de dépendance des attributs. C'est à dire qu'en fonction de la valeur de certains attributs, d'autre deviennent important ou pas pour comparer les objets.

AII.2.5 Structure de données et représentation des classes dans Cobweb

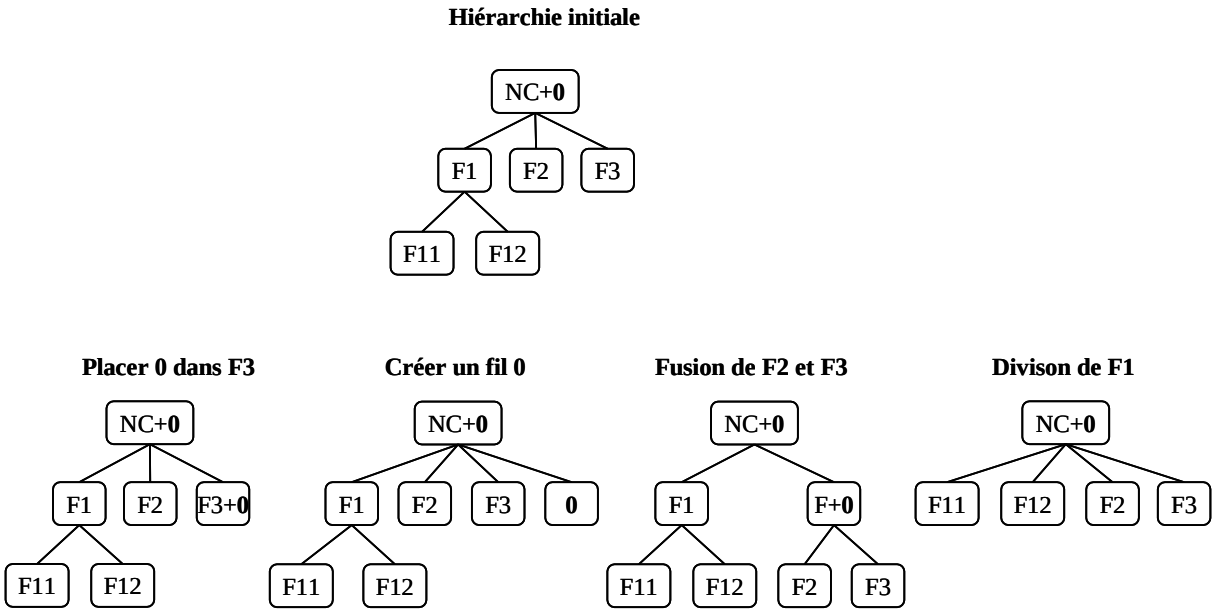
A partir de la liste des observations {o1, o2} présentées dans cet ordre, le système Cobweb élabore la hiérarchie ci-dessous. Les nœuds sont numérotés dans l'ordre de création.

	Taille		**Couleur**		**Forme**		
	Grand	Petit	Blanc	Noir	Rond	Carré	Triangle
o1	1		1		1		
o2		1		1			1



AII.2.6 Les opérateurs de restructuration dans Cobweb

Pour lutter contre les problèmes de minima locaux propres aux méthodes de gradients, ainsi que la sensibilité à l'ordre de présentation des observations, le programme utilise des opérateurs de restructuration : placer, créer, fusion et division.



Annexe IV.1 : Corpus de règles de classification

B1 est une base de règles représentative des connaissances supplémentaires que les experts sont capables de formuler lors de l'étape de validation du cycle de classification. Les règles de classification se répartissent en deux catégories principales : les règles de regroupement et les règles de codification. **Les règles de regroupement** portent sur les objets et indiquent comment les regrouper. Toutes les règles de B1 sont des règles regroupement, sauf les règles n° 2, 3, 4, 15, 17 et 18. **Les règles de codification** portent sur les modalités des variables descriptives. Elles indiquent la prépondérance de certaines combinaisons de valeurs pour comparer les objets. Ce sont les règles 2, 3 et 4. La règle 15 n'est pas une règle de classification. Elle définit un groupe d'objet ne devant pas être classifiés. Une requête dans la base de données permet de les retirer de l'ensemble des objets à classifier. Les règles 17 et 18 doivent être reformulées de façon à ce que les experts caractérisent en terme d'attributs les objets désignés par la phrase : « les éléments de la famille à laquelle elles ont été affectées par la classification ». La base B1 ci-dessous est définie par rapport aux variables descriptives L/D, FE, FI US et FO.

1 Deux pièces ayant des modalités L/D différentes ne peuvent pas appartenir à la même famille.

2 Si L/D=0 alors FE=2 ou FE =5 est plus important que FE=2 ou FE=4 pour comparer deux pièces

3 Si L/D=0 alors FE=2 ou 5 est pus important que FI=1 ou 4 pour déterminer la similarité de deux pièces.

4 Si L/D=0 alors les modalités FI=1 et FI=4 peuvent être groupées en une seule FI=1ou4 pour comparer deux pièces

5 Si L/D=0 alors les pièces telles que FE =1 ou 4, FE=0, FI=2 et FO=5 ne peuvent pas cohabiter au sein d'une même famille

6 Les pièces telles que L/D=0 et FE=0 et FI=1 et US=0 et FO=0 doivent participer à une même famille.

7 Les pièces telles que L/D=0 et FE=0 et FI=1 et US=0 et FO=0 définissent une famille.

8 (FE=7) et (L/D=0) est une fonction discriminante pour les familles.

9 Les pièces telles que L/D=0 et FO=6 doivent appartenir à la même famille.

10 Les pièces telles que L/D=0 et FO=6 et FE=7 doivent appartenir à la même famille.

11 Les pièces telles que L/D=0 et FO=6 et non FE=7 doivent appartenir à la même famille.

12 Les pièces telles que L/D=0 et US=6 appartiennent à la même famille.

13 Les pièces telles que L/D=0 et FI=1 et FE=1 et US=0 et FO=0 forment une famille.

14 Les pièces telles que L/D=0 et (FE=4ou1) et FI=1 et FO=2 et US=0 forment une famille.

15 Les pièces telles que L/D=0 et FI=4 et FE=4 et FO=0 et US =0 ne doivent pas être classifiée (trop spécifiques).

16 Les pièces telles que L/D=0 FI=0 et FE=0 et US=0 doivent appartenir à la même famille.

17 Les pièces telles que L/D=0 et FI=0 et FE=1 et US=7 et FO=0 ne doivent pas cohabiter avec les éléments de la famille à laquelle elles ont été affectées par la classification.

18 Les pièces telles que L/D=0 et FI=4 et FE=1 et US= 0 et FO=0 ne doivent pas cohabiter avec les éléments de la famille à laquelle elles ont été affectées par la classification.

Annexe IV.2 : Compatibilité d'une relation d'appartenance et d'une partition

Une fonction d'appartenance est compatible avec une partition s'il est possible de remplacer les " - " du tableau d'appartenance par des " 1 " ou des " 0 " de façon à trouver une relation d'équivalence. D'une manière plus formelle :

U est compatible avec une partition	⇔	**Il existe une partition P telle que** $\underline{\text{💣}}(Ar \oplus P) = \varnothing$

Nous considérons que la fonction d'appartenance Ar ne contient aucune incohérence. Dans ces conditions, Ar donne des informations pour l'ensemble des couples auxquels elle associe les valeurs 1 ou 0. Cette information peut être enrichie en considérant que l'objectif de Ar est de définir une partition de l'ensemble des objets. Considérons les trois opérateurs définis au chapitre IV.4.3.3.2 :

FR : la fermeture réflexive d'une fonction d'appartenance
FS : la fermeture symétrique d'une fonction d'appartenance
FT : la fermeture transitive d'une fonction d'appartenance

1 : Si l'application sur Ar de l'un de ces opérateurs engendre des erreurs, alors quelque soit P une partition de Ω : $\underline{\text{💣}}(Ar \oplus P) \neq \varnothing$. Car :

- si $\underline{\text{💣}}(FR(Ar)) \neq \varnothing$ alors Ar comprend au moins un 0 sur la diagonale qui sera en contradiction avec les 1 de n'importe quelle partition;
- si $\underline{\text{💣}}(FS(Ar)) \neq \varnothing$ alors Ar comprend au moins un couple (w1,w2) tel que Ar(w1,w2)=1 et Ar(w2,w1)=0 ou bien tel que Ar(w1,w2)=0 et Ar(w2,w1)=1. Toute partition est symétrique et sera donc en contradiction avec un de ces couples.
- si $\underline{\text{💣}}(FT(Ar)) \neq \varnothing$ alors Ar comprend au moins un triplet (w1,w2,w3) qui ne respecte pas la transitivité. Un des couples engendré par ce triplet sera forcément incohérent avec une partition.

2 : Inversement, si aucune de ces fermetures ne présente d'incohérence, c'est donc que la restriction de Ar à l'ensemble E des couples auxquels elle associe les valeurs 1 ou 0 est une relation d'équivalence. Il existe donc une partition telle que $\underline{\text{💣}}(Ar \oplus P) = \varnothing$. Par exemple, il suffit de considérer l'ensemble des partitions suivant :

Soit Ar_E : la restriction de Ar à l'ensemble E
Soit $P_{\Omega-E}$: une partition définie sur l'ensemble Ω-E

$Ar_E \cup P_{\Omega-E}$ est une partition telle que $\underline{\text{💣}}((Ar_E \cup P_{\Omega-E}) \oplus P) = \varnothing$.

Annexe IV.3 : Les fermetures transitives d'une fonction d'appartenance

Soit : Ar : une fonction d'appartenance.
F1 : la fonction d'inférence de la 1-transitivité
F0 : la fonction d'inférence de la 0-transitivité

L'étude ci-dessous apour objectif de définir l'ordre d'application de F1 et F0 sur Ar afin d'évaluer la compatibilité d'une fonction d'appartenance avec une partition.

On remarque facilement que F1(F0(Ar)) ≠ F0(F1(Ar). Par exemple, considérons le triplet d'objet (a, b, c) et le tableau d'appartenance suivant :

Appliquée sur Ar :
F1 génère une incohérence pour le couple (a, b)
F0 génère soit une incohérence pour le couple (a, c) ou bien pour le couple (b, c).

Les incohérences ne permettent pas de faire des inférences. Donc, si l'on calcule la fermeture transitive d'une fonction d'appartenance en appliquant d'abord F0, l'on obtiendra des solutions qui sont fonction de l'ordre de lecture des informations du tableau d'appartenance. Par contre, si l'on applique d'abord F1, alors F0 ne génère plus d'incohérences et la solution est unique. En effet, F0 génère des incohérences si et seulement si :

∃(a, b, c) tels que Ar(a, b)=0, Ar(b, c)=1 et Ar(a, c)=1.

Or si F1 a déjà été appliquée sur la fonction d'appartenance AR, ce type de configuration n'existe plus. L'information Ar(a, b)=0 est considérée comme incohérente. Les fonctions d'inférences et les opérateurs d'unions ne permettent pas de faire des inférences à partir d'une information incohérente. C'est pourquoi, nous calculerons la fermeture transitive des fonctions d'appartenance en appliquant successivement la fermeture 1-transitive puis la fermeture 0-transitive.

Annexe V.2 : Tableau de données

La partition Pi résulte d'une sériation sur le code principal avec alpha=0,h3
La partition Pf a été réalisées manuellement par les experts à partir de Pi. C'est la partition recherchée.

Chaque objet est nommé par un numéros.
Pour faciliter notre travail, nous renumérotons les objets en les triant par ordre croissant sur les familles de F2, et appellons Ω l'ensemble des objets ainsi nommés.

Ω	Désignation	Plan	PRINCIPAL					ADD.			SUPPL.			Pf	Pi
1	Bouchon M16x1	3U061	0	2	0	3	0	0		0	0	0	0	0	0
2	Couvercle sup. GV	3U297	0	4	4	3	4	5	0	5	1		5	0	0
3	Bague de centrage	3U298	0	1	1	1	5	5		5	3		3	0	0
4	Anneau	3U438	0	4	1	0	0	4	6	0	0	0	3	0	6
5	Rampe de graissage	3U538	2	2	1	0	1	0		0	3		2	0	9
6	Vis d'ancrage	3U734	2	5	0	0	0	1	1	0	9	2	1	0	10
7	Collet de butée	4J249	0	4	1	0	0	2	2	5	1		3	0	6
8	Collet de butée	4J682	0	4	1	0	0	5	2	5	2	1	3	0	3
9	Plaquette de frein	4J763	0	1	1	3	4	2		0	0	0	2	0	0
10	Gicleur 80 Sp	5E623	2	4	1	0	0	0		1	2		5	0	16
11	Flasques S14-4	6B063	0	0	4	4	2	6	0	9	2	2	4	0	0
12	Bague S14-4	6B066	0	0	1	7	0	3	0	5	0	0	1	0	6
13	Moyeu S14-4	6B070	0	4	4	6	2	5	0	5	4	3	6	0	0
14	Bague	6B089	0	0	1	7	0	4	0	5	1	0	1	0	6
15	Couvercle GV	3U183	0	1	2	1	2	4		1	1		3	1	1
16	Couvercle GV	3U299	0	1	2	9	2	4	5	0	2	1	2	1	1
17	Ecrou	3U451	0	0	2	3	3	3	0	0	1		1	1	0
18	Ecrou GV	4J143	0	0	2	0	2	2	0	1	0		1	1	2
19	Ecrou M72x2	4J327	0	1	2	3	1	3	0	1	1	0	1	1	0
20	Ecrou M64x2 D	4J509	0	0	2	0	2	3	0	1	0		1	1	2
21	Ecrou	4J571	0	0	2	0	2	4	0	1	3		1	1	2
22	Ecrou GV	4J575	0	0	2	0	2	3	0	1	1	1	1	1	2
23	Ecrou GV	4J721	0	0	2	0	2	3	0	0	0	1	1	1	1
24	Ecrou GV2	4J766	0	0	2	0	2	4	0	0	1	1	1	1	2
25	Ecrou R S14-4	6B067	0	0	2	0	2	3	0	5	0	0	1	1	2
26	Couvercle GV	3U210	0	0	0	0	2	4	0	0	2	1	0	2	2
27	Couvercle	3U497	0	0	0	0	2	3	6	1	0	1	0	2	2
28	Capot PV 22	3U503	0	0	0	0	2	5	6	1	0		0	2	2
29	Couvercle PV	3U602	0	0	0	0	2	6	6	0	0	2	0	2	2
30	Couvercle GV	3U603	0	0	0	0	2	5	0	0	0	1	0	2	2
31	Couvercle PV	3U640	0	0	0	0	2	4	6	1	1		0	2	2
32	Couvercle PV	3U643	0	0	0	0	0	2	6	0	0		0	2	6
33	Couvercle GV	3U644	0	0	0	0	0	3	6	0	0		0	2	6
34	Tole sur PV32	3U777	0	0	0	0	1	4		1	0	1	0	2	0
35	Couvercle PV	3U896	0	0	0	0	2	6	0	0	0		0	2	2
36	Entretoise	3U256	0	1	1	0	2	6	6	5	3	4	3	3	3
37	Déflecteur	3U301	0	1	1	0	0	5	0	2	1		2	3	5
38	Faux moyeu	3U576	0	1	1	0	2	5	0	9	2	3	2	3	3
39	Support pompe	3U590	0	1	1	0	2	6	1	5	1		2	3	3
40	Bague	4H031	0	1	1	0	2	5	0	1	2	2	2	3	3

Ω	Désignation	Plan	PRINCIPAL					ADD.			SUPPL.			Pf	Pi
41	Bague	4H489	0	1	1	0	2	5	2	1	1	2	2	3	3
42	Bague	4H724	0	1	1	0	2	5	2	1	1	1	2	3	3
43	Douille	4J272	0	1	1	0	0	4	3	5	2	0	2	3	3
44	Douille	4J273	0	1	1	0	0	3	3	5	2	0	2	3	3
45	Bague	4J330	0	1	1	0	2	5	0	1	0	1	2	3	3
46	Douille	4J432	0	1	1	0	0	3	3	5	2	0	2	3	3
47	Douille	4J708	0	1	1	0	0	3	3	5	2		2	3	3
48	Disque d'arret	4J724	0	1	1	0	2	4	2	1	1	2	2	3	3
49	Douille	4J729	0	1	1	0	0	3	3	5	3		2	3	3
50	Adaptateur S14-4	6B064	0	1	4	0	2	5	0	9	2	2	5	3	3
51	Bride HPS 16	6B086	0	1	4	0	2	6	0	9	2	2	5	3	3
52	Collier HPS 16	6B088	0	1	1	0	2	6	0	9	1	1	3	3	3
53	Bague HPS 16	6B100	0	1	1	0	2	4	0	1	1	1	2	3	3
54	Forge moyeu	8H645	0	1	1	0	0	5		0	4	4	3	3	3
55	Forge adaptateur	8H647	0	1	1	0	0	4		0	3	1	3	3	3
56	Forge bride	8H672	0	1	1	0	0	6		0	2	3	3	3	3
57	Forge roue GV2	8H704	0	1	1	0	0	6		0	5		2	3	3
58	Forge roue menée	8H713	0	1	1	0	0	6		0	3		2	3	3
59	Entretoise	3U382	0	0	1	6	0	1	2	0	0		1	4	6
60	Entretoise	3U641	0	0	1	6	0	2	2	0	3		1	4	6
61	Entretoise	3U703	0	0	1	6	0	2	2	0	3		1	4	6
62	Entretoise vireur	3U864	0	0	1	6	0	2	2	2	2		1	4	6
63	Entretoise	3U926	0	0	1	6	0	2	1	5	2		1	4	6
64	Entretoise	6B083	1	0	1	6	0	2	2	0	2		1	4	0
65	Collet	4J266	0	7	1	0	0	3	2	5	1	1	2	5	6
66	Collet	4J274	0	7	1	1	3	5	3	5	2	2	5	5	13
67	Collet	4J275	0	7	1	1	4	5	3	5	2	2	5	5	13
68	Collet	4J278	0	7	1	1	3	4	3	5	2	1	5	5	13
69	Collet	4J326	0	7	1	1	4	4	3	5	2	1	5	5	13
70	Collet	4J341	0	7	1	1	0	5	0	5	2	0	3	5	13
71	Collet	4J349	0	7	1	1	4	4	3	5	2	1	4	5	13
72	Collet	4J350	0	7	1	1	0	4	3	5	2	1	4	5	13
73	Collet	4J416	0	7	1	1	0	5	3	5	3	1	4	5	13
74	Collet	4J419	0	7	1	1	0	5	3	5	3	2	5	5	13
75	Collet	4J502	0	7	1	1	0	3	3	5	2	1	4	5	13
76	Collet	4J564	0	7	1	1	0	6	3	5	3	3	5	5	13
77	Collet	4J699	0	7	1	1	0	5	0	9	2	1	5	5	13
78	Collet	4J700	0	7	1	1	4	5	0	9	2	1	5	5	13
79	Collet	4J703	0	7	1	1	0	4	0	9	2	1	5	5	13
80	Collet	4J704	0	7	1	1	9	4	0	9	2	1	5	5	13
81	Collet	4J711	0	7	1	1	0	6	0	9	2	2	5	5	13
82	Collet	4J714	0	7	1	1	0	5	0	9	2	1	5	5	13
83	Collet	4J727	0	7	1	1	0	6	3	5	3	3	3	5	13
84	Collet	4J787	0	7	1	1	0	4	0	9	1	1	5	5	13
85	Collet	4J788	0	7	1	1	0	4	0	9	1	1	5	5	13
86	Bague	3U302	0	0	1	0	0	3	2	5	2		1	6	6
87	Entretoise	3U303	0	0	1	0	0	3	0	5	0		1	6	6
88	Cale de réglage	3U442	0	0	1	0	0	5	6	0	2		1	6	6
89	Rondelle d'ancrage	3U733	0	0	1	0	0	5	6	0	1	2	1	6	6
90	Bague	4J195	0	0	1	0	0	3	0	5	0		1	6	6
91	Frette	4J211	0	0	1	0	0	4	3	2	1		2	6	6
92	Frette	4J213	0	0	1	0	0	4	3	2	2		2	6	6
93	Frette	4J240	0	0	1	0	0	5	3	2	2		2	6	6
94	Frette	4J241	0	0	1	0	0	4	3	2	2		2	6	6
95	Frette	4J242	0	0	1	0	0	3	3	2	2		2	6	6
96	Frette	4J417	0	0	1	0	0	5	3	2	2	1	2	6	6

Ω	Désignation	Plan	PRINCIPAL					ADD.			SUPPL.			Pf	Pi
97	Frette	4J420	0	0	1	0	0	5	3	2	2	1	2	6	6
98	Frette	4J504	0	0	1	0	0	3	3	2	1	0	2	6	6
99	Bloc Run-Out	4J505	0	0	1	0	0	2	2	5	0	0	1	6	6
100	Disque	4J520	0	0	1	0	0	2	2	5	0		2	6	6
101	Frette	4J565	0	0	1	0	0	5	3	2	2	1	2	6	6
102	Collet	4J677	0	0	1	0	0	2	0	5	0	0	1	6	6
103	Frette	4J701	0	0	1	0	0	5	0	2	2	1	2	6	6
104	Frette	4J705	0	0	1	0	0	3	0	2	1	1	2	6	6
105	Frette	4J712	0	0	1	0	0	5	0	2	2	1	2	6	6
106	Frette	4J715	0	0	1	0	0	4	0	2	1	1	2	6	6
107	Frette	4J791	0	0	1	0	0	4	0	2	1	1	2	6	6
108	Rondelle	6B112	0	0	1	0	0	1		0	0		1	6	6
109	Forge adaptateur	8H646	0	0	1	0	0	5		0	2	2	2	6	6
110	Forge protecteur	8H673	0	0	1	0	0	6		0	3	2	1	6	6
111	Forge bague	8H674	0	0	1	0	0	4		0	2	1	1	6	6
112	Forge collier	8H675	0	0	1	0	0	6		0	2	2	1	6	6
113	Forge bague	8H676	0	0	1	0	0	4		0	2	1	1	6	6
114	Couvrcle GV	3U323	0	4	1	0	2	5	6	1	2		3	7	7
115	Masse add. GV	3U756	0	0	1	0	2	5	0	9	1	1	2	7	2
116	Masse add. PV	3U757	0	0	4	0	2	6		9	5	6	3	7	2
117	Bride reduite	3U800	0	0	1	0	2	3	6	0	1	0	1	7	2
118	Diaphragme	3U836	0	0	1	0	2	4	6	1	0		1	7	2
119	Bague	4F526	0	0	1	0	2	5	1	1	1	2	1	7	2
120	Frette de Butée	4J072	0	0	1	0	2	4	0	5	2	1	1	7	2
121	Collet de butée	4J220	0	0	1	0	2	4	2	5	2		1	7	2
122	Plaque d'arret	4J333	0	0	1	0	2	5	0	5	1	1	1	7	2
123	Plaque d'arret	4J333	0	0	1	0	2	4	0	5	0	1	1	7	2
124	Bague	4J589	0	0	1	0	2	4	0	1	0	1	1	7	3
125	Frette de Butée	4J683	0	0	1	0	2	5	2	5	1	1	1	7	2
126	Bague de manchon	4J844	0	0	1	0	2	6	0	4	3	3	2	7	2
127	Protecteur Ext HPS-S14-4	6B069	0	0	4	0	2	5	0	9	1	1	3	7	2
128	Protecteur HPS 16	6B087	0	0	4	0	2	6	0	9	1	1	3	7	2
129	Bague filetée taraudé	3U470	1	2	2	0	0	1	1	0	1		1	8	8
130	Raccord	3U556	1	0	2	0	0	0		0	1	0	2	8	8
131	Raccord	3U557	1	0	2	0	0	1		0	1	0	2	8	8
132	Douille	4J668	1	4	2	0	0	1	2	1	2	0	4	8	8
133	Manchon TMBD 30	6B008	1	4	6	1	6	4	0	9	4	1	6	9	23
134	Manchon TMBD 30	6B009	1	4	6	1	6	4	0	9	4	1	6	9	23
135	Manchon d'acc. TMED 70	6B015	1	4	6	1	6	5	0	5	5	2	6	9	23
136	Manchon d'acc. TMED 60	6B016	1	4	6	1	6	5	3	9	5	1	6	9	23
137	Manchon TCD 90	6B040	0	1	6	1	6	6	0	9	3	3	3	9	4
138	Manchon TCD 90	6B041	0	1	1	1	6	6	0	9	3	3	3	9	13
139	Moyeu TFD 90	6B116	1	4	6	0	6	5	3	5	6	5	6	9	23
140	Moyeu TFD 90	6B117	1	1	6	0	6	5	3	5	6	5	6	9	24
141	Forge roue arbrée	8H613	2	4	0	0	0	7		0	8	7	6	10	10
142	Forge flexible	8H614	2	4	0	0	0	7		0	6	4	2	10	10
143	Forge pignon	8H617	2	4	0	0	0	5		0	8	6	6	10	10
144	Forge flexible PV	8H618	2	4	0	0	0	6		0	8	6	5	10	10
145	Forge torsible	8H621	2	1	0	0	0	5		0	8	6	1	10	10
146	Forge flexible	8H623	2	4	0	0	0	5		0	7	5	2	10	10
147	Forge flexible PV	8H624	2	4	0	0	0	7		0	9	8	3	10	10
148	Forge roue arbrée	8H625	2	4	0	0	0	7		0	8		6	10	10
149	Forge flexible GV	8H627	2	4	0	0	0	6		0	9	7	2	10	10
150	Forge pignon	8H628	2	4	0	0	0	4		0	7	4	3	10	10
151	Forge arbre de roue	8H629	2	4	0	0	0	5		0	8	6	6	10	10
152	Forge flexible PV	8H638	2	4	0	0	0	6		0	8	6	3	10	10

Ω	Désignation	Plan	PRINCIPAL					ADD.			SUPPL.			Pf	Pi
153	Forge flexible PV	8H639	2	4	0	0	0	6		0	8	7	3	10	10
154	Forge pignon	8H640	2	4	0	0	0	5		0	8	6	2	10	10
155	Forge roue arbrée	8H641	2	4	0	0	0	6		0	8	7	2	10	10
156	Forge roue arbrée	8H648	2	4	0	0	0	6		0	8	7	3	10	10
157	Forge flexible	8H653	2	4	0	0	0	6		0	8	6	2	10	10
158	Forge flexible PV	8H656	2	4	0	0	0	6		0	8	7	3	10	10
159	Forge pignon	8H657	2	4	0	0	0	5		0	8	7	4	10	10
160	Forge flexible	8H661	2	4	0	0	0	5		0	7	5	4	10	10
161	Forge flexible PV	8H664	2	4	0	0	0	6		0	8	7	3	10	10
162	Forge arbre de roue	8H667	2	4	0	0	0	6		0	8	7	6	10	10
163	Forge pignon	8H668	2	4	0	0	0	5		0	7	6	4	10	10
164	Forge roue arbrée	8H669	2	4	0	0	0	7		0	8	7	6	10	10
165	Forge arbre de roue	8H670	2	4	0	0	0	6		0	8	8	4	10	10
166	Forge roue arbrée	8H682	2	4	1	0	0	7		0	8	9	6	10	10
167	Forge flexible PV	8H683	2	4	0	0	0	6		0	8	7	3	10	10
168	Forge arbre pignon	8H685	2	4	0	0	0	6		0	8		3	10	10
169	Forge roue arbrée	8H686	2	4	0	0	0	7		0	8		2	10	10
170	Forge arbre de roue	8H687	2	4	0	0	0	6		0	8	7	6	10	10
171	Forge flexible PV	8H692	2	4	0	0	0	6		0	8	7	4	10	10
172	Forge flexible PV	8H693	2	4	0	0	0	6		0	8	7	4	10	10
173	Forge pignon	8H703	2	4	0	0	0	6		0	8		2	10	10
174	Forge arbre de roue	8H711	2	1	0	0	0	6		0	8	8	1	10	10
175	Forge arbre de roue	8H712	2	1	0	0	0	6		0	8	8	1	10	10
176	Forge arbre de roue	8H753	2	4	0	0	0	6		0	8	8	6	10	10
177	Goupille	4G056	2	0	5	1	0	1	0	0	3	0	2	11	11
178	Goupille	4G452	2	0	5	1	0	1	0	0	3	0	2	11	11
179	Goupille	4H703	2	0	5	1	0	0	0	0	3	0	2	11	11
180	Goupille	4J060	2	0	5	1	0	0	0	0	2	0	2	11	11
181	Flexible	4J225	2	4	5	1	4	5	0	9	8	5	6	12	12
182	Torsible	4J226	2	4	5	1	2	5	0	9	7	4	6	12	12
183	Flexible	4J318	2	4	5	4	4	5		9	8	6	8	12	14
184	Torsible	4J321	2	4	5	1	2	6	0	9	8	6	6	12	12
185	Flexible	4J400	2	4	5	1	2	6	0	9	8	6	7	12	12
186	Flexible	4J401	2	4	5	1	2	6	0	9	8	6	7	12	12
187	Torsible	4J557	2	4	5	1	2	5	0	9	7	4	6	12	12
188	Flexible	4J563	2	4	5	1	2	6	0	9	8	7	6	12	12
189	Flexible	4J581	2	4	5	1	2	6	0	9	8	6	7	12	12
190	Flexible	4J649	2	4	5	1	4	6	0	9	8	6	6	12	12
191	Flexible	4J759	2	4	5	1	2	6	0	6	8	6	7	12	12
192	Flexible	4J803	2	5	5	4	4	6	0	6	8	7	8	12	14
193	Roue	4J269	0	7	1	1	6	8	3	5	5	9	5	13	13
194	Roue	4J304	0	4	1	1	6	7	2	5	6	8	3	13	13
195	Roue	4J312	0	0	1	1	6	6	2	5	3	4	1	13	13
196	Roue	4J317	0	0	1	1	6	7	2	5	6	7	1	13	13
197	Roue	4J354	0	7	1	1	6	8	0	5	5	9	5	13	13
198	Roue	4J378	0	0	1	1	6	8	2	5	5	8	1	13	13
199	Roue	4J389	0	7	1	1	6	8	3	5	6	8	5	13	13
200	Roue	4J393	0	7	1	1	6	8	3	5	6	9	5	13	13
201	Roue	4J397	0	7	1	1	6	8	3	5	6	9	5	13	13
202	Roue	4J407	0	0	1	1	6	7	2	5	5	7	1	13	13
203	Roue	4J414	0	7	1	1	6	6	2	5	3	4	3	13	13
204	Roue	4J425	0	4	1	1	6	8	3	5	5	9	3	13	13
205	Roue	4J434	0	4	1	1	6	8	3	5	4	8	3	13	13
206	Roue	4J472	0	4	1	1	6	6	2	5	5	6	3	13	13
207	Roue	4J518	0	0	1	1	6	8	2	5	6	9	1	13	13
208	Roue	4J572	0	4	1	1	6	7		5	6	6	3	13	13

Ω	Désignation	Plan	PRINCIPAL					ADD.			SUPPL.			Pf	Pi
209	Roue	4J578	0	0	1	1	6	8	2	5	6		1	13	13
210	Roue	4J603	0	4	1	1	6	6	2	5	4	5	3	13	13
211	Roue	4J616	0	0	1	1	6	7	3	5	6	8	1	13	13
212	Roue	4J619	0	0	1	1	6	8	2	5	5	9	1	13	13
213	Roue	4J620	0	0	1	1	6	8	2	5	5	9	1	13	13
214	Roue	4J632	0	7	1	1	6	8	3	5	5	9	5	13	13
215	Roue	4J638	0	7	1	1	6	7	3	5	4	6	5	13	13
216	Roue	4J657	0	7	1	1	6	6	2	5	3	5	3	13	13
217	Roue	4J685	0	0	1	1	6	8	2	5	6	8	1	13	13
218	Roue	4J690	0	0	1	1	6	5	2	5	3	3	1	13	13
219	Roue	4J743	0	0	1	1	6	7	2	5	6	8	1	13	13
220	Roue	4J794	0	4	1	1	6	8	2	5	6	9	4	13	13
221	Moyeu TMBD 30	6B006	1	4	1	6	6	3		9	3		4	14	22
222	Moyeu TMBD 30	6B007	1	4	1	6	6	3		9	3		4	14	22
223	Moyeu d'accouplement	6B023	1	4	1	6	6	4		9	4		4	14	22
224	Moyeu d'acc. TRED 70	6B024	1	4	1	6	6	4		9	4		4	14	22
225	Moyeu entretoise	6B042	1	4	4	1	6	5	3	9	7	4	6	14	23
226	Moyeu entretoise	6B044	2	4	4	1	6	5	3	9	8	5	6	14	0
227	Moyeu TMED 30	6B055	1	4	7	1	6	3	3	7	3	1	4	14	23
228	Moyeu TMED 30	6B056	1	4	7	1	6	3	3	5	3	1	4	14	23
229	Manchon TFD 90	6B115	1	4	4	1	6	5	0	9	6	4	5	14	23
230	Moyeu TMED 30	6B123	1	4	7	1	6	3	3	9	3	1	4	14	23
231	Moyeu TMED 30	6B124	1	4	7	1	6	3	3	9	3	1	4	14	23
232	Moyeu TMED 40	6B128	1	4	7	1	6	3	3	9	3	1	4	14	23
233	Moyeu TMED 40	6B129	1	4	7	1	6	3	3	9	3	2	4	14	23
234	Moyeu TMED 60	6B134	1	4	7	1	5	4	3	9	4	2	4	14	23
235	Moyeu TMED 60	6B135	1	4	7	1	6	4	3	9	4	2	4	14	23
236	Moyeu TMED 40	6B154	1	4	7	1	6	3	3	9	3	1	4	14	23
237	Moyeu TMED 40	6B155	1	4	7	1	6	3	3	9	3	1	4	14	23
238	Moyeu d'acc. TMED 70	6B165	1	4	7	1	6	4		5	4		4	14	23
239	Moyeu d'acc. TMED 70	6B166	1	4	7	1	6	4		5	4		4	14	23
240	Bride adaptateur	6B001	1	4	4	4	2	6	0	9	5	3	7	15	21
241	Bride adaptateur	6B002	1	4	4	0	2	5	0	9	5	2	7	15	21
242	Entretoise TMED 70	6B025	1	4	4	1	2	5	0	9	6	2	6	15	21
243	Entretoise	6B057	1	4	4	1	2	4	0	9	6	1	6	15	21
244	Adaptateur HPS-S14-4	6B065	0	4	1	4	2	4	0	9	3	1	6	15	7
245	Entretoise S14-4	6B068	2	4	4	0	2	4	0	9	7	2	8	15	0
246	Adaptateur HPS 16	6B090	1	4	4	4	2	4	0	5	6	2	8	15	21
247	Entretoise TMED 30	6B125	1	4	4	1	2	4	0	9	6	1	6	15	21
248	Entretoise TMED 40	6B130	1	4	4	1	2	4	0	9	6	2	6	15	21
249	Entretoise TMED 60	6B136	1	4	4	1	2	5	0	9	6	2	6	15	21
250	Entretoise TMED 40	6B156	1	4	4	1	2	4	0	9	6	2	6	15	21
251	Entretoise TMED 60	6B161	1	4	4	1	2	5	0	9	6	2	6	15	21
252	Entretoise TMED 70	6B167	1	4	4	1	2	5	0	9	7	3	6	15	21
253	Vis	4J370	2	5	5	2	0	1	0	1	5	0	6	16	0
254	Vis	4J431	2	5	2	1	0	2	0	1	5	1	5	16	15
255	Vis	4J442	2	5	2	1	0	2	0	1	5	1	5	16	15
256	Vis	4J560	2	5	2	1	0	2	0	1	5	1	4	16	15
257	Vis	4J577	2	5	2	1	0	2	0	1	5	1	4	16	15
258	Vis	4J645	2	6	5	8	0	2	0	0	5		7	16	0
259	Vis	4J709	2	5	1	6	0	1	0	1	4		4	16	16
260	Vis	4J710	2	5	1	6	0	1	0	1	4		5	16	16
261	Vis	4J749	2	5	5	1	3	2	0	1	5	1	6	16	0
262	Vis d'accouplement	6B048	2	2	1	6	0	1		1	4		2	16	16
263	Vis d'accouplement	6B049	2	2	1	6	0	1		1	4		2	16	16
264	Vis	6B071	2	2	1	6	0	0		1	3		2	16	16

Ω	Désignation	Plan	PRINCIPAL					ADD.			SUPPL.			Pf	Pi
265	Vis	6B106	2	5	0	0	0	1		0	3		2	16	10
266	Vis	6B150	2	2	1	6	0	0		1	2		2	16	16
267	Vis	6B151	2	2	1	6	0	0		1	2		2	16	16
268	Mannequin GV	4J843	1	1	7	0	0	4	0	2	4	3	2	17	17
269	Mannequin GV	4J851	1	1	7	0	0	4	0	2	3	2	2	17	17
270	Mannequin PV	4J852	1	1	7	0	0	4	0	2	4	2	2	17	17
271	Forge manchon	8H619	1	1	1	0	0	6		0	6	5	2	17	17
272	Forge manchon	8H654	1	1	1	0	0	6		0	6	5	2	17	17
273	Forge manchon PV	8H658	1	1	1	0	0	6		0	6	6	2	17	17
274	Forge moyeu HPS 12	8H677	1	1	1	0	0	5		0	5	4	2	17	17
275	Forge manchon	8H710	1	1	1	0	0	6		0	6		2	17	17
276	Gicleur type M159	5E318	1	5	1	3	5	0		0	1		5	18	18
277	Gicleur type M207	5E321	1	5	1	3	5	1		0	1		5	18	18
278	Gicleur type M190	5E344	1	5	1	3	5	1		0	1		5	18	18
279	Gicleur	5E346	1	5	4	6	0	0		0	0		6	18	19
280	Gicleur type M12159	5E440	1	5	1	3	5	0	2	0	1		5	18	18
281	Gicleur type M879	5E664	1	5	4	6	0	0		0	0		6	18	19
282	Forge arbre de roue	8H616	2	4	1	0	0	7		0	6	6	4	19	10
283	Forge roue arbrée	8H620	2	4	1	0	0	7		0	7	8	7	19	10
284	Forge roue arbrée	8H622	2	4	1	0	0	7		0	7	8	7	19	10
285	Forge arbre de roue	8H637	2	4	1	0	0	6		0	8	7	5	19	10
286	Forge roue arbrée	8H643	2	4	1	1	0	7		0	8	9	4	19	20
287	Forge entretoise	8H644	2	4	1	0	0	4		0	7	4	2	19	10
288	Forge arbre de roue	8H655	2	4	1	0	0	6		0	8	7	5	19	10
289	Forge arbre de roue	8H663	2	4	1	0	0	6		0	8	7	5	19	10
290	Forge adaptateur	8H671	1	4	1	0	0	4		0	6	3	5	19	25
291	Forge entretoise HPS12	8H678	2	4	1	0	0	4		0	7	4	5	19	10
292	Forge moyeu TFD90	8H679	1	0	1	0	0	6		0	6	6	2	19	6
293	Forge manchon TFD90	8H680	1	4	1	0	0	5		0	6	5	2	19	25
294	Forge pignon	8H681	2	4	0	0	0	6		0	8	8	2	19	10
295	Forge moyeu TFD90	8H688	1	0	1	0	0	6		0	6	6	2	19	6
296	Forge arbre de roue	8H690	2	4	1	0	0	6		0	8	7	5	19	10
297	Forge arbre de roue	8H691	2	4	1	0	0	6		0	8	7	5	19	10

Annexe V.2 : Tableau de données des codes projetés.

N désigne le nom du code projeté (il s'agit d'un numéros)
L/D, FE, FI, US et FO désignent les variables descriptives.
Pf : désigne le numéros de la classe où les objets sont rangées dans la partition finalle validée par les experts.

Ce tableau est une projection du tableau de données initiales donné dans l'annexe V.1
Par exemple : le code N°14 désigne l'ensemble des objets caractérisés par L/D=0 FE=0 FI=0 US=0 et FO=2.
Ce code correspond à l'ensemble des objets { 26, 27, 28, 29, 30, 31, 35} du tableau de données initial.

N	L/D	FE	FI	US	FO	Pf
1	0	2	0	3	0	0
2	0	4	4	3	4	0
3	0	1	1	1	5	0
4	0	4	1	0	0	0
5	0	1	1	3	4	0
6	0	0	4	4	2	0
7	0	0	1	7	0	0
8	0	4	4	6	2	0
9	0	1	2	1	2	1
10	0	1	2	9	2	1
11	0	0	2	3	3	1
12	0	0	2	0	2	1
13	0	1	2	3	1	1
14	0	0	0	0	2	2
15	0	0	0	0	0	2
16	0	0	0	0	1	2
17	0	1	1	0	2	3
18	0	1	1	0	0	3
19	0	1	4	0	2	3
20	0	0	1	6	0	4
21	0	7	1	0	0	5
22	0	7	1	1	3	5
23	0	7	1	1	4	5
24	0	7	1	1	0	5
25	0	7	1	1	9	5
26	0	0	1	0	0	6
27	0	0	1	0	2	7
28	0	0	4	0	2	7
29	1	2	2	0	0	8
30	1	0	2	0	0	8
31	1	4	2	0	0	8
32	1	4	6	1	6	9
33	0	1	6	1	6	9
34	1	4	6	0	6	9
35	1	1	6	0	6	9
36	2	4	0	0	0	10
37	2	1	0	0	0	10
38	2	4	1	0	0	10
39	2	0	5	1	0	11
40	2	4	5	1	4	12
41	2	4	5	1	2	12
42	2	4	5	4	4	12
43	2	5	5	4	4	12
44	0	7	1	1	6	13
45	0	4	1	1	6	13
46	0	0	1	1	6	13
47	1	4	1	6	6	14
48	1	4	4	1	6	14
49	2	4	4	1	6	14
50	1	4	7	1	6	14
51	1	4	7	1	5	14
52	1	4	4	4	2	15
53	1	4	4	0	2	15
54	1	4	4	1	2	15
55	2	5	5	2	0	16
56	2	5	2	1	0	16
57	2	6	5	8	0	16
58	2	5	1	6	0	16
59	2	5	5	1	3	16
60	2	2	1	6	0	16
61	2	5	0	0	0	16
62	1	1	7	0	0	17
63	1	1	1	0	0	17
64	1	5	1	3	5	18
65	1	5	4	6	0	18
66	2	4	1	0	0	19
67	2	4	1	1	0	19