ED n°431 : Information, Communication, Modélisation, Simulation

N° attribué par la bibliothèque
|__|__|__|__|__|__|__|__|__|__|

T H È S E

pour obtenir le grade de
Docteur de l'École des Mines de Paris
Spécialité "Informatique"

présentée et soutenue publiquement par
Jean VILLERD

le 19 novembre 2008

Représentations visuelles adaptatives de connaissances associant projection multidimensionnelle (MDS) et analyse de concepts formels (FCA)

Directeur de thèse : Michel Crampes

Jury

Marianne Huchard	Professeur Université Montpellier 2, LIRMM	Présidente
Guy Mélançon	Professeur Université Bordeaux 1, LABRI	Rapporteur
Amedeo Napoli	Directeur de Recherche CNRS, LORIA	Rapporteur
Michel Crampes	Maître-assistant (HDR) École des Mines d'Alès, LGI2P	Directeur, Examinateur
Robert Mahl	Professeur École des Mines de Paris, CRI	Co-directeur, Examinateur
Sylvie Ranwez	Maître-assistant École des Mines d'Alès, LGI2P	Encadrante, Examinatrice

Mis en page avec la classe thloria.

Remerciements

Je tiens à remercier en premier lieu les personnes qui m'ont encadré et accordé leur confiance pendant mes trois années de thèse. Michel Crampes, mon directeur de thèse, pour son enthousiasme, sa disponibilité et la liberté dont j'ai bénéficié dans mes recherches. Sylvie Ranwez, mon encadrante de proximité, pour son ouverture d'esprit, sa gentillesse, sa patience et son soutien sans faille, y compris durant les moments difficiles. Toujours à l'écoute de mes interrogations en recherche, elle a également guidé mes premiers pas en enseignement. Les échanges avec Robert Mahl, mon co-directeur de thèse à l'École des Mines de Paris, m'ont aidé à prendre le recul nécessaire à chaque étape importante de mon travail de thèse.

Je remercie les membres du jury qui m'ont fait l'honneur d'accepter d'évaluer mes travaux. Guy Mélançon et Amedeo Napoli qui ont accepté la charge de rapporteur et dont les remarques enrichissantes ont contribué à améliorer ce mémoire. Marianne Huchard qui a accepté de présider ce jury et à qui je dois ma première expérience en recherche. Plusieurs années après avoir encadré mes travaux de DEA, dans un autre laboratoire et sur une autre thématique, je la remercie chaleureusement d'avoir accepté d'examiner mon travail de thèse.

Je remercie également Yannick Vimont, directeur de la Recherche de l'École des Mines d'Alès ainsi que l'ensemble du personnel du centre de recherche LGI2P qui m'a accueilli durant ces trois ans. J'adresse un remerciement particulier à Gérard Dray, Jacky Montmain et Pascal Poncelet pour leurs conseils et leur soutien dans les ultimes phases de rédaction de ce mémoire. Je remercie de plus ceux qui m'ont aidé à préparer mes présentations orales : Anne-Lise Courbis, Thomas Lambolais, Françoise Armand.

Je remercie encore mes collègues thésards, notamment Fabien Jalabert, Nicolas Desnos, Fady Hamoui, Imane Anoir, Huaxi Yulin Zhang et Reena Shetty à qui mon niveau d'anglais doit une fière chandelle. Les conseils des anciens et le soutien des nouveaux m'ont été précieux.

Je remercie évidemment mes parents, qui m'ont soutenu tout au long de mes études, je leur suis aujourd'hui reconnaissant de m'avoir « poussé » un peu plus qu'à mon goût à certaines époques... Mes amis enfin, Alban, Laurent, Tophe et toute la famille Nicq, Yamina, qui m'ont supporté et me supportent toujours dans tous les sens du terme.

O Heil dem Tranke ! Heil seinem Saft ! Heil seines Zaubers hehrer Kraft ! Durch des Todes Tor, wo er mir floß, weit und offen er mir erschloß, darin ich sonst nur träumend gewacht, das Wunderreich der Nacht ; von dem Bild in des Herzens bergendem Schrein scheucht' er des Tages täuschenden Schein, daß nachtsichtig mein Auge wahr es zu sehen tauge. [...] *Wer des Todes Nacht liebend erschaut, wem sie ihr tief Geheimnis vertraut : des Tages Lügen, Ruhm und Ehr, Macht und Gewinn, so schimmernd hehr, wie eitler Staub der Sonnen sind sie vor dem zersponnen ! In des Tages eitlem Wähnen bleibt ihm ein einzig Sehnen, das Sehnen hin zur heil'gen Nacht, wo ur-ewig, einzig wahr, Liebeswonne ihm lacht !*

Ô gloire au philtre ! Gloire à sa saveur ! Gloire aux vertus sublimes de sa magie ! Par la porte de la mort où il a coulé pour moi, il m'a ouvert tout grand le royaume merveilleux de la nuit où je n'avais jamais veillé qu'en rêve. De l'image enclose au profond de mon cœur, il a chassé la clarté décevante du jour, pour que mes yeux, dans cette quête nocturne, puissent voir cette image dans sa vérité. [...] *Pour celui qui contemple avec amour la nuit de la mort, pour celui auquel elle a confié son profond secret, pour celui-là, mensonges du jour, gloire et honneurs, pouvoir et fortune, dans tout leur éclat superbe, sont dissipés comme vaine poussière de soleils ! Dans les chimères dérisoires du jour, une unique aspiration lui reste : l'aspiration à la sainte nuit où, de toute éternité, seule véridique, l'extase de l'amour le fait tressaillir !*

Richard Wagner, *Tristan und Isolde, acte* ii *scène* ii

Table des matières

Introduction

Les travaux présentés dans ce manuscrit ont été menés au sein du Laboratoire de Génie Informatique et d'Ingénierie de Production (LGI2P) de l'École des Mines d'Alès, de septembre 2005 à septembre 2008.

1 Contexte de l'étude

Les outils de recherche d'information sont confrontés à un accroissement constant à la fois du volume des données accessibles (nombre d'objets) et de leur dimensionnalité (nombre d'attributs). La traditionnelle liste de résultats ne suffit plus et un réel besoin en nouvelles techniques de représentation visuelle, capables de gérer des données nombreuses et multidimensionnelles, émerge. Les principales difficultés résident dans :

- le volume important des données à visualiser,
- l'hétérogénéité des structures des données (tabulaires, de type graphe),
- l'hétérogénéité des attributs (binaires, nominaux, numériques).

Ces nouvelles techniques doivent également permettre d'une part d'appréhender les données de manière globale, en révélant les tendances et la structure générales et d'autre part de pouvoir observer de façon détaillée un ensemble plus restreint de données selon un certain point de vue correspondant à des dimensions particulières.

2 Objectifs et approche

Nous nous sommes fixé pour objectif d'apporter des solutions aux trois problèmes précités en nous efforçant de nous inscrire dans une démarche de formalisation. En effet, plusieurs représentants de la communauté de visualisation de données ont pointé le manque de formalisation par lequel, selon eux, la discipline pèche. N'étant pas spécifiées formellement, les techniques de visualisation sont difficilement comparables et leur efficacité difficilement mesurable. Avant même d'élaborer nos solutions, notre première tâche consiste à formaliser notre environnement de visualisation Molage et l'ensemble des techniques de visualisation qu'il met en œuvre. Les solutions apportées pourront alors être spécifiées en suivant cette formalisation.
Les premières solutions élaborées nous ont amenés à adopter une approche de visualisation *overview + detail*, consistant en une vue globale reflétant la structure générale des données, et une vue locale représentant de manière détaillée les objets correspondant à un élément de structure sélectionné sur la vue globale. Dans le cadre de cette approche, l'utilisateur doit être assisté dans sa tâche d'exploration de l'information par une articulation judicieuse entre vue globale et vues locales maintenant sa carte mentale et par la suggestion de parcours cohérents à travers les données. La méthode de navigation que nous proposons utilise les techniques de FCA – *Formal Concept Analysis* ou Analyse de concepts formels – associées à des techniques de visualisation multidimensionnelles MDS – *MultiDimensional Scaling* ou Échelonnage multidimensionnel – pour suggérer des parcours de navigation. Une attention particulière est portée aux problèmes liés aux données manquantes, d'une part, et aux données indexées sur des dimensions mixtes (binaires, nominales, continues), d'autre part.

3 Structure du mémoire

La figure 1 illustre la structure de la thèse et l'enchaînement des chapitres, depuis la problématique liée aux données initiales, jusqu'aux contributions en terme de méthodes de visualisation, en passant par les réalisations qui nous ont permis d'identifier les verrous à lever.

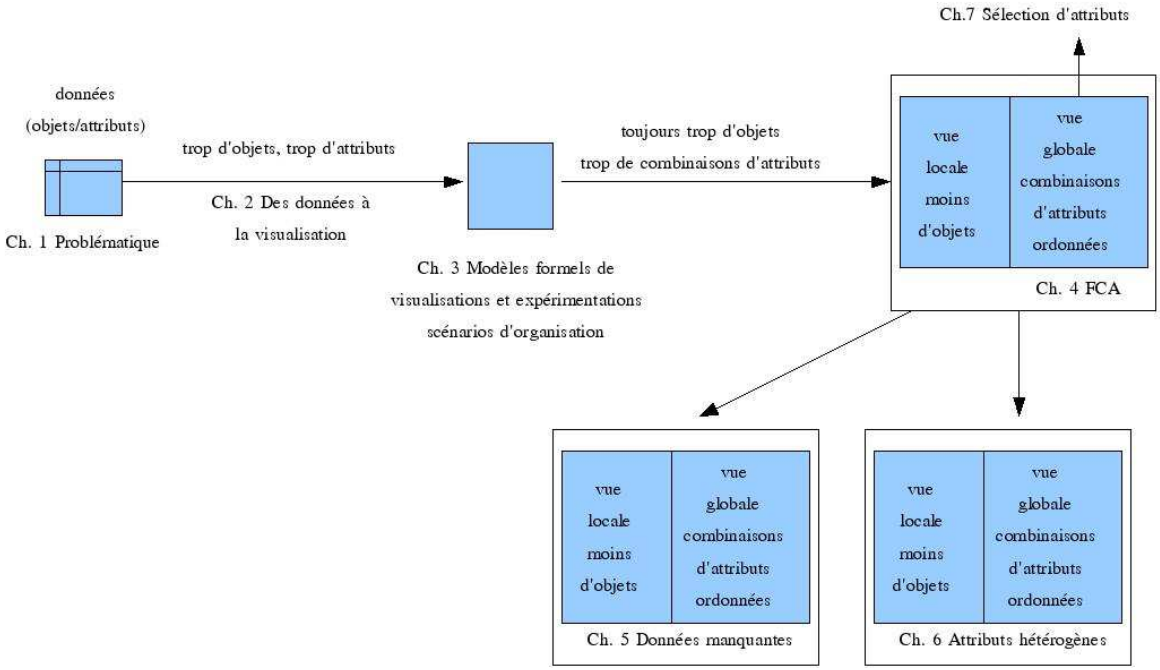


FIG. 1 – Structure de la thèse et objectifs de chaque chapitre

Le point de départ de nos travaux est l'accroissement en volume des données produites et rendues massivement accessibles par la baisse du coût de stockage. Le chapitre 1 expose les conséquences de cet accroissement en volume sur les outils permettant de manipuler et d'accéder aux données : le volume de données devient tel que seules des représentations visuelles adéquates permettent de naviguer. Ces représentations visuelles doivent résoudre trois problèmes :

– le volume important des données à visualiser,
– l'hétérogénéité des structures des données (tabulaires, de type graphe),
– l'hétérogénéité des attributs (binaires, nominaux, numériques).

Une fois la problématique établie, le chapitre 2 revient sur l'historique de la visualisation de données et expose les problèmes auxquels la communauté est actuellement confrontée par manque de formalisation. Le processus de construction d'une visualisation à partir des données est illustré par le modèle de Card-Chi. Une typologie des données et de leurs structures est établie et les solutions de visualisation correspondantes sont exposées.

La seconde partie présente un modèle formel fondé sur le paradigme *Force-directed placement* (FDP), également appelé « modèle des ressorts », pour la spécification des visualisations. Deux patrons de visualisation, appelés *scénarios*, sont identifiés, correspondant chacun à une structure précise des données à représenter. Ces données, provenant de deux projets de recherche distincts, sont visualisées via notre environnement FDP MOLAGE en implémentant les spécifications formelles des scénarios d'organisation. Des solutions partielles sont apportées aux verrous introduits dans la problématique et un nouveau verrou émerge : celui de la gestion des données manquantes.

Ces premières réalisations montrent la nécessité de réduire le nombre d'objets simultanément affichés. Une stratégie de visualisation *overview + detail* est adoptée afin de n'afficher qu'un nombre restreint d'objets sur la vue locale (*detail*) et de naviguer sur la vue globale. Le contenu

de cette vue globale est extrait par des techniques de Formal Concept Analysis (FCA), introduites au chapitre 4.

Les chapitres 5 et 6 proposent des solutions aux problèmes concernant respectivement les données manquantes et l'hétérogénéité des attributs. Ces deux solutions sont fondées sur la même approche *overview + detail* et montrent le caractère complémentaire de FCA et des techniques de visualisation.

Enfin, alors que dans les solutions proposées au cours des précédents chapitres, les attributs constituaient le point de départ de la navigation, le chapitre 7 se place dans le cas où un sous-ensemble d'objets constitue ce point de départ. Un processus de sélection d'attributs permet d'identifier les attributs pertinents pour la description de ces objets et les situe parmi le reste des données.

En proposant un cadre formel comme base de nos solutions, nous espérons contribuer à l'effort de formalisation attendu par la communauté visualisation et ouvrir ainsi de nouvelles perspectives de recherche.

Première partie

Problématique et État de l'art

1

Problématique

Sommaire

Le besoin récurrent en outils de visualisation d'information exprimé dans l'introduction trouve son origine dans l'accroissement significatif des capacités de stockage numérique. Ainsi, la quantité d'information stockée au cours de la seule année 2002 a été évaluée [LVSC03], tous formats confondus, à 5 exaoctets (5.10^{18} octets) et devait augmenter de plus de 30% chaque année. Concernant les données informatisées, une des conséquences de cette situation est le besoin d'interfaces adaptées permettant à l'utilisateur d'appréhender de grands volumes de données, les interfaces utilisées jusqu'à présent ne pouvant plus remplir leur fonction avec la même efficacité. Considérant le cas pratique des disques durs équipant les ordinateurs personnels, le volume de ceux-ci est passé en vingt ans de quelques mégaoctets à plusieurs centaines de gigaoctets. Les outils de gestion de fichiers ont dû s'adapter à ce changement d'échelle en proposant de nouvelles façons de représenter un système de fichiers. Ainsi, lors de son lancement en 1986, l'interface à base de listes textuelles du célèbre gestionnaire de fichiers Norton Commander était suffisante pour appréhender globalement un volume de quelques mégaoctets (voir figure 1.1 haut). L'accroissement des capacités de stockage a entraîné un accroissement du nombre de fichiers et parallèlement des hiérarchies de répertoires. De nouvelles formes de représentation sont apparues (voir figure 1.1 bas) délaissant les listes textuelles. Un changement de paradigme s'est en effet opéré : le caractère visuel est le dénominateur commun de ces nouvelles représentations.

L'accroissement conjoint du volume des données et du nombre de dimensions de ces données a été qualifié par Richard Bellman de *Curse of dimensionality* [Bel57] (fléau de la dimension). Ainsi, l'enjeu des nouvelles formes de représentations visuelles est double. Elles doivent pouvoir gérer à la fois le volume important des données et leur nombre croissant de dimensions. Ces dimensions peuvent être de natures diverses : attributs binaires ou numériques. La structure des données peut également varier : tableaux objets/attributs, relations binaires entre objets, arbres. Les solutions visuelles doivent à la fois être adaptées à la nature des dimensions et de la structure des données, et être capable de représenter des données aux dimensions et à la structure

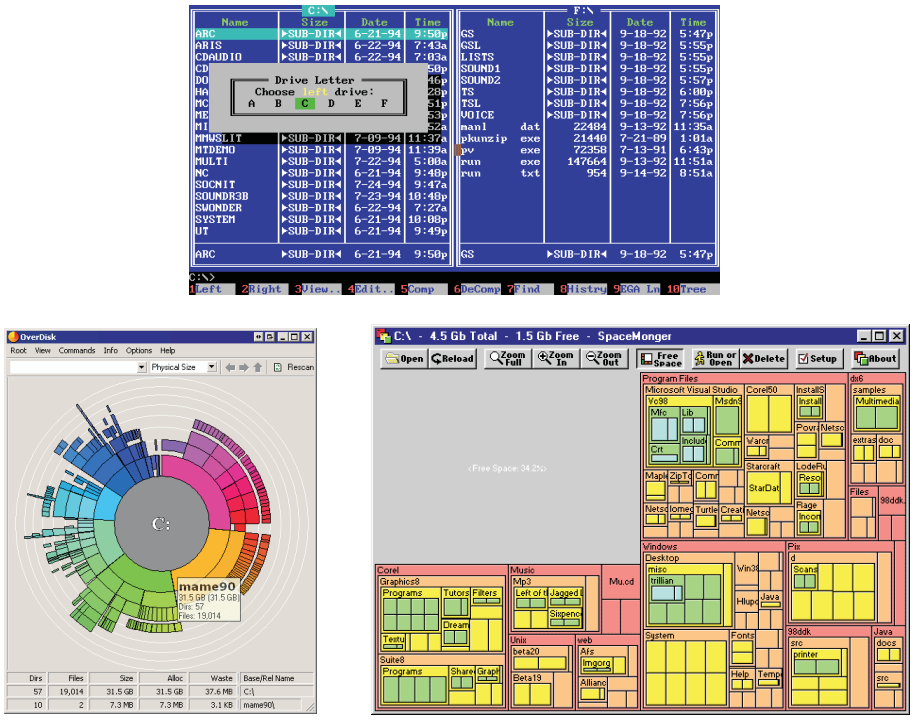


FIG. 1.1 – Évolution des interfaces des gestionnaires de fichiers : en haut NORTON COMMANDER, en bas à gauche OVERDISK, en bas à droite SPACEMONGER

hétérogènes.

Avant de parcourir plus en détails, dans le chapitre suivant, les techniques de visualisation, nous nous penchons d'abord sur la description des données.

Les données manipulées ici peuvent être décrites comme un ensemble d'objets, un ensemble d'attributs (ou dimensions), des relations objet-objet, objet-attribut et attribut-attribut. Un objet correspond à un individu pour les statisticiens, un attribut est une dimension à laquelle est associé un ensemble de valeurs. Un objet o est valué sur un attribut a lorsque cet objet est associé à une valeur particulière v de l'attribut. On dit alors que l'objet o a pour valeur v sur l'attribut a. Le contexte d'accroissement et d'hétérogénéité des dimensions évoqué précédemment se caractérise donc par un grand nombre d'objets, valués sur un grand nombre d'attributs de natures différentes. La section suivante présente une typologie des natures d'attributs. La structure des données, quant à elle, se définit en fonction des relations existant entre objets et sera étudiée au cours du chapitre 2.

1.1 Attributs et mesures

Les attributs associés aux objets peuvent être répartis selon une typologie dépendant de la nature de leurs valeurs (ou mesures) respectives et des propriétés de celles-ci. La valeur d'un attribut peut en effet prendre la forme d'un nombre (une hauteur exprimée en mètres, le dossard d'un joueur de football, le nombre d'enfants d'une famille), d'une chaîne de caractères (un nom, une couleur, une appréciation) ou d'un caractère symbolisant les valeurs *vrai* ou *faux*. Les liens entre un attribut et ses mesures ont été étudiés par la branche théorique de la métrologie

(*measurement theory*) [KS71, Han96]. Ces travaux sont utiles en analyse de données et en statistique descriptive dans la mesure où, pour tirer des conclusions sur un attribut, il faut prendre en compte la nature de la correspondance entre l'attribut et ses mesures [Sar95]. Stanley Smith Stevens a ainsi identifié quatre niveaux ou échelles de mesures se distinguant par les propriétés des ensembles de nombres ou de symboles constituant les mesures [Ste46]. Le tableau 1.1 associe à chaque échelle un exemple de mesures et un indicateur de tendance centrale. Notons que l'ensemble des indicateurs de tendance centrale pour une échelle considérée inclut les indicateurs spécifiés pour les échelles la précédant dans le tableau. Ainsi il est possible de calculer la médiane d'un ensemble de mesures d'intervalles tandis que la notion de moyenne arithmétique n'a pas de sens pour des mesures ordinales.

échelle	exemple	tendance centrale
nominale	{bleu, blanc, rouge}	mode
ordinale	{passable, bien, très bien}	médiane
intervalles	température en degrés Celsius	moyenne arithmétique, écart-type
rapports	température en kelvins	moyenne géométrique

TAB. 1.1 – Échelles de mesures

Nous détaillons dans la suite chaque échelle en précisant notamment les indicateurs de tendance centrale associés, les transformations possibles sur les mesures et la structure mathématique associée.

1.2 Échelle nominale

Une échelle nominale est un ensemble non ordonné de valeurs, pouvant être considérées comme des étiquettes. Ces valeurs se présentent généralement sous forme de chaînes de caractères mais pas toujours : les numéros attribués aux joueurs d'une équipe de football constituent en effet une échelle nominale, bien que les symboles utilisés soient des entiers naturels. Les transformations basées sur l'existence d'une relation d'ordre sur $\mathbb{N}$ sont alors proscrites, ces entiers ne jouant, dans le cas considéré, que le rôle d'identifiants. Notons que la prise en compte de la sémantique des entités, au sens large, manipulées se révèle d'ores et déjà importante. Voir à ce sujet l'article satirique de Frederick Lord [Lor70]. Pour chacune des échelles présentées dans la suite, nous précisons leurs caractéristiques mathématiques. Pour l'échelle nominale, ce sont les suivantes :

- *indicateurs de tendance centrale :* mode.
- *transformations possibles :* permutations une à une.
- *structure mathématique :* ensemble non ordonné.

1.3 Échelle ordinale

Une échelle ordinale est un ensemble de valeurs muni d'un ordre total. Comme pour les échelles nominales, les valeurs sont généralement des chaînes de caractères.

- *indicateurs de tendance centrale :* mode, médiane.
- *transformations possibles :* transformations monotones croissantes.
- *structure mathématique :* ensemble totalement ordonné.

1.4 Échelle d'intervalles

Les valeurs d'une échelle d'intervalles sont totalement ordonnées et l'intervalle entre deux valeurs est quantifiable. Les valeurs sont nécessairement des nombres et la soustraction de deux valeurs a un sens. Notons que la différence entre les échelles ordinales et d'intervalles réside dans ce dernier point. La valeur zéro étant fixée arbitrairement, l'addition de deux valeurs n'a pas de sens.

- *indicateurs de tendance centrale :* mode, médiane, moyenne arithmétique, écart-type.
- *transformations possibles :* toute transformation affine t telle que $t(m) = c \times m + d$ où c et d sont des constantes et m une valeur de l'échelle.
- *structure mathématique :* espace affine de dimension 1.

1.5 Échelle de rapports

La valeur zéro étant fixée de façon non arbitraire, le rapport m_1/m_2 entre deux valeurs a un sens, de même que l'addition, la multiplication et la division. La plupart des grandeurs physiques, telles que la masse, la longueur ou l'énergie sont mesurées sur des échelles de rapport. C'est également le cas d'une température mesurée en kelvins mais pas d'une température mesurée en degrés Celsius dont le zéro a été fixé de manière arbitraire et non selon un zéro absolu.

- *indicateurs de tendance centrale :* mode, médiane, moyennes arithmétique et géométrique, écart-type.
- *transformations possibles :* toute transformation linéaire t telle que $t(m) = c \times m$ où c est une constante et m une valeur de l'échelle.
- *structure mathématique :* corps.

1.6 Autres échelles

Les mesures **binaires** sont généralement considérées comme appartenant à une échelle nominale à deux valeurs possibles. La typologie présentée dans le tableau 1.1 regroupe les quatres échelles communément admises. On pourra cependant rencontrer dans la littérature les échelles de **log-intervalles** et **absolues** dont l'étude sort du cadre de ce manuscrit. En effet, le débat sur la validité des échelles de Stevens n'est pas encore clos au sein de la communauté de métrologie théorique [Lor70, Dun84, Mic86, VW93].

On s'autorisera dans la suite l'abus de langage consistant à qualifier un attribut du nom de l'échelle à laquelle appartiennent ses valeurs. On parlera ainsi d'attributs binaires, nominaux et ordinaux. De plus nous regrouperons sous le terme d'attribut numérique les attributs d'intervalles et de rapports.

1.7 Conclusion

L'accroissement du volume des données stockées et de leurs dimensions révèle un besoin crucial en techniques de représentations visuelles. De plus, les données sont caractérisées par une hétérogénéité en terme de structure d'une part, et en terme de nature des dimensions d'autre part. La problématique que nous traitons ici est donc double. Il s'agit de proposer des solutions de représentation visuelle capables de gérer :

- un volume important de données,

– la cohabitation d'attributs et de structures de natures différentes.

Le chapitre suivant présente un état de l'art des techniques de visualisation de données existantes.

2

Des données à la visualisation

Sommaire

À partir de 2004 fut publiée une série d'articles partageant les deux constats suivants : d'une part la recherche en visualisation arrive à la fin d'une étape de son développement, d'autre part la communauté aura besoin de se doter d'outils théoriques et formels pour entamer l'étape suivante. Au cours d'une table ronde intitulée *Is there science in visualization ?* [JKKK+06], organisée dans le cadre de la conférence IEEE Visualization, Wes Bethel pose le problème en ces termes :

> *Many in the "hard sciences" view Computer Science as a "Johnny-come-lately"*[1] *and lacking rigor in terms of scientific methodology ; the field of visualization is often viewed by outsiders as being even "more soft" than pure Computer Science. As a result, there is a "credibility gap" between a segment of our customer population – scientific researchers – and us.*

Afin de comprendre les origines de ce « fossé de crédibilité », nous retracerons dans cette section l'histoire de la visualisation au sens large, et de la visualisation d'information en particulier. Dans ce domaine hautement empirique, les définitions ont souvent été énoncées *a posteriori*, aussi nous les présenterons au fur et à mesure de l'historique. Celui-ci débute par une synthèse des grandes étapes de l'histoire de la représentation visuelle de connaissances de l'Antiquité à nos jours, puis détaille les différentes phases de développement qu'a connu la visualisation assistée par ordinateur. On pourra consulter [Fri06] pour un historique plus complet de la visualisation de données en général et [WB94] pour la visualisation de données assistée par ordinateur en particulier.

[1] parvenu

2.1 Historique et définitions

L'utilisation de métaphores visuelles pour exprimer des connaissances remonte à l'Antiquité. Pour Pythagore, arithmétique et géométrie sont sœurs : chaque point représentant une unité, on distingue des entiers triangles (1,3,6,10...) de la forme $n = \sum_{k=1}^{n-1} k$ et des entiers carrés (1,4,9,16...). En raisonnant graphiquement, les pythagoriciens ont démontré que tout entier carré est la somme de deux entiers triangles successifs. Par la suite, les premières représentations visuelles comme support pour le raisonnement ont été les diagrammes géométriques, les positions des astres et les cartes géographiques. Le développement, à partir du XVI^{e} siècle, de nouvelles techniques et instruments nécessaires à l'expansion maritime de l'Europe fut accompagné de représentations graphiques plus précises dont l'invention de l'imprimerie par Gutenberg en 1436 favorisa le déploiement. Sous l'impulsion de cette expansion territoriale, le XVII^{e} siècle est marqué par un grand renouveau des sciences en Europe (Kepler, Galilée, Newton, Descartes, Pascal, Leibniz) qui amènera à la révolution copernicienne et à l'avènement de l'héliocentrisme. Cette effervescence scientifique conduit à la mise en place de nouvelles formes de communication autres que les exposés oraux donnant lieu à d'onéreux voyages. Ainsi le premier périodique scientifique, intitulé *Le Journal des savants*, paraît à Paris en janvier 1665. Les premières représentations visuelles de grandeurs physiques mesurées, issues des progrès scientifiques récents, s'inscrivent dans cet effort de communication : le premier graphique statistique connu réalisé par Michael van Langren en 1644 (figure 2.1), le premier graphe d'une fonction mathématique par Christiaan Huygens en 1669 et la première carte météorologique connue par Edmond Halley en 1686.

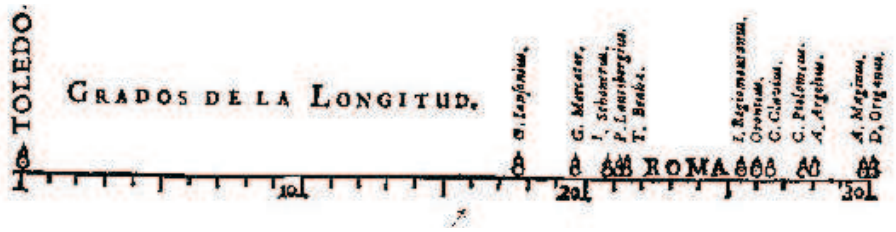


FIG. 2.1 – Graphique de Michael van Langren de 1644 représentant douze mesures de la différence en longitude entre Tolède et Rome. Le nom de l'auteur de la mesure accompagne chaque point. Cette représentation illustre bien mieux qu'une table la dispersion des mesures. La valeur exacte (16 ° 30') est indiquée par une flèche.

Au XVIII^{e} siècle, la visualisation s'étend à des données plus abstraites issues de l'économie, de la démographique et des statistiques tandis que des innovations techniques comme la lithographie facilitent son utilisation. La première moitié du XIX^{e} siècle voit l'explosion de nouvelles formes de graphiques statistiques et l'apparition de « cartes thématiques » utilisant le support d'une carte géographique pour présenter et localiser une information d'origine le plus souvent statistique (figures 2.2 et 2.3). La plupart des formes de graphiques statistiques utilisées aujourd'hui ont été introduites à cette époque, notamment les graphes en barres et les graphes circulaires par William Playfair (1759-1823).

La seconde moitié du XIX^{e} siècle est considérée par [Fri06] comme l'âge d'or de la visualisation de données. Le recours aux statistiques se généralise dans le cadre de la révolution industrielle afin d'exploiter la quantité massive de données liées aux problèmes sanitaires, commerciaux et de transports. Deux exemples majeurs de visualisation ont été réalisés à cette époque, illustrant deux problématiques spécifiques de la cartographie. Le premier est la carte utilisée par le Dr.

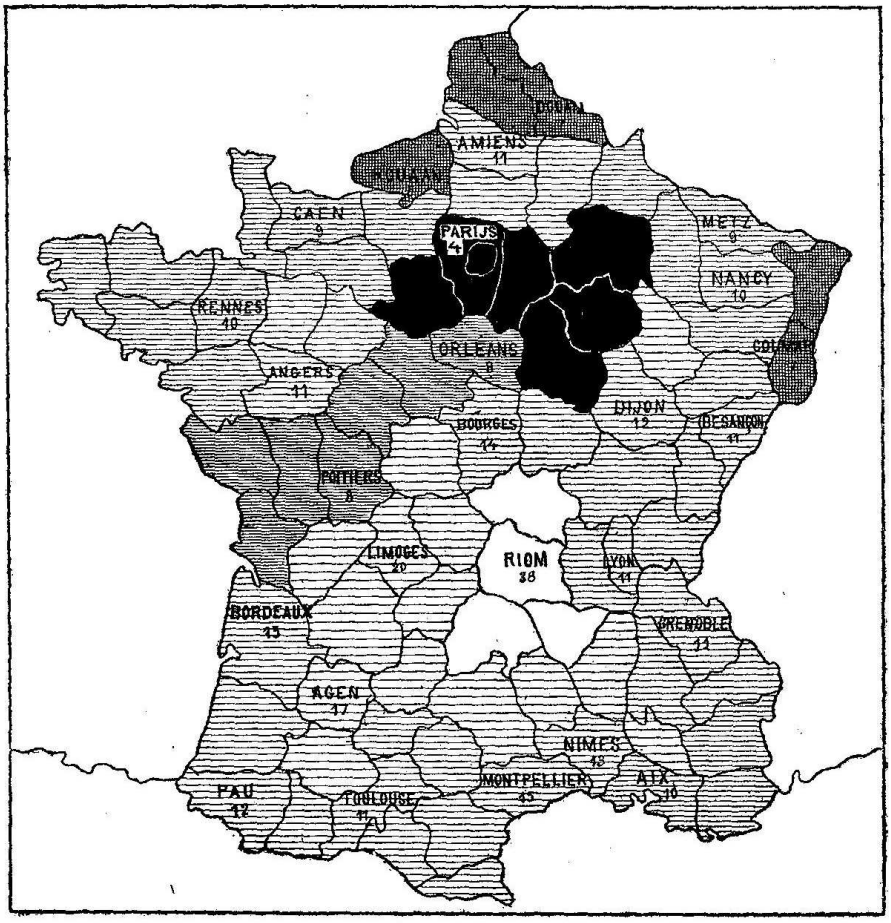
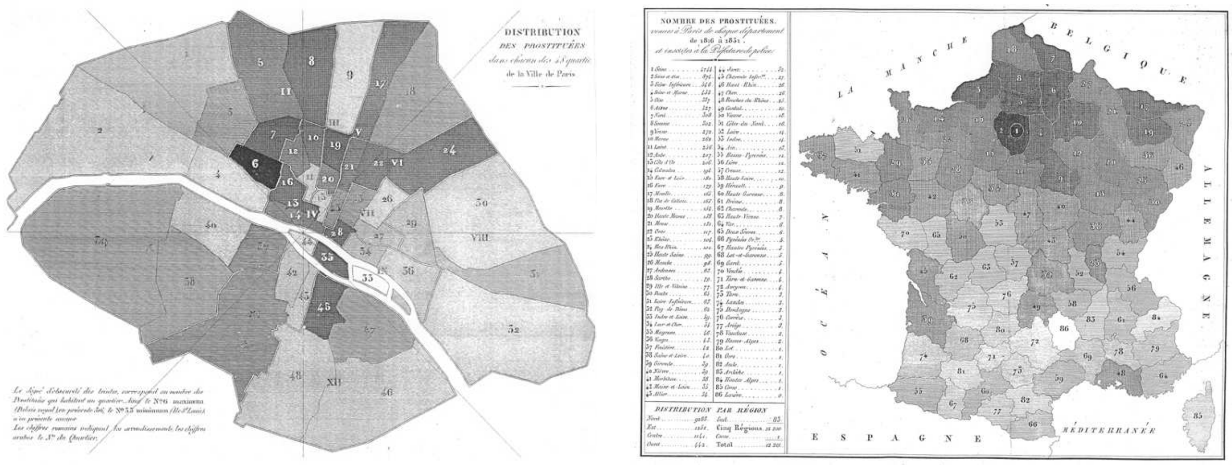


FIG. 2.2 – « Carte de la France obscure et de la France éclairée ». Carte thématique de Charles Dupin (1819) illustrant le taux d'illettrisme par département au moyen d'un dégradé de couleur et considérée comme la première carte statistique connue.

John Snow lors de l'épidémie de choléra qui frappa Londres en 1854 (figure 2.4). En reportant simultanément sur une carte du quartier de Soho les positions des cas de choléra et des pompes à eau, Snow observa que la plupart des malades étaient concentrés dans une zone dont la pompe de la rue Broad Street était le centre. Cette pompe infectée, responsable de la mort de plus de 500 londoniens en dix jours, fut donc identifiée grâce à la représentation visuelle et cartographique de données statistiques. Il est entendu que ce n'est pas la carte en elle-même qui a produit l'information « la pompe de Broad Street est la pompe la plus proche du plus grand nombre de cas de choléra », celle-ci aurait pu être déduite d'une table contenant les distances entre chaque cas et chaque pompe. Le grand atout de la carte réside dans sa faculté à abstraire et révéler de façon immédiate une information contenue de manière non explicite dans les données, résumée ainsi par Playfair : « faire en sorte que les données parlent aux yeux ».

Le second exemple fréquemment cité est la carte de la campagne de Russie réalisée par Charles-Joseph Minard en 1869 (figure 2.21) considéré par Edward Tufte comme « probalement le meilleur graphique statistique jamais dessiné » [Tuf83]. Minard parvient en effet à combiner plusieurs informations de natures différentes sur un seul graphique tout en préservant la cohérence



(a) (b)

FIG. 2.3 – Deux cartes thématiques de Duchâtelet (1836) reprenant le principe de dégradé de Dupin. *Distribution des prostitués dans chacun des 48 quartiers de la ville de Paris* (a). « Le degré d'obscurité des teintes correspond au nombre des prostitués qui habitent un quartier. Ainsi le n°6 maximum (Palais Royal) en présente 56, le n°33 (Île S^{t} Louis) n'en présente aucune. Les chiffres romains indiquent les arrondissements, les chiffres arabes le numéro du quartier. *Nombre de prostitués venues à Paris de chaque département de 1816 à 1831 et inscrites à la Préfecture de Police* (b).

et la lisibilité de celui-ci. Cela est dû à la pertinence du choix du dispositif visuel associé à chaque information. Le temps est représenté par l'axe des abscisses, l'effectif de l'armée par la largeur de la courbe, le sens aller/retour par sa couleur et une courbe supplémentaire permet de suivre l'évolution de la température en fonction du temps, représentée parallèlement à la courbe principale.

Si le début du XX^{e} siècle connaît un ralentissement des innovations graphiques, les techniques existantes se perfectionnent et leur utilisation se démocratise. Ainsi la carte du métro londonien subit des transformations tendant à optimiser son utilité pour le voyageur en se détachant du modèle de la cartographie géographique. La carte de 1887 (figure 2.18) superpose le tracé des lignes de métro sur un plan des rues. Le trajet géographique des lignes est fidèlement reproduit mais il se révèle ardu d'établir la liste des correspondances à suivre pour se rendre d'une station à une autre, les lignes étant d'une couleur identique et entremêlées. Le voyageur est probablement contraint de suivre les lignes du doigt et d'essayer plusieurs trajets avant de choisir le plus simple. Les différentes ligne se distinguent plus facilement sur la carte de 1932 (figure 2.19) qui a affecté à chaque ligne une couleur spécifique. Les informations superflues comme le plan des rues, qui n'est guère utile à l'intérieur du métro, ont été supprimées. La carte de 1933 (figure 2.20) peut paraître à première vue très proche de la précédente et constitue pourtant une évolution majeure. L'élément qui fait se ressembler ces deux cartes est l'utilisation des couleurs précédemment citée et la suppression des informations géographiques relatives au plan de la surface. Cependant ce dernier point n'avait pas été mené jusqu'à son terme sur la carte de 1932. En effet celle-ci pourrait être le calque apposé sur le plan des rues pour réaliser la carte de 1887 car elle restitue fidèlement le parcours géographique des lignes de métro. Pourtant si on pouvait finalement trouver une utilité

FIG. 2.4 – Carte de John Snow (1855) illustrant la répartition géographique des cas de choléra et des pompes à eau dans le quartier de Soho à Londres lors de l'épidémie de 1854.

à cette fidélité dans la carte de 1877 – pour se repérer une fois sorti du métro par exemple – elle n'en a plus aucune hors du contexte géographique que constituait le plan des rues. La carte de 1933 quant à elle s'abstrait presque totalement de ces considérations géographiques et ne conserve que les repères fondamentaux : les points cardinaux pour indiquer la direction globale d'une ligne et la Tamise comme repère relatif des positions des stations à l'intérieur de la ville. Il est alors bien plus aisé d'écrire le nom des stations de manière lisible et d'identifier les correspondances nécessaires à un trajet. On pourrait objecter la perte de distances physiques observables entre les stations ; toutefois pour un voyageur les distances dans le métro se compte plus en nombre de stations qu'en mètres, or sur la carte de 1933, l'espacement constant entre stations permet de compter les stations bien plus rapidement que sur les précédentes cartes. Cet exemple montre qu'il est fondamental de prendre en compte le contexte d'utilisation lors de la conception d'une représentation visuelle.

Au début de la seconde moitié du XX^{e} siècle, de nouveaux horizons s'ouvrent avec l'apparition des premiers ordinateurs qui rendent possible le traitement rapide et automatisé de données. Le développement de l'analyse de données exploratoire, dans laquelle les représentations graphiques jouent un rôle fondamental, de même que les recherches de Jacques Bertin sur la formalisation du langage visuel, démontrent un regain d'intérêt pour la visualisation de données.

Nous présentons dans la suite les développements qu'a connu la visualisation de données à

l'ère informatique.

À une époque où les capacités de calcul et donc de production de données sont encore limitées, les statisticiens sont une des rares communautés à manipuler de grands volumes de données et les premiers demandeurs en méthodes et outils capables de les représenter visuellement. Il s'agissait généralement de graphiques à deux dimensions dessinés en couleurs sur du papier millimétré destinés à mettre en évidence certaines caractéristiques remarquables, à illustrer un discours ou un raisonnement et à étayer les conclusions. Jacques Bertin est un des premiers à s'intéresser à une formalisation des représentations visuelles. Dans son ouvrage *Sémiologie graphique* [Ber67] paru en 1967, il définit les sept variables rétiniennes élémentaires à la base de tout codage visuel : la position, la taille, la couleur, la forme, la texture, l'intensité et l'orientation. Particulièrement visionnaires, ses travaux constituent aujourd'hui encore le socle de la formalisation des représentations visuelles. Les premières tentatives de visualisation de données multidimensionnelles voient le jour au début des années 1970 : les diagrammes en radar (*starplot*) apparaissent en 1971 et les visages de Chernoff en 1973 [Che73]. Dans ce dernier cas, il s'agit d'affecter chacune des dimensions à une caractéristique d'un visage (espacement des yeux, taille du nez, courbure de la bouche). Le principe est le même que pour les diagrammes en radar mais s'appuie sur une métaphore visuelle concrète permettant de représenter la nature positive ou négative d'une information. Ainsi dans l'exemple illustré par la figure 2.22 (p.36) le taux de chômage est affecté à la courbure de la bouche de telle sorte qu'un taux de chômage bas évoque un visage heureux ◡̈ et inversement ◠̈, à la manière des émoticônes (*smileys*).

La publication de l'article *The Future of Data Analysis* [Tuk62] du statisticien John Tukey constitue un acte fondateur de l'appropriation des représentations visuelles de données par les scientifiques. Tukey est le père de l'analyse de données exploratoire [Tuk77], qui se distingue de l'analyse de données traditionnelle qualifiée d'analyse confirmatoire. Il considère que les statisticiens ont développé beaucoup de méthodes, d'outils et de techniques pour tester des hypothèses sur des données (analyse de données confirmatoire) mais que peu se sont intéressés aux moyens d'utiliser les données pour suggérer ces hypothèses. Les représentations graphiques sont au centre de l'analyse de données exploratoire dans la mesure où elles permettent de révéler la structure des données et de suggérer les modèles statistiques à tester. L'ordinateur personnel, dont les premiers modèles apparaissent à la fin des années 1970, dote les scientifiques d'un puissant outil qui leur permet de mettre en pratique les préceptes de Tukey : les données peuvent être visualisées en temps réel à chaque étape du processus d'analyse. L'ouvrage de Edward Tufte, *The visual display of quantitative information*, paru en 1983, revient sur les origines de la visualisation au travers d'exemples historiques et expose certains principes généraux à respecter, comme la proscription d'éléments graphiques redondants ou superflus (*chartjunks*) qu'il généralise à travers la notion de « rapport données-encre » (*data-ink ratio*). Cette prise de recul, signe de maturité de la discipline, aboutit en 1987 à l'atelier *Visualization in scientific computing* organisé par la National Science Foundation [MDB87] qui consacre la visualisation de données comme un domaine de recherche à part entière [Tho05, JMM$^+$].

Arrivés à cette étape cruciale de l'historique il nous semble opportun d'exposer les différentes propositions qui ont été émises pour définir ce nouveau domaine de recherche. Concernant la visualisation au sens le plus large, la définition suivante a été proposée par Colin Ware.

> *A graphical representation of data or concepts, which is either an internal construct of the mind or an external artifact supporting decision making.*
>
> Colin Ware [War04]

Jarke van Wijk rappelle toutefois que le terme « visualisation » est ambigu [vW05]. Il peut renvoyer à la fois à un domaine de recherche, à une technologie, à une technique particulière ou à un résultat visuel. En outre, la visualisation ne peut être abordée séparément de la finalité pour laquelle elle est employée. Au cours du temps, l'emploi de la visualisation a varié selon les besoins des utilisateurs, conduisant à l'émergence de sous-domaines. Au cours de l'historique, nous avons ainsi pu croiser les termes de représentation visuelle, représentation graphique, diagramme, visualisation de données, visualisation d'information, visualisation scientifique. Les trois derniers termes désignent des disciplines scientifiques dont le but est l'étude et la production de représentations visuelles ou graphiques sous des formes diverses telles que des diagrammes, des cartes, etc. Nous nous penchons à présent sur les différences entre ces trois disciplines. Selon Michael Friendly et Daniel Denis, la discipline la plus générale est la visualisation d'information, les deux autres se distinguant par la nature et l'usage de l'information représentée.

> ***Information visualization*** *is the broadest term that could be taken to subsume all the developments described here. At this level, almost anything, if sufficiently organized, is information of a sort. [...]* ***scientific visualization*** *[...] is primarily concerned with the visualization of 3D phenomena (architectural, meterological, medical, biological, etc.), where the emphasis is on realistic renderings of volumes, surfaces, illumination sources, and so forth, perhaps with a dynamic (time) component. [...] Instead, we focus on the slightly narrower domain of* ***data visualization****, the science of visual representation of "'data"', defined as information which has been abstracted in some schematic form.*
>
> Michael Friendly et Daniel Denis [FD02]

Chaomei Chen, quant à lui, considère que la visualisation d'information manipule des données abstraites, non spatiales et multidimensionnelles par opposition à la visualisation scientifique.

> *Visual representations of the semantics, or meaning, of information. In contrast to scientific visualization, information visualization typically deals with nonnumeric, nonspatial, and high-dimensional data.*
>
> Chaomei Chen [Che05]

Keim, Mansmann, Schneidewind et Ziegler d'une part ; ainsi que Card, Mackinlay et Shneiderman d'autre part, retiennent le caractère abstrait des données manipulées en visualisation d'information et intègrent la notion d'interactivité dans leurs définitions.

> *Information visualization (InfoVis) is the communication of abstract data through the use of interactive visual interfaces.*
>
> Keim, Mansmann, Schneidewind, Ziegler [KMSZ06]

> *The use of computer-supported, interactive, visual representations of abstract data to amplify cognition.*
>
> Card, Mackinlay, Shneiderman [CMS99]

Gee, Yu et Grinstein, adoptent quant à eux une approche plus fonctionnelle reprenant les variables rétiniennes de Bertin. Comme Friendly et Denis, ils ne posent aucune condition sur la nature des données représentées.

> *Information visualizations attempt to efficiently map data variables onto visual dimensions in order to create graphic representations.*
>
> Gee, Yu, Grinstein [GYG05]

Si la définition de la visualisation scientifique semble faire l'objet d'un consensus, il ne semble pas en aller de même pour la visualisation de données et la visualisation d'information. Dans la suite du présent mémoire, nous laisserons la visualisation scientifique de côté et confondrons visualisation de données et d'information en adoptant la définition de Catherine Plaisant, orientée analyse de données exploratoire.

> *Compact graphical presentation and user interface for*
> *– manipulating large numbers of items ;*
> *– possibly extracted from far larger datasets ;*
> *enables users to make*
> *– discoveries,*
> *– decisions, or*
> *– explanations*
> *about*
> *– patterns (trend, cluster, gap, outlier...),*
> *– groups of items, or*
> *– individual items.*
>
> Plaisant [Pla04]

À présent que la visualisation d'information a été définie, revenons un bref instant à l'historique. La période de la fin des années 1980 et du début des années 1990 est très prolifique pour la communauté qui vient de naître en terme de nouvelles propositions techniques. Cependant peu d'efforts semblent être mobilisés pour doter la visualisation d'information d'une formalisation permettant de spécifier, évaluer, comparer et reproduire des résultats. Or, cette formalisation est nécessaire à la résolution de notre problématique : la visualisation de données caractérisées par des attributs et des structures hétérogènes. Le processus de visualisation doit être formalisé, de même que les types d'attributs et de structure, afin de pouvoir adapter les techniques de visualisation existantes à la résolution de notre problème. Nous listons dans la suite les travaux ayant trait à la formalisation de la visualisation de données.

2.2 Formalisation

Jacques Bertin fut un des premiers à s'intéresser à la formalisation des représentations visuelles dès les années 1960 mais il faut attendre vingt ans pour que Jock Mackinlay mette en pratique ces travaux dans le cadre de l'outil de génération automatique de « présentations » nommé APT (*A Presentation Tool)* [Mac86]. Mackinlay étend la liste des variables rétiniennes de Bertin et définit un ensemble de règles de composition graphique permettant de représenter des données relationnelles en 2D. Leland Wilkinson a développé un langage de spécification et une interface permettant d'interpréter des specifications et de générer des représentations [Wil05]. D'autres recherches s'intéressent à une formalisation guidée par l'usage (la tâche que l'utilisateur cherche à réaliser) et intégrent la question de l'interaction avec l'utilisateur [Shn96]. Les travaux

précités concernent un ou plusieurs des cinq facteurs autour desquels s'articule la problématique de la conception des représentations visuelles : les données, la tâche, l'interaction, le niveau d'expertise de l'utilisateur et le contexte d'utilisation [PHP03]. Ces cinq facteurs influencent les choix à effectuer à chaque étape du processus de visualisation décrit par Ed Chi et John Riedel [CR98, Chi00], d'une part, et par Stuart Card, Jock Mackinlay et Ben Shneiderman, d'autre part [CMS99]. Ces deux modèles peuvent être généralisés par le modèle simplifié qu'illustre la figure 2.5.

Dans la suite nous examinons chacune des étapes du processus de visualisation.

2.2.1 Modèle de Card-Chi

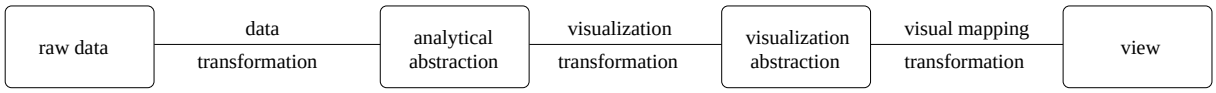


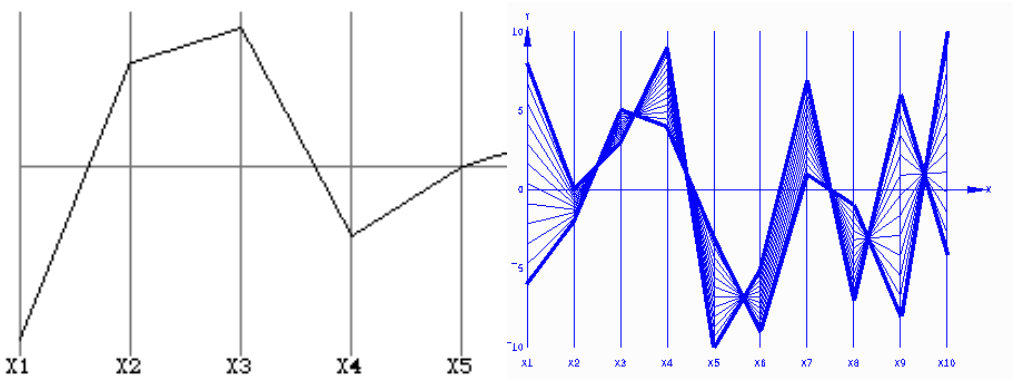
FIG. 2.5 – Modèle de processus de visualisation *data state reference model* commun à Card *et al.* [CMS99] et Chi *et al.* [CR98, Chi00].

Données brutes

La nature et la structure des données à représenter sont déterminantes dans les choix à effectuer pour construire une représentation visuelle efficiente. En effet, à chacune des quatre structures-types énumérées ci-après correspondent des patrons de visualisation et d'interaction [Shn96, Nor05].

Structure tabulaire Les données se présentent sous la forme de tableaux dans lesquels les lignes sont des objets (ou individus) et les colonnes des attributs. Elles définissent un espace multidimensionnel dans lequel chaque attribut constitue une dimension et chaque objet un point de cet espace. Une représentation visuelle intuitive est le nuage de points en système de coordonnées cartésien dans lequel chaque axe est associé à une dimension, cependant seuls trois attributs au plus peuvent être représentés simultanément. Les coordonnées parallèles (*parallel coordinates*) [Ins97] constituent une alternative permettant de représenter l'intégralité des dimensions de l'espace. Les axes des attributs sont représentés par des lignes verticales parallèles. Un objet est alors illustré par la ligne brisée passant par les valeurs respectives de l'objet sur chaque ligne verticale. Des propriétés particulières peuvent être observées entre deux attributs dont les axes sont adjacents. Si les lignes (représentant les objets) sont parallèles entre les deux axes, une relation linéaire positive existe entre les deux attributs x_i et x_{i+1}. Dans le cas contraire, ils sont en relation linéaire négative. Bien entendu ces interprétations dépendent du choix de l'échelle de valeurs et de l'orientation des valeurs sur les axes. La figure 2.6 illustre un exemple d'utilisation des coordonnées parallèles. Un autre exemple de technique permettant de représenter l'intégralité des dimensions est la matrice de nuages de points (*scatterplot matrix*) [Cle93] (cf. fig. 2.7). Il s'agit d'une matrice carrée de taille $|A|^2$ (où $|A|$ désigne le nombre d'attributs) dans laquelle la case ij contient le nuage de points avec l'attribut a_i en abscisses et a_j en ordonnées. Ces deux solutions représentant explicitement l'intégralité des attributs présentent deux inconvénients : d'une part elles ne sont pas adaptées aux attributs binaires ou nominaux, d'autre part elles deviennent difficilement lisibles lorsque le nombre d'objets ou le nombre d'attributs augmente. Des techniques

d'interaction, telles que le *brushing* sur les matrices de nuages de points [BC87, EDF08], ont été introduites afin de palier ce second problèmes.



(a) (b)

FIG. 2.6 – Représentation en coordonnées parallèles d'un point dans un espace à 6 dimensions (a) et d'un ensemble de points dans un espace à 10 dimensions (b). Des lignes parallèles entre deux attributs adjacents indiquent une relation linéaire positive (x_6 et x_7). Des lignes se croisant indiquent une relation linéaire négative (x_7 et x_8).

Structure de graphe Une structure de graphe est utilisée pour représenter des relations (arêtes ou arcs) entre objets (sommets). Au même titre que les objets, les relations peuvent contenir des attributs. Un ensemble de pages web reliées par des hyperliens est un exemple de données présentant une structure de graphe. Les arbres constituant un cas particulier très répandu et possèdant des propriétés spécifiques liées à leur caractère hiérarchique sont l'objet d'un intérêt particulier. On distingue deux approches dans la manière de les représenter visuellement : l'une fondée sur l'utilisation de liens, l'autre de conteneurs. Dans la première approche, les objets sont représentés par des nœuds reliés entre eux. Lorsque le nombre d'objets est important, cette représentation est souvent associée à des dispositifs d'interaction de type « *focus + context* » visant à optimiser la lisibilité des nœuds au voisinage du nœud courant [CMS99]. Parmi les techniques de visualisation d'arbres fondée sur des liens, citons SpaceTree [PGB02] (cf. fig. 2.8), HyperbolicTree [LRP95] (cf. fig. 2.9), ConeTree [RMC91]. La seconde approche exploite l'aspect ensembliste de la hiérarchie et reprend le principe des diagrammes de Venn. L'exemple le plus connu est TreeMap de Ben Shneiderman [Shn92]. Les objets sont représentés par des rectangles et le rectangle d'un objet fils est contenu dans le rectangle de son père (cf. fig 2.10). Concernant les structures de graphes qui ne sont pas des arbres, la technique de visualisation couramment utilisée est une représentation nœuds/liens, similaire à la première approche présentée pour les arbres. De nouvelles contraintes émergent toutefois, comme le croisement d'arêtes qu'il convient de minimiser. La communauté « dessin de graphes » (*graph drawing*) a produit de nombreux résultats applicables à la visualisation d'information [DBETT94, HMM00b].

Structure de type collection de documents Apparues avec l'accroissement des capacités de stockage, les collections de textes, d'images ou de musique enregistrée, sont complexes à visualiser de par le caractère hétérogène des objets et la cohabitation de différentes structures de données précédemment citées. Ce type de structure hétérogène se rapproche de la problématique

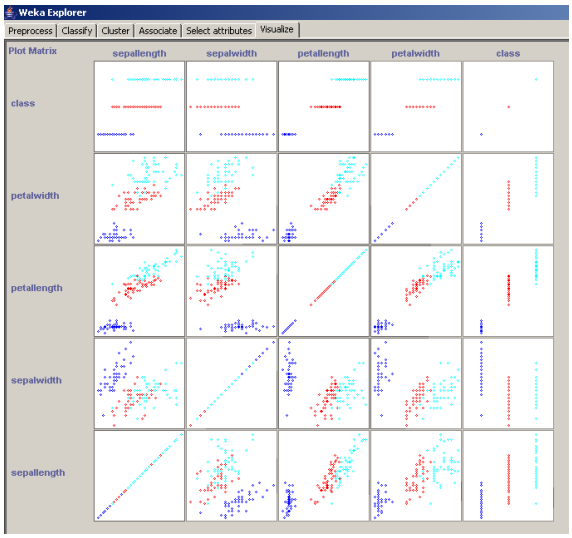


FIG. 2.7 – Matrice de nuages de points de données tabulaires comportant cinq attributs (jeu de données *Iris*).

traitée dans le présent manuscrit. Ainsi, dans une collection de documents textuels, certaines données ont une structure d'arbre (la table des matières d'un recueil) d'autres une structure tabulaire (les métadonnées telles que le nombre de pages, le prix) d'autres encore une structure de graphe (les citations entre documents). De nombreuses solutions de visualisation existent [TRFR06], nous en donnons ici quelques exemples répartis en deux familles : les visualisations fondées sur une représentation des similarités entre éléments de la collection et celles fondées sur une représentation explicite de la structure de la collection.
La première famille vise à produire une vue d'ensemble de la collection et à organiser les documents de façon à refléter leurs similarités respectives. Les documents dont le contenu est similaire sont visuellement proches de telle sorte que l'utilisateur identifie des *clusters* de documents similaires. Le calcul de similarité entre contenus dépend naturellement de la nature de ce contenu. Les techniques utilisées sont généralement issues de la communauté « recherche d'information » (*information retrieval*). L'utilisation d'une métaphore géographique (distances entre documents) amène à employer les termes de « cartes » ou « paysages » pour désigner ces représentations. Parmi les outils appartenant à cette famille, citons Lighthouse [LA00] et DocCube [MCDA03].
Dans les solutions de visualisation appartenant à la seconde famille, l'utilisateur accède aux documents à travers une représentation explicite de la structure de la collection. Un élement de cette structure constitue un point d'entrée vers un sous-ensemble de documents. Ces sous-ensembles peuvent correspondre aux *clusters* de documents similaires identifiables grâce aux techniques de la première famille. Ainsi, Grokker présente les résultats d'une recherche de pages web sous la forme de cercles représentant des *clusters* de pages similaires. À la manière des TreeMaps, un cercle contient un ensemble de sous-*clusters* (cf. fig. 2.11). La collection peut également être structurée à l'aide d'une représentation du domaine de connaissances considéré (cf. ClusterMap [FSvH06]), sous la forme d'une ontologie par exemple. Un élément de la structure est alors un concept ontologique et le sous-ensemble de documents l'ensemble des documents indexés par ce concept (cf. OntoExplo [HP05]). La visualisation de collections de documents rejoint notre problématique et en constitue un cas particulier. Nous retrouvons en effet la question de l'hétérogénéité de la structure où cohabitent des données à structure tabulaire et à structure de graphe.

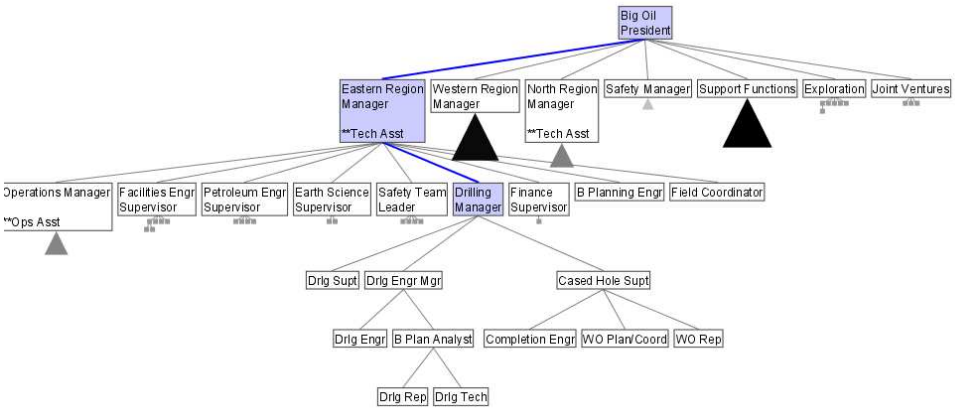


FIG. 2.8 – SpaceTree représente l'arbre de manière traditionnelle mais permet de ne développer qu'une branche à la fois afin de réduire la taille de la représentation. Une prévisualisation miniaturisée des fils d'un nœud permet de maintenir une vue générale de l'arbre.

Les deux familles de solutions exposées se distinguent par le choix effectué en faveur de l'une ou l'autre des structures. La première tire parti de la structure tabulaire pour adopter une visualisation fondée sur une métaphore géographique, tandis que la seconde met l'accent sur la structure de graphe. Notons que ce choix se fait au détriment de l'autre structure. Ainsi, Grokker exploite la structure tabulaire des pages web pour identifier des clusters mais ces similarités entre pages ne sont pas explicitement représentées. Seule la hiérarchie des clusters est visible et il n'est pas possible d'observer le degré de similarité entre deux pages appartenant à deux clusters distincts.

Transformation des données

Cette étape vise à extraire des données l'information à visualiser. Dans le cas d'une collection de documents textuels, le calcul de distance entre documents s'effectue lors de cette étape. Dans le cas de données de type tabulaire, des prétraitements, visant par exemple à sélectionner certains attributs ou à normaliser des valeurs numériques, peuvent être effectués.

Abstraction analytique

L'abstraction analytique est la forme que prennent les données après les éventuelles transformations précédentes. Pour les données de type graphes, ce sera un ensemble de nœuds et de liens. Pour des données de type tabulaire contenant $|O|$ objets et $|A|$ attributs, ce peut être un ensemble de $|O|$ vecteurs à $|A|$ composantes.

Transformation en vue de la visualisation

Cette étape a pour but de produire une représentation visualisable des données. Ainsi une matrice de distance est calculée entre les vecteurs de données tabulaires. Cette étape permet également de filtrer les objets à représenter.

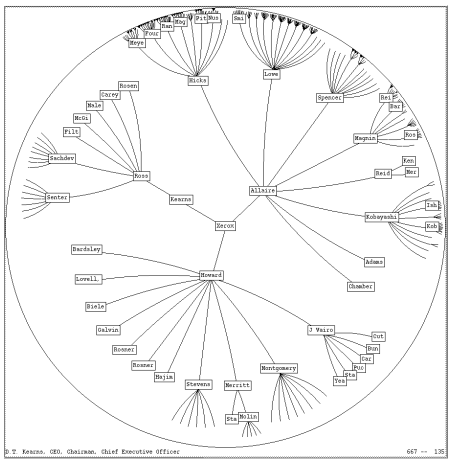


FIG. 2.9 – Représentation d'un arbre par HyperbolicTree. L'arbre est projeté sur un plan hyperbolique, lui-même associé à un espace de représentation circulaire. Le nœud sélectionné est placé au centre de l'espace de représentation et n'est pas nécessairement la racine de l'arbre.

Abstraction visualisable

Une abstraction visualisable est une forme directement visualisable des données.

Rendu visuel

Cette étape met en correspondance d'une part les données, et d'autre part les entités visuelles qui forment la visualisation proprement dite. Ainsi, considérant des données tabulaires représentées par un nuage de points dans un repère cartésien, chaque objet est associé à un point et chacune des deux dimensions choisies à un axe du repère. De même, pour des données sous forme d'arbre, cette étape déterminera le choix de la représentation visuelle (TreeMap ou ConeTree par exemple).

Vue

Une vue est le résultat final du processus, il s'agit de la visualisation à proprement parler. La vue constitue l'interface permettant à l'utilisateur d'observer et d'interpréter les données.

Le modèle de Card-Chi décrit le processus de visualisation aboutissant à une vue, à partir des données, et en fonction de choix de spécification effectués à chaque étape. Toutefois ce modèle ne décrit pas la façon dont l'utilisateur interagit avec la vue, afin d'accroître sa connaissance sur les données. Ce dernier est important car pour évaluer l'efficacité d'une visualisation, un modèle doit permettre de décrire son utilisation et surtout de mesurer et prédire son apport pour l'utilisateur. Jarke J. van Wijk a posé les bases d'un modèle prenant en compte non seulement la construction de la vue en fonction des données mais également les interactions avec l'utilisateur [vW05].

2.2.2 Modèle de van Wijk

L'objectif du modèle de van Wijk est plus de décrire le contexte dans lequel la visualisation est générée et utilisée, que de représenter les étapes du processus de construction de la visualisation à partir des données comme le fait le modèle de Card-Chi. Le modèle de van Wijk est illustré

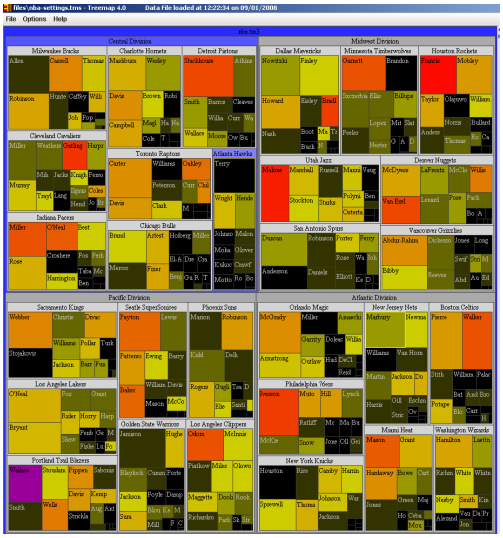


FIG. 2.10 – TreeMap représente les nœuds d'un arbre sous la forme de rectangles et les arêtes par le positionnement du rectangle représentant un fils à l'intérieur du rectangle représentant son père. Les feuilles de l'arbre représenté ici sont les joueurs du championnat de basket-ball américain. Chaque joueur est contenu dans le rectangle représentant son équipe (son père dans l'arbre), elle-même contenu dans le rectangle représentant la division dans laquelle elle évolue.

par la figure 2.12. Il se veut applicable à tout type de visualisation, qu'elle soit visualisation d'information ou visualisation scientifique. Les carrés représentent du contenu et les cercles des processus transformant ce contenu. Le processus central est le processus de visualisation V :

$$I(t) = V(D, S, t)$$

Les données D sont transformées selon des spécifications S en une représentation visuelle dynamique $I(t)$. Aucune hypothèse n'est posée quant à la structure des données D, le but du modèle étant d'être le plus général possible. Les spécifications S doivent être vues comme une configuration pouvant inclure des informations relatives au matériel utilisé comme aux algorithmes et techniques appliqués. L'observation, par l'utilisateur, de l'image I, entraîne une modification de l'ensemble de ses connaissances K :

$$\frac{dK}{dt} = P(I, K)$$

Ce gain en connaissances dépend de l'image I, des connaissances préalables K, et des facultés de perception et de cognition P de l'utilisateur. Le volume de connaissances courant s'exprime en intégrant sur le temps :

$$K(t) = K_0 + \int_0^t P(I, K, t)dt$$

où K_0 désigne les connaissances initiales. Le processus $E(K)$ représente l'exploration interactive, qui intervient lorsque l'utilisateur adapte les specifications S, en fonction des connaissances acquises K, afin de poursuivre l'exploration des données :

$$\frac{dS}{dt} = E(K)$$

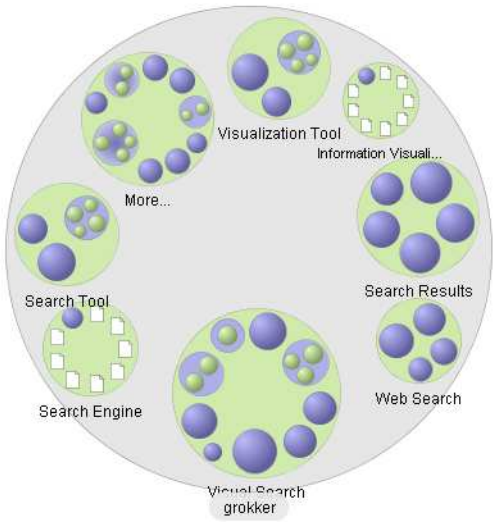


FIG. 2.11 – L'outil de recherche web Grokker présente les résultats d'une requête sous la forme d'une hiérarchie de *clusters* de pages web représentés par des cercles.

Les spécifications courantes s'expriment en intégrant en fonction du temps, pour prendre en compte l'ensemble des modifications successives :

$$S(t) = S_0 + \int_0^t E(K)dt$$

où S_0 désigne les spécifications initiales.

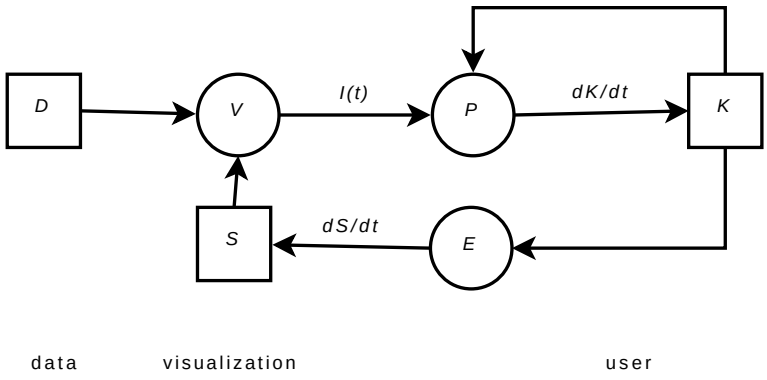


FIG. 2.12 – Modèle de visualisation de Jarke J. van Wijk

Le modèle de van Wijk pose les premières bases d'une formalisation rigoureuse de la visualisation, en réponse au « fossé de crédibilité » évoqué au début de ce chapitre. Bien que le présent manuscrit ne porte pas explicitement sur cette problématique, les solutions de visualisation que nous proposons mettent en œuvre des techniques formelles d'analyse de données. Nous nous inscrivons donc dans cet effort de formalisation des techniques de visualisation.

Nous présentons à présent les trois principaux patrons de navigation, correspondant au processus E dans le modèle de van Wijk. La navigation consiste à adapter les spécifications de

la vue afin d'obtenir un éclairage différent sur les données. Un système de navigation se révèle indispensable lorsque le volume ou la complexité des données croît.

2.2.3 Stratégies de navigation

Les trois patrons de navigation présentés ci-après, *zoom + pan*, *overview + detail* et *focus + context*, ont en commun de présenter dans un premier temps une visualisation globale mais simplifiée des données, puis diffèrent sur la stratégie employée pour afficher les détails relatifs à un sous-ensemble des données.

Zoom + pan

Cette stratégie consiste à zoomer sur la vue globale afin d'afficher les détails d'une zone de la vue globale. Elle présente l'avantage d'offrir une navigation souple et progressive, cependant la vue globale n'est plus présente dès que le zoom est enclenché puisqu'un espace de représentation unique est utilisé.

Overview + detail

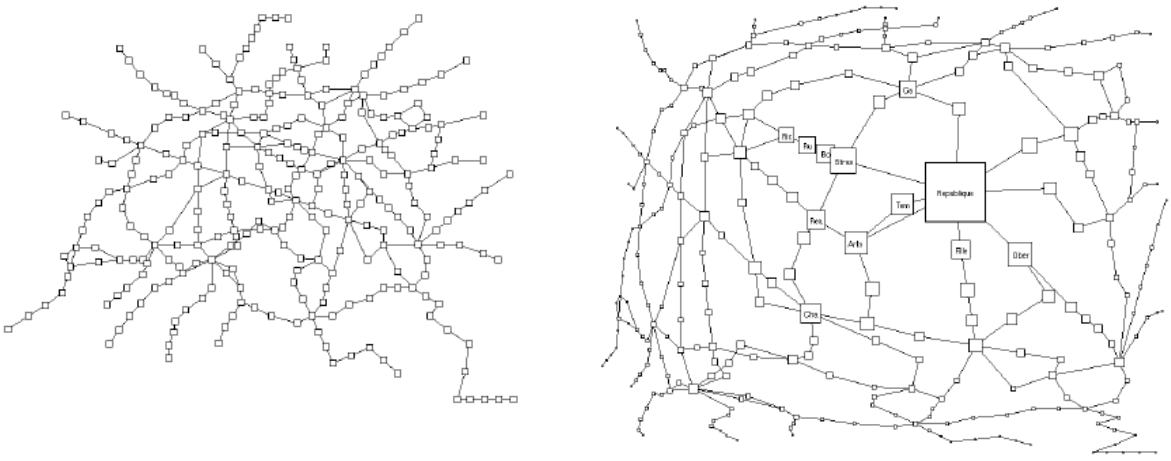
Cette stratégie utilise plusieurs espaces de représentation afin d'afficher simultanément une vue globale et une vue locale. Un repère visuel sur la vue globale permet d'identifier la zone affichée en détails par la vue locale. Ainsi la vue globale est toujours présente et l'utilisateur peut explorer les données en profondeur grâce à la vue locale tout en conservant une vision d'ensemble. L'inconvénient est la surchage cognitive induite par la nécessité d'appréhender simultanément deux vues différentes. Plus généralement, l'emploi de plusieurs vues distinctes, mettant en œuvre des techniques différentes, afin de représenter une même entité, constitue un système à vues multiples. Lorsqu'elles sont coordonnées, i.e. lorsqu'une action de l'utilisateur sur une vue entraîne la mise à jour des autres vues, on parle de vues multiples synchrones ou *multiple coordinated views* [BWK00, AA07].

Focus + context

Cette stratégie reprend le principe de maintenir la vue globale présente mais la vue locale y est incrustée. On distingue d'une part les approches consistant à appliquer une déformation au niveau de la représentation, afin de zoomer localement sur le contenu d'une zone à la manière d'une loupe (cf. fig. 2.13) [SB92, SR92, SSTR93, FK95] ; et d'autre part les approches appliquant un filtre au niveau des données, défini en fonction d'un centre d'intérêt manifesté par l'utilisateur (*degree of interest*) [Fur86, FWR99].

2.2.4 Choix effectués

Parmi les diverses techniques et stratégies présentées dans ce chapitre, nous revenons sur celles qui seront mises en œuvre dans la suite du présent manuscrit. Les données qui font l'objet de notre problématique se caractérisent par une structure hétérogène mêlant structure de type tabulaire et structure de type graphe. Les collections de documents présentent elles-aussi une structure hétérogène, cependant nous avons vu que les solutions existantes ne représentent explicitement que l'une ou l'autre de ces structures. Il s'agit généralement de solutions *ad hoc* où le choix de la structure représentée est fondé sur la nature des documents. Notre problématique étant plus générale, nous ne souhaitons pas à ce stade privilégier une structure au détriment de l'autre. Nous



(a) (b)

FIG. 2.13 – Utilisation du *fisheye* pour la navigation *focus + context*. La figure (a) représente le plan général du métro parisien sous sa forme traditionnelle. La figure (b) illustre le résultat de l'application du *fisheye* lors de la sélection de la station *République* [SB94]. Les détails du réseau à proximité de cette station, comme le nom des stations, sont affichés, tout en conservant une vision globale du réseau. Toutefois l'effet de distorsion rend l'interprétation de certains aspects difficiles, notamment l'orientation des différentes lignes de métro.

nous orientons donc vers une séparation des deux structures, suivant le schéma de la figure 2.14 issu de [KMS02]. Deux abstractions analytiques sont extraites : à partir des données tabulaires, un ensemble d'objets munis d'un vecteur d'attributs d'une part, et d'autre part un ensemble de sommets et d'arêtes représentant les relations entre objets. La première permet de définir un espace multidimensionnel et d'adopter une métaphore géographique pour représenter des proximités entre objets en fonction de leur vecteur d'attributs. La seconde permet de représenter des liens entre objets à l'intérieur de cet espace multidimensionnel.

La section suivante présente les différentes techniques permettant de calculer des similarités entre objets à partir d'un vecteur d'attributs, afin de construire une représentation des objets fondée sur une métaphore géographique.

2.3 Dissimilarité, distance et visualisation de proximités

Une visualisation reposant sur une métaphore géographique met en correspondance deux grandeurs : la *proximité* entre deux objets au regard de leurs attributs, et la *distance* entre ces mêmes objets dans un plan. Il est donc nécessaire de quantifier la proximité entre les objets à représenter. On préférera mesurer la dissimilarité plutôt que la similarité puisque la mesure sera utilisée pour représenter une proximité géométrique. En effet plus les objets sont similaires, plus ils devront être proches. Le calcul de la dissimilarité entre objets dépend de la nature des attributs sur lesquels ils sont valués. Après avoir rappelé les définitions formelles de la dissimilarité et de la distance, nous exposons dans la suite de cette section les fonctions de dissimilarité [And73, NC05] applicables selon les types d'attributs présentés dans la section 1.1.

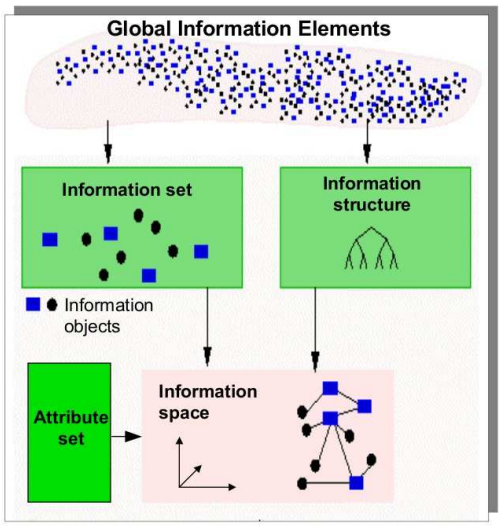


FIG. 2.14 – Principe de séparation objets/structure [KMSZ06].

2.3.1 Définitions

En mathématiques, une *distance* d est définie sur un ensemble E et est une fonction $d : E \times E \longrightarrow \mathbb{R}$ telle que, quels que soient x,y,z de E, les propriétés suivantes sont vérifiées :

$$\begin{aligned} d(x,y) &\geq 0 && \text{(positivité)} \\ d(x,y) = 0 \Rightarrow x &= y && \text{(séparation)} \\ d(x,y) &= d(y,x) && \text{(symétrie)} \\ d(x,y) &\leq d(x,z) + d(z,y) && \text{(inégalité triangulaire)} \end{aligned}$$

Dans l'espace euclidien $\mathbb{R}^n$, la distance la plus intuitive est la distance euclidienne mais elle n'est pas la seule utilisable. En effet, d'autres distances dans $\mathbb{R}^n$ peuvent être définies à partir d'autres normes vectorielles que $||\vec{xy}|| = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}, x, y \in \mathbb{R}^2$, sur laquelle repose la distance euclienne et qui provient du théorème de Pythagore. La distance de Minkowski généralise la distance euclidienne à d'autres normes.

$$\begin{aligned} & & d(x,y) &= \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p} && \text{distance de Minkowski} \\ p = 1 & & d(x,y) &= \sum_{i=1}^{n} |x_i - y_i| && \text{distance de Manhattan} \\ p = 2 & & d(x,y) &= \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2} && \text{distance euclidienne} \\ p \to \infty & & d(x,y) &= \sup_i |x_i - y_i| && \text{distance de Chebyshev} \end{aligned}$$

On ne s'intéressera dans la suite qu'à la distance euclidienne.

En statistique, une *dissimilarité* δ est définie sur un ensemble fini O à n éléments numérotés $\{1, \ldots, i, \ldots, n\}$ et est une fonction $\delta : O \times O \longrightarrow \mathbb{R}$ telle que, pour tout i et j :

$$\delta_{ij} \quad \geq \quad 0$$

$$\begin{aligned} \delta_{ii} &= 0 \\ \delta_{ij} &= \delta_{ji} \end{aligned}$$

Une dissimilarité est dite *métrique* [GL86] si pour tout i, j et k éléments de O on a $\delta_{ij} \leq \delta_{ik} + \delta_{kj}$. Une dissimilarité est dite *euclidienne* [GL86] s'il existe n points dans un espace euclidien dont les distances deux à deux sont exactement les dissimilarités considérées. Seule une dissimilarité euclidienne peut donc être directement assimilée à une distance et être utilisée pour spatialiser les objets. Toutefois il existe une constante additive permettant de rendre euclidienne une dissimilarité [Tor58]. Le calcul de cette constante a fait l'objet de nombreux travaux dépassant le cadre du présent rapport.

2.3.2 Fonctions de dissimilarité usuelles

On notera x_a la valeur de l'attribut a pour l'objet x et δ^a_{xy} la dissimilarité entre deux objets x et y selon l'attribut a. Afin de d'homogénéiser les domaines de valeurs des fonctions de dissimilarité, celles-ci peuvent être normalisées entre 0 et 1. Les normalisations simples présentées dans la suite fonctionnent par translation et homothétie, il existe toutefois des tranformations plus perfectionnées tenant compte notamment de la distribution des valeurs sur l'intervalle [HMM00a].

Attributs numériques

Considérant n objets et un attribut numérique a, la dissimilarité entre deux objets x et y est définie par :

$$\delta^a_{xy} = |x_a - y_a|$$

et peut être normalisée comme suit :

$$\delta^a_{xy} = \frac{|x_a - y_a|}{R_a}$$

où $R_a = \max_k k_a - \min_k k_a$ désigne le domaine de valeurs de a et permet de vérifier $\delta^a_{xy} \in [0, 1]$.

Attributs ordinaux

Les valeurs d'un attribut ordinal étant ordonnées, e.g. *passable* > *bien* > *excellent*, il est possible d'affecter à chacune un entier dans $\mathbb{N}^\star$ correspondant à son rang : $r(passable) = 1$, $r(bien) = 2$, $r(excellent) = 3$. Dès lors, on se ramène à une dissimilarité entre attributs numériques :

$$\delta^a_{xy} = |r(x_a) - r(y_a)|$$

à normaliser en divisant par $k_a - 1$ où $k_a = \max_z r(z_a)$ désigne le nombre de rangs distincts.

Attributs nominaux

Les valeurs d'un attribut nominal n'étant pas ordonnées, la dissimilarité est simplement :

$$\delta^a_{xy} = \begin{cases} 0 & \text{si } x_a = y_a \\ 1 & \text{sinon} \end{cases}$$

Attributs binaires

Un attribut binaire étant un cas particulier d'attribut nominal à deux modalités, la dissimilarité nominale peut être employée si les deux modalités ont la même importance sémantique, comme dans le cas d'un attribut *genre : masculin/féminin*. Toutefois il arrive souvent que la sémantique d'un attribut binaire consiste en la présence ou l'absence d'une caractéristique, e.g. *signes particuliers : oui/non* où seules les valeurs positives sont significatives, le partage de la valeur *non* ne rapprochant pas réellement les objets d'un point de vue sémantique. On parle alors d'information asymétrique puisque les deux modalité n'ont pas la même importance sémantique. Dans ce cas la dissimilarité est définie comme suit, considérant que 1 représente la modalité positive :

$$\delta_{xy}^{a} = \begin{cases} 0 & \text{si } x_a = y_a = 1 \\ 1 & \text{sinon} \end{cases}$$

Vecteurs d'attributs hétérogènes

Les dissimilarités précédentes permettent de comparer deux objets selon un seul attribut d'un type particulier. Or, dans les données qui seront examinées dans la suite du présent manuscrit, les objets sont souvent valués sur un vecteur d'attributs. Gower [Gow71] a défini un coefficient général de similarité à partir duquel une dissimilarité générale peut être construite [Bas00] :

$$\delta_{xy} = \frac{\sum_{a=1}^{q} w_{xy}^{a} w_a \delta_{xy}^{a}}{\sum_{a=1}^{q} w_{xy}^{a} w_a}$$

où x et y sont décrits sur un vecteur de q attributs. w_{xy}^{a} prend la valeur 1 si x et y sont comparables sur l'attribut a, 0 sinon. Deux objets sont comparables sur un attribut s'ils possèdent tous deux une valeur pour cette attribut. Si cette valeur est manquante pour au moins l'un d'eux, l'attribut n'est pas pris en compte dans le calcul de la dissimilarité générale. $w_a \in [0, 1]$ permet de pondérer la contribution de chaque attribut.

Cas des vecteurs d'attributs binaires asymétriques

En présence de vecteurs d'attributs binaires asymétriques creux, i.e. présentant un grand nombre de valeurs négatives non signifiantes, celles-ci font tendre la dissimilarité générale de Gower vers 1, éclipsant les valeurs positives seules signifiantes. Une alternative consiste à utiliser la dissimilarité de Jaccard [Jac01] qui pour comparer deux vecteurs ne prend en considération que les attributs présentant une valeur positive sur au moins un des deux vecteurs. Cette dissimilarité, qui est une distance, notée $J_\delta(x, y) = 1 - J(x, y)$ est construite à partir de l'indice de similarité de Jaccard $J(x, y)$ avec :

$$J(x, y) = \frac{|x \cap y|}{|x \cup y|} = \frac{M_{11}}{M_{01} + M_{10} + M_{11}}$$

où M_{ij} désigne le nombre d'attributs ayant la valeur i sur le vecteur x et j sur le vecteur j. Ainsi, M_{11} représente le nombre d'attributs positifs sur les deux vecteurs. On note que le nombre d'attributs négatifs sur les deux vecteurs M_{00} n'intervient pas. Ainsi, pour $x = \{1, 1, 1, 1, 0, 0, 0, 0\}$ et $y = \{1, 1, 1, 0, 0, 0, 0, 0\}$, la dissimilarité de Jaccard vaut $J_\delta(x, y) = 1 - \frac{3}{0+1+3} = \frac{1}{4}$ lorsque la dissimilarité de Gower donne $\delta_{xy} = \frac{5}{8}$. On peut toutefois ramener Gower à Jaccard en posant $w_{xy}^{a} = 0$ pour les attributs a ayant des valeurs négatives sur x et y.

2.3.3 Techniques de projection

Une fois la matrice de distances entre objets calculée, la projection consiste à associer à chaque objet un point muni de coordonnées cartésiennes de telle sorte que la disposition des objets sur le plan représente le plus fidèlement possible leurs dissimilarités respectives. Formellement, les techniques présentées ici prennent en entrée un ensemble d'objets munis d'un vecteur d'attributs à n éléments (l'espace d'origine à n dimensions) et donnent en sortie le même ensemble d'objets munis d'un vecteur à 2 dimensions (le plan). Ce sont des techniques de réduction dimensionnelle visant à réduire le nombre de dimensions des données, que nous appliquons au cas particulier où l'espace d'arrivée est de dimension 2.

Multidimensional Scaling

Le Multidimensional Scaling (MDS) [KW78, BG97, BG03] est une technique de réduction multidimensionnelle couramment utilisée en visualisation d'information pour projeter des dissimilarités entre objets sur le plan [SF03]. MDS repose sur une approche itérative illustré par la figure 2.15.

1. initialisation (aléatoire ou non) des coordonnées des objets dans le plan,
2. calcul des distances entre objets dans le plan,
3. calcul des disparités entre les distances dans le plan et les distances dans l'espace d'origine,
4. calcul d'une fonction de coût à partir de ces disparités,
5. mise à jour des coordonnées des objets dans le plan et retour à l'étape 2.

Cet algorithme général a donné lieu à de nombreuses implémentations différentes [Bas00], notamment concernant la fonction de coût employée. Notre choix s'est porté sur l'implémentation de MDS par modèle de ressorts (*spring model*) [BSL+01, Cha96], étant la plus simple et la plus adaptée pour la visualisation d'information interactive. L'appellation « modèle de ressorts » vient des travaux de Peter Eades [Ead84] en dessin de graphes. Eades présente une technique fondée sur une analogie physique avec un système d'anneaux reliés par des ressorts. Les anneaux correspondent aux sommets du graphe à représenter et les ressorts aux arêtes. La longueur au repos d'un ressort prend la valeur de la distance désirée entre les deux sommets qu'il relie, dans notre cas il s'agit de la distance dans l'espace d'origine. Les positions des anneaux sont initialisées aléatoirement, les ressorts se trouvant alors soit compressés, soit étirés. En relâchant les positions des anneaux, les forces d'attraction et de répulsion exercées par les ressorts amènent l'ensemble des anneaux à une position d'équilibre. Cette configuration d'équilibre n'est malheureusement pas nécessairement la configuration d'énergie minimale du système (i.e. de tension minimale entre les ressorts). Toutefois cette technique a donné de bons résultats en dessin de graphes, notamment pour résoudre des contraintes esthétiques telles que la symétrie ou la longueur uniforme des arêtes. De par l'analogie physique, ce type d'algorithme est également appelé *force-directed placement*. Le modèle de ressort est appliqué au MDS en générant des forces proportionnelles aux différences entre les distances entre objets dans l'espace d'origine (distance désirée) et dans le plan (distance constatée), de sorte que le système fasse tendre les distances entre objets dans le plan vers les distances dans l'espace d'origine (cf. fig. 2.16). Une fonction de coût, généralement appelée *stress*, est introduite pour indiquer le niveau d'énergie du système. Cette fonction peut être définie comme suit [Cha96]

$$Stress = \frac{\sum_{i<j} (d_{ij} - \delta_{ij})^2}{\sum_{i<j} \delta_{ij}^2}$$

où d_{ij} représente la distance dans l'espace d'origine entre les objets i et j et δ_{ij} la distance sur le plan. Les forces entre atomes sont recalculées à chaque itération. La force exercée sur un objet i par un objet j est proportionnelle à $|d_{ij} - \delta_{ij}|$. Pour N objets, $N(N-1)$ forces doivent donc être calculées à chaque itération. Le système est dit stable lorsque le gain en stress devient négligeable entre deux itérations : $|stress^t - stress^{t-1}| < \epsilon$. Le nombre d'itérations nécessaire pour parvenir à l'équilibre est en $O(N^3)$ mais la variante présentée par Chalmers [Cha96] permet de le réduire en $O(N^2)$ en réduisant le nombre de forces calculées à chaque itération. À chaque objet i sont associés deux ensembles d'objets V_i et S_i. À chaque itération, un nombre donné d'objets, choisis aléatoirement, sont affectés à S_i et leurs distances désirées à i sont comparées avec celles des objets contenus dans V_i. Si la distance à i d'un objet de S_i est inférieure à au moins une des distances des objets de V_i à i, il est ajouté à V_i et retiré de S_i. Les forces ne sont alors calculées qu'entre i et les objets de V_i et S_i. Au fur et à mesure des itérations, V_i contient les plus proches voisins de i. Ainsi, le calcul des forces se concentre sur les voisins de i et sur quelques objets choisis aléatoirement. Les tailles de V_i et S_i étant bornées par des constantes, le nombre de forces calculées à chaque itération est ainsi réduit de $N(N-1)$ à $N(V_{max} + S_{max})$.

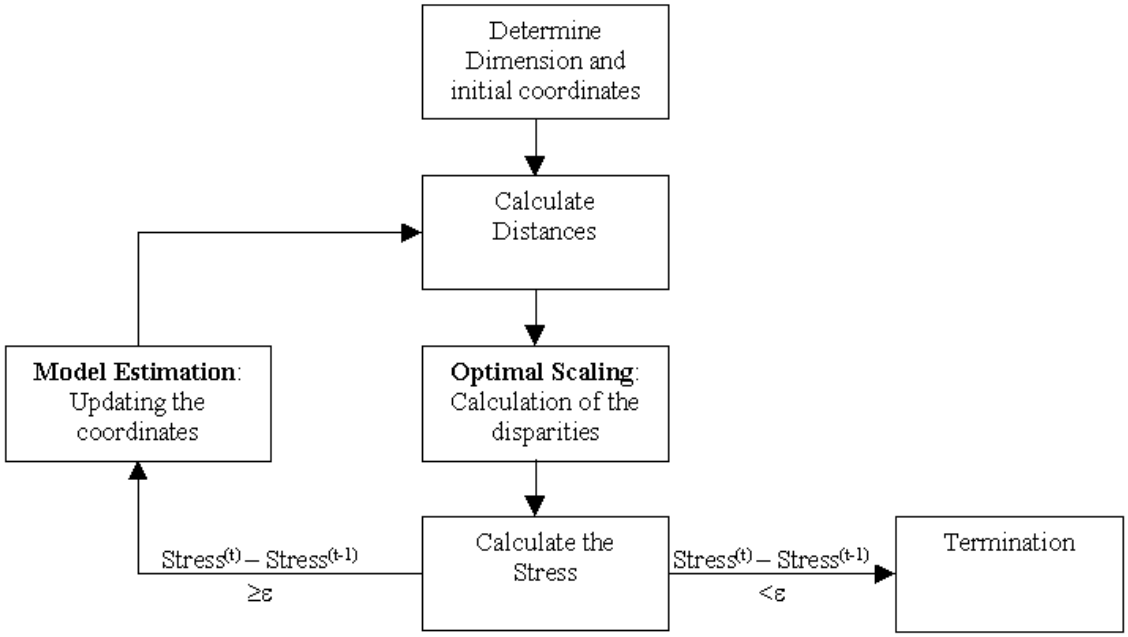


FIG. 2.15 – Algorithme général MDS [VDD00].

Cartes de Kohonen

Une carte de Kohonen [Koh01], également appelée *self-organizing map* (SOM), est obtenue par l'application d'un algorithme d'apprentissage non supervisé sur un réseau de neurones. Le réseau de neurones prend la forme d'une grille représentant le plan. Chaque neurone est associé à un vecteur à n dimensions. Les vecteurs des neurones sont d'abord intialisés aléatoirement. L'algorithme consiste à sélectionner un objet à projeter et à déterminer le neurone dont les valeurs du vecteur sont les plus similaires, i.e. à déterminer le neurone dont la distance entre son

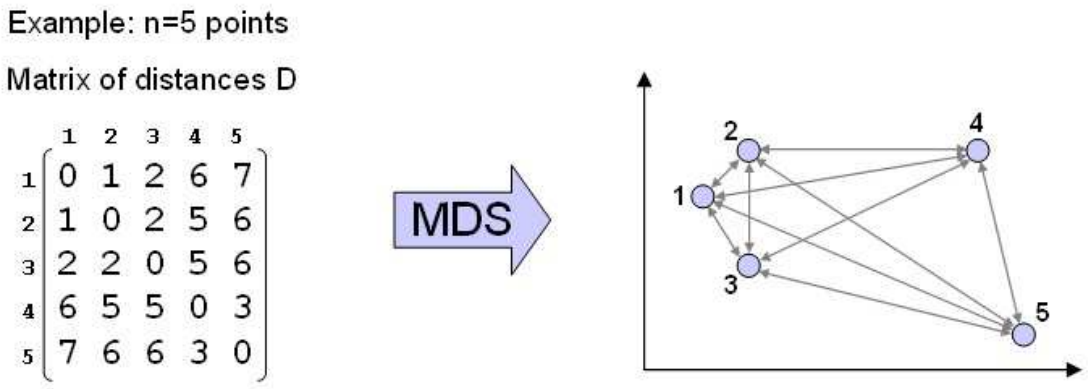


FIG. 2.16 – Multidimensional scaling. L'algorithme consiste à déterminer une configuration des objets dans le plan de sorte que leurs distances relatives dans l'espace d'origine soient respectées au mieux sur le plan.

vecteur et le vecteur de l'objet est minimale. On nomme ce neurone « neurone gagnant ». L'idée est d'affecter le vecteur de l'objet au neurone gagnant et de modifier les vecteurs des neurones voisins de façon à ce qu'ils deviennent similaires à celui du neurone gagnant. L'algorithme itère ensuite en sélectionnant un nouvel objet à projeter. Ainsi si le second objet sélectionné est similaire au premier, son neurone gagnant se trouvera parmi les voisins du neurone gagnant précédent. L'appartenance des neurones au voisinage du neurone gagnant est évaluée par une fonction de voisinage, généralement gaussienne et centrée sur le neurone gagnant. La mise à jour des vecteurs des neurones voisins est alors fonction de leur degré de voisinage : plus on s'éloigne du neurone gagnant, plus les modifications apportées aux vecteurs s'atténuent.

L'inconvénient des cartes de Kohonen réside dans leur tendance à répartir les objets projetés sur la totalité de la grille de neurones, au détriment des distances relatives entre objets. Lorsque MDS, dont le principe même est le maintien des distances entre objets, crée des *clusters* denses d'objets similaires, ceux-ci se retrouvent « étalés » sur une carte de Kohonen [SF03].

Analyse en composantes principales

L'analyse en composantes principales (ACP) [EP88] calcule, à partir d'un ensemble d'objets à n dimensions, des vecteurs à n dimensions appelés « composantes principales ». Ces vecteurs représentent les directions de variance maximale, i.e. les directions dans l'espace d'origine sur lesquelles les objets sont les plus étalés et donc rendant compte au mieux de la dispersion des objets. Les composantes principales sont calculées de telle sorte que :

- elles sont ordonnées par variance décroissante ;
- elles forment une base orthonormale, i.e. elles sont perpendiculaires une à une et de norme unitaire, et donc non corrélées.

Les deux premières composantes principales définissent le plan sur lequel les objets seront projetés. Les coordonnées d'un objet sur ce plan s'obtiennent en calculant le produit scalaire de l'objet sur chacune des deux composantes (cf. fig. 2.17).

L'avantage de l'ACP réside dans le fait que les axes du plan de projection sont des combinaisons linéaires des dimensions d'origine. Ainsi il est possible d'interpréter les positions des

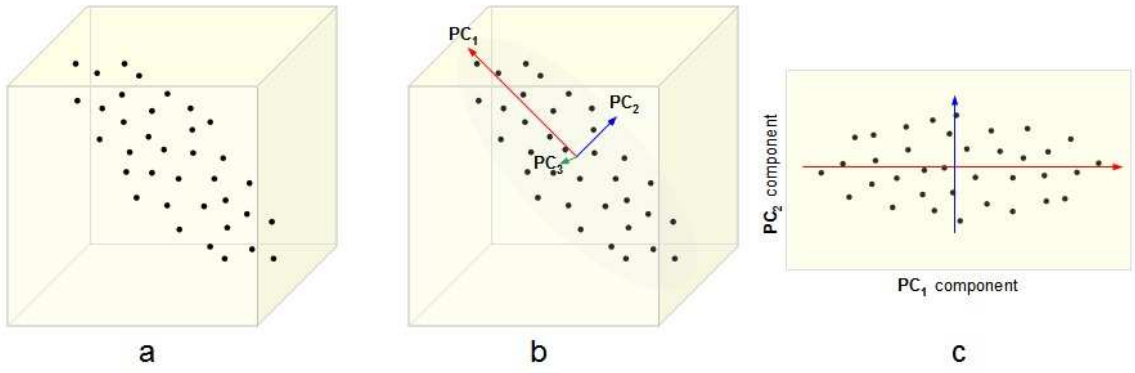


FIG. 2.17 – Analyse en composantes principales. (a) l'espace d'origine à 3 dimensions et l'ensemble des objets. (b) trois composantes principales ordonnées par variance décroissante. (c) le plan de projection défini par les deux premières composantes principales et les objets projetés sur ce plan.

objets sur le plan en fonction des dimensions d'origine et d'évaluer la contribution de chacune de ces dimensions sur les axes, autrement dit de « sémantiser » l'espace de représentation. En MDS comme sur une de Kohonen, les axes du plan n'ont aucun lien avec les dimensions d'origine puisque MDS ne s'intéresse qu'aux distances relatives entre objets, et les cartes de Kohonen aux distances entre les objets et des neurones initialisés aléatoirement. Toutefois, le caractère non itératif de l'ACP ne permet pas à l'utilisateur d'observer le déroulement du processus.

Notre choix s'est porté sur MDS par modèle de ressorts. Sa plus grande fidélité aux distances relatives entre objets nous l'a fait préférer aux cartes de Kohonen et son caractère souple et itératif, donc dynamique, à l'ACP.

2.4 Conclusion

Au cours de ce chapitre, nous avons présenté les travaux relatifs à la formalisation du processus de visualisation. Nous avons vu que des techniques de visualisation différentes sont employées en fonction de la structure des données : tabulaire ou de type graphe. Nous avons présenté les techniques de visualisation existantes pour chacune de ces deux structures sans trouver de solution satisfaisante pour le cas des structures hétérogènes. Les collections de documents présentent une structure hétérogène mêlant les deux structures précédentes mais les solutions de visualisation existantes mettent l'accent sur l'une au détriment de l'autre. Nous nous sommes orientés vers une séparation des deux structures et une visualisation exploitant simultanément la structure tabulaire via une métaphore géographique par projection MDS, et la structure de type graphe via la représentation de liens entre objets. Nous nous sommes ensuite penchés sur le calcul des distances entre objets, nécessaire à la visualisation de la structure tabulaire et avons pointé le rôle joué par la nature des attributs dans ce calcul.
La partie suivante présente nos contributions et les solutions proposées aux problèmes introduits dans le chapitre 1 : la visualisation de données volumineuses comportant des structures et des attributs de nature hétérogène. Le premier chapitre de cette partie s'inscrit dans l'effort de formalisation évoqué dès le début du présent chapitre en proposant un modèle formel du paradigme

FDP. Ce modèle sera utilisé pour spécifier des patrons de visualisation, identifiés à partir de visualisations élaborées dans le cadre de deux projets de recherche, et implémentés dans notre environnement FDP Molage.

Fig. 2.18 – Carte du métro de Londres (1887).

FIG. 2.19 – Carte du métro de Londres par Frederick H. Stingemore (1932).

FIG. 2.20 – Carte du métro de Londres par Harry Beck (1933), réalisée en s'inspirant de schémas électriques (angles à 45 °).

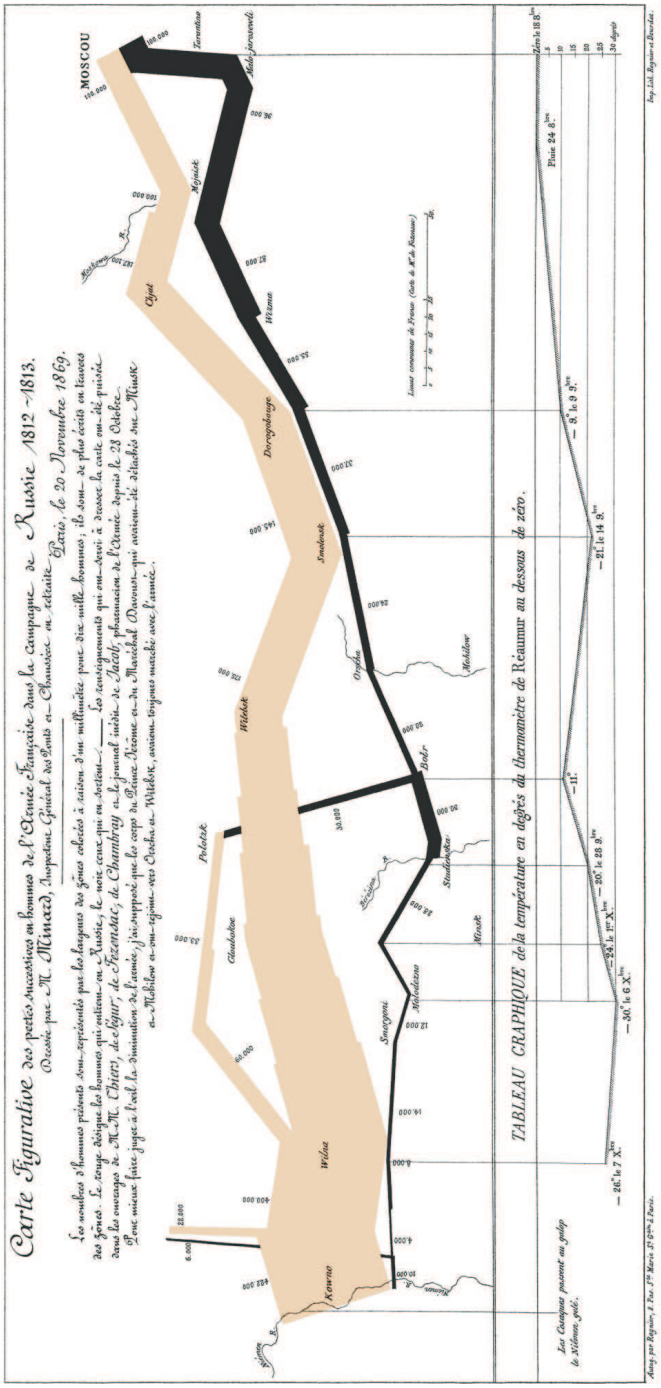


FIG. 2.21 – Carte figurative des pertes successives en hommes de l'armée française dans la campagne de Russie (1812-1813) par Charles-Joseph Minard.

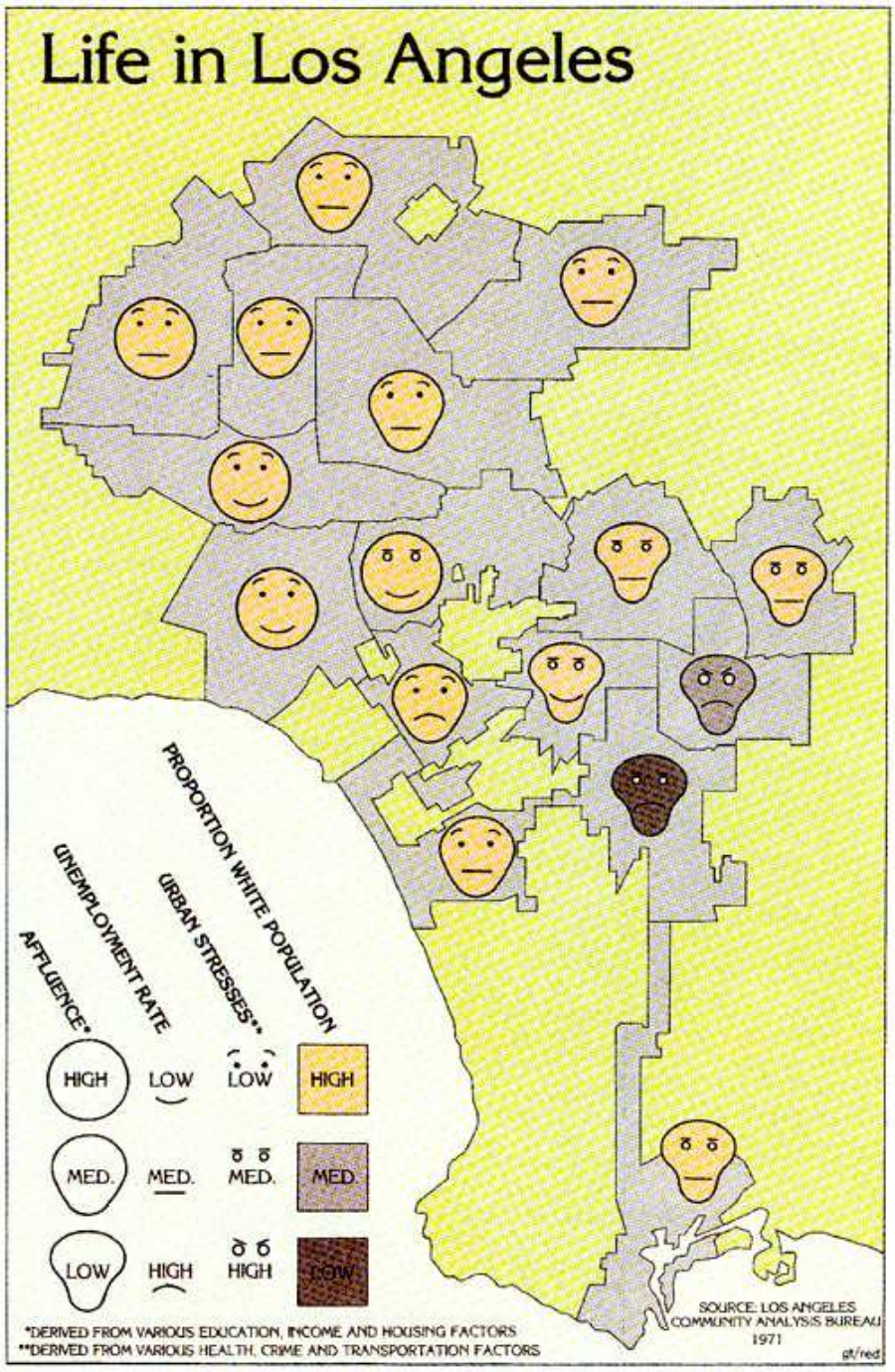


FIG. 2.22 – Carte d'Eugene Turner, géographe à la California State University, Northridge utilisant des visages de Chernoff pour représenter simultanément quatre variables – le niveau de vie (éducation, revenus, logement), le taux de chômage, le niveau de stress urbain (santé, criminalité, transports) et la proportion de blancs – pour chaque quartier de Los Angeles, réalisée en 1979 à partir de données de 1971. L'utilisation des visages de Chernoff révèle immédiatement que les noirs sont concentrés dans deux des seize quartiers et qu'ils subissent les conditions de vie les plus difficiles.

Deuxième partie

Contributions et réalisations

3

Modèles formels de visualisations et expérimentations

Sommaire

Nous abordons dans ce chapitre la question du « fossé de crédibilité » évoqué au cours du chapitre précédent en proposant des spécifications de visualisations s'appuyant sur un modèle formel. Les visualisations présentées dans ce chapitre ont été conçues dans le cadre de deux projets de recherche présentant chacun des structures de données spécifiques. Le modèle formel utilisé pour spécifier ces visualisations décrit les mécanismes mis en œuvre dans le cadre du paradigme de visualisation FDP reposant sur un modèle de forces. Les visualisations ainsi spécifiées ont été implémentées dans l'environnement de visualisation Molage, aux développements récents duquel nous avons participé. Ces expérimentations montrent la pertinence de la formalisation des visualisations en fonction de la structure des données, dans la mesure où

des patrons de visualisation peuvent être dégagés et réutilisés. Nous verrons ainsi qu'un patron de visualisation spécifié dans le cadre du premier projet de recherche a pu être réutilisé dans le second. MOLAGE est un outil de visualisation de données utilisant un modèle de ressorts (FDP) pour animer et positionner dans le plan de l'écran des objets (dénommés « atomes ») représentant des informations ou des connaissances. Le principe FDP consiste à associer entre deux objets reliés pour quelque raison sémantique un ressort dont la longueur est prédéfinie. Éventuellement, pour répondre à certains critères esthétiques comme par exemple la répartition spatiale harmonieuse des objets [DBETT94], une force de répulsion est simulée. En cela MOLAGE ne se distingue pas de nombreuses autres applications basées sur l'algorithme FDP. Cependant, de manière originale, elle simule aussi d'autres forces qui sont très rarement évoquées dans les autres applications. Ainsi la force *limite* permet d'éviter la superposition d'objets tout en ne les éloignant pas, tandis qu'une force *propriété* calcule comme le nous verrons la distance entre les objets pour les positionner à une distance correspondant à leur éloignement, en application du principe de projection MDS. Ces forces sont applicables à une hiérarchie de types d'objets. La combinaison de ces forces offre une grande richesse d'organisation qui est complétée par une approche de contrôle des mouvements d'objets à l'aide d'une interface tournée vers des envois de message aux objets ou aux classes d'objet. On pilote alors l'organisation des atomes de la même façon qu'on enverrait des instructions à des agents.Au final la richesse des organisations possibles s'obtient par une séquence d'instructions envoyées. La recherche de ces instructions et de leur ordonnancement en fonction des données à visualiser constitue à présent l'axe de recherche principal et s'inscrit dans le même effort de formalisation. Un « scénario » est une suite ordonnée d'instructions aux agents sur les forces à déclencher. Pour l'instant pilotés à la main, ils feront ensuite l'objet d'une description en XML et d'une mise en œuvre par un moteur en cours d'implémentation afin de les automatiser.
Les entités du modèle formel que nous présentons dans la section suivante s'appuient sur les objets et dispositifs visuels manipulés dans MOLAGE.

3.1 Entités du modèle formel

3.1.1 Objets, attributs et relations

MOLAGE permet de visualiser des données de type tabulaire et de type graphe. Les entités relatives à ces deux structures sont détaillées dans la suite de cette section.

Données tabulaires

Rappelons que des données tabulaires se présentent sous la forme d'un tableau objets/attributs. Un **objet** est une entité d'information décrite par un ensemble d'**attributs**. Soient O un ensemble d'objets et A un ensemble d'attributs. La valuation des objets sur les attributs est représentée par le graphe biparti $G(V, E)$ où $V = O \cup A$ et $E \subseteq O \times A$. Notons que G n'est pas nécessairement biparti complet, ce qui signifie qu'un objet n'est pas obligatoirement relié à tous les attributs. Le poids associé à une arête $e = (o, a)$ correspond à la **valeur** de l'attribut a pour l'objet o. On notera cette valeur $a(o)$ ou $o.a$. L'ensemble des attributs valués sur un objet o sera noté $o.\vec{a}$.

Un attribut est associé à un ensemble de valeurs possibles. On distingue des **types d'attributs** selon la nature de ces valeurs (cf. section 1.1). On manipulera dans la suite les types d'attributs binaire, nominatif, ordinal et numérique. Un ensemble d'objets représentant sémantiquement la même nature d'information définit un **type d'objets**. MOLAGE offre ainsi la possibilité de typer les objets afin de leur affecter un traitement spécifique.

propriété	valeurs	description
x	[0,500]	position en x
y	[0,500]	position en y
xfixed	{true,false}	position fixée en x
yfixed	{true,false}	position fixée en y
fixed	{true,false}	position fixée en x et y
shape	{circle,triangle,square,image}	forme de l'atome
size	[0,100]	taille en pixels (côté ou diamètre)
color	(R,G,B)	sauf pour forme image
transparency	[0,1]	opacité
label	`texte`	étiquette
labelVisible	{true,false}	affichage de l'étiquette
label2	`texte`	étiquette 2
label2Visible	{true,false}	affichage de l'étiquette 2
visible	{true,false}	affichage de l'atome

TAB. 3.1 – Propriétés rétiniennes d'un atome

Données de type graphe

Une relation entre deux objets est notée $R(p, q)$ et $R(O_1, O_2)$ désigne la relation existant entre les ensembles d'objets O_1 et O_2, i.e. $R(O_1, O_2) := \{(o_1, o_2) \in O_1 \times O_2 | R(o_1, o_2)\}$

3.1.2 Atomes et liaisons

Un **atome** est l'entité visuelle élémentaire de Molage. Un objet est généralement représenté visuellement par un atome. Ainsi un atome est associé, comme un objet, à un vecteur d'attributs valués. Un atome possède un ensemble de propriétés correspondant aux propriétés rétiniennes de Bertin et décrites par le tableau 3.1.

Un **type d'atomes** regroupe un sous-ensemble d'atomes et représente généralement un type d'objets. Une **liaison** est une relation entre atomes utilisée pour représenter une relation entre objets.

3.1.3 Forces

Une force est un mécanisme mis en œuvre pour disposer visuellement un ensemble d'atomes. Nous détaillons dans la suite les quatre types de forces utilisées dans Molage.

Force propriété

La force propriété a pour but de représenter visuellement les similarités entre atomes en fonction de leurs vecteurs d'attributs. Une distance théorique est calculée pour chaque paire d'atomes (cf section 2.3) selon l'approche *Multidimensional Scaling* (MDS) (cf. 2.3.3) puis projetée sur le plan par l'algorithme *Force Directed Placement* (FDP) de Peter Eades basé sur un modèle physique de ressorts consistant à minimiser une fonction de stress global pour parvenir à une position d'équilibre correspondant à une disposition des atomes sur le plan respectant au mieux les distances théoriques.

L'application de la force propriété sur un ensemble d'atomes O valués sur un ensemble d'attributs A consiste en les étapes successives :

1. calcul des distances $O \times O$ selon A,
2. mise à jour des positions de O par FDP.

Seul un sous-ensemble d'attributs $A_0 \subseteq A$ peut être pris en compte dans le calcul de distances (MDS sélectif) :

1. calcul des distances $O \times O$ selon A_0,
2. mise à jour des positions de O par FDP.

La force peut être appliquée afin de positionner un type d'atomes $Q \subseteq O$ d'après les positions d'un type $P \subseteq O$. Il s'agit alors de mettre à jour les positions de Q de telle sorte qu'elles respectent au mieux les distances calculées pour paire (p, q) selon A_0 :

1. calcul des distances $P \times Q$ selon A_0,
2. mise à jour des positions de Q par FDP.

Notons que les positions P restent inchangées et que les similarités entre atomes de Q ne sont pas prises en compte dans le calcul de distance. On appelle P l'**origine** de la force et Q la **destination**. Le respect des distances entre Q peut être ajouté :

1. calcul des distances $P \times Q$ selon A_0,
2. calcul des distances $Q \times Q$ selon A_0,
3. mise à jour des positions de Q par FDP.

Dans ce dernier cas, les positions P restent encore inchangées et Q se positionnent en respectant à la fois les distances des paires (p, q) et des paires (q_i, q_j).

Dans la suite nous noterons l'application d'une force propriété sur un type d'atomes destination Q à partir d'un type origine P selon un ensemble d'attributs A_0 par : $F_P(Q, P, A_0)$.

Force liaison

La force liaison organise les atomes par FDP de manière à respecter au mieux les poids des liaisons entre atomes qui sont assimilés à des distances. La force liaison s'applique sur l'ensemble des atomes munis de liaisons sans distinction entre types : $F_L(O)$.

Force limite

La force limite organise les atomes de telle sorte que chaque atome destination $q \in Q$ soit à une distance minimum δ de chaque atome origine $q \in P$. Elle sera notée : $F_\Delta(Q, P, \delta)$. La force limite peut être appliquée à l'ensemble des atomes : $F_\Delta(O, O, \delta)$.

Force empathie

La force empathie permet de regrouper des atomes d'un type P en les forçant à se rapprocher les uns des autres : $F_E(P)$.

3.1.4 Lentilles

Une **lentille** permet de définir et modifier les propriétés rétiniennes d'un type d'atome. Une lentille l est donc un vecteur de valeurs pour ces propriétés rétiniennes, e.g. l.shape=circle. Chaque type d'atome est associé à une lentille globale et à une lentille de proximité. La lentille globale définit les valeurs par défaut des propriétés tandis que la lentille de proximité contient les valeurs prises lorsque le curseur s'approche d'un atome du type considéré à une distance donnée δ dépendant du type d'atomes. On spécifie les lentilles d'un type d'atomes P par : $Lens(P, l_{glob}, l_{prox}, \delta)$. Ainsi la spécification $Lens(P, l_1, l_2, 10)$ avec l_1.shape=circle et l_2.shape=triangle signifie que la forme d'un atome $p \in P$ sera circulaire par défaut et deviendra triangulaire lorsque le curseur s'approchera à moins de 10 pixels de p.

Une **lentille topologique** est une lentille de proximité particulière permettant à un atome en mode proximité de propager cet état aux atomes avec lesquels il est relié par des liaisons. Cette propagation est répercutée récursivement jusqu'à un nombre de pas n donné. $LensTopo(P, Q_1 \cup \ldots \cup Q_k, n)$ spécifie une lentille topologique pour le type P qui active les lentilles de proximité sur un pas de n. Le second argument restreint la propagation à un ensemble de types qui seront seuls concernés par la propagation.

On appelle paysage l'ensemble des atomes organisé selon un enchaînement de forces.

Au terme de cette section, nous avons à notre disposition :

- une caractérisation formelle des données, de leur structure et la nature de leurs attributs (cf. chapitres 1 et 2 ;
- une caractérisation formelle des entités et dispositifs visuels disponibles dans Molage.

Une représentation visuelle peut donc être spécifiée sous la forme d'un appariement entre ces deux caractérisations [CRV+06b], comme nous le verrons dans les sections suivantes qui décrivent les premières visualisations réalisées.

3.2 Modèle formel et visualisation d'une collection musicale

3.2.1 Le projet de recherche Savic

Nous exposons dans cette section les paysages réalisés dans le cadre du projet Savic. Savic est un projet ANR-RIAM réalisé en partenariat avec la société Netia[2] et s'intéressant à la gestion d'une collection musicale à travers une interface visuelle. L'outil devait permettre d'accomplir les tâches suivantes :

- disposer d'une vue d'ensemble de la collection ;
- accéder à un morceau à partir de son interprète ou de son compositeur ;
- trouver d'autres morceaux d'un même compositeur ou d'un même interprète ;
- observer des proximités entre morceaux selon des critères de ressentis émotionnels (*moods*) ;
- indexer un nouveau morceau en fonction de sa proximité avec des morceaux existants ;
- observer le cheminement d'une *playslist* à travers la collection ;
- définir un modèle de *playlist* et générer de nouvelles *playlists*.

Un « ressenti émotionnel » est un descripteur désignant une sensation éprouvée par l'auditeur lors de l'écoute du morceau. 24 de ces descripteurs ont été identifiés lors d'un précédent projet avec Netia [CRV+06a, CVER07]. Chaque morceau est indexé sur les 24 descripteurs par une valeur entre 0 et 100 décrivant sa position par rapport aux deux ressentis extrêmes. Ainsi, pour le descripteur *Rythme*, la valeur 0 correspond au ressenti *calme* et la valeur 100 à *énergique*. Une *playlist* est une succession de morceaux dont l'ordre est supposé refléter la façon dont l'utilisateur

[2] `http://www.netia.fr`

fait évoluer le climat musical, en commençant par des morceaux très calmes puis en terminant après une gradation par des morceaux énergiques par exemple. La génération automatique de *playlists* à partir d'une *playlist* d'apprentissage consiste à reproduire la même évolution au moyen de nouveaux morceaux. Les travaux ont consisté à enrichir l'interface du précédent projet afin d'obtenir une interface adaptative permettant notamment d'analyser visuellement les apports des différents descripteurs aux proximités entre morceaux.

Nous nous trouvons clairement en présence de données à structure tabulaire : elles se présentent sous la forme d'un tableau objets (morceaux) - attributs (*moods*) et l'objectif premier est de visualiser les proximités entre morceaux. Nous mettons donc en œuvre une visualisation à métaphore géographique via une projection MDS.

3.2.2 Mise en œuvre de la projection MDS

L'objectif est de représenter les similarités entre morceaux en fonction d'un ensemble de ressentis émotionnels A. Les objets sont les morceaux et les attributs les ressentis émotionnels valués entre 0 et 100. Chaque objet est associé à un atome de type M muni de son vecteur d'attributs. On définit les lentilles suivantes :

	l_1	l_2
visible	true	true
shape	image	image
size	10	10
transparency	1	1
labelVisible	**false**	**true**

La valeur *image* pour la propriété *shape* signifie que l'atome sera représenté par une image correspondant ici à la pochette du morceau associé à l'atome. La propriété *label* contient le titre du morceau. Le paysage est spécifié comme suit :

$$Lens(M, l_1, l_2, 5)$$
$$F_P(M, M, A)$$
$$F_\Delta(M, M, 5)$$

et est illustré par la figure 3.1. Les morceaux sont spatialisés selon leur similarité sur l'ensemble des attributs grâce à F_P, F_Δ permet d'éviter que les titres proches se superposent et l_2 affiche le titre du morceau au passage du curseur.

Ce premier paysage remplit l'objectif de représenter les similarités entre morceaux. Toutefois, ces proximités sont calculées une fois pour toutes sur l'ensemble des attributs ce qui empêche l'analyse de la contribution de chaque attributs. En d'autres termes il n'est pas possible de répondre à la question : *pourquoi ces deux titres sont-ils proches ?* L'interface visuelle n'est finalement exploitée qu'en tant que représentation globale de la collection musicale et ne sert pas de support à des tâches d'analyse du contenu de la collection.

Afin de pallier ce manque de sémantique du paysage, on introduit un nouveau type d'atome appelé *indicateur* et noté I. Un atome indicateur est créé pour chaque attribut et est uniquement valué sur cet attribut avec la valeur maximale 100. L'idée est de placer les atomes indicateurs aux positions proches des atomes M dotés d'une valeur proche de 100 sur leurs attributs respectifs. Pour ce faire on active une force propriété supplémentaire $F_P(I, M, A)$. Le résultat est illustré par la figure 3.2. Les atomes indicateurs se disposent sans perturber les positions des atomes M. Ainsi on trouve l'indicateur *Rythme* sur la droite du paysage, près de morceaux ayant pour *Rythme*

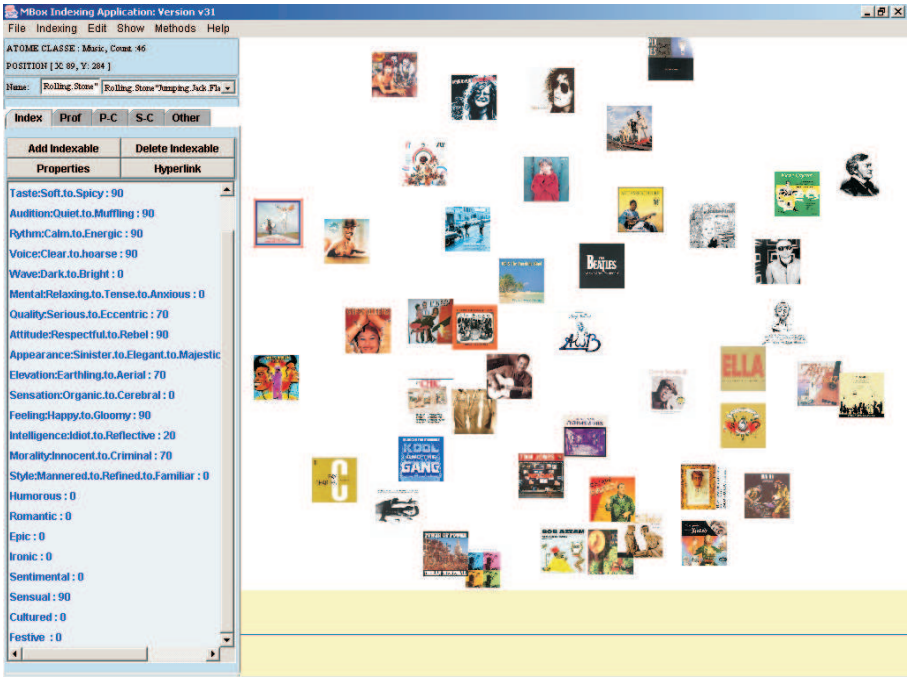


FIG. 3.1 – Premier paysage représentant une collection musicale

la valeur 100 correspondant au ressenti émotionnel *énergique* tels que *Sex machine* de James Brown. L'étude des positions relatives des indicateurs peut également se révéler intéressante : l'indicateur *Rythme* est diamétralement opposé à *Allure*. Cela signifie que les morceaux ayant la valeur 100 pour *Allure*, exprimant le ressenti émotionnel *majestueux*, ont a priori une valeur faible pour *Rythme*, correspondant à *calme*. Cette possible corrélation négative entre ces deux attributs est un exemple d'information nouvelle suscitée par l'utilisation d'une représentation visuelle de la collection.

Ces deux paysages permettent de représenter des similarités entre objets mais deviennent difficilement exploitables lorsque le nombre d'objets augmente. De plus, ils n'offrent pas la possibilité de rechercher des morceaux par d'autres attributs tels que compositeur, interprète, date de composition, etc.

3.2.3 Intégration de nouveaux attributs nominaux

L'introduction de trois nouveaux attributs *compositeur*, *interprète* et *date de composition* conduit à reconsidérer l'ensemble du paysage afin de les représenter et de les exploiter. En effet, *compositeur* et *interprète* sont des attributs nominaux, à la différence de *date* et des 24 attributs d'origine qui sont numériques. Nous avons vu qu'il était possible d'intégrer des attributs nominaux dans le calcul de distance entre objets (cf. 2.3.2), cependant ces attributs sont généralement utilisés comme des points d'entrée dans une collection musicale. En effet, l'utilisateur se sert de ces attributs pour trouver des morceaux dans la collection et il est nécessaire qu'ils soient explicitement représentés sur le paysage. On opère donc une transformation consistant à réifier chaque valeur d'attribut nominatif pour en faire un objet. Ainsi les valeurs *Mozart* et *Beethoven* de l'attribut *compositeur* deviennent deux objets de type *Compositeur*. Les relations *R(Musique,Compositeur)* et *R(Musique,Interpète)* sont introduites afin de conserver les associa-

FIG. 3.2 – Deuxième paysage représentant une collection musicale

tions entre morceaux, compositeurs et interprètes. Un nouvel attribut *alpha* est introduit pour les types *Compositeur* et *Interprète* contenant le rang de la première lettre de leur nom dans l'aphabet. Ainsi, $albeniz.alpha = 1$ et $bach.alpha = 2$. De plus un attribut *nais* (date de naissance) est défini pour les compositeurs. Un type d'atome est défini pour chaque type d'objet. Après transformation des attributs nominaux en objets, les données ne présentent plus une structure exclusivement tabulaire. Une structure de graphe existe en effet entre les objets de types Musique, Compositeur et Interpète. Nous expliquons dans la suite comment représenter conjointement ces deux structures.

La disposition des compositeurs selon leur date de naissance selon x sur la ligne la plus au sud du paysage s'effectue de la manière suivante.

1. Introduction d'un atome indicateur $i_c \in I$ valué sur *nais* avec $i_c.nais = \min_{c \in Comp} c.nais$. Cet indicateur est donc valué avec la valeur minimum pour *nais*. Il est placé à l'extrême sud-ouest du paysage : $i_c.x = 0$, $i_c.y = 500$ et est fixe et invisible : $i_c.fixed = true$, $i_c.visible = false$.
2. Placement de tous les atomes Compositeur au sud du paysage.
   $Compo.x = 0$, $Compo.y = 0$.
3. Blocage des atomes Compositeur en y.
   $Compo.yfixed = true$.
4. Disposition des Compositeurs selon l'attribut *nais* par rapport à l'indicateur.
   $F_P(Compo, I, nais)$.
5. Application de la force limite $F_\Delta(Compo, Compo, 5)$ afin que les compositeurs nés la même année ne se chevauchent pas.

Le même principe est employé pour disposer les interprètes horizontalement par ordre alphabétique sur la ligne la plus au sud du paysage. Les morceaux, quant à eux, sont dans un premier

temps disposés horizontalement selon leur date de composition puis fixés en x. La force propriété les dispose en y selon leur vecteurs de *moods*. La figure 3.3 illustre le résultat obtenu.

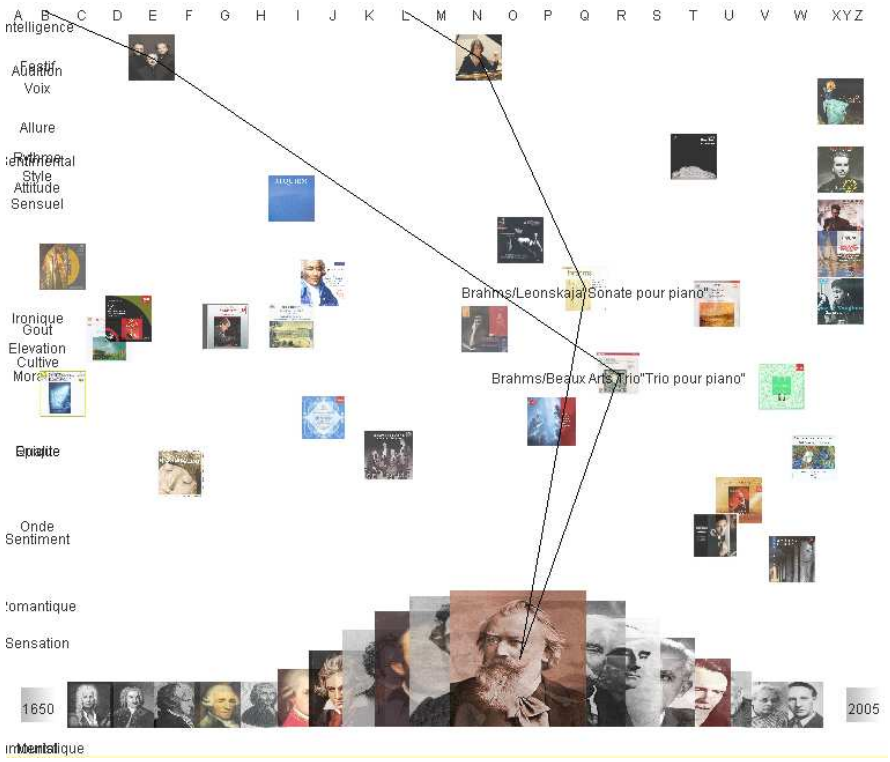


FIG. 3.3 – Intégration des attributs nominaux *compositeur* et *interprète*. Chacune des valeurs de ces attributs a été transformée en un objet et mise en relation avec les morceaux possédant cette valeur. Il en résulte l'apparition d'une nouvelle structure de type graphe aux côtés de la structure tabulaire initiale.

3.2.4 Bilan

Nous avons dans un premier temps réalisé une visualisation à partir des seules données tabulaires initiales. Cette visualisation a permis d'observer les proximités entre morceaux et d'utiliser le dispositif de projection MDS sélective afin de n'observer les proximités qu'en fonction d'un sous-ensemble d'attributs. L'introduction de trois attributs nominaux nous a amenés à transformer leurs valeurs en objets et à introduire une structure de graphe. Cet exemple montre la visualisation conjointe de deux structures différentes, d'une part, et une façon de gérer l'hétérogénéité des natures d'attributs d'autre part. Enfin, l'appariement entre les objets et les entités visuelles permet de spécifier formellement cette visualisation.

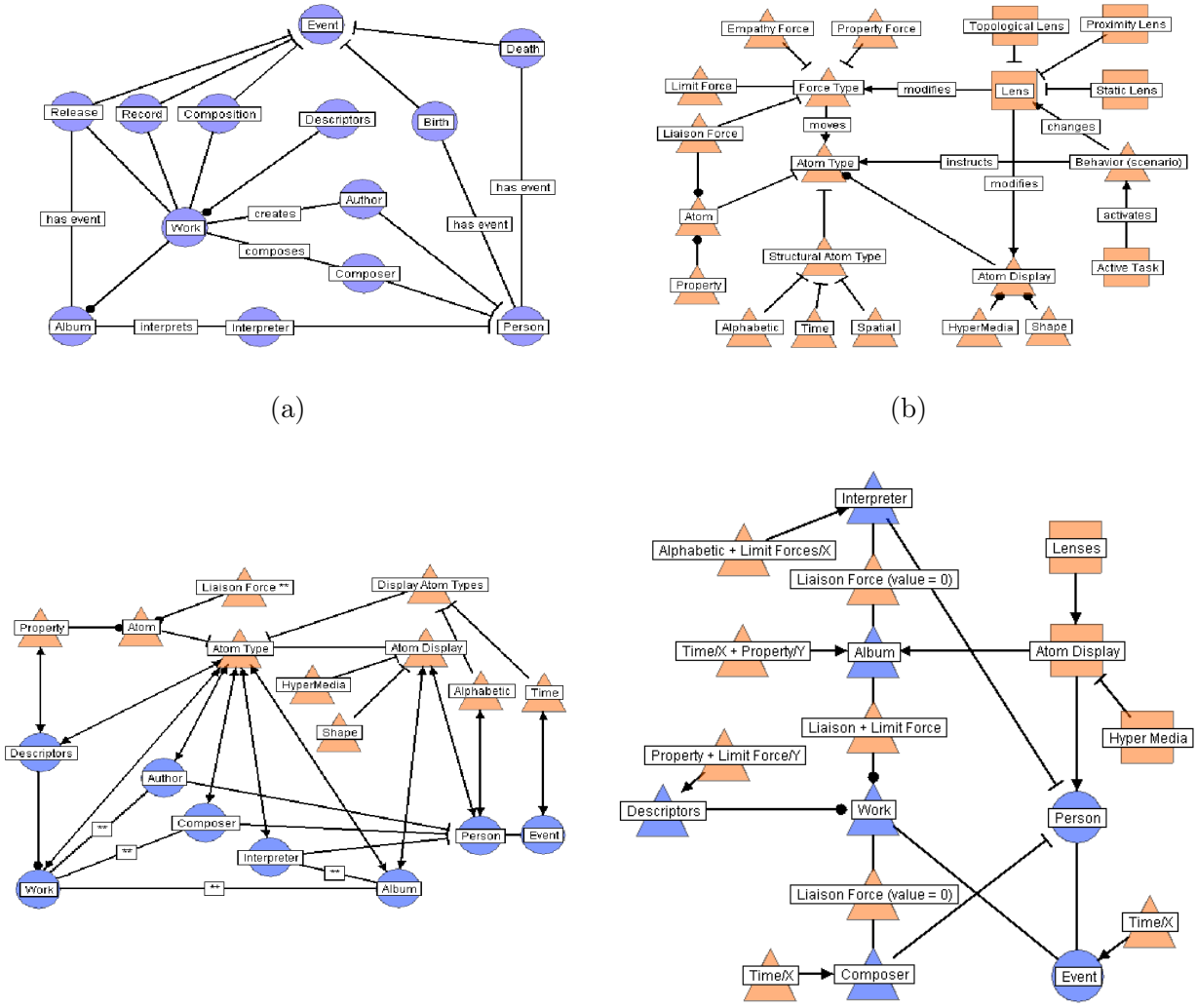


FIG. 3.4 – Spécification de la visualisation de la figure 3.3 par appariement entre les objets du domaine (a), les entités et dispositifs visuels de MOLAGE (b). La figure (c) montre les appariements spécifiés et (d) les scénarios d'organisation mis en œuvre pour disposer les entités sur le plan.

3.3 Modèle formel et visualisation d'une base documentaire scientifique

3.3.1 Le projet de recherche ToxNuc-E

Le projet ToxNuc-E[3] est un programme national de recherche sur la toxicologie nucléaire environnementale qui fédère 300 chercheurs du Cea, du Cnrs, de l'Inra et de l'Inserm avec le soutien du ministère de la Recherche. Le projet est divisé en douze sous-projets pluridisciplinaires. L'objectif global du point de vue de la gestion des connaissances est de tirer parti de cette pluridisciplinarité en mettant en évidence des rapprochements possibles entre acteurs de disciplines différentes travaillant sur des thématiques proches. Ceci implique la mise en place d'un référentiel de connaissances partagé centré sur une ontologie générale de la toxicologie nucléaire environnementale servant de support d'indexation pour l'ensemble des publications produites dans le cadre du projet. L'identification d'un ensemble de documents de référence pour chaque sous-projet permet d'indexer également les sous-projets selon les concepts de l'ontologie. Il est alors possible de quantifier l'adéquation entre un document et un sous-projet d'après leurs indexations respectives. Ainsi à chaque document est associé un vecteur à douze composantes contenant le pourcentage d'adéquation du document pour chaque sous-projet. Les chercheurs sont à leur tour indexés sur ce vecteur dont les valeurs sont calculées à partir des vecteurs de leurs publications. La construction de l'ontologie et la conception de l'indexation ont été réalisées par Reena Shetty [SRQ06]. Notre contribution a consisté à fournir une interface visuelle permettant d'exploiter les résultats de cette indexation afin de :

- naviguer dans la base documentaire ;
- favoriser l'émergence de nouvelles connaissances sur la base elle-même.

Le premier objectif recouvre les attentes habituelles d'un outil d'exploration documentaire, telles que la recherche d'un document à partir d'un auteur, d'un concept, d'une date, la recommandation par similarité, et s'adresse à l'ensemble des acteurs du projet. Le second est plus inattendu et nécessite quelques précisions. Il s'agit d'observer l'évolution du projet et les rapports entre les acteurs à partir de cette base documentaire. L'interface doit notamment permettre de représenter les proximités entre chercheurs et d'identifier des regroupements comme la découverte d'un cluster de chercheurs aux thématiques proches bien que issus de disciplines différentes et affectés à des sous-projets différents, des publications aux thématiques proches de celles d'un sous-projet autre que celui auquel elle est rattachée, etc. Ces informations concernent en priorité la direction de programme. Notre cahier des charges spécifiait donc deux usages différents d'une même source de données. Nous avons été amenés à produire des interfaces différentes selon ces usages.

3.3.2 Représentation explicite d'une structure de type graphe

Concernant le second usage, les données à représenter sont purement tabulaires : on observe des proximités entre chercheurs et entre publications. Les données à représenter pour le premier usage, quant à elles, ont une structure assez similaire à celle de la collection musicale : les documents munis de leurs vecteurs présentent une structure tabulaire, les relations document/concept et document/chercheur une structure de type graphe. Une nouveauté réside dans l'existence d'une troisième structure de graphe autour de la relation concept/concept. Contrairement à celles figurant dans la collection musicale, cette structure de graphe existe entre objets d'un même type et représente une information existant indépendamment des autres objets. En effet, les structures de graphes présentes dans la collection musicale, de même que la structure

[3] `http://www.toxnuc-e.org`

concept/document, peuvent être vues comme la valuation d'un objet (document ou morceau) sur un attribut nominal (concept ou compositeur). Les liens entre objets concept sont représentés explicitement et les objets concept sont disposés en fonction de ces liens par une force liaison. Nous détaillons dans la suite la visualisation correspondant au premier usage.

Fig. 3.5 – Interface textuelle de la base documentaire ToxNuc-E

La consultation de la base documentaire se faisait à l'origine au moyen d'une liste textuelle illustrée par la figure 3.5. L'interface visuelle développée au cours du projet a été conçue pour permettre à la fois de rechercher un document précis à partir d'une de ses propriétés (auteur, date, mots-clef) et de naviguer librement dans la base. Ainsi sur la représentation visuelle illustrée par la figure 3.6 figurent trois types de données : des auteurs, des documents et des concepts ontologiques. Les auteurs sont disposés verticalement par ordre alphabétique, les documents par année de parution et les concepts sous forme de graphe reflétant la relation *est_un*. Cette figure montre le résultat d'une recherche de document suivie d'une navigation par mot-clef. L'utilisateur recherchant l'article *Rivasseau2006* est parti de la liste alphabétique verticale afin d'afficher les auteurs dont le nom débute par la lettre R. L'approche du pointeur sur la lettre R a fait émerger de la colonne de silhouettes semi-transparentes les auteurs concernés, rendant leur silhouette opaque et affichant leur nom. En se déplaçant vers la silhouette sous-titrée *Rivasseau*, les articles de ce dernier ont émergé de la colonne de documents, révélant l'article recherché. En pointant sur l'article, les mots-clef associés émergent de l'ontologie du domaine représentée par le réseau de concepts sur la droite de l'écran. L'utilisateur peut dès lors explorer la base à la recherche de documents partageant ces mots-clef. L'article *Djelabi2006* possédant le mot-clef *leaf* est ainsi apparu, de même que l'ensemble de ses auteurs. À partir d'un document particulier, l'auteur peut donc explorer progressivement la base à la découverte d'articles et d'auteurs liés à sa requête initiale.

Nous détaillons à présent l'organisation des concepts de l'ontologie. Les types d'atomes *Concept*, *Document* et *Auteurs* sont introduits. Les deux premiers sont disposés selon le même

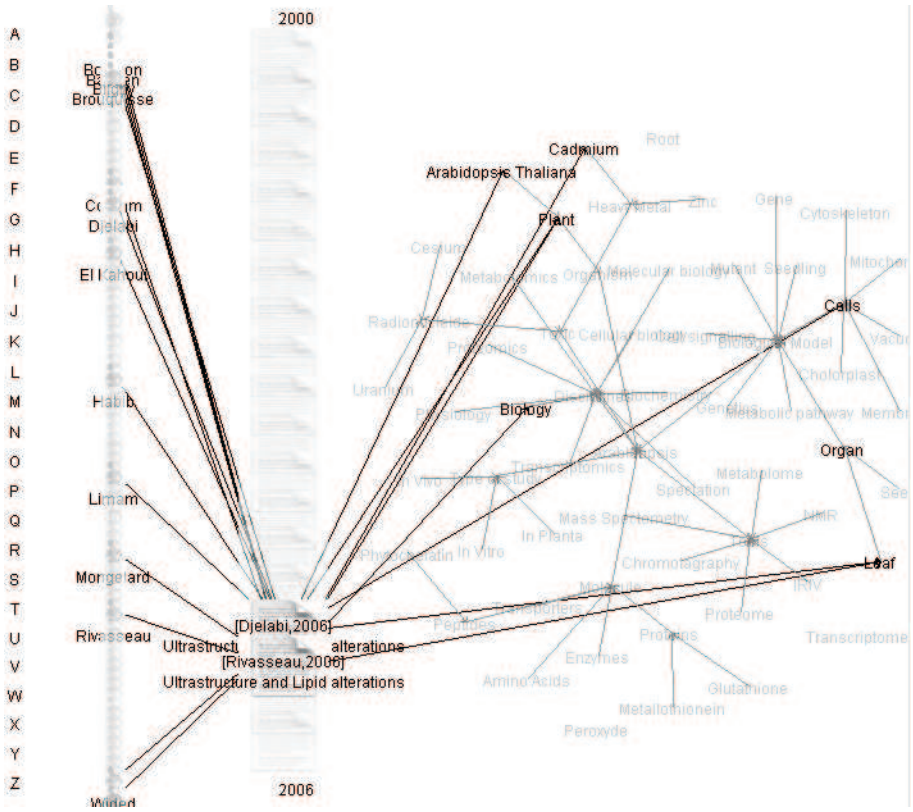


FIG. 3.6 – Interface visuelle de la base documentaire TOXNUC-E

principe que les compositeurs dans l'application musicale. Les atomes *Concept*, quant à eux, sont disposés de façon à reconstituer le graphe de l'ontologie. Une relation binaire $R(Concept, Concept)$ existe entre les atomes *Concept* qui correspond à la relation *est_un*. Le scénario d'organisation consiste à affecter une longueur identique à chaque liaison entre atomes *Concept*, puis à déclencher une force liaison afin qu'ils se disposent conformément à ces liaisons, enfin à déclencher une force limite permettant aux atomes non reliés de ne pas se superposer.

1. tous les atomes *Concept* sont affectés à une même position.
   $Concept.x = 300$, $Concept.y = 250$
2. toutes les liaisons ont une longueur identique.
   $R(Concept, Concept).long = 20$
3. déclenchement de la force liaison.
   $F_L(Concept)$
4. déclenchement de la force limite.
   $F_\Delta(Concept, Concept, 5)$

Les lentilles statique et de proximité affectées aux atomes *Concept* diffèrent par la valeur de la propriété *transparency*, qui vaut 0,5 pour la lentille statique et 0 pour la lentille de proximité. Ainsi l'étiquette d'un atome *Concept*, portant le nom du concept ontologique, devient opaque lorsque la lentille de proximité est activée. Afin de faire ressortir, à partir d'un concept, les documents liés, une lentille topologique est définie : $LensTopo(Concept, Document, 1)$, qui permet, lorsque la lentille de proximité d'un concept est activée, de déclencher la lentille de proximité des documents auxquels le concept est relié.

3.3.3 Bilan

Cette réalisation nous a permis de nous pencher sur la visualisation d'une structure de graphe indépendante, ne résultant pas de la transformation d'un attribut nominal en objets. Les concepts de l'ontologie sont disposés par l'application combinée d'une force liaison et d'une force limite qui a pour effet de recréer visuellement la structure de graphe. Leur mise en œuvre a été spécifiée en fonction des spécifications formelles des dispositifs visuels présents dans *Molage*, comme l'avait été la disposition des atomes *Compositeur* dans l'application musicale. Dans le cas de la collection musicale, la structure de graphe à représenter était la relation binaire *R(Musique,Compositeur)*. Aucune relation n'existait entre atomes *Compositeur* puisque ces atomes résultaient de la transformation d'un attribut nominal en objets. Nous les avions alors disposés horizontalement selon la valeur de leur attribut *dateNaissance*. Dans le cas de la base documentaire, la structure de graphe *R(Document,Concept)*, similaire à *R(Musique,Compositeur)*, était complétée par une relation *R(Concept,Concept)* existant par ailleurs. Les atomes *Concept* ont été disposés de façon à représenter visuellement cette relation.

Le premier cas correspond à un graphe biparti entre les deux types d'atomes *Musique* et *Compositeur*, puisqu'il n'existe pas de relation entre atomes *Musique* ni entre atomes *Compositeur*. Dans le second cas, le graphe de la relation entre les atomes *Document* et *Concept* n'est pas biparti puisqu'il existe une relation entre atomes *Concept*. Les scénarios d'organisation que nous avons formalisés pour chacun de ces cas peuvent être généralisés et considérés comme des « patrons » de visualisation à appliquer selon la nature des relations existant entre deux types d'atomes. Nous avons décrit dans [CRV+06b] une méthode permettant de spécifier ces scénarios d'organisation en fonction des données à représenter. Cette méthode consiste à

(a) décrire les données à représenter,
(b) décrire les entités et dispositifs visuels disponibles dans MOLAGE,
(c) définir un appariement entre les données et ces entités,
(d) définir les scénarios d'organisation.

La figure 3.4 illustre ces différentes étapes. Telle que décrite dans [CRV+06b], cette méthode n'est qu'une aide à la conception de visualisations, la construction effective de la visualisation nécessitant un travail manuel de positionnement des atomes et de déclenchement des forces. La formalisation introduite dans le présent chapitre ouvre la voie à une mise en correspondance automatique entre les entités du domaine à représenter et les entités visuelles sans faire appel à un processus manuel. Ce dernier point permettant, à terme, d'automatiser les scénarios d'organisation.

3.4 Synthèse des verrous identifiés

Outre les problèmes du volume des données et de l'hétérogénéité de leur structure et de la nature de leur attributs, qui constituent notre problématique, ces premières réalisations nous ont permis d'identifier un verrou supplémentaire, celui des données manquantes.

3.4.1 MDS sélective et données manquantes

Le premier verrou considéré concerne la mise en œuvre de la force propriété sur des données manquantes. Nous rappelons que la force propriété dispose les atomes selon une matrice de distances calculée par projection MDS des vecteurs d'attributs, la distance utilisée étant la distance euclidienne. Or il peut exister des objets qui ne sont pas valués sur tous les attributs, la distance entre deux objets étant alors calculée en fonction des attributs sur lesquels ils sont tous deux

	a_1	a_2	a_3
o_1	5		
o_2	5	10	
o_3		10	20

(a)

d_{ij}	o_1	o_2	o_3
o_1	0	0	?
o_2		0	0
o_3			0

(b)

TAB. 3.2 – Matrice objets/attributs comportant des données manquantes (a) et matrice de distances associée (b)

O/A	a_1	a_2	a_3	a_4	a_5	a_6
o_1	5					
o_2	5	10				
o_3		10	20	50	30	40
o_4		9	20	50	30	40

(a)

d_{ij}	o_1	o_2	o_3	o_4
o_1	0	0 (1)	? (0)	? (0)
o_2		0	0 (1)	1 (1)
o_3			0	1 (5)
o_4				0

(b)

TAB. 3.3 – Matrice objets/attributs comportant des données manquantes (a) et matrice de distances associée (b) avec entre parenthèses le nombre d'attributs impliqués dans le calcul de distance

valués. L'exemple suivant montre que cette gestion des données manquantes introduit un biais dans l'interprétation des positions des atomes par l'utilisateur. Le tableau 3.2 (a) montre trois objets dont la valuation sur trois attributs est incomplète et la matrice de distances associée (b). En ne prenant en compte que les attributs communs, o_2 est à une distance nulle des deux autres et la distance entre o_1 et o_3 ne peut être calculée puisqu'ils n'ont aucun attribut valué en commun. On a donc $d(o_1, o_2) = d(o_2, o_3) = 0$ ce qui conduit à représenter les trois objets par trois atomes empilés à la même position. L'utilisateur, qui n'a pas accès aux données initiales et qui interprète les distances entre atomes comme une représentation des dissimilarités entre objets, en conclut que les trois objets que représentent les atomes sont identiques, ce qui n'est pas le cas.

Une variante de ce biais est illustrée en étendant l'exemple comme le montre le tableau 3.3. Les objets o_3 et o_4 partagent les mêmes valeurs sur quatre des cinq attributs sur lesquels ils sont valués et ont des valeurs proches sur le cinquième. Toutefois, comme le montre la matrice de distances (cf. tab 3.3 (b)), on a $d(o_3, o_4) > d(o_2, o_3) = 0$, o_2 et o_3 partageant un seul attribut pour lequel ils ont la même valeur. Ainsi, pour l'utilisateur, tous les objets sont identiques sauf o_4 qui est à l'écart des autres, malgré le fait que o_3 et o_4 sont identiques pour quatre des cinq attributs qu'ils partagent. Notons encore que o_4 se trouve à la même distance de o_2 que de o_3 quand o_3 a quatre attributs identiques à o_4 alors que o_2 n'en a aucun. Notons enfin que o_4 est encore à une distance de 1 de o_1 (du fait de $d(o_1, o_2) = 0$) alors que o_4 et o_1 sont incomparables.

Nous avons indiqué qu'il était possible de spécifier une force propriété de telle sorte que seul un sous-ensemble d'attributs soit pris en compte dans le calcul des distances entre objets. Ainsi, si nous écartons a_1 et a_2 du calcul de distance on a cette fois $d(o_3, o_4) = 0$ et les autres couples deviennent incomparables (cf. tab 3.4). Ce dispositif appelé « MDS sélectif » peut résoudre le problème des données manquantes à deux conditions :

1. que l'utilisateur sache quels attributs sélectionner pour obtenir une matrice de distance cohérente,

	a_1	a_2	a_3	a_4	a_5	a_6
o_1	5					
o_2	5	10				
o_3		10	20	50	30	40
o_4		9	20	50	30	40

(a)

d_{ij}	o_1	o_2	o_3	o_4
o_1	0	?	?	?
o_2		0	?	?
o_3			0	0
o_4				0

(b)

TAB. 3.4 – Matrice objets/attributs comportant des données manquantes (a) et matrice de distances associée (b) calculée en ne prenant en compte que les attributs non grisés

2. que les objets qui ne sont valués sur aucun des attributs sélectionnés (o_1 et o_2 dans l'exemple du tableau 3.4) ne soient pas représentés.

En effet, l'utilisateur n'ayant *a priori* pas accès aux données brutes, celui-ci ne peut en déduire les sous-ensembles d'attributs susceptibles de produire une matrice de distances non biaisée (condition 1). De plus, les objets non valués sur les attributs sélectionnés et donc qu'on ne peut comparer ne doivent pas apparaître (condition 2). Nous verrons dans le chapitre 5 comment nous utilisons les techniques de *Formal concept analysis* pour présenter des sous-ensembles objets/attributs comparables.

3.4.2 Attributs hétérogènes

Un second verrou est la visualisation d'objets valués sur un ensemble d'attributs « mixte », i.e. comportant des attributs numériques, binaires, nominaux et ordinaux. Le tableau 3.5 montre un tel cas. À première vue le seul attribut non numérique est *origine* qui est nominal. Cet attribut pourrait être transformé en affectant une valeur numérique à chacune de ses modalités. On pourrait ainsi définir Europe=0, Japon=50 et USA=100 mais alors, considérant trois voitures identiques selon les autres attributs, la voiture américaine se retrouverait plus proche de la japonaise que de l'européenne. La numérisation des attributs nominaux n'est donc pas une solution satisfaisante et introduit un biais. On peut également s'interroger sur l'attribut *nombre de cylindres*. Bien qu'il soit numérique, n'est-il pas ici vu comme un attribut ordinal ? Le nombre de cylindres n'est pas *mesuré* sur une voiture mais permet plutôt de définir des catégories. Nous proposons de séparer le traitement des attributs selon leur nature et avons élaboré une méthode *overview + detail* que nous exposons dans le chapitre 6.

3.4.3 Hétérogénéité de la structure

Au cours de ce chapitre, nous avons proposé des solutions permettant de représenter conjointement des structures tabulaires et des structures de type graphe. Nous verrons dans les chapitres suivants comme ces solutions seront utilisées dans le cadre de la résolution des autres verrous.

3.4.4 Volume des données

Les exemples de la collection musicale et de la base documentaire, sur lesquels les solutions apportées au verrou précédent ont été illustrées, n'étaient pas caractérisés par un volume de données particulièrement important. Or, ces solutions peuvent s'avérer inopérantes lorsque le nombre d'objets ou de liaisons affichés devient trop important. Nous verrons dans les chapitres

	consommation (miles/gallon)	nombre de cylindres	cylindrée (pouces^3)	puissance (ch)	poids (pounds)	accélération (pieds/s^2)	millésime	origine
Chevrolet Chevelle Malibu	18	8	307	130	3504	12.0	1970	USA
Citroën DS-21 Pallas		4	133	115	3090	17.5	1970	Europe
Ford Mustang Boss 302		8	302	140	3353	8	1970	USA
Toyota Corolla	29	4	97	75	2171	16	1975	Japon
Renault 5 GTL	36	4	79	58	1825	18	1977	Europe

TAB. 3.5 – Exemple de données « mixtes » comportannt des attributs hétérogènes, extrait de [AN07].

suivants comment l'adoption d'une stratégie *overview + detail* permet de réduire le volume des données affichées simultanément.

3.5 Conclusion

Dans ce chapitre nous avons présenté un modèle formel du paradigme de visualisation FDP implémenté par MOLAGE. Nous avons spécifié les visualisations mises en œuvre dans deux projets de recherche en utilisant ce modèle formel. Nous avons vu que les spécifications dépendaient du type de structure des données et avons réutilisé les spécifications réalisées dans le cadre du premier projet pour la réalisation du second. Les solutions de visualisation présentées sont toutefois rendues inefficaces en présence d'un volume de données trop important. Enfin, un nouveau verrou est apparu : le problème des données manquantes. Dans le chapitre suivant, nous introduisons les techniques de Formal Concept Analysis, qui seront ensuite utilisées pour résoudre le problème des données manquantes puis celui de l'hétérogénéité des attributs.

4

Analyse de concepts formels

Sommaire

Nous présentons dans ce chapitre les principes de base de l'Analyse de concepts formels, généralement désignée par le terme anglophone *Formal Concept Analysis* (FCA). FCA est un ensemble de techniques d'analyse de données visant à identifier des regroupements objets/attributs, appelés concepts formels, et à ordonner ces regroupements sous la forme d'un treillis. Nous commençons par une description intuitive de FCA puis donnons les définitions formelles associées. Nous détaillons plus particulièrement certains aspects ayant trait aux types des attributs avant de présenter les applications de FCA en recherche d'information.

4.1 Approche intuitive

Les données manipulées par FCA sont de type tabulaire (cf. section 2.2.1) et prennent la forme d'un tableau objets/attributs. Dans un premier temps nous nous restreindrons aux attributs binaires. La case (i, j) est marquée d'une croix si l'objet i possède l'attribut j. Ainsi dans l'exemple décrit par la figure 4.1, l'objet *lion* possède les attributs *chasse* et *mammifère*. Ce tableau est appelé *contexte formel*. L'objectif est d'identifier les regroupements objets/attributs tels que tous les objets du regroupement possèdent tous les attributs du regroupement et vice-versa.

4.1.1 Concepts, extensions, intensions et treillis de concepts

Les regroupements sont appelés *concepts formels*. On appelle l'ensemble des objets O_i (resp. des attributs A_i) d'un concept $c = (O_i, A_i)$ l'extension (resp. l'intension) de ce concept. Ainsi huit

Animaux	chasse	vole	oiseau	mammifère
lion	×			×
moineau		×	×	
aigle	×	×	×	
lapin				×
autruche			×	

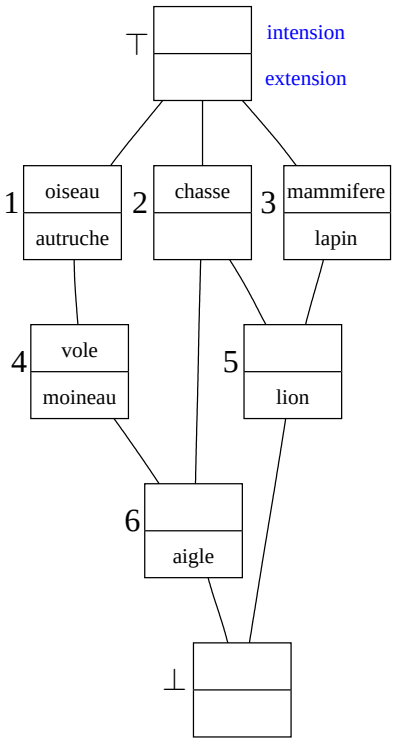


FIG. 4.1 – Contexte formel *Animaux* et treillis de concepts associé

concepts ont été identifiés à partir du contexte *Animaux* (cf. fig. 4.1), six concepts numérotés et deux concepts particuliers notés $\top$ et $\bot$ que nous détaillons plus loin. Le tableau 4.1 détaille les extensions et les intensions de ces concepts. Ceux-ci sont ordonnés par les inclusions réciproques de leurs extensions et intensions. Ainsi le concept 4 est inférieur au concept 1 puisque l'extension de 4 {*moineau, aigle*} est incluse dans l'extension de 1 {*moineau, aigle, autruche*}. Inversement, l'intension de 1 {*oiseau*} est incluse dans l'intension de 4 {*oiseau, vole*}. Un concept A est donc supérieur à un concept B si l'extension de B est incluse dans celle de A et si l'intension de A est incluse dans celle de B. Nous verrons dans la section 4.2 que les concepts sont extraits de telle façon que si l'une de ces conditions est vérfiée, l'autre l'est toujours également. Certains concepts sont incomparables : ainsi aucune relation d'inclusion n'est identifiable entre les intensions et extensions des concepts 1 et 2. L'ordre partiel ainsi introduit est complété par un concept maximal $\top$ (top) et un concept minimal $\bot$ (bottom) afin de former un treillis appelé *treillis de Galois* ou *treillis de concepts*. Le concept $\top$ est défini comme contenant tous les objets en extension et le concept $\bot$ comme contenant tous les attributs en intension. L'intension de $\top$ contient donc les attributs possédés par tous les objets et l'extension de $\bot$ les objets possédant tous les attributs. Dans notre exemple l'intension de $\top$ et l'extension de $\bot$ sont toutes deux vides.

4.1.2 Représentation graphique, extensions et intensions réduites

Le treillis de concepts associé à un contexte formel est généralement représenté graphiquement par un diagramme de Hasse. Il s'agit d'un graphe dans lequel les sommets représentent les concepts et les arêtes l'ordre partiel entre concepts réduit transitivement. La figure 4.1 représente le treillis de concepts associé au contexte *Animaux* sous la forme d'un diagramme de Hasse. Le concept $\top$ est toujours situé à l'extrémité supérieure du diagramme et $\bot$ à l'extrémité inférieure.

concept	extension	intension
$\top$	lion, moineau, aigle, lapin, autruche	$\emptyset$
1	moineau, aigle, **autruche**	**oiseau**
2	lion, aigle	**chasse**
3	lion, **lapin**	**mammifère**
4	**moineau**, aigle	oiseau, **vole**
5	**lion**	chasse, mammifère
6	**aigle**	oiseau, vole, chasse
$\bot$	$\emptyset$	oiseau, vole, chasse, mammifère

TAB. 4.1 – Extensions et intensions des concepts de *Animaux*, en gras les éléments des extensions et intensions réduites

Les concepts sont organisés de façon à ce que le nombre d'objets en extension diminue en suivant un chemin de $\top$ vers $\bot$ et réciproquement que le nombre d'attributs en intension augmente. Un concept prend la forme d'une boîte dont la partie haute représente l'intension et la partie basse l'extension. On notera que leur contenu ne correspond pas aux intensions et extensions relevées dans le tableau 4.1. En effet, afin de ne pas surcharger le diagramme de Hasse, seules les intensions et extensions *réduites* ont été représentées.

L'intension réduite d'un concept contient seulement les attributs qui n'apparaissent pas dans les intensions des concepts supérieurs. Réciproquement, l'extension réduite d'un concept contient seulement les objets qui n'apparaissent pas dans les extensions des concepts inférieurs. Ainsi, l'intension réduite du concept 4 est $\{vole\}$ puisque, pami les attributs de l'intension complète de 4 $\{oiseau,\ vole\}$, cet attribut est le seul qui n'apparaît pas dans les intensions de 1 ni de $\top$ (cf. tableau 4.1). Son extension réduite est $\{moineau\}$ car c'est le seul objet de son extension qui n'apparaît pas en extension de 6 ni de $\bot$. L'algorithme 4.1 décrit ce calcul. Le choix de ne représenter que les extensions et intensions réduites sur le diagramme de Hasse permet d'identifier facilement les relations de dépendance entre attributs : le fait que *vole* apparaît dans un sous-concept de celui où figure *oiseau* signifie que seuls les oiseaux volent mais que tous les oiseaux ne volent pas, sinon ces deux attributs apparaîtraient dans le même concept. Ces relations sont appelées *implications* et seront abordées dans la section 4.2.

ALGORITHME 4.1 : calcul des extension et intension réduites d'un concept $c = (O_i, A_i)$

Entrées : $c = (O_i, A_i)$, Sup_c (concepts supérieurs), Inf_c (concepts inférieurs)
Sorties : Les extension et intension réduites O_i^s et A_i^s de C

$A_i^s \longleftarrow A_i$
$O_i^s \longleftarrow O_i$
pour chaque $c_j = (O_j, A_j) \in Sup_c$ **faire**
  $A_i^s \longleftarrow A_i^s \setminus A_j$
fin
pour chaque $c_j = (O_j, A_j) \in Inf_c$ **faire**
  $O_i^s \longleftarrow O_i^s \setminus O_j$
fin

On notera que certains concepts ont une intension ou une extension réduite vide. L'intension complète du concept 5 est $\{chasse,\ mammifère\}$ et son extension complète (dont la réduite est

d'ailleurs identique) est {*lion*}. Les deux attributs en intension apparaissent dans des concepts supérieurs car il existe d'autres objets possédant soit l'un soit l'autre mais pas les deux. Ainsi le concept 5 ne fait qu'associer deux attributs présents dans des concepts supérieurs. Sémantiquement, un concept dont l'intension réduite est vide correspond à un regroupement d'objets qu'aucun attribut ne caractérise de manière exclusive. À l'inverse, considérant le concept 4, son extension {*moineau, aigle*} est caractérisée de manière exclusive par l'intension réduite {*vole*} : ce sont les seuls objets à posséder cet attribut. Un concept avec une intension réduite vide indique donc un ensemble d'objets pouvant être décrit par une conjonction d'attributs mais ne possédant pas d'attribut (donc de nom) leur étant propre. Ceci est particulièrement intéressant car pouvant correspondre à un regroupement inattendu selon le contexte d'application. Le concept 2 présente quant à lui une extension réduite vide. Cela est dû au fait que les objets de son extension complète {*lion, aigle*} possèdent d'autres attributs outre ceux présents dans l'intension {*chasse*} et apparaissent donc dans les intensions réduites de concepts inférieurs. En d'autres termes, il n'existe aucun objet ne possédant que *chasse*.

4.2 Approche formelle, définitions et notations

Cette section reprend de manière formelle les notions abordées dans la section précédente et introduit de nouveaux aspects de FCA.

Un **contexte formel** est un triplet $\mathbb{K} = (O, A, I)$ où O est un ensemble d'objets, A un ensemble d'attributs et $I \subseteq O \times A$ une relation binaire entre objets et attributs. Pour un ensemble d'objets $O_i \subseteq O$ et un ensemble d'attributs $A_i \subseteq A$, on définit l'ensemble des attributs communs à tous les objets de O_i par :

$$f : 2^O \to 2^A \qquad f(X) = \{a \in A | \forall o \in X, (o, a) \in I\}$$

et l'ensemble des objets possédant tous les attributs de A_i par :

$$g : 2^A \to 2^O \qquad g(Y) = \{o \in O | \forall a \in Y, (o, a) \in I\}$$

Le couple (f, g) définit une **connexion de Galois** entre les ensembles ordonnés $(2^O, \subseteq)$ et $(2^A, \subseteq)$. Un **concept formel** du contexte $\mathbb{K}$ est un couple (O_i, A_i) tel que :

$$O_i \in 2^O,\ A_i \in 2^A,\ A_i = f(O_i),\ O_i = g(A_i)$$

O_i est appelé l'**extension** et A_i l'**intension** du concept (O_i, A_i). Soit $\mathcal{C}_\mathbb{K}$ l'ensemble des concepts de $\mathbb{K}$ et soit $\leq_\mathbb{K}$ l'ordre partiel défini comme suit, considérant deux concepts (O_i, A_i) et (O_j, A_j).

$$(O_j, A_j) \leq_\mathbb{K} (O_i, A_i) \Leftrightarrow O_j \subseteq O_i \Leftrightarrow A_j \supseteq A_i$$

Le couple $\mathcal{L}_\mathbb{K} = (\mathcal{C}_\mathbb{K}, \leq_\mathbb{K})$ est le **treillis de concepts** ou **treillis de concepts** associé à $\mathbb{K}$. Deux concepts $c_i, c_j \in \mathcal{C}_\mathbb{K}$ sont **voisins directs** si

$$c_j \leq_\mathbb{K} c_i \quad c_j \neq c_i \qquad \nexists c_k | c_j \leq_\mathbb{K} c_k \leq_\mathbb{K} c_i \quad c_k \neq c_j \neq c_i$$

La relation de voisinage direct est notée $c_j \prec_\mathbb{K} c_i$ et est orientée ; on dit que c_j est un **fils direct** de c_i et c_i est un **père direct** de c_j. Une **implication** entre deux ensembles d'attributs A_i et A_j, notée $A_j \longrightarrow A_i$, existe lorsque tous les objets possédant A_j possèdent également A_i, i.e. $g(A_j) \subseteq g(A_i)$. Les implications suivantes ont été extraites du contexte illustré par la figure 4.1.

$$\begin{aligned} \text{vole} &\longrightarrow \text{oiseau} \\ \text{chasse, vole} &\longrightarrow \text{oiseau} \\ \text{chasse, oiseau} &\longrightarrow \text{vole} \end{aligned}$$

Le tableau 4.2 résume les notations introduites.

	notation utilisée	notation de [GW99]
contexte formel	$\mathbb{K} = (O, A, I)$	$\mathbb{K} = (G, M, I)$
ensemble des concepts	$\mathcal{C}_{\mathbb{K}}$	$\mathfrak{B}(G, M, I)$
treillis de concepts	$\mathcal{L}_{\mathbb{K}} = (\mathcal{C}_{\mathbb{K}}, \leq_{\mathbb{K}})$	$\underline{\mathfrak{B}}(G, M, I)$
voisinage direct	$c_j \prec_{\mathbb{K}} c_i$	$c_j \prec_{\mathbb{K}} c_i$
extension du concept $c = (O_i, A_i)$	$\text{ext}(c) = O_i$	$\text{ext}(c)$
intension du concept $c = (O_i, A_i)$	$\text{int}(c) = A_i$	$\text{int}(c)$
extension réduite de $c = (O_i, A_i)$	$\text{ext}_s(c) = O_i^s$	
intension réduite de $c = (O_i, A_i)$	$\text{int}_s(c) = A_i^s$	

TAB. 4.2 – Synthèse des notations utilisées dans le présent manuscrit et de celles introduites par [GW99]

4.3 Contextes multivalués et échelles conceptuelles

Dans les sections précédentes, nous nous sommes restreints aux attributs binaires. La présente section traite de la prise en charge d'attributs non binaires.

Un **contexte multivalué** est un tuple $(O, A, (V_a)_{a\in A}, I)$ où V_a est un ensemble de **valeurs** pour chaque $a \in A$ et $I \subseteq O \times \bigcup_{a\in A}(\{a\} \times V_a)$ est une relation telle que $(o, a, v_1) \in I$ et $(o, a, v_2) \in I$ implique $v_1 = v_2$. Une **échelle conceptuelle** pour un attribut $a \in A$ est un contexte monovalué (i.e. binaire) $\mathbb{S}_a = (O_a, A_a, I_a)$ avec $V_a \subseteq O_a$. Le contexte $\mathbb{R}_a = (O, A_a, J_a)$ avec

$$(o, b) \in J_a \iff \exists v \in V_a | (o, a, v) \in I \wedge (v, b) \in I_a$$

est une **échelle réalisée** pour l'attribut a. Le **contexte dérivé** $\mathbb{D}$ de $(O, A, (V_a)_{a\in A}, I)$ selon les échelles conceptuelles $(\mathbb{S}_a)_{a\in A}$ est le contexte monovalué (O, B, J) tel que $B = \bigcup_{a\in A}(\{a\} \times A_a)$ avec

$$oJ(a, b) \iff \exists v \in V_a | (o, a, v) \in I \wedge (v, b) \in I_a$$

Afin de fixer les esprits, nous détaillons le processus de transformation d'un contexte multivalué en un contexte dérivé monovalué sur un exemple simple. Le tableau 4.3 décrit le contexte multivalué $\mathbb{K}$ muni des deux attributs non binaires *sexe* et *âge*. Les valeurs de *sexe* sont $V_{sexe} = \{M, F, ?\}$ et celles de *âge* sont $V_{âge} = \{17, 21, 50, 66, 88, 90, ?\}$. Deux échelles conceptuelles $\mathbb{S}_{sexe}$ et $\mathbb{S}_{âge}$ vont donc être construites. L'échelle $\mathbb{S}_{sexe}$ est un contexte monovalué dans lequel les objets sont les valeurs V_{sexe} et les attributs sont des attributs binaires nouvellement introduits. L'échelle a pour but de mettre en correspondance les valeurs de V_{sexe} avec ces nouveaux attributs binaires. Le choix des nouveaux attributs binaires dépend de l'information que l'on souhaite voir représentée dans le contexte dérivé final et, en ce sens, le choix d'une échelle n'est pas automatique même s'il existe des patrons d'échelles en fonction de la nature des attributs multivalués [GW99]. Concernant l'échelle V_{sexe}, l'information que l'on souhaite voir apparaître dans le contexte dérivé final est l'appartenance des individus à l'un ou l'autre des deux sexes. Lorsque cette information n'est pas fournie par le contexte multivalué initial, i.e. lorsqu'un individu a la valeur ? pour l'attribut *sexe*, on souhaite que cette absence d'information se traduise par l'absence d'appartenance aux deux sexes, et non par l'appartenance à un troisième sexe « inconnu ». Autrement dit, on souhaite que dans le contexte dérivé final, les individus masculins aient une croix pour un nouvel attribut binaire M, les individus féminins une croix pour un nouvel attribut binaire F et les individus de sexe inconnu n'aient de croix ni pour M ni pour F. On ne souhaite pas avoir un troisième attribut binaire ? qui soit marqué d'une croix pour les individus de sexe inconnu. Le

$\mathbb{K}$	sexe	âge
Adam	M	21
Betty	F	50
Chris	?	66
Dora	F	88
Eva	F	17
Fred	M	?
George	M	90
Harry	M	50

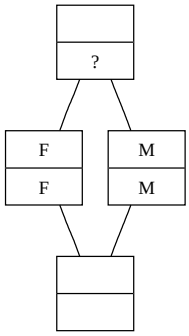
$\mathbb{S}_{sexe}$	M	F
M	×	
F		×
?		

$\mathbb{S}_{\hat{a}ge}$	≥ 18	≥ 40	≥ 65	> 65	> 80
17	×	×	×		
21		×	×		
50			×		
66				×	
88				×	×
90				×	×
?					

TAB. 4.3 – Le contexte multi-valué $\mathbb{K}$ est composé d'un attribut nominal (sexe) et d'un attribut numérique (âge). Chacune des échelles conceptuelles $\mathbb{S}_{sexe}$ et $\mathbb{S}_{\hat{a}ge}$ a pour objets les valeurs V_{sexe} et $V_{\hat{a}ge}$ et pour attributs de nouveaux attributs binaires qui deviendront les attributs du contexte dérivé.

$\mathbb{S}_{sexe}$	M	F
M	×	
F		×
?		

(a)



(b)

TAB. 4.4 – Échelle conceptuelle $\mathbb{S}_{sexe}$ (a) et treillis de concepts associé (b)

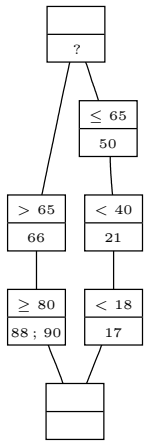
tableau 4.3 (a) montre l'échelle $\mathbb{S}_{sexe}$ dans laquelle les valeurs M et F sont associées aux deux nouveaux attributs portant le même nom tandis que la valeur ? n'est associée à aucun de ces attributs. Le tableau 4.3 (b) montre le treillis de concepts généré à partir de $\mathbb{S}_{sexe}$. La valeur ? apparaît dans l'extension réduite du concept $\top$ puisque ? est le seul objet à ne posséder aucun attribut.

Le choix des attributs binaires pour l'échelle $\mathbb{S}_{\hat{a}ge}$ s'est porté sur une dichotomie autour de la valeur 65 : deux attributs ≤ 65 et > 65 permettent de répartir les valeurs $V_{\hat{a}ge}$ entre ces deux pôles. De plus les attributs < 18 et < 40 d'une part, et l'attribut ≥ 80 d'autre part, séparent les individus à l'intérieur de ces pôles. La valeur ? n'est associée à aucun attribut binaire. Le tableau 4.5 montre $\mathbb{S}_{\hat{a}ge}$ et le treillis associé où apparaît clairement la séparation des valeurs en deux pôles autour de la valeur 65.

Une fois les deux échelles conceptuelles $\mathbb{S}_{sexe}$ et $\mathbb{S}_{\hat{a}ge}$ spécifiées, l'étape suivante consiste à générer les échelles réalisées $\mathbb{R}_{sexe}$ et $\mathbb{R}_{\hat{a}ge}$. Une échelle réalisée est un contexte monovalué dans lequel les attributs sont les attributs de l'échelle conceptuelle et les objets sont les objets du contexte multivalué initial $\mathbb{K}$. Une échelle réalisée se déduit automatiquement de $\mathbb{K}$ et de l'échelle conceptuelle correspondante. Considérant $\mathbb{R}_{\hat{a}ge}$, il s'agit de reporter pour chaque objet de $\mathbb{K}$ la

$\mathbb{S}_{\hat{a}ge}$	18 $\vee$	40 $\vee$	65 $\vee\vert$	65 $\wedge$	80 $\wedge\vert$
17	×	×	×		
21		×	×		
50			×		
66				×	
88				×	×
90				×	×
?					

(a)



(b)

Tab. 4.5 – Échelle conceptuelle $\mathbb{S}_{\hat{a}ge}$ (a) et treillis de concepts associé (b)

ligne de $\mathbb{S}_{\hat{a}ge}$ correspondant à sa valeur multivaluée pour l'attribut *âge*. Ainsi la ligne de l'objet *Eva* dans $\mathbb{R}_{\hat{a}ge}$ correspond à la ligne de la valeur 17 dans $\mathbb{S}_{\hat{a}ge}$. Le tableau 4.6 montre pour chaque échelle conceptuelle l'échelle réalisée associée.

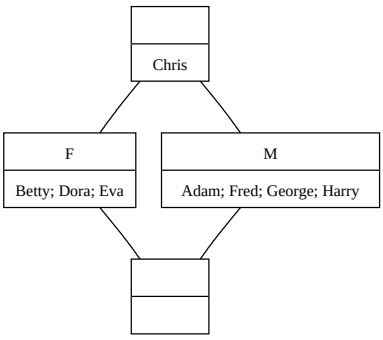
Le contexte dérivé est construit en « concaténant » les échelles réalisées de sorte que les objets du contexte initial $\mathbb{K}$ soient valués sur l'ensemble des nouveaux attributs binaires introduits par les échelles conceptuelles. Le tableau 4.7 (a) montre le contexte binaire $\mathbb{D}$ dérivé de $\mathbb{K}$ selon les échelles conceptuelles $\mathbb{S}_{sexe}$ et $\mathbb{S}_{\hat{a}ge}$. Le treillis de concepts associé (cf. tab 4.7 (b)) est bien moins lisible que les treillis des échelles réalisées pris séparément. La séparation homme/femme tout comme les deux catégories d'âge ≤ 65 et > 65 n'apparaissent pas clairement car les attributs provenant des deux échelles sont entremêlés. De plus, le nombre de concepts extraits du contexte dérivé est supérieur à la somme du nombre de concepts de chaque échelle réalisée : $|\mathcal{C}_{\mathbb{D}}| > |\mathcal{C}_{\mathbb{R}_{sexe}}| + |\mathcal{C}_{\mathbb{R}_{\hat{a}ge}}|$. En effet, certains concepts d'une échelle réalisée éclatent car les objets en extension ne partagent pas les mêmes attributs issus de l'autre échelle. Ainsi, le concept $(\{Betty, Dora, Eva\}, \{F\})$ de $\mathbb{R}_{sexe}$ ne rassemble plus les trois individus féminins puisque ceux-ci ne partagent pas les mêmes attributs issus de $\mathbb{S}_{\hat{a}ge}$. Cet accroissement du nombre de concepts vient de l'accroissement du nombre d'attributs induit par la discrétisation des attributs multivalués : le nombre d'attributs est passé de 2 pour $\mathbb{K}$ à 7 pour $\mathbb{D}$. Afin de pallier ce problème de lisibilité des treillis associés aux contextes dérivés, une variante du diagramme de Hasse, appelée *nested-line diagrams* a été introduite et sera présentée en détails dans la section 6.1. Auparavant nous exposons les principaux patrons d'échelles conceptuelles.

4.4 Variantes

L'inconvénient majeur de FCA est la taille du treillis qui augmente considérablement en fonction du nombre d'objets et d'attributs. En effet, pour un contexte à $|O|$ objets et $|A|$ attributs, le nombre de concepts du treillis associé est en $O(2^{\min(|O|,|A|)})$. Les performances et complexités des différents algorithmes de génération de treillis ont été étudiés dans [KO02]. Plusieurs solutions ont été avancées afin de restreindre le nombre de concepts.

$\mathbb{R}_{sexe}$	M	F
Adam	×	
Betty		×
Chris		
Dora		×
Eva		×
Fred	×	
George	×	
Harry	×	

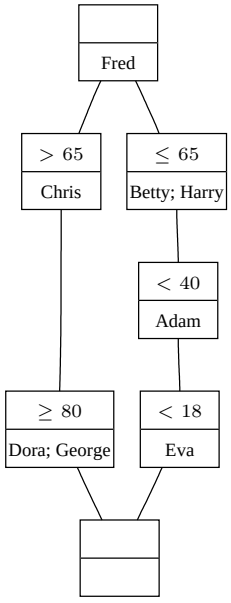
(a)



(b)

$\mathbb{R}_{\hat{a}ge}$	18 $\vee$	40 $\vee$	65 $\vee\vert$	65 $\wedge$	80 $\wedge\vert$
Adam		×	×		
Betty			×		
Chris				×	
Dora				×	×
Eva	×	×	×		
Fred					
George				×	×
Harry			×		

(c)

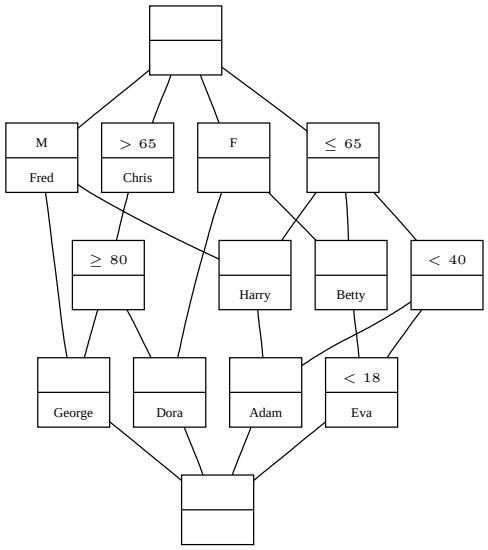


(d)

TAB. 4.6 – Échelles conceptuelles et échelles réalisées correspondantes

$\mathbb{D}$	sexe : M	sexe : F	âge : <18	âge : <40	âge : ≤ 65	âge : >65	âge : ≥ 80
Adam	×			×	×		
Betty		×			×		
Chris						×	
Dora		×				×	×
Eva		×	×	×	×		
Fred	×						
George	×					×	×
Harry	×				×		

(a)



(b)

Tab. 4.7 – Contexte dérivé final

4.4.1 Treillis iceberg

Le principe des treillis iceberg [STB+02] est de ne conserver que la partie « émergée » du treillis en supprimant les concepts contenant peu d'objets en extension. Seuls les concepts dont le nombre d'objets en extension dépasse un seuil donné, fonction du nombre total d'objets, sont conservés. Les treillis iceberg peuvent être vus comme une représentation condensée des motifs fréquents, ou *itemsets*, présents dans le contexte formel et de ce fait ont trouvé de nombreuses applications en fouille de données. Considérant un contexte $\mathbb{K} = (O, A, I)$, le support d'un ensemble d'attributs $A_i \subseteq A$ (un motif) est défini par $\text{supp}(A_i) = |g(A_i)|/|O|$. A_i est un ensemble d'attributs fréquent si $\text{supp}(A_i) \geq \text{minsupp}$ avec $\text{minsupp} \in [0, 1]$. Un concept (O_i, A_i) est un concept fréquent si son intension A_i est fréquente. Le treillis iceberg du contexte $\mathbb{K}$ contient l'ensemble des concepts fréquents de $\mathbb{K}$.

4.4.2 Sous-hiérarchie de Galois

Intuitivement, une sous-hérarchie de Galois [GM93, HDL00] correspond au treillis privé des concepts dont l'extension réduite ou l'intension réduite est vide. La sous-hiérarchie de Galois a l'avantage d'être plus légère puisque le nombre de concepts y est borné par $|O|+|A|$. Considérant un contexte $\mathbb{K} = (O, A, I)$, où O désigne l'ensemble des objtes, A l'ensemble des attributs et I une relation binaire entre O et A, les concepts $\mathcal{C}^O = \{\gamma_o = (g(f(o)), f(o)) \mid o \in O\}$ sont appelés **concepts objets**, et les concepts $\mathcal{C}^A = \{\mu_a = (g(a), f(g(a))) \mid a \in A\}$ **concepts attributs**. Le concept objet γ_o, associé à un objet o, est le plus petit concept avec o en extension. Réciproquement, le concept attribut μ_a, associé à un attribut a, est le plus grand concept avec a en intension. La *sous-hiérarchie de Galois* associée à $\mathbb{K}$ est constituée de l'ensemble des concepts $\mathcal{C}^O \cup \mathcal{C}^A$ muni de l'ordre suivant : soient (O_j, A_j) et (O_i, A_i) deux concepts de $\mathcal{C}^O \cup \mathcal{C}^A$,

$$(O_j, A_j) < (O_i, A_i) \Longleftrightarrow O_j \subset O_i \Longleftrightarrow A_j \supset A_i$$

Les sous-hiérarchies de Galois ont fait l'objet de plusieurs applications, notamment en génie logiciel pour la restructuration de hiérarchies de classes en modélisation objet. Plusieurs algorithmes de génération de sous-hiérarchies de Galois ont été élaborés et implémentés dans GALICIA [VGRH03].

4.5 Outils

Dans cette section, nous passons en revue les outils permettant d'éditer des contextes formels, de construire les treillis de concepts associés et de générer les implications. La question de la visualisation du treillis sera traitée à part. Nous nous restreignons ici aux outils libres les plus couramment utilisés en recherche académique, une liste plus complète incluant les logiciels commerciaux peut être consultée sur la *FCA homepage*[4].

CONIMP
- format entrée contexte : `cxt`
- format sortie contexte : `cxt`
- format entrée treillis : non
- format sortie treillis : `bgr`
- visualisation : non
- adresse : `www.mathematik.tu-darmstadt.de/~burmeister/`

GALICIA
- format entrée contexte : `slf`, `bin.xml`, `ibm`
- format sortie contexte : `slf`, `bin.xml`
- format entrée treillis : `lat.xml`
- format sortie treillis : `lat.xml`
- visualisation : oui
- adresse : `www.iro.umontreal.ca/~galicia/`

CONEXP
- format entrée contexte : `cex`, `cxt`, `csv`, `oal`
- format sortie contexte : `cex`, `cxt`
- format entrée treillis : `cex`
- format sortie treillis : `cex`
- visualisation : oui
- adresse : `conexp.sourceforge.net/`

TOSCANAJ ET SIENA
- format entrée contexte : `csx`, `cxt`, `csc` (format ANACONDA), `xml` (format CERNATO)
- format sortie contexte : `csx`
- format entrée treillis : `csx`
- format sortie treillis : `csx`
- visualisation : oui, y compris *nested-line diagrams*
- adresse : `toscanaj.sourceforge.net/`

[4] `www.upriss.org.uk/fca/`

Nous avons utilisé ces différents outils au cours de la mise au point de nos solutions. En particulier, ils nous ont permis de construire les treillis que nous manipulons et visualisons par la suite.

4.6 Applications en recherche d'information

Les techniques de FCA sont couramment utilisées en recherche d'information [Pri06]. Depuis les premiers travaux de [GPG89] sur un système de recherche d'information documentaire basé sur un treillis document/terme, de nombreuses recherches ont été entreprises et ont abouti à des résultats significatifs. Carpineto et Romano [CR04] ont relevé que, outre leur utilité en classification, les treillis de concepts pouvaient servir de support à l'utilisateur pour la formulation de requêtes et la navigation parmi les résultats. Leur moteur de recherche web CREDO[5] construit un contexte formel dans lequel les objets sont les pages retournées et les attributs les termes indexant ces pages. L'interface présente sous forme d'arbre les concepts de plus haut niveau du treillis associé. L'utilisateur a ainsi une vision globale de l'espace de recherche et peut naviguer parmi les pages retournées selon les termes qui les indexent (cf. fig. 4.2). FOOCA [Koe06] reprend ce principe et permet à l'utilisateur de raffiner ses requêtes en lui proposant d'agir sur le contexte formel associé à ses résultats (cf. fig. 4.3). À partir d'une requête transformée en concept formel, BR-EXPLORER [MDNST05, MDNST06] identifie un concept-pivot dans le treillis construit à partir des documents-objets de la base. L'ensemble des résultats retournés est construit et ordonné pas à pas en remontant les concepts pères du pivot. L'application des techniques de FCA aux réultats de moteurs de recherche web a connu de nouveaux développement récemment avec SEARCHSLEUTH [DE07, DDE08] qui est centré sur le concept correspondant aux termes recherchés, permettant des opérations de raffinement plus précises dont une navigation par concepts voisins. Bien que des recherches aient été menées sur l'intuitivité du treillis en tant que représentation visuelle [EDB04], dans les outils précédemment cités le treillis sous-jacent n'est pas explicitement présenté à l'utilisateur. Ainsi, SEARCHSLEUTH reprend le principe de navigation de IMAGESLEUTH[6] [DVE06], outil de navigation dans une collection d'images indexées par mots-clef. L'interface affiche les images correspondant à l'extension du concept sélectionné et propose une liste de mots-clef à retirer de l'intension (afin de remonter dans le treillis vers des concepts plus généraux contenant plus d'images et moins de mots-clef) et une liste de mots-clef à ajouter (les nouveaux mots-clefs apparaissant dans les intensions des concepts inférieurs contenant plus de mots-clef et moins d'images). Ce parcours à travers les concepts du treillis assure une navigation progressive dans la base : à chaque pas l'utilisateur choisit soit d'ajouter soit d'enlever des images. Il peut ainsi observer les conséquences successives de l'ajout ou de la suppression de mots-clef de sa requête sans que son ensemble de résultats soit totalement bouleversé (cf. fig. 4.4).

4.7 Conclusion

Les techniques de FCA permettent d'extraire et de classer des regroupements pertinents objets/attributs à partir de données tabulaires. Ces concepts ordonnés peuvent être vus comme une représentation de la structure des données d'origine. Ainsi plusieurs applications les utilisent dans le domaine de la recherche d'information afin de classer et naviguer parmi les résultats d'une requête. Typiquement, chaque concept du treillis contient en extension un sous-ensemble

[5] `credo.fub.it`

[6] `www.kvocentral.org/software/imagesleuth.html`

de résultats qui est présenté sous forme de liste. Nous proposons quant à nous d'utiliser cette structure extraite par FCA pour construire une visualisation de type *overview + context* (cf. 2.2.3) des données tabulaires d'origine. Le treillis constitue la vue globale (*overview*) des données, et la vue locale est une projection MDS des objets contenus en extension d'un concept sélectionné par l'utilisateur. Cette méthode est présentée dans les chapitres suivants.

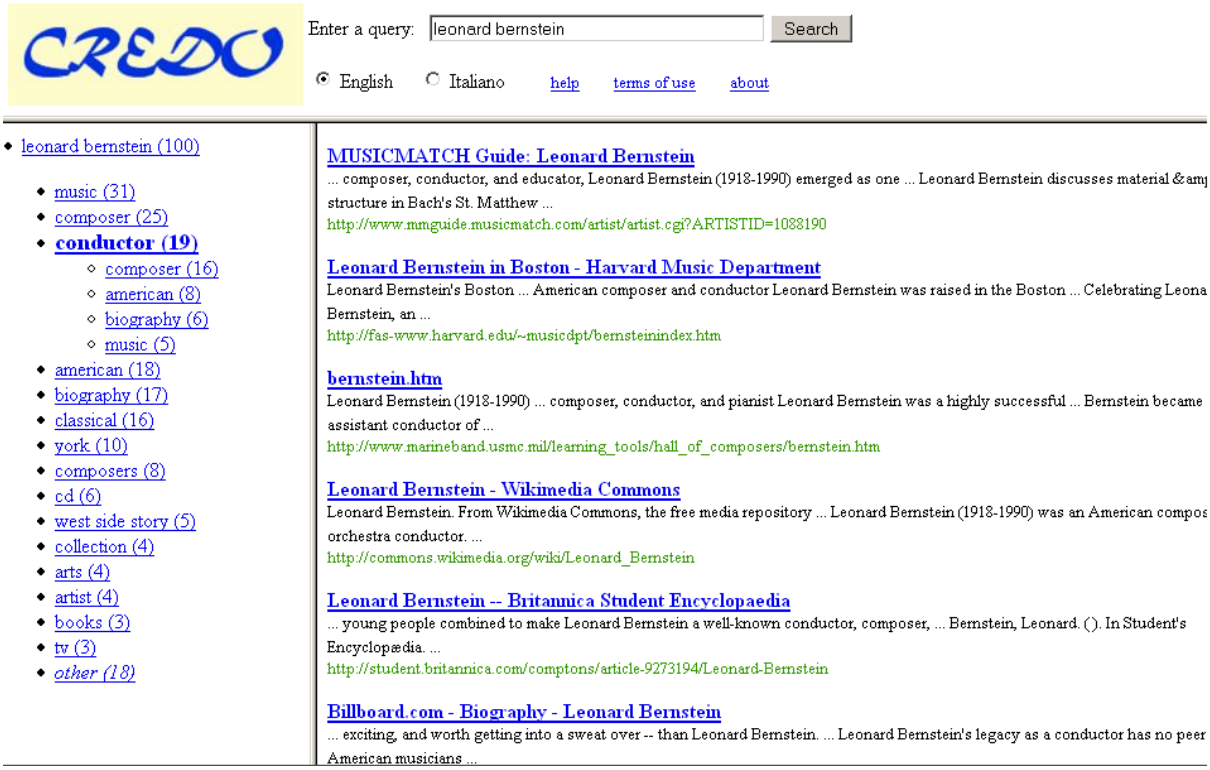


FIG. 4.2 – Outil de recherche web CREDO [CR04]. Les pages retournées par le moteur de recherche sont utilisées pour construire un contexte formel dans lequel les objets sont les pages et les attributs leurs mots-clef. Les concepts du treillis engendré contiennent des ensembles de pages en extension et des mots-clef en intension. Le panneau de gauche présente les intensions réduites des concepts du premier niveau du treillis (les fils directs de $\top$). Lorsque l'utilisateur sélectionne une intension, les pages en extension du concept sont listées sur le panneau de droite et les intensions des fils directs du concept sélectionné sont affichées sur le tableau de gauche. Ainsi, les résultats de la requête *leonard + bernstein* ont généré, entre autres, deux concepts comportant *composer* pour l'un, et *conductor*, pour l'autre, en intension réduite. Ces deux concepts contiennent en extension les pages comportant respectivement le mot-clef *composer* et le mot-clef *conductor*. L'utilisateur a sélectionné *conductor* et on voit apparaître quatre fils du concept *conductor*. Parmi ceux-ci, le fils étiquetté *composer* contient les pages comportant à la fois les mots-clef *composer* et *conductor*.

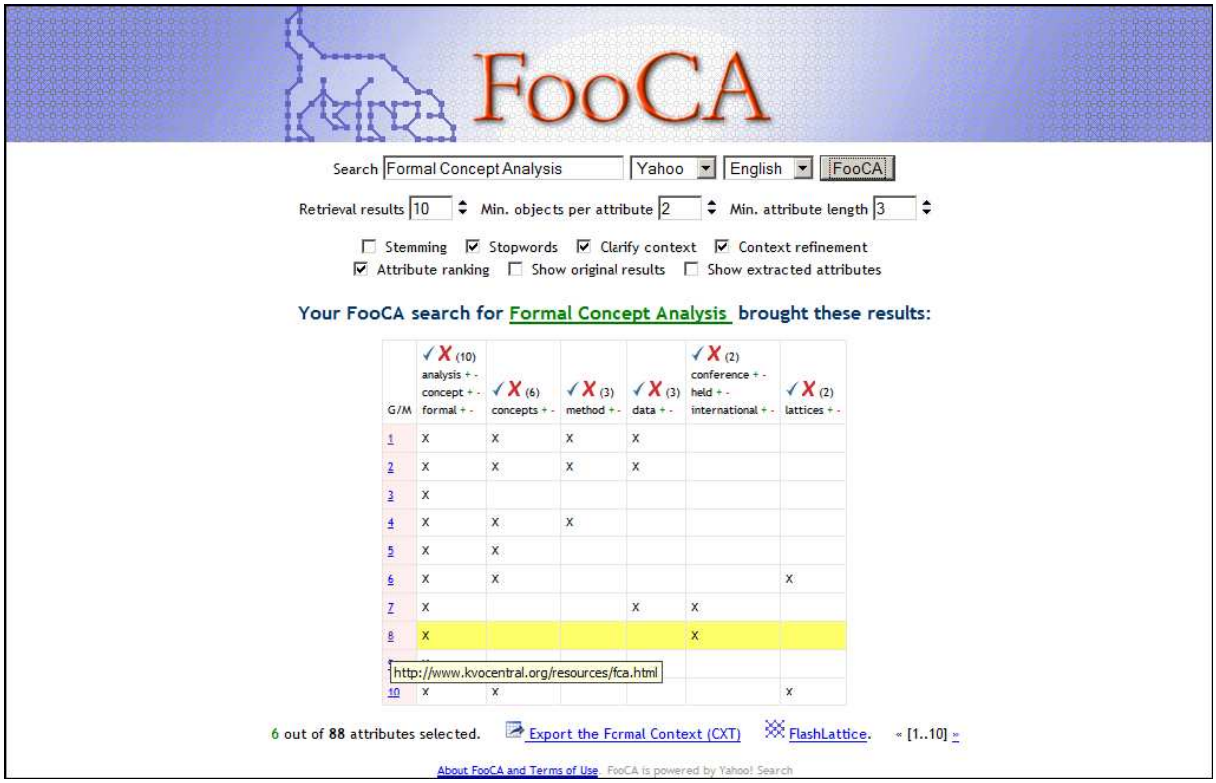


FIG. 4.3 – FOOCA [Koe06] reprend le principe de CREDO en permettant à l'utilisateur de manipuler le contexte formel extrait des résultats de la requête pour affiner sa recherche.

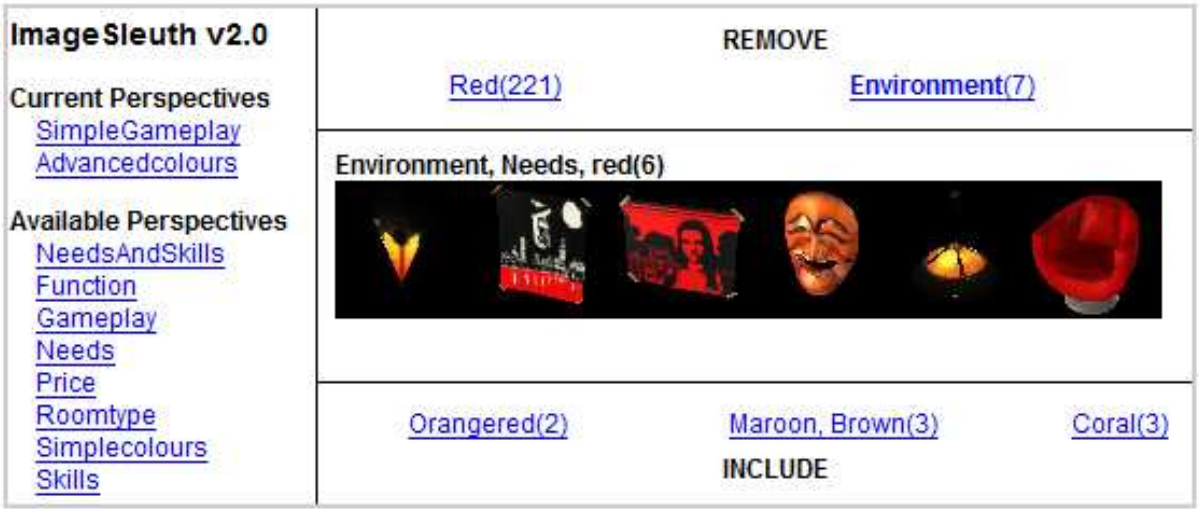


FIG. 4.4 – IMAGESLEUTH est un outil de navigation dans une base d'images [DVE06]. Les images et les mots-clef sur lesquels elles sont indexés définissent un contexte formel. La navigation s'effectue en parcourant les concepts générés, à la manière de CREDO. Dans l'exemple ci-dessus, le concept courant comprend les mots-clef {*environment, needs, red*} en intension et les six images formant son extension sont affichées. Le système propose de naviguer vers un concept père direct en retirant de l'intension soit *red*, soit *environment* ; ou bien vers un fils direct en ajoutant à l'intension les mots-clef *orangered*, *brown* ou *coral*. Le panneau de gauche permet de sélectionner un sous-ensemble de mots-clef, correspondant à un certain point de vue, pour construire le contexte formel.

5

Projection MDS sélective de données creuses assistée par FCA

Sommaire

DANS ce chapitre nous présentons une solution au verrou formulé dans la section 3.4.1. Rappelons que ce verrou concerne la mise en œuvre de la force propriété (projection MDS) sur des matrices objets/attributs comportant des données manquantes. Nous avons vu que le calcul de distance entre objets était biaisé lorsque qu'il existait des données manquantes sur un ou plusieurs des attributs pris en compte dans le calcul de distance. Nous avons présenté le principe « MDS sélectif » permettant de ne prendre en compte qu'un sous-ensemble d'attributs dans le calcul de distance entre objets. L'objectif est d'identifier des couples objets/attributs (O_i, A_i) tels qu'il n'existe aucun attribut de A_i dont la valeur est manquante pour un objet de O_i. Le résultat du calcul des distances entre objets de O_i selon les attributs de A_i est alors valide. Nous proposons d'utiliser les techniques de FCA afin d'identifier, à partir des données brutes, l'ensemble des couples (O_i, A_i).

5.1 Identification des couples

Nous nous restreignons dans ce chapitre aux attributs numériques, le cas des données mixtes étant traité dans le chapitre suivant. Nous prendrons comme exemple un jeu de données issu de [AN07] ayant trait aux caractéristiques de 205 modèles de véhicules immatriculés aux États-Unis en 1985. Le tableau 5.1 présente la liste des attributs et la distribution des valeurs manquantes.

Un contexte formel binaire $\mathbb{K}$ est construit pour lequel l'ensemble des objets O est l'ensemble des 205 véhicules et A l'ensemble des attributs. L'objectif est d'identifier des couples

attribut		manquantes	traduction
a_1	normalized losses	41	indicateur destiné aux assureurs
a_2	wheelbase	0	empattement
a_3	length	0	longueur
a_4	width	0	largeur
a_5	height	0	hauteur
a_6	curb weight	0	poids avec carburant
a_7	engine size	0	cylindrée
a_8	bore	4	alésage (diamètre d'un piston)
a_9	stroke	4	course (distance parcourue par un piston durant un cycle)
a_{10}	compression ratio	0	taux de compression
a_{11}	horsepower	2	puissance
a_{12}	peak rpm	2	vitesse de rotation maximale (tr/min)
a_{13}	city mpg	0	consommation urbaine
a_{14}	highway mpg	0	consommation routière
a_{15}	price	4	prix de vente

TAB. 5.1 – Liste des attributs avec le nombre d'objets pour lesquels l'information est manquante.

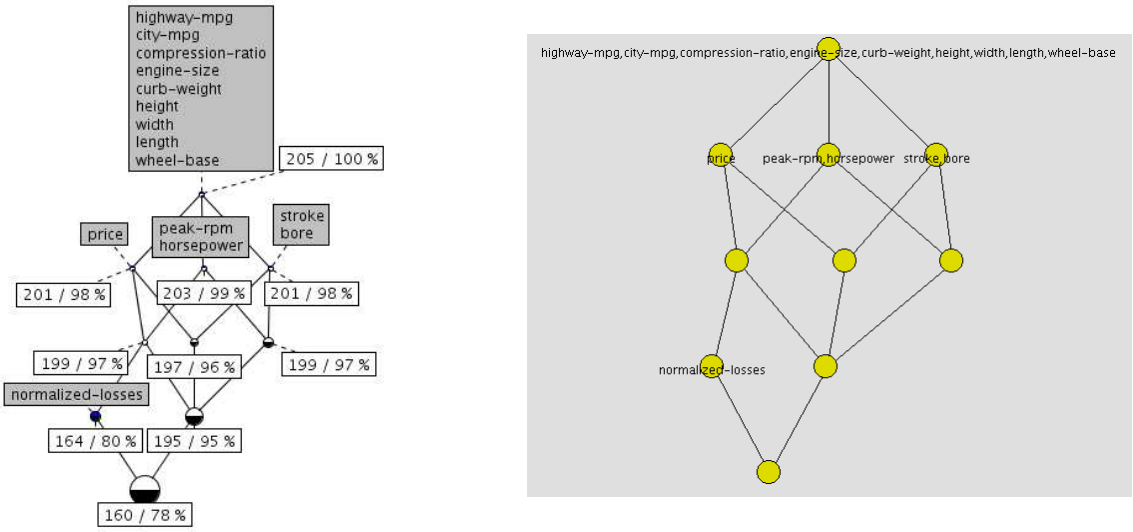
$o \in \text{ext}(c)$	o est comparable avec les objets de $\text{ext}(c)$ selon les attributs de $\text{int}(c)$
$o \in \text{ext}_s(c)$	o n'est pas valué sur d'autres attributs que ceux de $\text{int}(c)$
$a \in \text{int}(c)$	les objets de $\text{ext}(c)$ possèdent tous une valeur pour a
$a \in \text{int}_s(c)$	a n'est pas valué sur d'autres objets que ceux de $\text{ext}(c)$

TAB. 5.2 – Interprétation des extension et intension d'un concept $c \in \mathcal{C}_{\mathbb{K}}$.

objets/attributs en fonction de la présence de valeurs sur les attributs. Ainsi $(o, a) \in I$ si l'objet o est valué sur l'attribut a. Autrement dit une cellule (i, j) n'a pas de croix si la valeur de a_j est manquante pour l'objet o_i. Le treillis de concepts associé est illustré par la figure 5.1. Chaque concept (O_i, A_i) représente l'information suivante : tous les objets de O_i sont valués sur tous les attributs de A_i, i.e. les objets de O_i sont comparables sur tous les attributs de A_i. Une projection MDS non biaisée peut alors être mise en œuvre. Le nombre de concepts identifiés est $|\mathcal{C}_{\mathbb{K}}| = 10$. L'interprétation de leurs extensions et intensions est résumée dans le tableau 5.2.

Plusieurs remarques peuvent être faites concernant certains concepts. On note tout d'abord que $\text{int}(\top) = \text{int}_s(\top) \neq \emptyset$. Cela signifie que les objets sont tous valués sur $\text{int}_s(\top)$. En effet, rappelons que par construction $\top = (f(O), O)$. Le sous-ensemble d'attributs $\text{int}_s(\top)$ contient les attributs sans aucune valeur manquante pour l'ensemble des objets. Cela est vérifiable en comparant $\text{int}_s(\top)$ avec la colonne « manquantes » du tableau 5.1. On note également $\text{ext}(\bot) = \text{ext}_s(\bot) \neq \emptyset$. Par construction, on a $\bot = (A, g(A))$. $\text{ext}_s(\bot)$ contient ainsi les objets valués sur tous les attributs, donc ne comportant aucune valeur manquante. On remarque que $|\text{ext}_s(\bot)| = 160$ et que les valeurs manquantes concernent $|O| - |\text{ext}_s(\bot)| = 45$ objets. On note encore que certains attributs apparaissent ensemble dans l'intension simplifiée d'un même concept. C'est le cas pour {*peak rpm, horsepower*} d'une part, et pour {*stroke, bore*} d'autre part, et cela signifie que lorsque l'information est manquante pour *stroke* sur un objet, elle l'est aussi pour *bore* sur le même objet. On observe l'implication

$$normalizedlosses \longrightarrow price, horsepower, peak\ rpm$$



(a) (b)

FIG. 5.1 – Treillis de concepts associé à $\mathbb{K}$. Chaque concept contient en extension un ensemble d'objets comparables sur l'ensemble d'attributs en intension. (a) Représentation produite par CONEXP : la partie supérieure (resp. inférieure) d'un concept est bleue (resp. noire) lorque son intension (resp. extension) simplifiée est non vide. Les étiquettes mentionnent les intensions simplifiées des concepts ainsi que le cardinal de leurs extensions et le pourcentage de chaque extension par rapport au nombre total d'objets. (b) Représentation produite par MOLAGE.

En effet, en tant qu'indicateur, on peut supposer que *normalized losses* est calculé en fonction du prix et de la puissance du véhicule, *peak rpm* apparaissant conséquemment aux implications réciproques $horsepower \leftrightarrows peak\ rpm$.

5.2 Organisation visuelle : principe général

Nous proposons à l'utilisateur une visualisation de type *overview + detail* (cf. section 2.2.3) illustrée par la figure 5.2. La vue globale (*overview*) représente le treillis. Chaque concept (O_i, A_i) correspond à une projection MDS non biaisée des objets en extension O_i sur les attributs en intension A_i, comme nous l'avons vu dans la section précédente. Lorsque l'utilisateur sélectionne un concept, la vue locale (*detail*) affiche la projection MDS correspondante. Le treillis, en tant que vue globale, permet donc de naviguer parmi les différentes projections MDS possibles. En sélectionnant le concept $\top$, l'utilisateur obtiendra la vision la plus générale mais la moins précise des similarités entre objets. En effet, la sélection de $\top$ entraîne l'affichage sur la vue locale de la totalité des objets projetés selon les attributs qu'ils ont tous en commun. Sur notre exemple, lorsque $\top$ est sélectionné, la vue locale affiche la totalité des 205 objets (extension de $\top$) projetés sur 9 attributs (intension de $\top$). La sélection de $\top$ a pour avantage de permettre l'observation des similarités entre tous les objets. Cependant, les similarités calculées entre certains objets ne

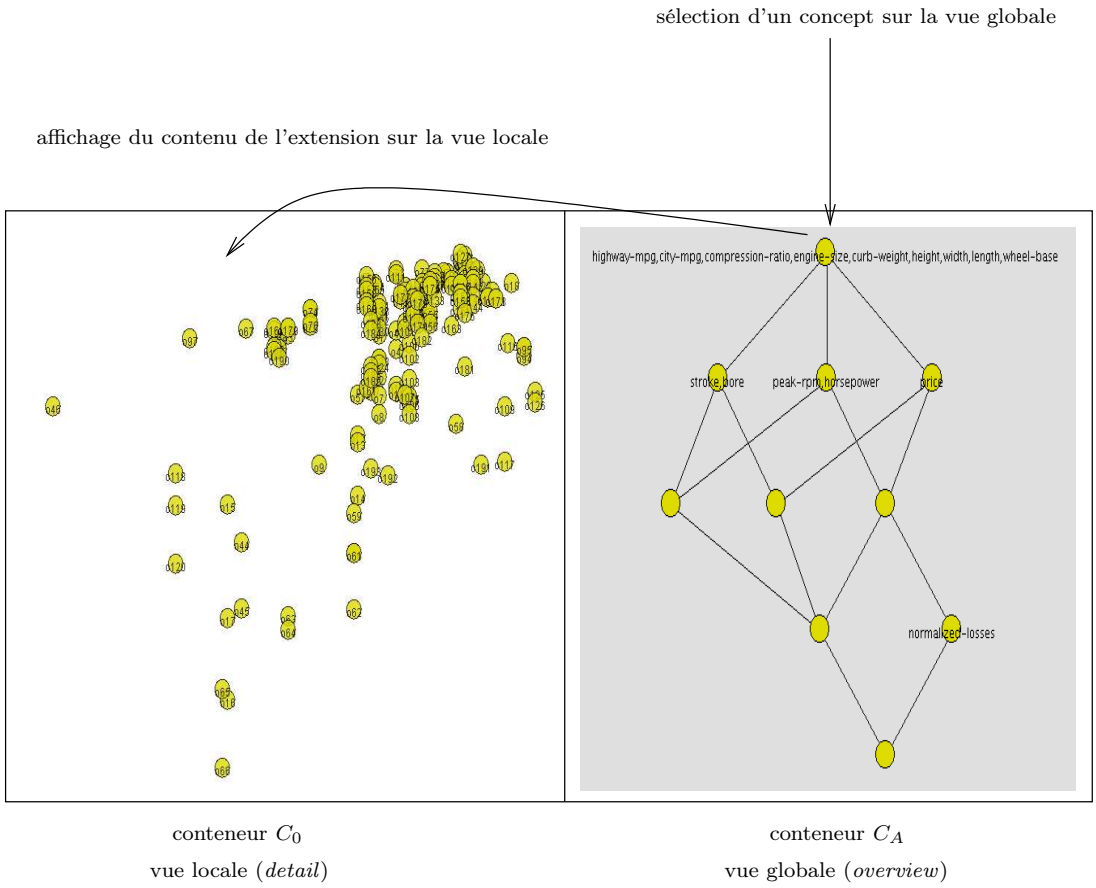


FIG. 5.2 – Principe général.

prennent pas en compte la totalité de l'information disponible puisqu'elles ne sont calculées que sur 9 des 15 attributs. En sélectionnant un des trois fils de $\top$, par exemple le concept contenant l'attribut *price*, l'utilisateur affine la précision de ces similarités, puisqu'elles seront calculées en tenant compte des valeurs sur l'attribut *price* mais restreint le nombre d'objets affichés à ceux valués sur cet attribut. Ainsi, plus l'utilisateur descend dans le treillis, plus les similarités affichées sur la vue locale seront précises et plus le nombre d'objets sera restreint. Nous présentons dans la suite la réalisation de cette visualisation *overview + detail* dans MOLAGE et introduisons le dispositif de conteneur permettant de séparer l'espace de représentation en deux vues distinctes.

5.2.1 Conteneurs

Un conteneur permet d'assigner des propriétés spécifiques à une zone rectangulaire de l'espace visuel. Un conteneur C est caractérisé par :

- deux couples (x_1, y_1) et (x_2, y_2) représentant deux points définissant la diagonale du rectangle,
- un ensemble d'atomes affectés au conteneur,
- une propriété *selected* renvoyant l'atome sélectionné par l'utilisateur.

Les positions des atomes affectés à C sont restreintes à la zone rectangulaire associée au conteneur. Des forces peuvent être déclenchées sur un conteneur, elles s'appliquent alors uniquement

aux atomes du conteneur.

5.2.2 Mise en œuvre dans MOLAGE

Dans cette section, nous explicitons le processus de construction de la visualisation à partir des données initiales en suivant les étapes du modèle de Card-Chi introduit dans la section 2.2.1.

Données brutes Les données brutes sont constituées de la matrice objets/attributs.

Abstraction analytique Un prétraitement extrait des données deux abstractions analytiques qui serviront à construire, d'une part, la vue locale, et d'autre part, la vue globale.
- vue globale : ensemble des concepts formant le treillis associé au contexte $\mathbb{K}$ où $(o, a) \in I$ si l'objet o est valué sur l'attribut a,
- vue locale : ensemble des objets munis de leurs vecteurs d'attributs, i.e. les lignes de la matrice de données initiales.

Abstraction visuelle
- vue globale : sommets et arêtes constituant le diagramme de Hasse du treillis,
- vue locale : matrice de distances entre objets.

Vue
- vue globale : conteneur C_A contenant des atomes de type *Concept* représentant le diagramme de Hasse du treillis,
- vue locale : conteneur C_O contenant des atomes de type *Car* disposés par projection MDS selon les attributs en intension de l'atome *Concept* sélectionné sur la vue globale.

Deux conteneurs sont définis afin de partitionner l'espace visuel en deux vues distinctes. L'une représente le treillis, l'autre les objets (voitures). La figure 5.3 illustre le processus de construction des deux vues en fonction des données brutes. D'une part (côté gauche de fig. 5.3) chaque ligne de la matrice initiale, correspondant à un objet o_i muni de ses valeurs d'attributs, est associée à un atome de type *Car* qui constitue sa représentation visuelle. D'autre part (côté droit de fig. 5.3) le contexte binaire $\mathbb{K}$ est construit à partir de la matrice initiale. Chaque concept $c_k \in \mathcal{C}_{\mathbb{K}}$, muni des quatre propriétés ext, ext_s, int et int_s, est associé à un atome de type *Concept*. Les atomes de types *Car* (resp. *Concept*) sont affectés à un conteneur C_O (resp. C_A) correspondant à la partie gauche (resp. droite) de l'espace visuel. Le tableau 5.3 résume la traduction des entités logiques en entités visuelles.

5.3 Organisation visuelle : détails sur C_O

Le conteneur C_O contient les atomes de type *Car* représentant les objets. Ces atomes sont disposés par MDS en fonction des attributs en intension de l'atome *Concept* sélectionné sur le conteneur C_A. L'atome sélectionné est $C_A.selected$. Les attributs en intension sont donc $C_A.selected.int$ et les objets comparables sur ces attributs sont $C_A.selected.ext$. Le conteneur C_O ne doit représenter que les atomes *Car* correspondant à $C_A.selected.ext$ et les disposer par MDS en déclenchant la force propriété $F_P(C_A.selected.ext, C_A.selected.ext, C_A.selected.int)$. Le conteneur C_O est donc mis à jour d'après l'atome sélectionné sur C_A en appliquant l'algorithme 5.1. Cette organisation fournit à l'utilisateur une interface lui permettant de naviguer parmi les

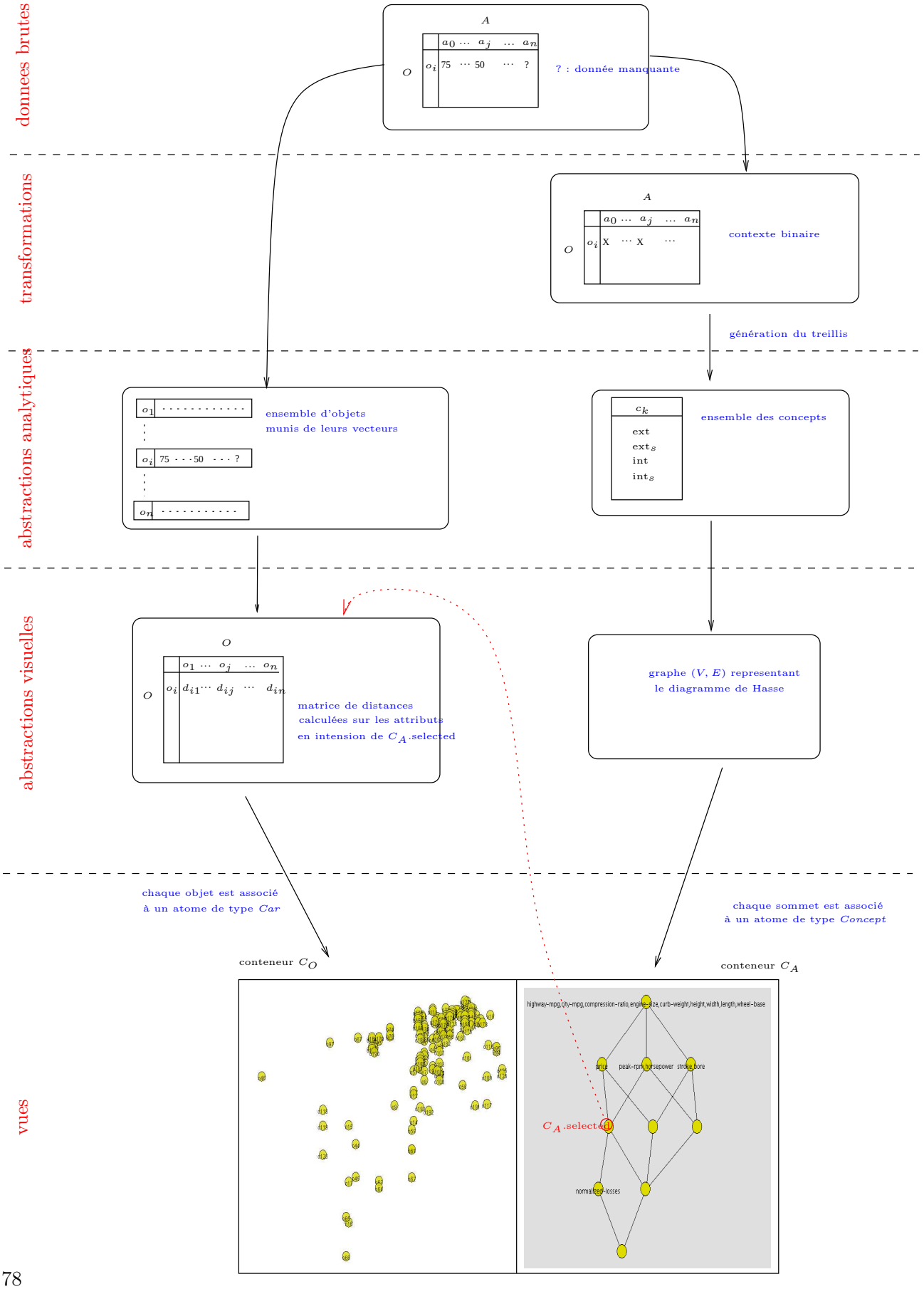


FIG. 5.3 – Processus de construction des vues

entité logique	entité visuelle
ensemble d'objets O	type d'atomes *Car*
objet $o \in O$	atome de type *Car*
ensemble des concepts $\mathcal{C}$	type d'atomes *Concept*
concept $c \in \mathcal{C}$	atome x de type *Concept*
$\text{ext}(c)$	$x.ext$
$\text{int}(c)$	$x.int$
$\text{ext}_s(c)$	$x.ext_s$
$\text{int}_s(c)$	$x.int_s$

TAB. 5.3 – Mise en œuvre dans Molage

objets en s'assurant de la comparabilité des attributs utilisés pour le calcul de distance entre objets.

ALGORITHME 5.1 : Mise à jour de la vue des objets (conteneur C_O) en fonction de l'atome sélectionné sur le treillis des attributs

Entrées : L'atome $C_A.selected$ sélectionné sur le treillis des attributs
Sorties : Le conteneur C_O mis à jour

$Q \leftarrow \emptyset$ `/* atomes qui seront disposés par MDS */`
pour chaque $o \in Car$ **faire** `/* pour chaque atome de type Car */`
  $o.visible \leftarrow$ FAUX
  pour chaque $p \in C_A.selected.ext$ **faire**
    si $o.name = p$ **alors** `/* si son nom apparait en extension */`
      $o.visible \leftarrow$ VRAI `/* il est visible */`
      $Q \leftarrow Q \cup \{o\}$ `/* il sera disposé par MDS */`
    fin
  fin
fin
`/* MDS entre atomes en extension selon les attributs en intension */`
$F_P(Q, Q, C_A.selected.int)$

5.4 Organisation visuelle : détails sur C_A

Le conteneur C_A contient les atomes de type *Concept*. Plusieurs solutions ont été expérimentées pour disposer ces atomes. La première consiste à reconstituer le diagramme de Hasse, la seconde à spécifier une distance entre atomes *Concept* et à les disposer selon cette distance.

5.4.1 Reconstitution du diagramme de Hasse

Nous avons vu que les atomes de type *Concept* étaient munis de quatre propriétés contenant les intensions et extensions des concepts qu'ils représentent. Deux informations supplémentaires sont nécessaires pour recréer le diagramme de Hasse à partir des atomes : la relation de voisinage et le niveau de chaque concept. La relation de voisinage est aisément représentée en spécifiant une liaison entre atomes représentant des concepts voisins directs, i.e. on pose $R(c, d)$ si $c \prec d$.

L'algorithme de dessin de diagramme de Hasse qui suit est inspiré de l'algorithme de Freese [Fre04]. Afin de maintenir l'orientation du diagramme, il est nécessaire de calculer le niveau de chaque concept. Bien que le diagramme de Hasse soit un graphe acyclique et non un arbre, nous définissons le niveau d'un concept c comme un entier $c.niv$ tel que :

- $\forall\, d \mid c \prec d,\ c.niv > d.niv$ (le niveau d'un père est strictement inférieur à celui de son fils),
- $\top.niv = 0$ (le concept racine est au niveau 0).

L'algorithme récursif 5.2 réalise le marquage des concepts dont les niveaux sont supposés initialisés à 0.

ALGORITHME 5.2 : MarquageNiveau(c , niv_p). Algorithme de marquage du niveau des concepts. Appel récursif à partir de MarquageNiveau($\top$, -1). Les $c.niv$ sont initialisés à 0.

Entrées : c : concept courant, niv_p : niveau du père appelant

si $c.niv = 0$ || $c.niv \leq niv_p$ **alors** `/* non exploré ou inférieur au père */`
  $c.niv \leftarrow niv_p + 1$
fin
pour chaque $d \prec c$ **faire**
  MarquageNiveau(d , $c.niv$) `/* appel récursif sur les fils */`
fin

Une fois les atomes marqués, on les dispose selon l'algorithme 5.3. Les atomes $\top$ et $\bot$ sont d'abord placés respectivement en haut et en bas du conteneur C_A. Puis les niveaux sont normalisés entre 0 et 100. Une force propriété est déclenchée afin que les atomes se regroupent selon leur niveau (MDS sur niv). Ils forment des clusters d'atomes dont les niveaux décroissent de haut en bas puisqu'ils se positionnent en fonction des positions fixes de $\top$ et $\bot$. Les atomes sont ensuite fixés en y afin qu'ils ne puissent se déplacer que sur x. Enfin, une force limite déploie horizontalement les atomes de chaque niveau.

ALGORITHME 5.3 : Disposition des atomes *Concept*

$\top.y \leftarrow \min_y$ `/* ⊤ en haut */`
$\top.fixed \leftarrow$ VRAI
$\bot.y \leftarrow \max_y$ `/* ⊥ en bas */`
$\bot.fixed \leftarrow$ VRAI
pour chaque $c \in Concept$ **faire**
  $c.niv \leftarrow \frac{100}{|C|}\, c.niv$ `/* normalisation entre 0 et 100 */`
fin
$F_P(Concept, Concept, niv)$ `/* regroupement en niveaux par MDS */`
pour chaque $c \in Concept$ **faire**
  $c.yfixed \leftarrow$ VRAI `/* blocage en y */`
fin
$F_\delta(Concept, Concept, 10)$ `/* déploiement en x par force limite */`

Afin d'afficher l'intension simplifiée d'un atome *Concept* au passage de la souris, une lentille de proximité $l_{prox}.label = \text{int}_s$ est spécifiée. Une lentille topologique $LensTopo(Concept, Concept, 1)$ peut être ajoutée afin d'afficher également les intensions simplifiées des voisins directs du concept survolé.

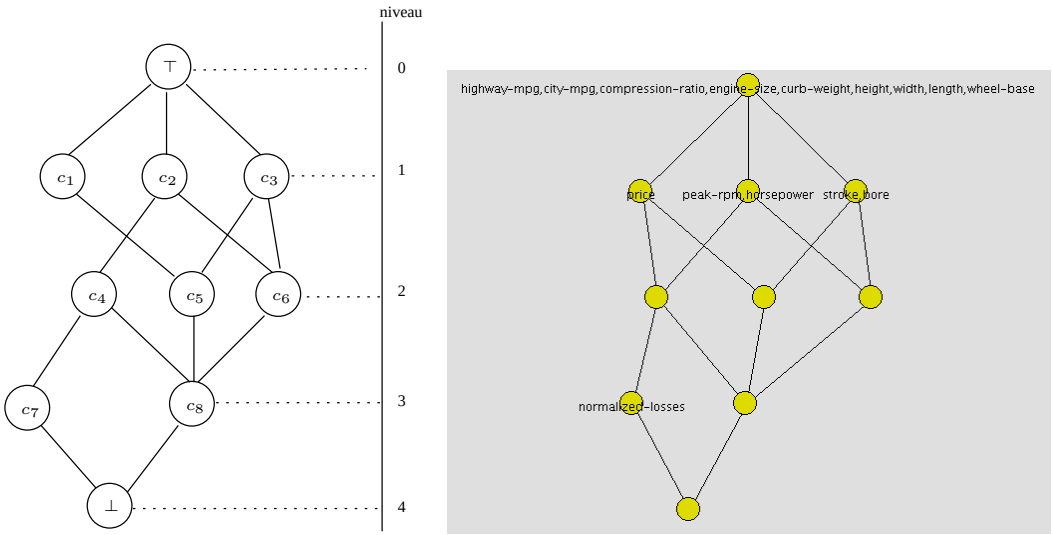


FIG. 5.4 – Marquage des niveaux

5.4.2 Distances euclidienne et de Jaccard

Une alternative consiste à calculer une matrice de distances entre concepts et à les disposer selon ces distances. Les tableaux 5.5 et 5.6 montrent respectivement les matrices de distances selon la distance euclidienne et la distance de Jaccard (cf. section 2.3.2). La figure 5.5 montre des résultats relativement proches. Notons que seules les intensions des concepts sont prises en compte dans le calcul des distances euclidiennes et de Jaccard telles que nous les employons ici. Relativement peu de travaux se sont penchés sur les mesures de similarité et de distance entre concepts. Toutefois, dans le cadre de l'élaboration de l'outil SEARCHSLEUTH [DE07], de nouvelles avancées ont été accomplies récemment. En effet, Dau, Ducrou et Eklund rappellent [DDE08] les deux mesures de distances entre concepts utilisées jusqu'à présent :

$$d_1((O_i, A_i), O_j, A_j)) = \frac{1}{2}\left(\frac{|O_i \cup O_j| - |O_i \cap O_j|}{|O_i \cup O_j|} + \frac{|A_i \cup A_j| - |A_i \cap A_j|}{|A_i \cup A_j|}\right)$$

$$d_2((O_i, A_i), (O_j, A_j)) = \frac{1}{2}\left(\frac{|O_i \cup O_j| - |O_i \cap O_j|}{|O|} + \frac{|A_i \cup A_j| - |A_i \cap A_j|}{|A|}\right)$$

et montrent que d_1 peut être considérée comme une distance locale, étant uniquement fondée sur les objets et attributs partagés par les deux concepts ; tandis que d_2 prend en compte le nombre total d'objets et d'attributs au dénominateur, et peut être considérée comme une distance globale. Notons que d_1 peut être déduite de la distance de Jaccard : $d_1((O_i, A_i), (O_j, A_j)) = \frac{1}{2}(J_\delta(O_i, O_j) + J_\delta(A_i, A_j))$.

5.5 Conclusion

Dans ce chapitre, nous avons proposé une solution au problème de la projection MDS sur des objets contenant des valeurs manquantes pour certains attributs. Nous avons introduit notre méthode de visualisation *overview + detail* consistant en :

– une vue globale représentant les différents couples attributs/objets pour lesquels une projection MDS non biaisée peut être générée,

$\mathcal{C}$	highway-mpg	city-mpg	compression-ratio	engine-size	curb-weight	height	length	wheel-base	price	peak-rpm	horsepower	stroke	bore	normalized-losses
$\top$	×	×	×	×	×	×	×	×						
c_1	×	×	×	×	×	×	×	×	×					
c_2	×	×	×	×	×	×	×	×		×	×			
c_3	×	×	×	×	×	×	×	×				×	×	
c_4	×	×	×	×	×	×	×	×	×	×	×			
c_5	×	×	×	×	×	×	×	×	×			×	×	
c_6	×	×	×	×	×	×	×	×		×	×	×	×	
c_7	×	×	×	×	×	×	×	×						×
c_8	×	×	×	×	×	×	×	×	×	×	×	×	×	
$\bot$	×	×	×	×	×	×	×	×	×	×	×	×	×	×

TAB. 5.4 – Intensions des concepts $\mathcal{C}$

	$\top$	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	$\bot$
$\top$	0	1	1,41	1,41	1,73	1,73	2	2	2,24	2,45
c_1		0	1,73	1,73	1,41	1,41	2,24	1,73	2	2,24
c_2			0	2	1	2,24	1,41	1,41	1,73	2
c_3				0	2,24	1	1,41	2,45	1,73	2
c_4					0	2	1,73	1	1,41	1,73
c_5						0	1,73	2,24	1,41	1,73
c_6							0	2	1	1,41
c_7								0	1,73	1,41
c_8									0	1
$\bot$										0

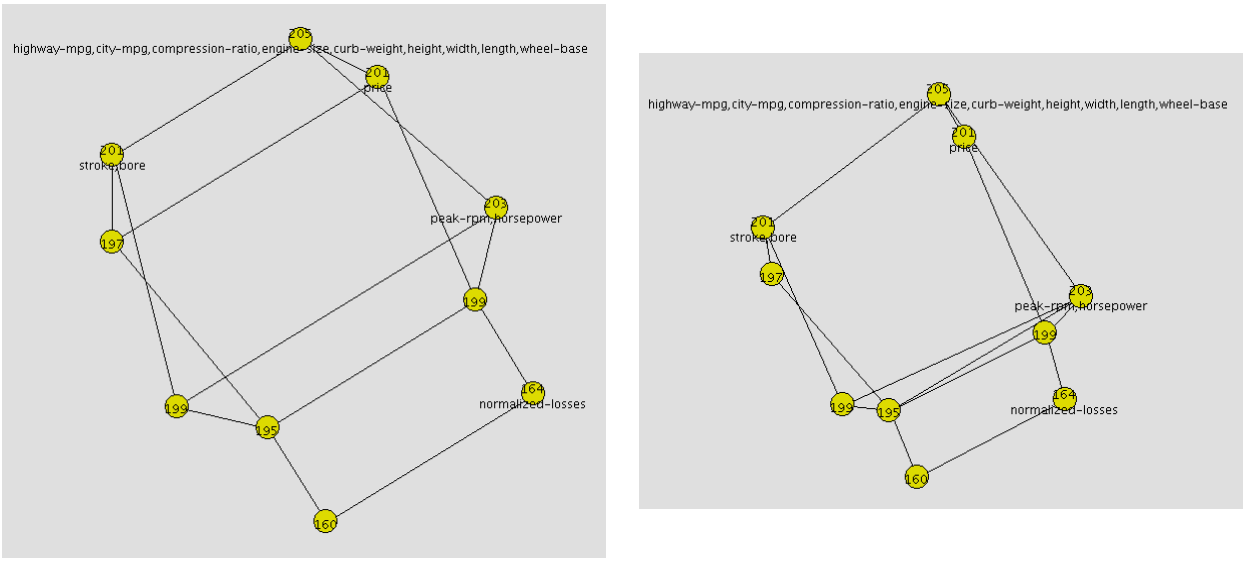
TAB. 5.5 – distance euclidienne entre concepts

	$\top$	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	$\bot$
$\top$	0	0,1	0,18	0,18	0,25	0,25	0,31	0,31	0,36	0,4
c_1		0	0,25	0,25	0,17	0,17	0,36	0,23	0,29	0,33
c_2			0	0,31	0,08	0,36	0,15	0,15	0,21	0,27
c_3				0	0,36	0,08	0,15	0,4	0,21	0,26
c_4					0	0,29	0,21	0,08	0,14	0,2
c_5						0	0,21	0,33	0,14	0,2
c_6							0	0,27	0,07	0,13
c_7								0	0,2	0,13
c_8									0	0,07
$\bot$										0

TAB. 5.6 – distance de Jaccard entre concepts

– une vue locale représentant les objets sans valeurs manquantes pour les attributs du couple sélectionné sur la vue globale, et projetés selon ces attributs.

Dans le chapitre suivant, nous reprenons ce principe *overview + detail* comme base de notre solution au problème de l'hétérogénéité des attributs.



(a) (b)

FIG. 5.5 – Spatialisation selon la distance euclidienne (a) et la distance de Jaccard (b)

6

Navigation visuelle *overview + detail* dans un contexte mixte

Sommaire

NOUS proposons dans ce chapitre une solution au verrou formulé dans la section 3.4.2, concernant la visualisation d'objets valués sur un ensemble d'attributs mixte, i.e. comportant des attributs de natures différentes, binaires, nominaux, ordinaux ou continus. Notre proposition repose toujours sur une visualisation *overview + detail*. Nous proposons d'utiliser les attributs binaires pour générer un treillis qui constituera la vue globale. En sélectionnant un concept, l'utilisateur déclenche l'affichage des objets en extension sur la vue locale, disposés par MDS selon les attributs continus. Nous revenons dans un premier temps sur le traitement des attributs non binaires en FCA et sur une variante du diagramme de Hasse, appelée *nested-line diagram* ou diagramme enchevêtré. Nous développons notre solution en deux temps. Tout d'abord, nous montrons comment la projection MDS constitue une alternative aux diagrammes enchevêtrés. Dans un second temps, nous présentons notre visualisation de type *overview + detail* en tant que telle. Nous expliquons enfin comment des informations sur les corrélations entre attributs peuvent être déduites de l'observation de la vue locale.

6.1 Diagrammes enchevêtrés (*nested-line diagrams*)

Les échelles conceptuelles (cf. section 4.3) permettent d'étendre FCA aux attributs non binaires mais entraînent une augmentation significative du nombre d'attributs binaires présents dans le contexte dérivé, donc du nombre de concepts dans le treillis associé. Les diagrammes

enchevêtrés peuvent apporter une solution à ce problème tout en offrant la possibilité d'observer les relations entre attributs multivalués. Un diagramme enchevêtré, ou *nested-line diagram*, est construit en partitionant l'ensemble des attributs multivalués en deux sous-ensembles appelés *facteurs*, puis en générant séparément les treillis des deux sous-contextes induits, enfin en remplaçant chaque nœud du treillis du premier facteur par le diagramme de Hasse du treillis du second facteur, de sorte que chaque nœud du premier facteur représente la distribution de son extension selon le second facteur. Cette opération est appelée *produit direct* des deux treillis. Nous revenons dans la suite sur l'exemple introduit dans la section 4.3. L'attribut *sexe* est choisi comme premier facteur et âge comme second facteur. Les deux sous-contextes induits sont donc les échelles réalisées $\mathbb{R}_{sexe}$ et $\mathbb{R}_{âge}$. La figure 6.1 représentent les treillis associés aux deux sous-contextes et la figure 6.2 le diagramme enchevêtré associé. Notons que les nœuds dont l'extension simplifiée est vide ne sont pas retirés afin de préserver la structure de l'échelle conceptuelle utilisée pour le second facteur. Ces figures ont été réalisées grâce à l'outil TOSCANAJ (cf. section 4.5) qui est, à notre connaissance, le seul à générer des diagrammes enchevêtrés.

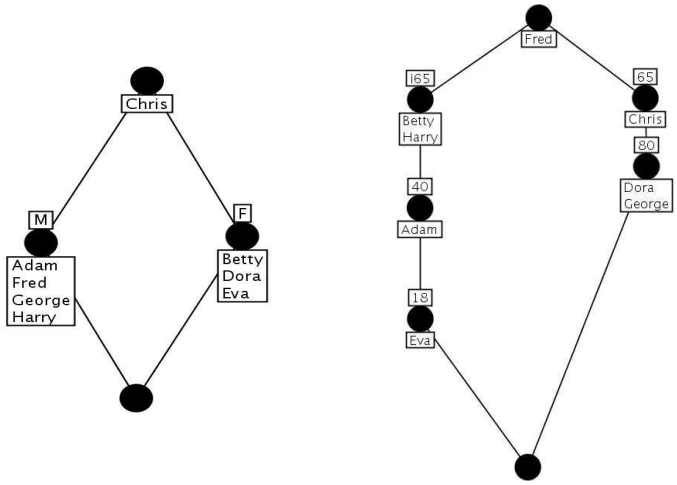


FIG. 6.1 – Diagrammes de Hasse des premier et second facteurs.

6.2 La projection MDS comme alternative aux diagrammes enchevêtrés

Combiner plus de deux échelles engendre des diagrammes encapsulés complexes utiles pour une analyse en profondeur mais peu adaptés à la navigation intuive que nous souhaitons fournir à l'utilisateur. De plus des incompréhensions peuvent survenir, y compris sur des diagrammes encapsulés simples. Ainsi sur le diagramme illustré par la figure 6.2, la dichotomie entre ≤ 65 et > 65 sur le second facteur âge apparaît clairement mais l'âge décroît sur la branche de gauche tandis qu'il croît sur celle de droite. Une telle représentation peut induire en erreur un utilisateur qui s'attend à une représentation intuitive reflétant la relation d'ordre sur $\mathbb{N}$. En outre la discrétisation des attributs numériques conduit inévitablement à une perte de précision en terme de similarité entre objets. L'alternative que nous proposons considère l'ensemble des attributs binaires comme premier facteur et l'ensemble des attributs non binaires comme second facteurs. Le treillis associé au sous-contexte induit par le premier facteur représente la structure de

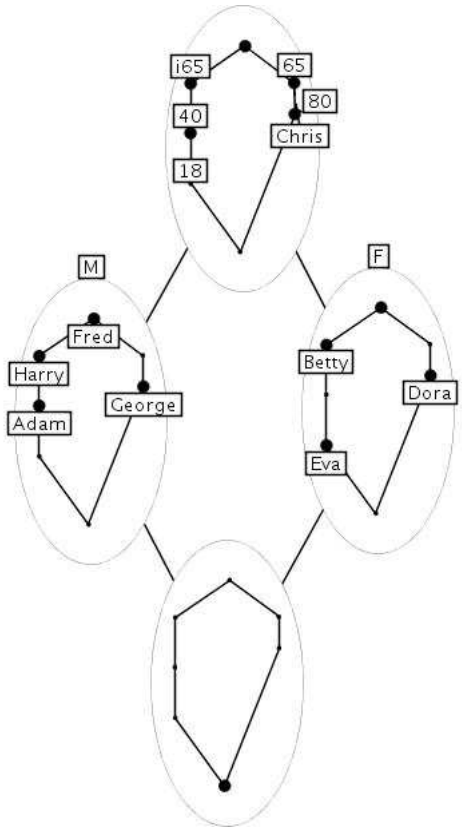


FIG. 6.2 – diagramme enchevêtré réalisé par TOSCANAJ. Les nœuds du treillis associé au premier facteur ont été remplacés par le treillis du second facteur.

la base, et chaque nœud de ce treillis est remplacé par une projection MDS des objets en extension selon les valeurs des attributs du second facteur. On peut objecter la possibilité que les attributs numériques soient discrétisés à dessein pour observer la répartition des objets selon certaines valeurs seuils. Une solution simple pour retrouver cette répartition dans une projection MDS est d'introduire des atomes indicateurs valués avec ces valeurs seuils. Dans notre exemple nous créons quatre indicateurs $i_{1...4}$ pour l'attribut *âge* avec $i_1.\hat{a}ge$=80, $i_2.\hat{a}ge$=65, etc. La projection MDS des objets et des indicateurs reflète à la fois les proximités des objets selon leur valeur sur *âge*, et leur répartition dichotomique autour de 65 tout en respectant la relation d'ordre sur $\mathbb{N}$ entre les valeurs seuils. La figure 6.3 illustre le résultat obtenu.

Dans les sous-sections suivantes, nous rappelons les échelles conceptuelles usuelles présentées dans [GW99] et précisons comment elles peuvent être représentées via des projections MDS.

6.2.1 Échelles nominale et dichotomique

Une échelle nominale est utilisée sur des attributs nominaux, i.e. dont les valeurs s'excluent les unes des autres (e.g. masculin, féminin, neutre). Le tableau 6.1(a) montre un contexte comprenant un attribut nominal N et le tableau 6.1(b) donne l'échelle nominale à appliquer sur cet attribut dont les valeurs possibles sont {a,b,c,d}. Le tableau 6.2(a) montre le contexte correspondant à l'échelle réalisée.

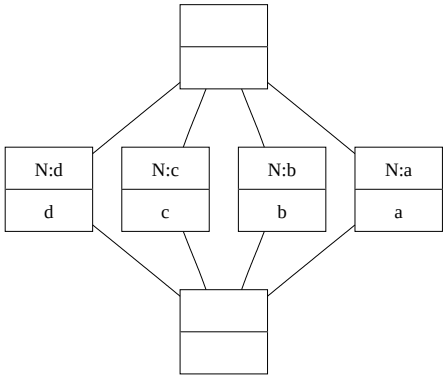
Les échelles nominales nous amènent à considérer des attributs multi-valués mais non nu-

$\mathbb{K}$	N
o_1	b
o_2	a
o_3	c
o_4	a
o_5	d

(a)

$\mathbb{S}_N$	N :a	N :b	N :c	N :d
a	×			
b		×		
c			×	
d				×

(b)

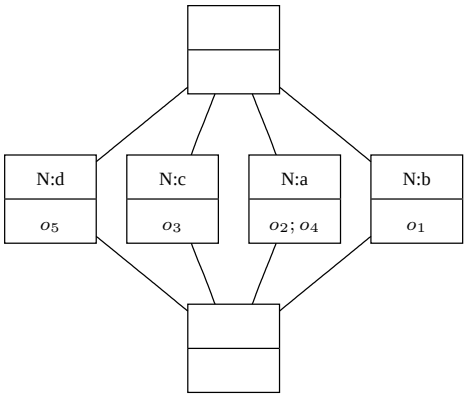


(c)

TAB. 6.1 – Contexte multivalué avec un attribut nominal N (a), échelle nominale $\mathbb{S}_N$ (b) et treillis de concepts associé à $\mathbb{S}_N$ (c).

$\mathbb{R}_N$	N :a	N :b	N :c	N :d
o_1		×		
o_2	×			
o_3			×	
o_4	×			
o_5				×

(a)



(b)

	N
o_1	33
o_2	0
o_3	66
o_4	0
o_5	100

(c)

TAB. 6.2 – Échelle réalisée $\mathbb{R}_N$ (a), treillis de concepts associé à $\mathbb{R}_N$ (b) et résultat du prétraitement de l'attribut N en vue d'une projection MDS (c).

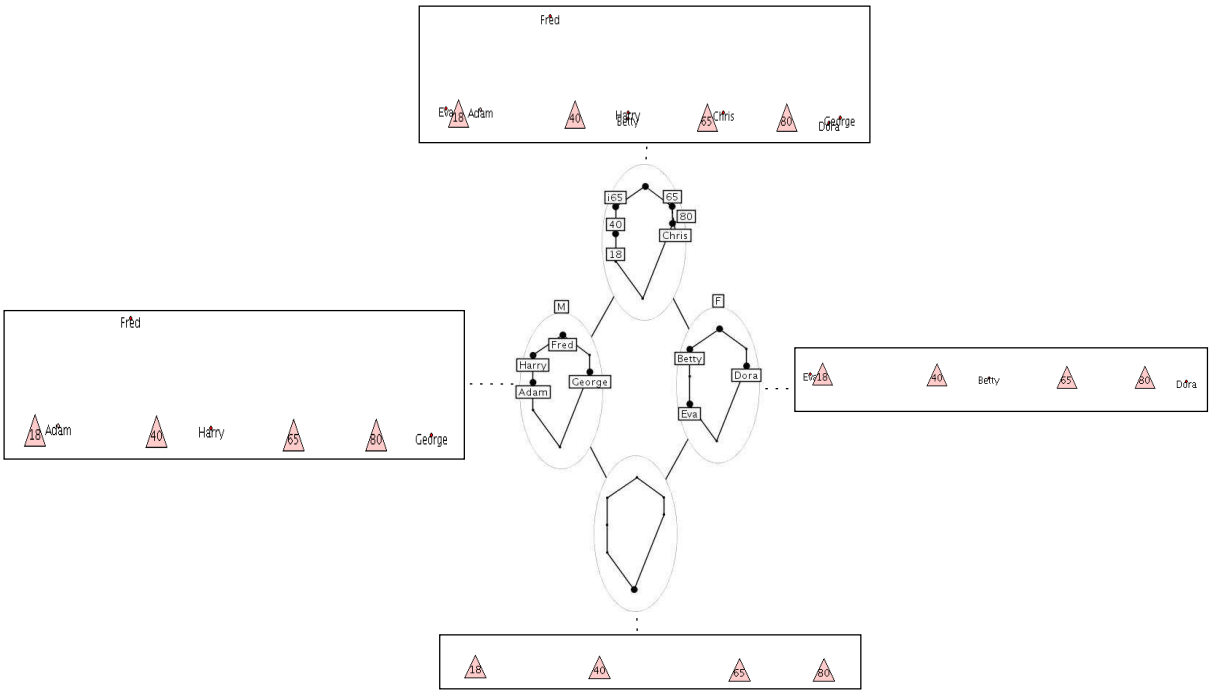


FIG. 6.3 – Les diagrammes encapsulés de la figure 6.2 sont replacées par des projections MDS dans lesquels les valeurs seuils sont représentées par des atomes indicateurs (triangles).

mériques. Un prétraitement est nécessaire pour projeter par MDS ces attributs. Une valeur numérique est associée à chaque valeur nominale, ainsi dans l'exemple du tableau 6.2, a=0, b=33, c=66 et d=100. Ces valeurs numériques remplacent les valeurs nominales pour les objets tel que le montre le tableau 6.2(c).

La projection MDS associée regroupe de manière effective les objets en quatre clusters puisque la distance euclidienne entre deux objets partageant la même valeur nominale (comme o_2 et o_4) est nulle. Pour une séparation optimale des objets en clusters, les valeurs numériques affectées aux valeurs nominales durant le prétraitement doivent être choisies en fonction du nombre de valeurs nominales différentes. Pour une projection des valeurs nominales sur une plage numérique allant de 0 à 100, les n valeurs numériques doivent être : $v_i = 100(\frac{i}{n})$ pour $i \in \{0 \dots n-1\}$. Notons que puisque les valeurs nominales ne sont pas ordonnées, l'indice i peut être affecté de manière aléatoire. Une échelle dichotomique est un cas particulier d'échelle nominale avec n=2.

6.2.2 Échelles ordinale et biordinale

Les échelles ordinales sont utilisées pour les attributs ordinaux, i.e. dont les valeurs sont ordonnées. Une échelle ordinale établit une hiérarchie entre les valeurs de l'attribut. On a ainsi l'implication *excellent* $\rightarrow$ *bien*. L'affectation d'un entier entre 0 et 100 à chacune des valeurs s'effectue de la même façon que pour les attributs nominaux, à la différence que les entiers affectés doivent refléter la relation d'ordre entre les valeurs. Ainsi, pour l'exemple décrit par le tableau 6.3, les entiers affectés aux valeurs de l'attribut ordinal *note* sont : *bien*=0, *très bien*=50 et *excellent*=100. Ainsi, on conserve *bien* $\leq$ *très bien* $\leq$ *excellent*.

Les échelles biordinales sont utilisées lorsque les valeurs d'un attribut sont séparées en deux

	note
o_1	bien
o_2	excellent
o_3	bien
o_4	très bien
o_5	bien

(a)

$\mathbb{S}_{note}$	note :excellent	note :très bien	note :bien
excellent	×	×	×
très bien		×	×
bien			×

(b)

	N
o_1	0
o_2	100
o_3	0
o_4	50
o_5	0

(c)

TAB. 6.3 – Contexte multivalué avec un attribut ordinal *note* (a), échelle conceptuelle $\mathbb{S}_note$ (b) et résultat du prétraitement de l'attribut *note* en vue d'une projection MDS (c).

pôles. Ainsi, un attribut dont les valeurs sont {*très lent, lent, rapide, très rapide*} suggère l'utilisation d'une échelle biordinale. En effet, l'application d'une échelle ordinale entraînerait l'implication *rapide* → *lent*, qui ne reflète pas la sémantique initiale des valeurs. Ces valeurs sont en fait réparties en deux pôles ordonnés : *lent* $\leq_1$ *très lent* d'une part, et *rapide* $\leq_2$ *très rapide* d'autre part. L'échelle biordinale consiste alors en deux échelles ordinales définies pour chacun des pôles. L'inconvénient de l'utilisation d'une échelle biordinale est la perte de l'ordre total qui existait par ailleurs sur les valeurs. En effet, même si les deux pôles ont chacun une sémantique particulière, les valeurs sont ordonnées par leurs niveaux relatifs de vitesse : *lent* $\leq_v$ *très lent* $\leq_v$ *rapide* $\leq_v$ *très rapide*. Or, cette information disparaît avec l'application d'une échelle biordinale qui ne retranscrit que les ordres partiels $\leq_1$ et $\leq_2$.

Revenons à l'exemple introduit dans la section 4.3. L'échelle conceptuelle appliquée à l'attribut *âge* et une échelle biordinale dont les deux pôles sont {18,40,65} d'une part, et {65,80} d'autre part (cf. tab. 4.5). Les deux pôles ordonnés apparaissent clairement sur le treillis autour de la valeur pivot 65. Cependant, l'ordre sur l'ensemble des valeurs n'est pas visible. L'avantage d'utiliser une projection MDS pour l'attribut *âge* est qu'une seule et même projection MDS contient les informations issues :

- de l'application d'une échelle ordinale sur *âge*,
- de l'application de l'échelle biordinale sur *âge* autour de la valeur pivot 65,
- de l'application des échelles biordinales sur *âge* autour des autres valeurs pivot possibles 18, 40 et 80.

En effet, la projection MDS conserve l'ordre total sur l'ensemble des valeurs, comme le ferait une échelle ordinale. De plus, la présence des valeurs-seuil 18, 40, 65 et 80, sous forme d'indicateurs, permet d'identifier facilement deux pôles autour d'une valeur-seuil. Les valeurs du pôle inférieur à 65 sont situées à gauche de l'indicateur 65, et celles du pôle supérieur à 65 son situées à sa droite, comme l'illustre la figure 6.4. Sur la même projection, on peut également observer les deux pôles autour de 40. Changer ainsi de valeur pivot aurait nécessité de générer une nouvelle échelle conceptuelle pour *âge* et de redessiner chaque diagramme encapsulé de la figure 6.2.

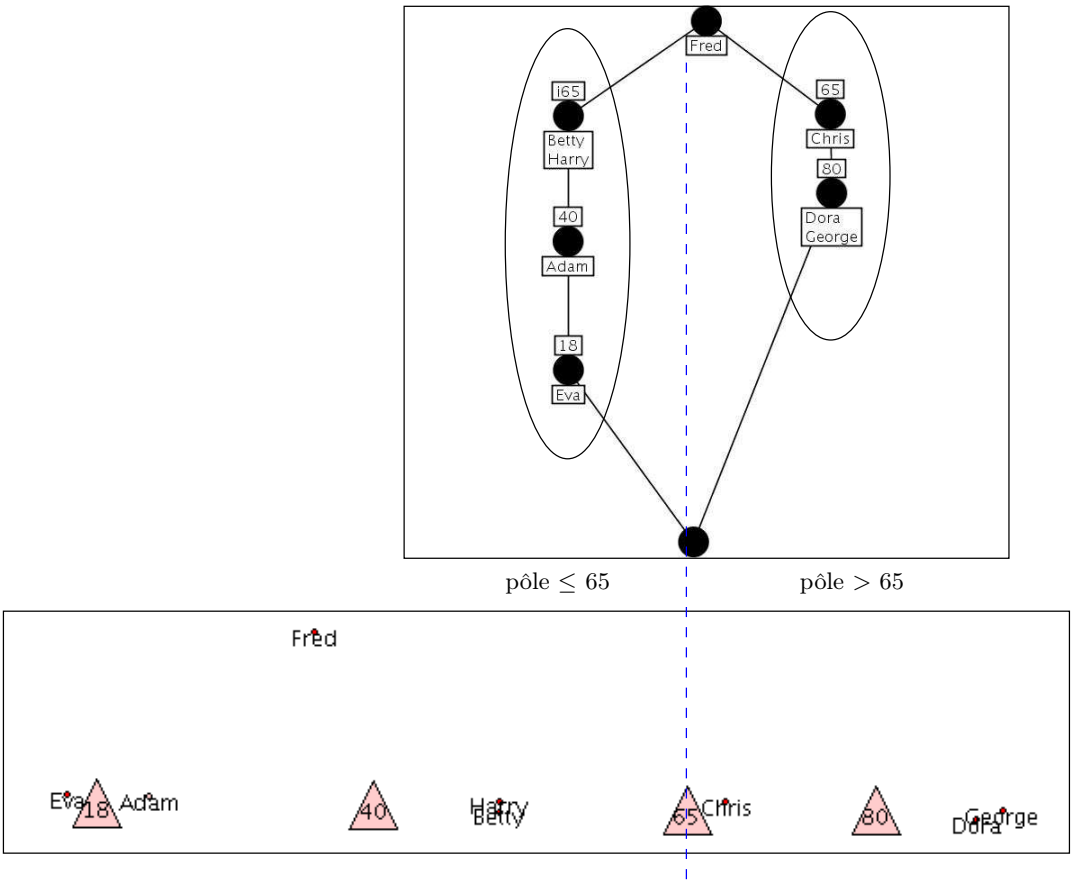


FIG. 6.4 – Lecture des pôles d'une échelle biordinale (haut) sur la projection MDS correspondante (bas). Un indicateur (triangle) représente chaque valeur seuil de l'échelle. L'échelle biordinale s'articule autour de la valeur pivot 65 qui définit deux pôles. Le pôle ≤ 65 (resp. > 65) se situe à gauche (resp. à droite) de l'indicateur 65 sur la projection MDS. De la même façon, on identifie visuellement les pôles autour des trois autres valeurs seuils. Changer ainsi de pivot nécessiterait de générer une nouvelle échelle biordinale. De plus, l'ordre total sur l'ensemble des valeurs des deux pôles est visible sur la projection MDS.

6.3 Interactions visuelles entre attributs non binaires

Dans les précédentes sous-sections, nous nous sommes intéressés aux projections MDS de seconds facteurs composés d'un seul attribut. Or les données brutes peuvent contenir plusieurs attributs non binaires. L'utilisateur peut vouloir observer les corrélations entre ces attributs. Utiliser des diagrammes enchevêtrés reviendrait à combiner plusieurs échelles conceptuelles qu'il faudrait recalculer pour chaque ajout ou retrait d'attribut dans le second facteur. À l'opposé, générer une projection MDS sur plusieurs attributs est très facile puisque MDS a justement été conçue pour gérer des données multidimensionnelles. Chaque objet, toujours représenté visuellement par un atome, est muni d'un vecteur dont chaque composante contient la valeur numérique d'un attribut non binaire. Notre outil de visualisation MOLAGE permet de sélectionner dynamiquement les composantes à prendre en compte dans le calcul de distance MDS entre objets (voir chapitre 3). L'utilisateur peut ainsi observer les proximités entre objets, les clusters et leurs

évolutions lorsqu'un attribut est sélectionné ou désélectionné.

L'exemple qui suit illustre notre méthode de visualisation et navigation pour contextes multivalués sur un exemple réel [AN07]. Les données sont constituées de 205 objets et 24 attributs et concernent les véhicules importés aux États-Unis. Tous les attributs sont non binaires : 14 sont numériques et 10 sont nominaux. Deux attributs nominaux *number-of-doors* et *number-of-cylinders* sont facilement numérisables. Nous avons finalement 16 attributs numériques et 8 attributs nominaux. Ces derniers seront utilisés comme premier facteur (le treillis associé sera utilisé comme vue globale) et les attributs numériques comme second facteur (projections MDS sélective sur ces attributs). La figure 6.5 illustre le principe d'interaction entre la vue globale et la vue locale, et la figure 6.6 les étapes de construction selon le modèle de Card-Chi.

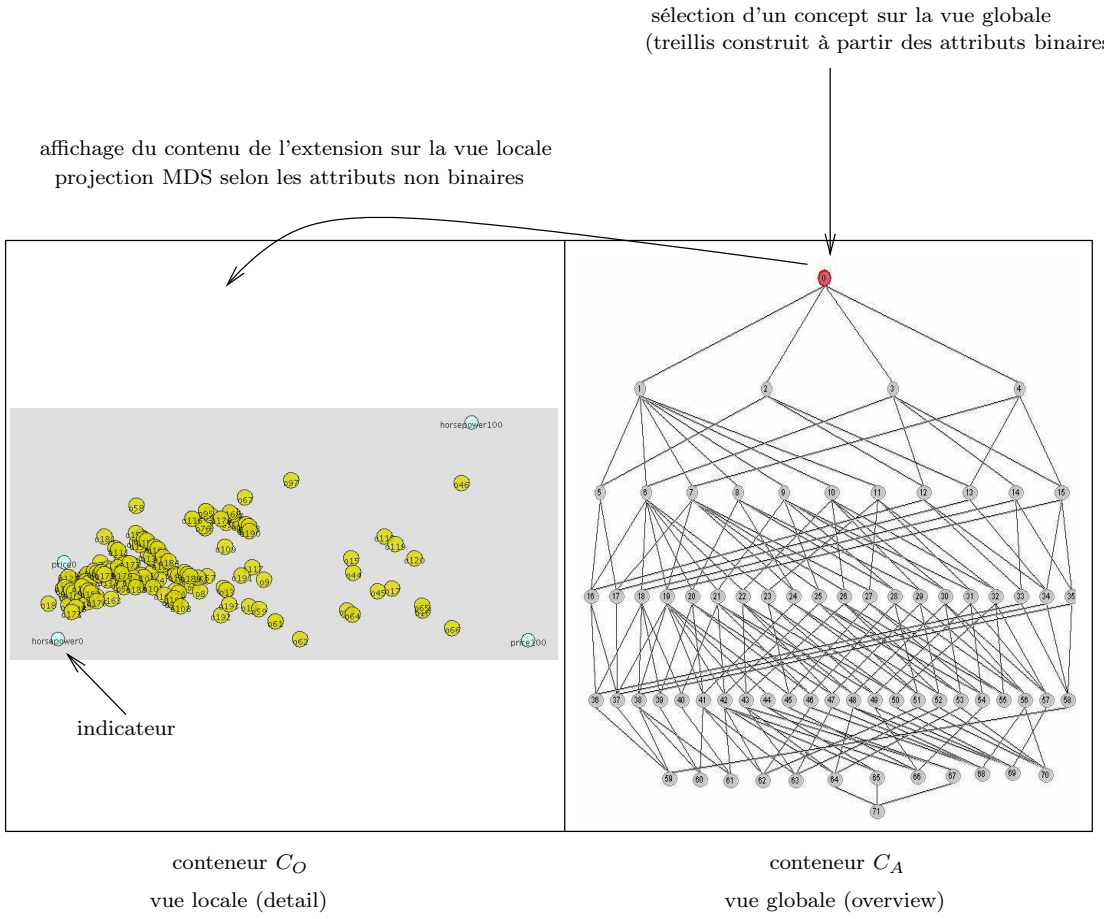


FIG. 6.5 – Principe d'interaction entre la vue globale et la vue locale.

6.3.1 Attributs du premier facteur et vue globale

Les attributs du premier facteur sont les 8 attributs nominaux tels que *name-of-constructor* ou *engine-location*. Une échelle nominale est appliquée à chacun de ces attributs. Le contexte dérivé contient 47 attributs binaires et le treillis associé 551 nœuds. Ceci illustre l'augmentation du nombre d'attributs binaires, et conséquemment du nombre de concepts du treillis, qu'entraîne le recours aux échelles conceptuelles. Comme nous l'avons dit précédemment, l'objectif de la vue

globale est de fournir une représentation de la structure de la base, aussi nous ne conservons que les concepts avec un support de 20% en extension (i.e. un concept est conservé si son extension contient au moins 20% du nombre d'objets total). Un treillis ainsi allégé est appelé treillis « iceberg » (cf. section 4.4.1). Le résultat obtenu est illustré par la figure 6.7.

6.3.2 Attributs du second facteur et vue locale

Les attributs du second facteur sont les 14 attributs numériques tels que *height*, *width*, *weight*, *horsepower*, *price*. Une projection MDS convient parfaitement pour comparer les objets sur ces attributs. Chaque attribut est normalisé entre 0 et 100 afin de maintenir un poids égal pour chaque dimension lors du calcul de la distance euclidienne. Rappelons que les axes n'ont pas de signification sur une projection MDS et que seules les distances entre objets sont signifiantes.

La figure 6.5 montre la vue locale du nœud racine. Tous les objets sont présents (tous les objets appartiennent au nœud racine) et l'utilisateur a choisi d'appliquer la projection MDS sur les attributs *horsepower* et *price*. Afin de pallier l'absence d'axes et d'observer la répartition des valeurs fortes et faibles sur *horsepower* et *price*, deux paires d'indicateurs sont introduits : les atomes *horsepower0*, *horsepower100*, *price0* et *price100*. *horsepower0* est valué uniquement sur *horsepower* avec la valeur 0, *horsepower100* avec la valeur 100, etc. La présence de ces indicateurs permet de sémantiser l'espace visuel et d'observer que les deux attributs *horsepower* et *price* sont globalement positivement corrélés. En effet, les indicateurs *horsepower0* et *price0* sont proches, ce qui signifie que les objets ayant une faible valeur sur l'un ont aussi une faible valeur sur l'autre. De plus, les indicateurs *horsepower100* et *price100*, même s'ils sont plus éloignés entre eux que les précédents, sont situés de telle sorte que l'on discerne pour les deux attributs une distribution similaire des valeurs, augmentant de la gauche vers la droite.

6.4 Corrélations entre attributs numériques

La figure 6.8(a) montre le même ensemble d'objets (nœud racine du treillis) projeté sur les trois attributs numériques *horsepower*, *price* et *wheelbase*. On observe visuellement que ce dernier est moins corrélé aux deux premiers que les deux premiers entre eux. Afin de quantifier ces indices visuels, on mesure la corrélation entre deux attributs a et b par l'angle α_{ab} formé par les vecteurs $\vec{a} = \overrightarrow{a_0 a_{100}}$ et $\vec{b} = \overrightarrow{b_0 b_{100}}$ [Sal89] (cf. fig. 6.8(b)) avec :

$$\alpha_{ab} = \arccos\left(\frac{\vec{a}.\vec{b}}{||\vec{a}||\,||\vec{b}||}\right)$$

Les attributs a et b sont positivement corrélés si $\alpha_{ab} = 0$, négativement corrélés si $\alpha_{ab} = \pi$ et non corrélés si α_{ab} prend les valeurs $\frac{\pi}{2}$ ou $\frac{3\pi}{2}$.

Cet indice de corrélation, mesuré à partir de la représentation visuelle des données, peut être confirmée grâce au coefficient de Pearson calculé à partir des données brutes. Le coefficient de Pearson $r_P(X,Y)$ entre deux variables $X(x_1,\ldots,x_n)$ et $Y(y_1,\ldots,y_n)$, où x_i désigne la valeur de X pour l'objet i et $\bar{x}$ la moyenne arithmétique de X, est défini par :

$$r_P(X,Y) = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$$

X/Y	Pearson	angle (Pearson)	angle (MDS)	écart
price/horsepower	0,81	0,63	0,68	+7,93%
price/wheelbase	0,58	0,95	0,99	+4,21%
horsepower/wheelbase	0,35	1,21	1,24	+2,48%

TAB. 6.4 – Comparaison de mesures de corrélation entre les attributs *price*, *horsepower* et *wheelbase*.

et prend la valeur 0 si X et Y ne sont pas corrélées, 1 si elles sont positivement corrélées, et -1 si elles sont négativement corrélées. D'autre part on a $r_P = \cos(\alpha_{XY})$ où α_{XY} désigne l'angle formé par les vecteurs multidimensionnels $\vec{X}$ et $\vec{Y}$. On peut donc estimer la fiabilité, en tant qu'indicateur de corrélation, de l'angle mesuré sur la projection MDS en le comparant avec la mesure de l'angle « théorique » calculé à partir du coefficient de Pearson. Le tableau 6.4 montre le résultat de cette comparaison.

Les angles mesurés sur la projection MDS semblaient indiquer que le couple d'attributs {*price, horsepower*} était le plus fortement corrélé. Cette intuition visuelle est confirmé par le coefficient de Pearson. De même, le couple {*horsepower, wheelbase*} paraissaient le moins corrélé avec un angle mesuré proche de $\frac{\pi}{2}$, intuition également confirmée. L'indice de corrélation fondé sur l'angle mesuré n'est cependant pas toujours fiable (8% d'écart pour {*price, horsepower*}) dans la mesure où l'on suppose une distribution linéaire des valeurs de l'attribut a sur le segment $[a_0 a_{100}]$. Or nous avons vu que les projections MDS ne permettaient pas de poser de telles hypothèses.

6.5 Projection MDS du premier facteur

Nous avons vu comment les attributs du premier facteur avaient été utilisés pour construire le treillis constituant la vue globale. On peut aussi s'intéresser à la visualisation de ces attributs nominaux par projection MDS, en utilisant les méthodes introduites dans ce chapitre à la section 6.2. Ainsi si l'on affecte des valeurs numériques aux deux valeurs que prend l'attribut *engine-location* : *rear* et *front* et que l'on projette les objets du nœud racine sur cet attribut, on obtient le résultat illustré par la figure 6.9. Deux clusters apparaissent clairement, correspondant à chaque valeur de l'attribut. Les utilisateurs peuvent voir du premier coup d'œil que bien plus de véhicule ont le moteur à l'avant.

6.6 Conclusion

Au cours de ce chapitre, deux contributions ont été présentées pour la prise en charge des attributs mixtes :

- une alternative aux diagrammes enchevêtrés par la représentation du second facteur via une projection MDS,
- une méthode de visualisation d'objets valués sur des attributs mixtes, fondée sur une séparation des rôles des attributs binaires, qui construisent la vue globale, et des attributs numériques, qui permettent d'observer les proximités entre objets de façon plus précise sur la vue locale.

Ces travaux ont fait l'objet d'une publication dans les actes de CLA 2007 [VRCC07] dont une version étendue a été acceptée pour publication dans un numéro spécial du *Journal of General Systems* consacré aux treillis de Galois et à leurs applications [VRCC08].

Jusqu'à présent les attributs constituaient le point d'entrée de la navigation, nous montrons dans le chapitre suivant l'adaptation de nos solutions au cas où ce point d'entrée est constitué par un sous-ensemble d'objets.

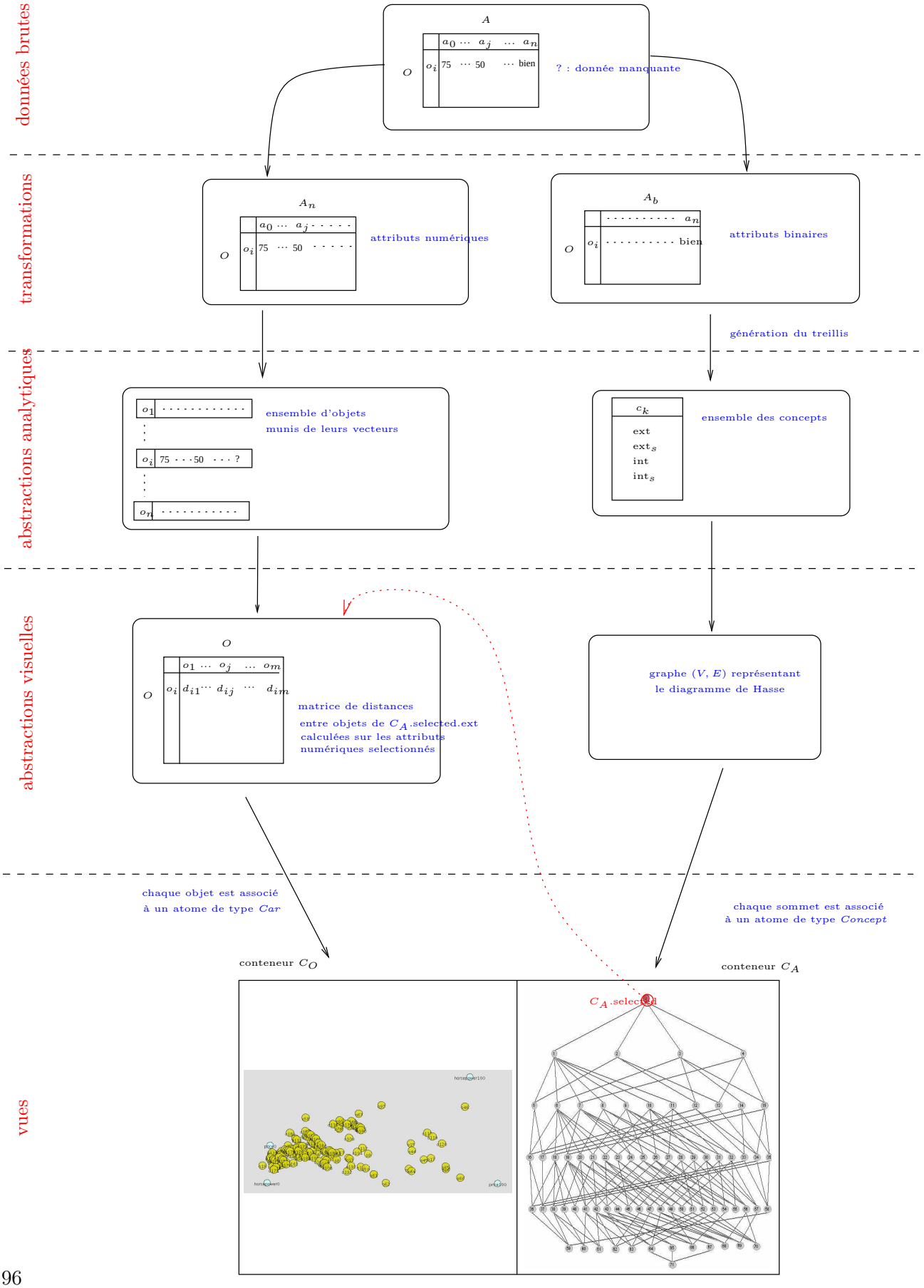
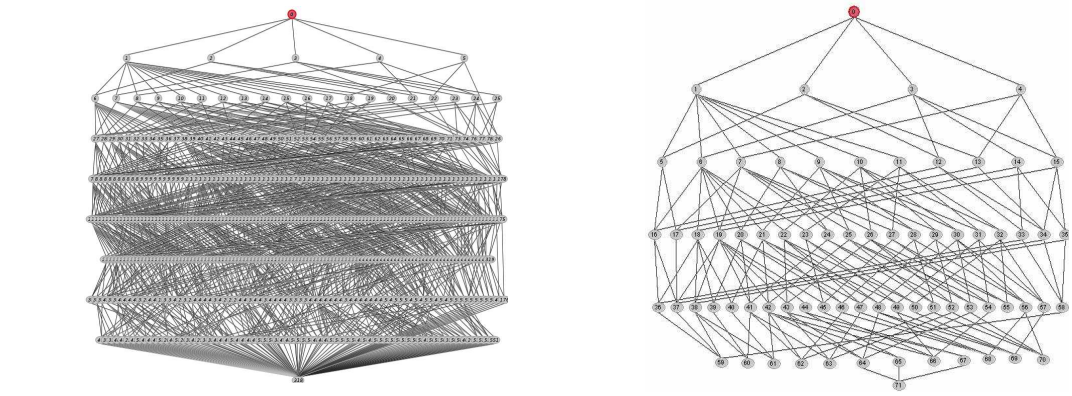
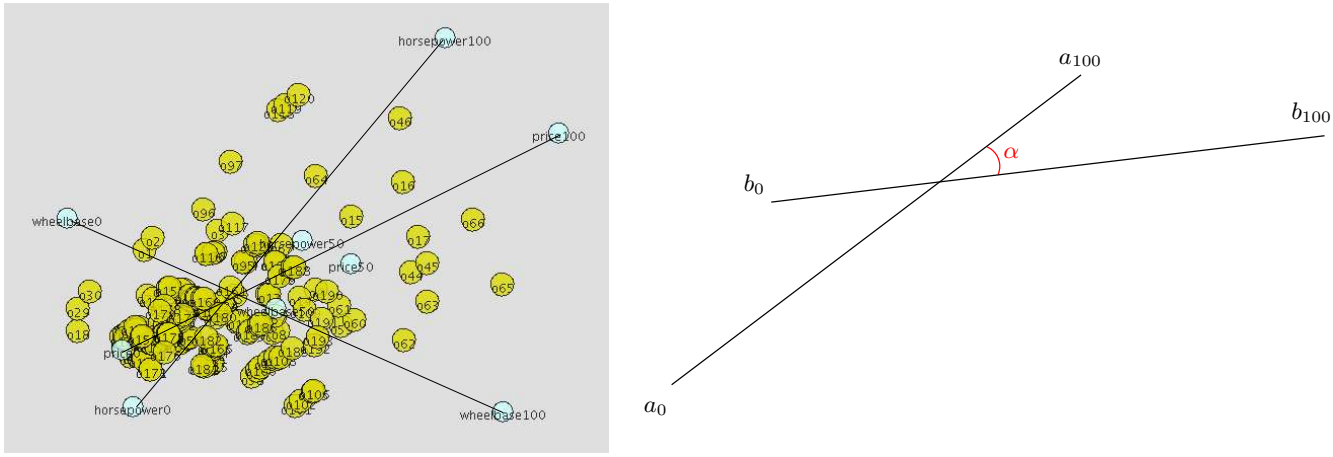


FIG. 6.6 – Étapes de construction selon le modèle de Card-Chi.



(a) (b)

FIG. 6.7 – Treillis de concepts complet (a) et treillis iceberg correspondant (support 20%) (b).



(a) (b)

FIG. 6.8 – Projection des objets de l'extension du nœud racine selon les attributs *horsepower*, *price* et *wheelbase* (a) α_{ab} est l'angle formé par les vecteurs $\vec{a}$ et $\vec{b}$ (b).

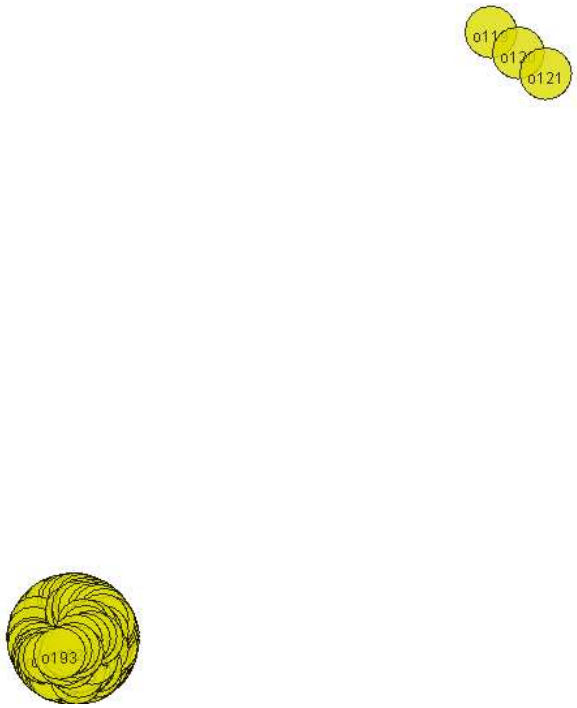


FIG. 6.9 – Projection des objets de l'extension du nœud racine selon l'attribut *engine-location*

7

Sélection d'attributs

Sommaire

DANS les chapitres précédents, nous avons présenté des méthodes permettant de visualiser des données caractérisées à la fois par un grand nombre d'objets et un grand nombre d'attributs. Nous avons vu comment réduire le nombre d'objets à afficher en adoptant une approche *overview + detail* dans laquelle seule une partie des objets est représentée dans une vue locale. Nous nous sommes également penchés sur la nécessité de réduire le nombre d'attributs sur lesquels les objets sont projetés, à cause des valeurs manquantes sur certains attributs. Cette réduction du nombre d'attributs revêtait alors un caractère purement utilitaire afin d'obtenir une projection MDS non biaisée. Elle peut pourtant se révéler utile dans une autre situation. Supposons qu'un utilisateur utilise notre approche *overview + detail* dans le but de naviguer parmi des données comportant des attributs hétérogènes. Selon la solution proposée au chapitre 6, l'utilisateur sélectionne un sous-ensemble d'attributs binaires, correspondant à l'intension d'un concept, sur la vue globale, puis observe le sous-ensemble d'objets possédant ces attributs binaires sur la vue locale. Cette solution répond au cas d'utilisation où les attributs binaires constituent le point d'entrée dans le processus de navigation. Toutefois, l'utilisateur peut souhaiter explorer les données à partir d'un sous-ensemble d'objets donné. S'il ne connaît pas les attributs binaires que ces objets partagent, notre solution ne lui permet pas d'identifier les concepts à sélectionner sur la vue globale pour observer ces objets. Nous exposons dans ce chapitre la mise en œuvre d'une technique de sélection d'attributs permettant, à partir d'un sous-ensemble d'objets, d'identifier ces sous-ensembles d'attributs binaires.

7.1 Sélection d'attributs : principes fondamentaux

La sélection d'attributs est couramment utilisée en apprentissage automatique comme un prétraitement permettant d'exclure les attributs non pertinents au regard d'une tâche d'appren-

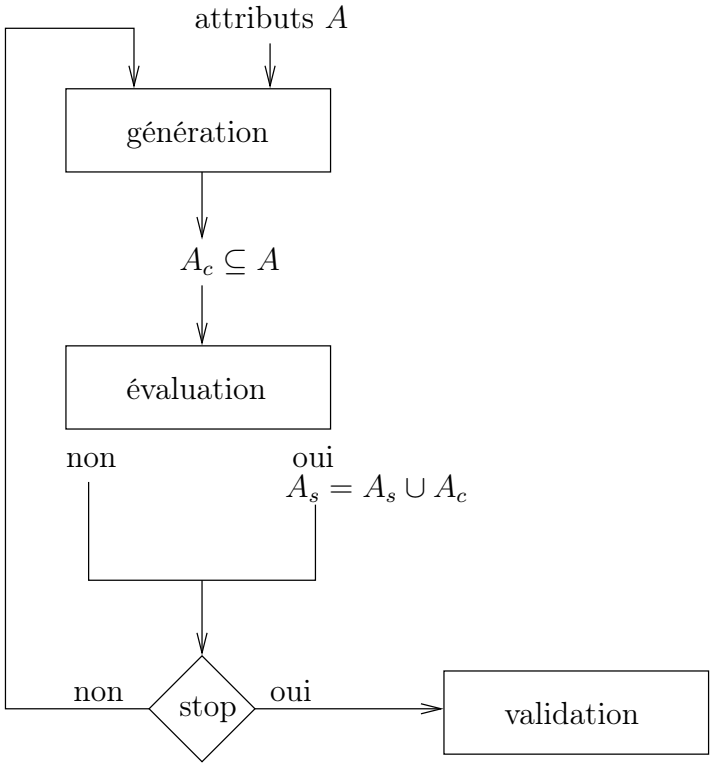


FIG. 7.1 – Processus de sélection d'un sous-ensemble d'attributs A_s parmi l'ensemble d'attributs A. À chaque itération, un ensemble candidat A_c est généré et évalué. La boucle se termine lorsque le critère d'arrêt est vérifié ou lorsqu'aucun nouveau candidat ne peut être généré. Les attributs sélectionnés A_s sont alors validés.

tissage à effectuer. Une description détaillée des diverses techniques de sélection d'attributs est présentée dans [GE03]. En particulier, IGLUE [NN97, NN98] est un système d'apprentissage à base d'instances utilisant des treillis de concepts pour accomplir une sélection d'attributs. Plus précisément, IGLUE génère, à partir des attributs binaires originaux pertinents, de nouveaux attributs numériques, plus adaptés à l'apprentissage à base d'instances. Notre objectif étant uniquement d'identifier les sous-ensembles d'attributs binaires pertinents, nous avons repris la phase d'identification des attributs binaires pertinents de IGLUE et l'avons adaptée à notre problématique. Nous revenons d'abord sur les différentes étapes d'une processus de sélection d'attributs.

Toutes les techniques suivent un même squelette qui résume le processus en quatre étapes :

1. génération d'un sous-ensemble d'attributs candidat,
2. évaluation du candidat,
3. évaluation du critère d'arrêt,
4. validation du sous-ensemble d'attributs sélectionné.

La figure 7.1 illustre le processus. Tout d'abord un ensemble d'attributs candidat A_c est généré selon la stratégie de recherche choisie ; la seconde étape évalue la pertinence de A_c puis le sélectionne ou le rejette. Si le critère d'arrêt n'est pas satisfait, un nouvel ensemble d'attributs candidat est généré et les étapes précédentes sont répétées. Le processus s'arrête lorsque le critère

(a)	age	spectacle prescription	astigmatism	tear-drop rate	decision
...	...	...	...	...	...
o_i	young	myope	no	reduced	no lenses
o_j	prepresbyopic	myope	no	reduced	no lenses
...	...	...	...	...	...

(b)	age :young	age :prepresbyopic	age :presbyopic	presc :myope	presc :hypermetrope	astigmatism :yes	astigmatism :no	tear-drop :normal	tear-drop :reduced	*decision :no lenses*
...	...	...	...	...	...	...	...	...	...	...
o_i	×			×			×		×	×
o_j		×		×			×		×	×
...	...	...	...	...	...	...	...	...	...	...

TAB. 7.1 – Extrait des données brutes de *Lenses* (a). Contexte dérivé après binarisation de l'attribut de classe *decision* et application d'échelles conceptuelles nominales (b).

d'arrêt est satisfait ou lorsque plus aucun nouveau candidat ne peut être généré. Une ultime étape facultative de validation permet généralement de classer les sous-ensembles sélectionnés.

Nous traitons ici de la sélection d'attributs comme prétraitement pour la classification. Notre sélection d'attributs est supervisée : l'appartenance des objets (des instances) aux classes est connue *a priori*. Plus précisément, et pour reprendre les termes de la question posée au début de ce chapitre, nous sommes en présence d'une seule classe dont les membres sont les objets initialement identifiés. Ces objets ont pu être identifiés en fonction des valeurs d'un attribut supplémentaire que nous appellerons attribut de classe.

7.2 Prétraitements et attribut de classe

Afin de fixer les esprits, nous nous reposerons tout au long de ce chapitre sur l'exemple du jeu de données *Lenses*, extrait de UCI Irvine [AN07]. Ce jeu de données contient 24 objets et quatre attributs nominaux. Un objet est un profil-type de patient décrit par les quatre attributs *age*, *spectacle prescription*, *astigmatism*, et *tear-drop rate*. Ces attributs décrivent les facteurs à prendre en compte dans le choix du type de verres de contact à prescrire. Un cinquième attribut nominal, *decision*, prend les valeurs *soft*, *hard* ou *no lenses* en fonction des valeurs sur les quatre premiers attributs. Le tableau 7.1(a) montre un extrait de ce jeu de données.

Nous nous donnons comme objectif de sélectionner, parmi les quatre premiers attributs, les combinaisons de valeurs conduisant à *no lenses* pour le cinquième attribut *decision*. Nous avons indiqué en préambule que nous souhaitions indiquer visuellement la localisation des attributs sélectionnés sur la structure de la base de données. Nous allons en effet utiliser le treillis de concepts associé à *Lens* et afficher les nœuds correspondant aux attributs sélectionnés. Le contexte formel

correspond aux quatre premiers attributs après application d'une échelle nominale sur chacun d'eux (cf. tab. 7.1(b)) et à la valeur de l'attribut de classe qui nous intéresse (*no lenses*) Les attributs bimodaux *astigmatism* et *tear-drop rate* pouvant être chacun codé par un seul attribut binaire, nous obtenons le contexte dérivé final du tableau 7.2.

7.3 Proposition

Les objets sont partitionnés en deux ensembles selon qu'ils possèdent ou non l'attribut binaire *no lenses*, les objets positifs O^+ qui le possèdent, et les objets négatifs O^- qui ne le possèdent pas. Nous proposons d'utiliser le treillis généré par le contexte dérivé final privé de l'attribut *no lenses* pour accomplir la sélection d'attributs en utilisant les intensions des concepts du treillis comme sous-ensembles candidats. L'attribut *no lenses* n'apparaît pas dans le treillis puisque l'on cherche justement à sélectionner les attributs qui lui sont liés, il ne peut donc pas faire partie des ensembles d'attributs candidats. Cependant l'information codée par *no lenses* est toujours présente au travers de la partition des objets en O^+ et O^-. La stratégie de recherche des sous-ensembles candidats consiste à parcourir le treillis par niveaux (*breadth-first traversal*) de $\top$ vers $\bot$, l'ensemble d'attributs candidat généré correspondant à l'intension du concept visité. Cette intension est évaluée en calculant l'entropie de Shannon [SW48] de l'extension du concept visité. L'entropie binaire sera minimale si tous les objets de l'extension sont soit tous positifs, soit tous négatifs. Si l'entropie est inférieure à un certain seuil et que les objets sont majoritairement positifs, l'intension du concept est sélectionné. Le critère d'arrêt est satisfait lorsque tous les concepts ont été traversés. Enfin les concepts sélectionnés sont coloriés sur le treillis présenté à l'utilisateur. Nous détaillons dans la suite chacune des étapes du processus.

7.3.1 Génération des sous-ensembles candidats

Considérant un contexte à n attributs, il existe 2^n sous-ensembles candidats potentiels. Une recherche exhaustive (génération des 2^n candidats) est donc fort coûteuse. Les stratégies permettant de réduire l'espace de recherche se répartissent en deux familles : complètes et séquentielles. Les stratégies complètes, comme *branch and bound*, assurent que tous les sous-ensembles optimaux seront explorés et proposés comme candidats. La taille de l'espace de recherche est toujours en $O(2^n)$ mais en pratique moins de sous-ensembles sont explorés. Les stratégies séquentielles, fondées pour la plupart sur l'approche gloutonne de l'algorithme du grimpeur (*greedy hill climbing approach*), explorent un espace de recherche en $O(n^2)$ ou moins mais la complétude n'est pas garantie. Afin d'éviter les optima locaux, une dose d'aléatoire peut être introduite dans les stratégies séquentielles. Quant à notre approche fondée sur le parcours du treillis, le nombre de sous-ensembles explorés est égal à la taille du treillis, i.e. $O(2^{\min(|A|,|O|)})$. Toutefois la taille du treillis étant inférieure en pratique, puisque seuls les sous-ensembles d'attributs pertinents au regard des sous-ensembles d'objets engendrent un concept, le nombre de sous-ensembles explorés est moindre.

7.3.2 Évaluation des sous-ensembles candidats

L'entropie binaire est utilisée pour évaluer la pertinence de l'intension du concept examiné. Elle mesure l'influence de l'intension sur la nature positive ou négative des objets de l'extension. Notons que cette évaluation ne tient pas compte du nombre d'objets en extension, permettant ainsi aux intensions de concepts proches de $\bot$ d'être sélectionnés. Formellement, l'entropie binaire

	age :young	age :prepresbyopic	age :presbyopic	presc :myope	presc :hypermetrope	astigmatism	tear-drop :reduced	*no lenses*
o_0	×			×			×	×
o_1	×			×				
o_2	×			×		×	×	×
o_3	×			×		×		
o_4	×				×		×	×
o_5	×				×			
o_6	×				×	×	×	×
o_7	×				×	×		
o_8		×		×			×	×
o_9		×		×				
o_{10}		×		×		×	×	×
o_{11}		×		×		×		
o_{12}		×			×		×	×
o_{13}		×			×			
o_{14}		×			×	×	×	×
o_{15}		×			×	×		×
o_{16}			×	×			×	×
o_{17}			×	×				×
o_{18}			×	×		×	×	×
o_{19}			×	×		×		
o_{20}			×		×		×	×
o_{21}			×		×			
o_{22}			×		×	×	×	×
o_{23}			×		×	×		×

TAB. 7.2 – Contexte dérivé final. L'attribut *no lenses* est utilisé comme attribut de classe.

du concept (O_i, A_i) est calculée comme suit :

$$H(O_i, A_i) = -\left(\frac{|O_i^+|}{|O_i|} \cdot \log_2\left(\frac{|O_i^+|}{|O_i|}\right) + \frac{|O_i^-|}{|O_i|} \cdot \log_2\left(\frac{|O_i^-|}{|O_i|}\right)\right)$$

où $|O_1^+|$ (resp. $|O_1^-|$) est le nombre d'objets positifs (resp. négatifs) de l'extension. Une entropie nulle survient lorsque les objets en extension sont soit tous positifs soit tous négatifs. L'objectif étant de sélectionner les attributs pertinents par rapport aux objets positifs (possédant l'attribut de classe *no lenses*), un concept (et donc le sous-ensemble d'attributs en intension) est dit optimal si la valeur de l'entropie binaire est inférieure à un seuil donné α est si les objets positifs représentent plus de la moitié de l'extension, soit : (O_i, A_i) est optimal si $H(O_i, A_i) \leq \alpha$ et $\frac{|O_i^-|}{|O_i|} < \frac{1}{2}$. Pour le présent exemple, nous posons $\alpha = 0$. Notons que si $H(O_i, A_i) = 0$ alors $\frac{|O_i^-|}{|O_i|} < \frac{1}{2} \Leftrightarrow O_i^- = \emptyset$. Le cas de $\alpha \neq 0$ sera discuté en fin de chapitre.

7.4 Exemple

Le treillis engendré par le contexte privé de l'attribut de classe *no lenses* (cf. tab. 7.2) comporte 50 concepts. Les objets positifs sont ceux qui possèdent l'attribut *no lenses*. La figure 7.2 montre le treillis sur lequel les nœuds carrés représentent les concepts optimaux. Durant le parcours par niveaux, le premier concept optimal identifié porte le numéro 3. Son intension est $A_3 = \{tear - drop : reduced\}$ et son extension O_3 contient douze objets positifs et un seul négatif. Son entropie binaire vaut donc :

$$H(O_3, A_3) = -\left(\frac{12}{12} \cdot \log_2\left(\frac{12}{12}\right) + \frac{0}{12} \cdot \log_2\left(\frac{0}{12}\right)\right) = 0$$

en prenant $O\log_2 0 = 0$ d'après la règle de L'Hôpital. Le fait que (O_3, A_3) est optimal peut être interprété ainsi : « seuls les objets positifs possèdent A_3 ». Dans le cas présent, cela signifie : « seuls les objets positifs possèdent $\{tear - drop : reduced\}$ » ou formellement :

$$\forall\, o \in O, \{tear - drop : reduced\} \in f(o) \Rightarrow o \in O^+$$

Si $O^+ - O_1 = \emptyset$, i.e. si tous les objets positifs appartiennent à l'extension du concept optimal, la réciproque est également vraie.

Un point important est que, grâce à la structure du treillis de concepts, tous les enfants d'un concept optimal sont aussi optimaux. Ainsi, considérant deux concepts $(O_j, A_j) \leq (O_i, A_i)$, si (O_i, A_i) est optimal alors $O_i \subseteq O^+$. Or $O_j \subseteq O_i$ d'après $\leq$ d'où $O_j \subseteq O^+$. Lorsqu'un concept optimal est identifié, la propriété précédente permet de retirer tous ses enfants de l'espace de recherche. Notons que cette propriété est uniquement vraie pour un seuil d'entropie $\alpha = 0$. Le treillis de la figure 7.2 répond bien à la question initiale « où se situent les objets préalablement identifiés dans la structure de la base et quels sont les attributs pertinents associés ? ».

7.5 Interprétation et implications

Les utilisateurs peuvent observer visuellement la répartition des objets positifs en fonction de la structure de la base, qui peut également servir à naviguer dans cette même base (cf. chapitres précédents). Les concepts optimaux peuvent également engendrer des implications (cf. section 4.2) entre leurs intensions et l'appartenance à la classe considérée. Ces implications ont

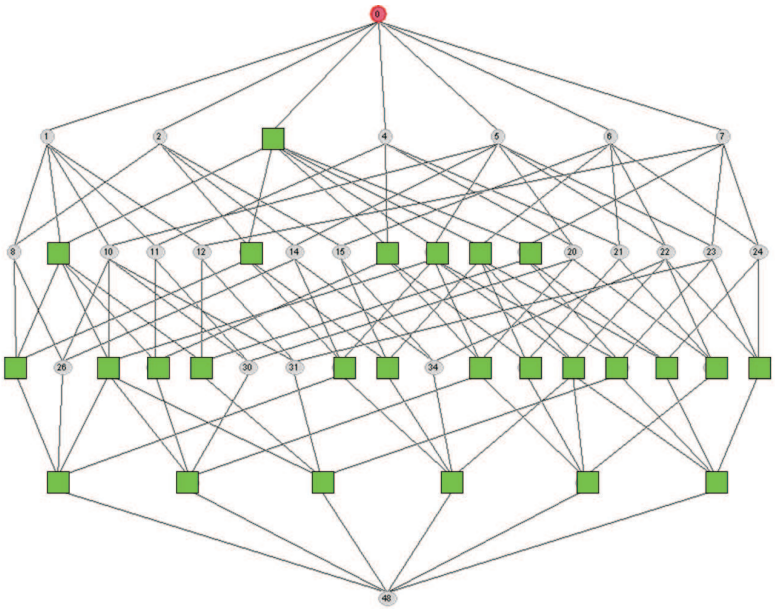


FIG. 7.2 – Treillis de concepts généré à partir du contexte dérivé final. Les nœuds carrés représentent les concepts formels optimaux dont les intensions forment les sous-ensembles d'attributs sélectionnés.

une confiance maximale puisque tous les objets en extension des concepts optimaux sont positifs. Leur support est le cardinal de l'extension. Notons que grâce au diagramme de Hasse, les utilisateurs peuvent évaluer ce support en fonction de la position relative du concept dans le treillis. Ainsi, considérant deux concepts optimaux $(O_j, A_j) \leq (O_i, A_i)$ et leurs implications $c_j : A_j \rightarrow class$ et $c_i : A_i \rightarrow class$, alors $support(c_j) \leq support(c_i)$ puisque $|O_j| \leq |O_i|$. Remarquons également que c_j est redondante avec c_i puisque $A_j \subseteq A_i$.
Puisque les enfants d'un concept optimal sont aussi optimaux (avec un seuil d'entropie $\alpha = 0$), lorsqu'un concept optimal apparaît parmi les fils directs de $\top$, comme dans le cas présent, le treillis peut être surchargé de nœuds carrés redondants. Il n'est pas aisé alors de séparer ces nœuds redondants de ceux qui ne sont pas fils d'un concept optimal. Nous proposons de colorier uniquement les nœuds non redondants, i.e. qui ne sont pas fils d'un concept optimal (cf. fig. 7.3). Seuls trois nœuds carrés demeurent : un numéroté 3 dont l'intension est {*tear-drop :reduced*} et deux nœuds numérotés 40 et 41 dont les intensions sont respectivement {*age :pre-presbyopic, astigmatism, prescription :hypermetrope*} et {*age :presbyopic, astigmatism, prescription :hypermetrope*}. Ces deux derniers nœuds étaient auparavant noyés parmi les nœuds redondants (cf. fig. 7.2). Notre processus de sélection d'attributs fournit visuellement les résultats suivants : les profils-types pour lesquels les verres de contact sont contrindiqués sont ceux qui ont un taux lacrimal trop faible, ou ceux qui ont la combinaison de facteurs correspondant aux nœuds 40 et 41.

7.6 Conclusion

Dans ce chapitre, nous avons présenté une méthode permettant, à partir d'un ensemble d'objets donné, de présenter à l'utilisateur les concepts qui lui permettront, à partir de la vue globale, d'observer ces objets sur la vue locale. Cette sélection d'attributs reprend le principe

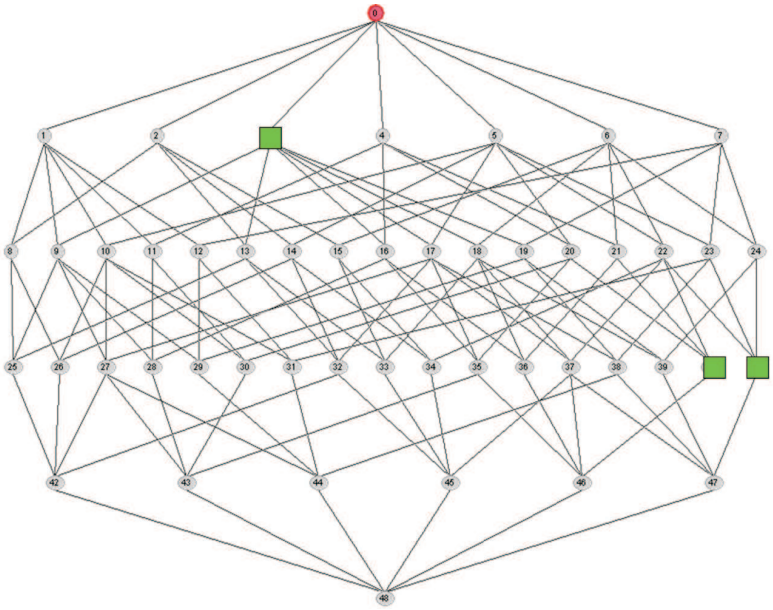


FIG. 7.3 – Les carrés redondants ont été retirés du treillis de la figure 7.2.

introduit dans IGLUE en l'adaptant à notre objectif et en se servant des propriétés visuelles du treillis pour comparer les positions relatives des concepts optimaux. Les travaux décrits dans ce chapitre ont fait l'objet d'une publication en article court dans les actes complémentaires (*supplementary proceedings*) de ICCS 2008 [VRC08].

Conclusion et perspectives

Conclusion

Les travaux présentés dans ce mémoire concernent la visualisation de données. Ce domaine est devenu un enjeu majeur face à l'accroissement considérable du volume et de la complexité des données que peut contenir ou auxquelles peut accéder tout ordinateur personnel. Autrefois circonscrite à un public restreint de gros consommateurs ou producteurs de données, tels que les statisticiens, la visualisation se retrouve aujourd'hui en première ligne dans le combat à mener contre le « fléau de la dimension ». En effet, seules des formes de représentations visuelles peuvent offrir une vue générale de ces ensembles de données volumineux. Les tableaux et listes de résultats ne sont plus adaptés à la navigation dans des collections de documents numériques – textes, images, sons – pouvant aisément atteindre plusieurs centaines de gigaoctets en ne considérant que les ordinateurs personnels. De plus, la recherche d'information dans ces grands volumes de documents repose sur des mécanismes d'indexation de plus en plus sophistiqués. Chaque document se trouve indexé sur un ensemble d'attributs de natures différentes, ajoutant au volume la complexité. Celle-ci est encore accrue sur le plan structurel par les diverses relations pouvant exister entre documents et entre attributs. Le développement des techniques de visualisation de données n'a malheureusement pas connu la même croissance fulgurante. De nombreux résultats ont été obtenus sur des types de données et de structures précis, apportant des solutions *ad hoc* à des problèmes restreints. Toutefois, le rôle prépondérant que la visualisation est amenée à jouer nécessite un saut qualitatif dans la manière de concevoir de nouvelles techniques de visualisation, de les spécifier et de les expérimenter. Ce « fossé de crédibilité » dont souffre la communauté ne peut se résoudre que par un effort de formalisation. De nombreux travaux ont été entrepris de ce sens, ouvrant la voie à la résolution formalisée de problèmes ouverts.

Nous nous sommes concentrés sur les problèmes de visualisation liés à la nature hétérogène des attributs et des structures de données.

Après avoir introduit la problématique du « fléau de la dimension » au cours du chapitre 1, nous nous sommes penchés sur l'historique de la visualisation de données dans le chapitre 2. Nous avons complété cet état de l'art par la présentation d'un nouvel enjeu pour la communauté : l'effort de formalisation nécessaire afin de résoudre le « fossé de crédibilité ». Dans cette perspective, le modèle de Card-Chi fournit une taxonomie des différents types de structures de données et précise les étapes du processus de visualisation. Nous avons repris les termes de cette taxonomie pour décrire les problèmes qui nous ont été posés dans le cadre de deux projets de recherche nécessitant la visualisation d'une collection musicale, pour l'un, et d'une base documentaire, pour l'autre.

Le chapitre 3 expose les solutions de visualisation que nous avons conçues dans le cadre de ces deux projets de recherche dont les données manipulées avaient en commun l'hétérogénéité de la nature de leurs attributs et de leurs structures. Nous avons d'abord présenté l'environnement de visualisation Molage fondé sur le modèle de forces FDP, et implémentant la technique de réduction multidimensionnelle MDS. D'autres types de forces peuvent être combinées afin de structurer la visualisation. Nous avons caractérisé ces dispositifs de manière formelle et avons utilisé cette formalisation dans la spécification des solutions proposées. Nous avons notamment spécifié les appariements entre données et entités visuelles d'une part, et la séquence de forces à activer pour réaliser une représentation de données à la structure hybride, à la fois tabulaire et de type graphe. Au terme de ce chapitre, nous avons récapitulé les verrous non encore résolus. Tout d'abord les solutions présentées dans ce chapitre ne conviennent pas à un volume important

d'objets affichés simultanément. Nous nous sommes alors orientés vers une approche *overview + detail* comprenant une vue globale reflétant la structure générale des données, et une vue locale représentant de manière détaillée un sous-ensemble d'objets sélectionné à partir de la vue globale. La présence de données manquantes constitue un autre verrou et nous avons montré que l'approche MDS peut être biaisée dans ce cas. Enfin concernant l'hétérogénéité des attributs, bien que des solutions aient été présentées dans le cadre de la collection musicale et de la base documentaire, elle n'est pas résolue dans le cas général. Les solutions apportées à ces verrous sont présentées dans les chapitres suivant et mettent en œuvre des techniques issues de l'analyse de concepts formels (FCA). Le chapitre 4 introduit les techniques de FCA utilisées dans la suite.

Le chapitre 5 présente la solution apportée au problème des données manquantes. Nous considérons des données tabulaires constituées d'attributs numériques pour lesquels des valeurs sont manquantes. Suivant notre approche *overview + detail*, la vue globale représente un treillis dans lequel un concepts (O, A) contient un sous-ensemble d'objets O possédant une valeur pour tous les attributs de A. Le treillis représente ainsi toutes les combinaisons de projections MDS non biaisées possibles. Lorsque l'utilisateur sélectionne un concept du treillis, la vue locale affiche les objets O projetés par MDS afin de représenter leurs similarités au regard des attributs A. Ces combinaisons de projection MDS non biaisées étant ordonnées grâce au treillis, l'utilisateur peut naviguer de manière progressive et cohérente parmi ces combinaisons.

Le chapitre 6 expose la solution apportée au problème de l'hétérogénéité des attributs. Nous présentons dans un premier temps les *nested-line diagrams* et les échelles conceptuelles, solutions apportées par l'approche FCA pour prendre en compte des attributs non binaires et représenter les treillis générés. Nous montrons en quoi la représentation par treillis encapsulés des *nested-line diagrams* peut induire des erreurs d'interprétation et proposons une alternative basée sur la substitution de projections MDS aux treillis encapsulés. Ceci nous amène à présenter notre solution consistant en une vue globale représentant le treillis construit à partir des attributs binaires, et une vue locale affichant les objets en extension par projection MDS sur les attributs non binaires.

Enfin, le chapitre 7 montre comment, à partir d'un sous-ensemble d'objets, sélectionner les sous-ensembles d'attributs binaires permettant de caractériser ce sous-ensemble d'objets. Les concepts correspondant aux sous-ensembles d'attributs identifiés sont affichés sur le treillis de la vue globale, et constituent des points d'entrée pour la navigation.

Perspectives

Automatisation des scénarios d'organisation dans Molage

La formalisation des entités et des dispositifs visuels présents dans Molage introduite dans le chapitre 3 s'inscrit dans la perspective d'une spécification formelle des représentations visuelles de données. Au cours de ce même chapitre, nous avons ainsi spécifié deux scénarios d'organisation sous la forme d'une séquence d'activation de forces. Des patrons de scénarios peuvent être élaborés en fonction du type de données à représenter. Combinée à notre méthode de conception fondée sur la spécification des appariements entre objets du domaine à représenter et entités visuelles, présentée dans [CRV+06b], cette formalisation ouvre la perspective de scénarios d'organisation décrits sous forme de scripts pouvant être déclenchés de manière automatique.

Dessin du treillis des objets concepts par MDS

Nous avons montré dans la section 5.4.1 comment, à partir des concepts d'un treillis, dessiner le diagramme de Hasse associé dans MOLAGE en suivant l'agorithme de Freese. Cette approche nécessite d'avoir préalablement calculé les concepts du treillis. Or, dans le cas d'objets décrits par des attributs binaires, en transformant un attribut binaire a_b en attribut numérique a_n avec $a_n = 100$ si l'objet possède a_b et 0 sinon, la projection MDS révèle des clusters d'objets correspondant aux concepts-objets (cf. section 4.4.2). En effet, les objets partageant exactement le même sous-ensemble d'attributs se regroupent. En restreignant la projection MDS à un sous-ensemble d'attributs $A_0 \subseteq A$, on obtient les concepts-objets du sous-contexte correspondant au contexte d'origine privé des attributs $A \setminus A_0$. Cette découverte est intéressante car elle ouvre la perspective d'une représentation visuelle du treillis des concepts-objets émergeant directement des objets sans calculs préalables.

Les travaux de recherche présentés dans cette thèse ont abouti, à partir de trois problèmes de visualisation – volume des données, hétérogénéité des attributs et des structures – à deux contributions. La première s'inscrit dans la démarche de formalisation entreprise par la communauté. Nous avons proposé un cadre formel pour la spécification de visualisations par FDP guidées par les données et l'avons mis en œuvre dans notre environnement MOLAGE. La seconde consiste en l'utilisation de techniques d'analyse de données formelles FCA dans le cadre de la visualisation de données hétérogènes. Les représentations visuelles ont pour but de révéler les structures et les tendances générales des données. Nous avons montré comment les techniques FCA permettent d'extraire ces structures et de fiabiliser les similarités observées par MDS.
Au delà de ces résultats, nous aurions souhaité aller plus loin et, de façon formelle, faire émerger de ces représentations visuelles des informations nouvelles que n'auraient pu extraire les techniques d'analyse de données traditionnelles. Malgré nos efforts en ce sens, les résultats auxquels nous sommes arrivés n'ont pas été probants. Mais, comme le rappelle Robert Spence dans [Spe01], le but de la visualisation d'information est essentiellement de fournir une image mentale exploitable à l'utilisateur. L'usage qui en est fait, en particulier la production de nouvelles connaissances, relève de l'activité cognitive humaine, pas de celle d'un ordinateur.
Nous espérons que les résultats que nous avons atteints contribuent de manière pertinente à la réalisation de cet objectif.

Bibliographie

[AA07] G. Andrienko and N. Andrienko. Coordinated Multiple Views : a Critical View. In *Proceedings of the Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization*, pages 72–74. IEEE Computer Society Washington, DC, USA, 2007.

[AN07] A. Asuncion and D.J. Newman. UCI Machine Learning Repository, 2007.

[And73] M.R. Anderberg. *Cluster analysis for applications*. Probability and Mathematical Statistics, New York : Academic Press, 1973.

[Bas00] W. Basalaj. Proximity Visualisation of Abstract Data. Technical report, University of Cambridge, 2000.

[BC87] R.A. Becker and W.S. Cleveland. Brushing Scatterplots. *Technometrics*, 29(2) :127–142, 1987.

[Bel57] R. Bellman. Dynamic Programming, 1957.

[Ber67] J. Bertin. *Sémiologie graphique*. Mouton, Paris, 1967.

[BG97] I. Borg and P.J.F. Groenen. *Modern multidimensional scaling*. Springer New York, 1997.

[BG03] I. Borg and P. Groenen. Modern Multidimensional Scaling : Theory and Applications. *Journal of Educational Measurement*, 40(3) :277–280, 2003.

[BSL+01] A. Buja, D.F. Swayne, M. Littman, N. Dean, and H. Hofmann. XGvis : Interactive Data Visualization with Multidimensional Scaling. *Journal of Computational and Graphical Statistics*, pages 1061–8600, 2001.

[BWK00] M.Q.W. Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the working conference on Advanced visual interfaces*, pages 110–119. ACM Press New York, NY, USA, 2000.

[Cha96] M. Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. In *Proceedings of the IEEE conference on Visualization'96*. IEEE Computer Society Press, 1996.

[Che73] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68(342) :361–367, 1973.

[Che05] C. Chen. Top 10 Unsolved Information Visualization Problems. *IEEE Computer Graphics and Applications*, pages 12–16, 2005.

[Chi00] E.H. Chi. A taxonomy of visualization techniques using the data state reference model. In *Proceedings of the 2000 IEEE Symposium on Information Visualization*, pages 69–75, 2000.

[Cle93] W.S. Cleveland. *Visualizing Data*. Hobart Press, 1993.

[CMS99] S.K. Card, J.D. Mackinlay, and B. Shneiderman. *Readings in information visualization : using vision to think*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1999.

[CR98] E.H. Chi and J.T. Riedl. An Operator Interaction Framework for Visualization Systems. *Proceedings of the 1998 IEEE Symposium on Information Visualization*, pages 63–70, 1998.

[CR04] C. Carpineto and G. Romano. Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO. *Journal of Universal Computer Science*, 10(8) :985–1013, 2004.

[CRV+06a] M. Crampes, S. Ranwez, F. Velickovski, C. Mooney, and N. Mille. An integrated visual approach for music indexing and dynamic playlist composition. *Proceedings of SPIE*, 6071 :103–118, 2006.

[CRV+06b] M. Crampes, S. Ranwez, J. Villerd, F. Velickovski, C. Mooney, A. Emery, and N. Mille. Concept maps for designing adaptive knowledge maps. *Information Visualization*, 5(3) :211, 2006.

[CVER07] M. Crampes, J. Villerd, A. Emery, and S. Ranwez. Automatic playlist composition in a dynamic music landscape. In *Proceedings of the 2007 international workshop on Semantically aware document processing and indexing*, pages 15–20. ACM Press New York, NY, USA, 2007.

[DBETT94] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis. Algorithms for drawing graphs : an annotated bibliography. *Computational Geometry : Theory and Applications*, 4(5) :235–282, 1994.

[DDE08] F. Dau, J. Ducrou, and P. Eklund. Concept Similarity and Related Categories in SearchSleuth. In *Proceedings of the 16th International Conference on Conceptual Structures (ICCS'08)*, volume 5113 of *LNCS - LNAI*, pages 255–268. Springer, 2008.

[DE07] J. Ducrou and P. Eklund. SearchSleuth : The Conceptual Neighbourhood of an Web Query. In *Proceedings of the 5th International Conference on Concept Lattices and Their Applications (CLA'07)*, volume 331, pages 253–263. CEUR Workshop Proceedings, 2007.

[Dun84] O.D. Duncan. *Notes on social measurement : historical and critical*. Russell Sage Foundation, New York, 1984.

[DVE06] J. Ducrou, B. Vormbrock, and P. Eklund. FCA-Based Browsing and Searching of a Collection of Images. In *Proceedings of the 14th International Conference in Conceptual Structures (ICCS'06)*, volume 4068 of *LNCS*, pages 203–214. Springer, 2006.

[Ead84] P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42 :149–160, 1984.

[EDB04] P. Eklund, J. Ducrou, and P. Brawn. Concept lattices for information visualization : Can novices read line diagrams. In *Proceedings of the 2nd International Conference on Formal Concept Analysis (ICFCA)*. Springer, 2004.

[EDF08] N. Elmqvist, P. Dragicevic, and J.D. Fekete. Rolling the Dice : Multidimensional Visual Exploration using Scatterplot Matrix Navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6) :1539–1148, 2008.

[EP88] B. Escofier and J. Pagès. *Analyses factorielles simples et multiples : objectifs, méthodes et interprétation*. Dunod, 1988.

[FD02] M. Friendly and D.J. Denis. Milestones in the History of Thematic Cartography, Statistical Graphics, and Data Visualization. 2002.

[FK95] A. Formella and J. Keller. Generalized Fisheye Views of Graphs. In *Proceedings of the Symposium on Graph Drawing*, pages 242–253. Springer-Verlag London, UK, 1995.

[Fre04] R. Freese. Automated Lattice Drawing. In *Proceedings of the 2nd International Conference on Formal Concept Analysis (ICFCA 2004)*. Springer, 2004.

[Fri06] M. Friendly. A brief history of data visualization. In C. Chen, W. Härdle, and A. Unwin, editors, *Handbook of Computational Statistics : Data Visualization*, volume 3. Springer-Verlag, Heidelberg, 2006.

[FSvH06] C. Fluit, M. Sabou, and F. van Harmelen. Ontology-Based Information Visualization : Toward Semantic Web Applications. *Visualizing the Semantic Web : Xml-based Internet And Information Visualization*, 2006.

[Fur86] G.W. Furnas. Generalized fisheye views. In *CHI '86 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 16–23, New York, NY, USA, 1986. ACM.

[FWR99] Y.H. Fua, M.O. Ward, and E.A. Rundensteiner. Navigating hierarchies with structure-based brushes. In *Proceedings of the 1999 IEEE Symposium on Information Visualization (Info Vis' 99)*, pages 58–64, 1999.

[GE03] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(7-8) :1157–1182, 2003.

[GL86] J.C. Gower and P. Legendre. Metric and Euclidean properties of dissimilarity coefficients. *Journal of Classification*, 3(1) :5–48, 1986.

[GM93] R. Godin and H. Mili. Building and Maintaining Analysis-Level Class Hierarchies Using Galois Lattices. In *Proceedings of the OOPSLA'93 Conference on Object-oriented Programming Systems, Languages and Applications*, pages 394–410, 1993.

[Gow71] J.C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 27(4) :857–871, 1971.

[GPG89] R. Godin, C. Pichet, and J. Gecsei. Design of a browsing interface for information retrieval. In *Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 32–39. ACM Press New York, NY, USA, 1989.

[GW99] B. Ganter and R. Wille. *Formal concept analysis*. Springer New York, 1999.

[GYG05] A.G. Gee, M. Yu, and G.G. Grinstein. Dynamic and Interactive Dimensional Anchors for Spring-Based Visualizations. Technical report, computer science, University of Massachussetts Lowell, 2005.

[Han96] D.J. Hand. Statistics and the theory of measurement. *Journal of the Royal Statistical Society A*, 159 :445–492, 1996.

[HDL00] M. Huchard, H. Dicky, and H. Leblanc. Galois lattice as a framework to specify building class hierarchies algorithms. *Theoretical Informatics and Applications*, 34(6) :521–548, 2000.

[HMM00a] I. Herman, M.S. Marshall, and G. Mélançon. Density functions for visual attributes and effective partitioningin graph visualization. In *Proceedings of the 2000 IEEE Symposium on Information Visualization (Info Vis' 00)*, pages 49–56, 2000.

[HMM00b] I. Herman, G. Mélançon, and M.S. Marshall. Graph Visualization and Navigation in Information Visualization : A Survey. *IEEE Transactions on Visualization and Computer Graphics*, pages 24–43, 2000.

[HP05] N. Hernandez and S. Poulain. Customizing information access according to domain and task knowledge : the ontoExplo system. *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 607–608, 2005.

[Ins97] A. Inselberg. Multidimensional detective. *IEEE Symposium on Information Visualization*, pages 100–107, 1997.

[Jac01] P. Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bull. Soc. Vaudoise Sci. Nat*, 37 :547–579, 1901.

[JKKK+06] T.J. Jankun-Kelly, R. Kosara, G. Kindlmann, C. North, C. Ware, and W. Bethel. Is there Science in Visualization. *IEEE Visualization Conference Compendium*, pages 68–71, 2006.

[JMM+] C. R. Johnson, R. Moorhead, T. Munzner, H. Pfister, P. Rheingans, and T. S. Yoo.

[KMS02] D.A. Keim, W. Müller, and H. Schumann. Visual Data Mining. *Eurographics 2002 State of the Art Reports*, 2002.

[KMSZ06] D.A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler. Challenges in Visual Data Analysis. *IEEE Information Visualization*, pages 9–16, 2006.

[KO02] S.O. Kuznetsov and S.A. Obiedkov. Comparing performance of algorithms for generating concept lattices. *Journal of Experimental & Theoretical Artificial Intelligence*, 14(2) :189–216, 2002.

[Koe06] B. Koester. Conceptual Knowledge Retrieval with FooCA : Improving Web Search Engine Results with Contexts and Concept Hierarchies. In *Proceedings of the 6th Industrial Conference on Data Mining, ICDM 2006*, volume 4065 of *LNCS*. Springer, 2006.

[Koh01] T. Kohonen. *Self-Organizing Maps*. Springer, 2001.

[KS71] D.H. Krantz and P. Suppes. *Foundations of measurement*. Academic Press, New York, 1971.

[KW78] J.B. Kruskal and M. Wish. *Multidimensional Scaling*. Sage Publications, 1978.

[LA00] A. Leuski and J. Allan. Lighthouse : Showing the Way to Relevant Information. In *Proceedings of the IEEE Symposium on Information Vizualization*, page 125, 2000.

[Lor70] F.M. Lord. On the statistical treatment of football numbers (1953). *Readings in Statistics*, 8 :750–751, 1970.

[LRP95] J. Lamping, R. Rao, and P. Pirolli. A focus+ context technique based on hyperbolic geometry for visualizing large hierarchies. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408, 1995.

[LVSC03] P. Lyman, H.R. Varian, K. Swearingen, and P. Charles. How Much Information ?, 2003.

[Mac86] J. Mackinlay. Automating the Design of Graphical Presentations of Relational Information. *ACM Transactions on Graphics*, 5(2), 1986.

[MCDA03] J. Mothe, C. Chrisment, B. Dousset, and J. Alau. DocCube : Multi-dimensional visualisation and exploration of large document sets. *JASTIS*, 54(7) :650–659, 2003.

[MDB87] B.H. McCormick, T.A. DeFanti, and M.D. Brown. Visualization in scientific computing. *IEEE Computer Graphics and Applications*, 21(6) :1–14, 1987.

[MDNST05] N. Messai, M. Devignes, A. Napoli, and M. Smail-Tabbone. Querying a Bioinformatic Data Sources Registry with Concept Lattices. *Lecture Notes in Computer Science*, 3596 :323, 2005.

[MDNST06] N. Messai, M.D. Devignes, A. Napoli, and M. Smaïl-Tabbone. BR-Explorer : An FCA-based algorithm for Information Retrieval. *Fourth Internationnal Conference on Concept Lattices and their Applications, CLA 2006, October 30th-November 1st, Yasmine Hammamet, Tunisia*, pages 285–290, 2006.

[Mic86] J. Michell. Measurement scales and statistics : a clash of paradigms. *Psychological bulletin.*, 100(3) :398–407, 1986.

[NC05] J.P. Nakache and J. Confais. *Approche pragmatique de la classification : arbres hiérarchiques, partitionnements*. Editions Technip, 2005.

[NN97] P. Njiwoua and E.M. Nguifo. IGLUE : an instance-based learning system over lattice theory. In *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence*, pages 75–76, 1997.

[NN98] E.M. Nguifo and P. Njiwoua. Using Lattice-Based Framework as a Tool for Feature Extraction. *European Conference on Machine Learning*, pages 304–309, 1998.

[Nor05] C. North. Information Visualization. *Handbook of Human Factors and Ergonomics, 3rd Edition, G. Salvendy (editor), New York : John Wiley & Sons*, 2005.

[PGB02] C. Plaisant, J. Grosjean, and B.B. Bederson. SpaceTree : Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. *Proceedings of the IEEE Symposium on Information Visualization (InfoVis' 02)*, 2002.

[PHP03] D. Pfitzner, V. Hobbs, and D. Powers. A unified taxonomic framework for information visualization. *Proceedings of the Australian symposium on Information visualisation*, 24 :57–66, 2003.

[Pla04] C. Plaisant. The challenge of information visualization evaluation. *Proceedings of the working conference on Advanced visual interfaces*, pages 109–116, 2004.

[Pri06] U. Priss. Formal concept analysis in information science. *Annual Review of Information Science and Technology*, 40 :521–543, 2006.

[RMC91] G.G. Robertson, J.D. Mackinlay, and S.K. Card. Cone Trees : animated 3D visualizations of hierarchical information. *Proceedings of the SIGCHI conference on Human factors in computing systems : Reaching through technology*, pages 189–194, 1991.

[Sal89] G. Salton. *Automatic text processing : the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.

[Sar95] W.S. Sarle. Measurement theory : Frequently asked questions. *Disseminations of the International Statistical Applications Institute*, 30 :61–66, 1995.

[SB92] M. Sarkar and M.H. Brown. Graphical fisheye views of graphs. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 83–91. ACM New York, NY, USA, 1992.

[SB94] M. Sarkar and M.H. Brown. Graphical fisheye views. *Commun. ACM*, 37(12) :73–83, 1994.

[SF03] A. Skupin and SI Fabrikant. Spatialization Methods : A Cartographic Research Agenda for Non-geographic Information Visualization. *Cartography and Geographic Information Science*, 30(2) :99–119, 2003.

[Shn92] B. Shneiderman. Tree visualization with tree-maps : 2-d space-filling approach. *ACM Transactions on Graphics (TOG)*, 11(1) :92–99, 1992.

[Shn96] B. Shneiderman. The Eyes Have It :A Task by Data Type Taxonomy for Information Visualizations. *IEEE Visual Languages*, pages 336–343, 1996.

[Spe01] R. Spence. *Information Visualization*. ACM Press Books, 2001.

[SR92] M. Sarkar and S.P. Reiss. Manipulating Screen Space with StretchTools : Visualizing Large Structures on Small Screens. 1992.

[SRQ06] R. Shetty, P.M. Riccio, and J. Quinqueton. Hybrid Model for Knowledge Representation. In *ICHIT 2006, International Conference on Hybrid Information Technology*. IEEE, 2006.

[SSTR93] M. Sarkar, S.S. Snibbe, O.J. Tversky, and S.P. Reiss. Stretching the rubber sheet : a metaphor for viewing large layouts on small screens. In *Proceedings of the 6th annual ACM symposium on User interface software and technology*, pages 81–91. ACM Press New York, NY, USA, 1993.

[STB+02] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, and L. Lakhal. Computing iceberg concept lattices with Titanic. *Data & Knowledge Engineering*, 42(2) :189–222, 2002.

[Ste46] S.S. Stevens. On the Theory of Scales of Measurement. *Science*, 103(2684) :677–680, 1946.

[SW48] C.E. Shannon and W. Weaver. A mathematical theory of communications. *Bell System Technical Journal*, 27(2) :632–656, 1948.

[Tho05] J.J. Thomas. *Illuminating the Path : The Research and Development Agenda for Visual Analytics*. IEEE Computer Society Press, 2005.

[Tor58] W.S. Torgerson. *Theory and methods of scaling*. Wiley, New York, 1958.

[TRFR06] C. Tricot, C. Roche, C.E. Foveau, and S. Reguigui. Cartographie sémantique de fonds numériques scientifiques et techniques. *Document numérique*, 9 :12–35, 2006.

[Tuf83] E.R. Tufte. *The visual display of quantitative information*. Graphics Press Cheshire, CT, USA, 1983.

[Tuk62] J.W. Tukey. The future of data analysis. *Annals of Mathematical Statistics*, 33(1) :1–67, 1962.

[Tuk77] J.W. Tukey. *Exploratory data analysis*. Addison-Wesley, 1977.

[VDD00] K. Van Deun and L. Delbeke. Multidimensional Scaling, Open and Distance Learning, Mathematical Psychology Belgium, University of Leuven. http ://www.mathpsyc.uni-bonn.de/doc/delbeke/delbeke.htm, 2000.

[VGRH03] P. Valtchev, D. Grosser, C. Roume, and M.R. Hacene. Galicia : an open platform for lattices. In *Using Conceptual Structures : Contributions to 11th Intl. Conference on Conceptual Structures (ICCS ?03)*, pages 241–254, 2003.

[VRC08] J. Villerd, S. Ranwez, and M. Crampes. Using Concept Lattices as a Visual Assistance for Attribute Selection. In *Supplementary Proceedings of ICCS 2008, Conceptual Structures : Knowledge Visualization and Reasoning*, volume 354 of *CEUR*, pages 41–48, 2008.

[VRCC07] J. Villerd, S. Ranwez, M. Crampes, and D. Carteret. Using Concept Lattices for Visual Navigation Assistance in Databases : Application to a Patent Database. In *Proceedings of CLA 2007*, volume 331 of *CEUR*, pages 88–99, 2007.

[VRCC08] J. Villerd, S. Ranwez, M. Crampes, and D. Carteret. Using Concept Lattices for Visual Navigation Assistance in Databases : Application to a Patent Database (extended version). *Journal of General Systems, special issue on Concept Lattices and Their Applications*, 2008. accepted.

[VW93] P.F. Velleman and L. Wilkinson. Nominal, ordinal, interval, and ratio typologies are misleading. *The American Statistician*, 47(1) :65–72, 1993.

[vW05] J.J. van Wijk. The Value of Visualization. In *Proceedings of IEEE Visualization*, pages 11–18. IEEE Computer Society, 2005.

[War04] C. Ware. *Information Visualization : Perception for Design.* Morgan Kaufmann, 2004.

[WB94] P.C. Wong and R.D. Bergeron. Years of Multidimensional Multivariate Visualization. *Scientific Visualization, Overviews, Methodologies, and Techniques*, pages 3–33, 1994.

[Wil05] L. Wilkinson. *The Grammar of Graphics.* Springer, 2005.

Résumé

Les outils de recherche d'information sont confrontés à un accroissement constant à la fois du volume et du nombre de dimensions des données accessibles. La traditionnelle liste de résultats ne suffit plus. Un réel besoin en nouvelles techniques de représentation visuelle émerge. Ces nouvelles techniques doivent permettre d'appréhender de manière globale des données nombreuses et multidimensionnelles, en révélant les tendances et la structure générales. On souhaite également pouvoir observer de façon détaillée un ensemble plus restreint de données selon un certain point de vue correspondant à des dimensions particulières.

Notre objectif principal est d'assister l'utilisateur dans sa tâche d'exploration de l'information par une articulation judicieuse entre vue globale et vues locales maintenant sa carte mentale. Pour atteindre cet objectif, nous allions des techniques d'analyse de données capables d'identifier des sous-ensembles pertinents, à des techniques de visualisation d'information permettant de naviguer dynamiquement et intuitivement parmi ces sous-ensembles. Une attention particulière est portée aux problèmes liés aux données manquantes, d'une part, et aux données indexées sur des dimensions mixtes (binaires, nominales, continues), d'autre part. De plus, conformément aux attentes de la communauté visualisation, nous définissons un cadre formel pour la spécification de visualisations à partir des données à représenter.

Concrètement, nous proposons une méthode de navigation originale associant des techniques de FCA (Formal Concept Analysis) et de visualisation multidimensionnelle MDS (MultiDimensional Scaling). Cette méthode s'appuie sur le paradigme de visualisation *overview + detail* constitué d'une vue globale révélant la structure des données et d'une vue locale affichant les détails d'un élément de la vue globale. Nous tirons parti des propriétés de regroupement du treillis de Galois en l'utilisant comme vue globale pour représenter la structure des données et suggérer des parcours cohérents. La vue locale représente les objets en extension d'un concept sélectionné, projetés par MDS.

Nous illustrons la pertinence de cette méthode sur des données concrètes, issues de nos partenariats industriels, et montrons en quoi les techniques de visualisation liées à FCA et la visualisation spatialisée de données par projection MDS, parfois jugées incompatibles, se révèlent complémentaires.

Mots-clés: Visualisation d'information, Formal Concept Analysis, Multidimensional Scaling

Abstract

Information retrieval tools are faced with the constant increase of data both in volume and in dimensionality and the traditional list of results no longer meet many applications' requirements. New visual representation techniques are needed. These new techniques have to provide an overview of large and multidimensional data sets that gives insights into the underlying trends and structures. They must also be able to represent, in detail, portions of the original data from different standpoints.

The aim is to assist the user in her data exploration task by designing a shrewd link between general and local views, that maintains her mental map. In order to achieve this goal, we develop a combination of data analysis techniques that identify pertinent portions of data as well as information visualization techniques that intuitively and dynamically explore these portions of data in detail. In addition, a formalization of the visualization process is needed. We introduce a formal frame that is used to specify visualizations from data structures.

Concretely, the solution proposed is an original navigation method that combines techniques from Formal Concept Analysis (FCA) and Multi-Dimensional Scaling (MDS) visualization approaches to suggest navigation paths in the data. This method is based on the "overview + detail" paradigm : One component is an overall view which summarises the underlying structure of the data. A second component is a local view showing an element of the overall view in detail. We take advantage of the classification skills of the Galois lattice by using it as the overall view that reveals the inner data structure and suggests possible navigation paths. The local view uses Multi-Dimensional Scaling to display the objects in the extent of a selected concept.

We illustrate and discuss the pertinence of our method on concrete data sets, provided by our industrial partners, and show how hybridisation of FCA and traditional data visualization approaches, which have sometimes been considered distinct or incompatible, can be complementary.

Keywords: Information Visualization, Formal Concept Analysis, Multidimensional Scaling