Conception et implémentation semi-automatique des entrepôts de données : application aux données écologiques

Lucile Sautot

Université de Bourgogne AgroParisTech

Thèse

présentée pour obtenir le titre de DOCTEUR en
Informatique

Conception et implémentation semi-automatique des entrepôts de données : Application aux données écologiques

Lucile Sautot

Soutenue publiquement le 09/10/2015 devant un jury composé de :

Rapporteur	Dr. Fadila Bentayeb, Pr. Gilles Zurfluh
Examinateur	Pr. Christophe Nicolle, Pr. Engelbert Mephu
Invité	Pr. Francis Aubert
Directeur de thèse	Pr. Bruno Faivre
Encadrant de thèse	Dr. Sandro Bimonte

Avant de commencer

Langue de la thèse

Cette thèse est une thèse sur articles. Elle est donc organisée de la façon suivante :

— Tout d'abord, dans une introduction générale, nous exposerons le contexte institutionnel et scientifique de cette thèse. En partant du contexte général de ce travail de recherche et du cas d'étude, nous exposerons les questions soulevées par la gestion et l'analyse de données écologiques par les outils d'informatique décisionnelle. Nous proposerons ainsi une problématique pour ce travail de recherche et nous définirons les objectifs de cette thèse. Cette première partie est en **français**.

— Dans une seconde partie, nous présenterons nos trois contributions principales. Pour chaque contributions, nous proposerons une synthèse en **français**, puis le texte en **anglais** de la publication associée.

— Pour finir, nous proposerons une conclusion générale et des perspectives de recherche dans une troisième partie rédigée en **français**.

Remerciements

Pour commencer, je souhaiterai remercier **Bruno Faivre**, mon directeur de thèse, qui m'a conseillée et soutenue durant ces trois ans de thèse. Sans lui, ce travail n'aurait pas été possible.

Je remercie également **Sandro Bimonte**, mon encadrant, qui a investi beaucoup de temps et d'énergie pour me faire progresser. Un grand merci pour cette collaboration fructueuse !

Mes plus sincères remerciements vont au **Dr. Fadila Bentayeb**, au **Pr. Gilles Zurfluh**, au **Pr. Engelbert Mephu Nguifo** et au **Pr. Christophe Nicolle**, rapporteurs et examinateurs de mon jury de thèse, qui m'ont fait l'honneur d'évaluer ma thèse. Je remercie également le **Dr. Jean Secondi** et le **Pr. Kokou Yetongnon**, membres de mon comité de thèse, pour leurs conseils. Enfin, je remercie tout particulièrement le **Pr. Francis Aubert** qui a accompagné mon projet de thèse, du dépôt de dossier de financement jusqu'à la soutenance.

Je remercie sincèrement mes collègues d'AgroSup Dijon et de l'UMR Biogéosciences grâce auxquels j'ai pu expérimenter le métier d'enseignante et développer une violente addiction au café. Merci à **Aurélie Khimoun**, **Anthony Ollivier**, **Stéphane Garnier**, **Nicolas Navarro**, **Annie Marchand**, **Isabelle Santacroce**, **Houda Bediaf**, **Simeng Han**, **Thomas Decourselle**, **Bastien Billiot**,

Rachid Sabre, **Jean-Pierre Lemière**, **Patricia Chavanelle**, et à tous ceux que j'ai oublié, pour vos encouragements, vos conseils, votre aide et tout ce que nous avons partagé.

En particulier, je remercie le **Pr. Paul Molin** pour son aide dans les moments difficiles. Je remercie également **Carmela Château** pour son investissement dans la relecture de mes publications. Pour finir, je remercie **Alexandre Gaudry** pour son soutien.

Enfin, mes remerciements les plus affectueux vont à **mes parents**, à ma soeur **Camille**, et à mes amies **Julie**, **Méline** et **Astrid**, qui m'ont patiemment demandé pendant trois ans si ma thèse avançait et qui ont gentiment écouté mes réponses à base de "Nan, je galère, je dois expliquer en anglais que les dimensions contextuelles d'un fait qui n'est pas le fait source sont les dimensions partagées par le fait source et un autre fait lié à la dimension cible qui ne sont pas la dimension cible ...". Et à **Ludo**, bien sûr, qui m'a supportée dans tous les états inhérents au statut de doctorante, de la fatigue au stress en passant par le questionnement existentiel. Merci chéri <3

Table des matières

Première partie

Introduction

Chapitre 1

Contexte et problématique de la thèse

1.1 Contexte

Les études écologiques ou agronomiques nécessitent de récolter des données concernant des organismes variés sur de grands intervalles de temps et de vastes zones géographiques. L'effort de collecte associé à ses données est important à plusieurs titres. Tout d'abord, le temps consacré à la collecte de données dans les sciences du vivant est particulièrement long. Par exemple, le jeu de données associé à notre cas d'étude, présenté dans le Chapitre 3 (page 35), a été collecté sur une durée totale de 21 ans. Deuxièmement, la collecte de ces données nécessite souvent du personnel qualifié. Troisième point, la mobilisation de personnel qualifié, pendant une longue durée, lors d'un recueil de données, implique des coûts importants. Par ailleurs, en écologie comme en agronomie, les études sur le terrain impliquent la prise en compte de nombreux facteurs. Ainsi, les données récoltées sont souvent des données complexes (issues de plusieurs sources, de plusieurs capteurs, ...) et multifactorielles (de nombreux paramètres sont pris en compte pour expliquer le phénomène d'intérêt). Ainsi, quand une étude est vouée à durer longtemps et à comprendre un phénomène complexe, le volume de données généré, en plus d'être coûteux à produire, peut s'avérer difficile à analyser voire à manipuler.

En résumé, dans le cadre d'une étude écologique de grande ampleur, l'acquisition des données, pour décrire les communautés et les terrains, est un travail important, qui génère un coût élevé. Ces données représentent un volume d'informations, dont la gestion peut s'avérer difficile, pour les raisons suivantes :

— *La subjectivité de l'expert.* En raison de sa connaissance du domaine, l'expert

s'attache à certaines informations et moins à d'autres, au risque de passer à côté de certaines variables essentielles à la compréhension des phénomènes. L'automatisation, au moins partielle, des traitements, permet d'éviter cette "mise à l'écart" d'informations a priori non pertinentes, qui peuvent se révéler finalement utiles.

— *Le coût.* L'acquisition et la manipulation des données ont un coût temporel et humain, donc financier. Or, ce coût peut devenir très important dans une étude à large échelle et devient donc souvent un facteur limitant.

— *Le savoir-faire dans la gestion de données.* Les écologues savent peu, voir ne savent pas, gérer des masses de données importantes. Leurs traitements sont effectués à l'aide de tableur, voir éventuellement de bases de données relationnelles. Or, ces technologies s'avèrent limitées dans le cadre d'une analyse de cette ampleur.

Il est donc intéressant de proposer aux scientifiques travaillant dans les sciences du vivant des systèmes d'information capable de stocker et de restituer leurs données, en particulier quand celles-ci présentent un volume important (Triplet and Butler, 2011).

Parmi les outils existants, les outils de l'informatique décisionnelle, notamment les systèmes d'analyse en ligne (*On-Line Analytical Processing : OLAP*), ont particulièrement retenu notre attention, car il s'agit de processus d'analyse de données sur de larges collections de données historiques (c'est-à-dire un entrepôt de données) afin d'offrir un support à la prise de décision (Hurtado et al., 1999; Jerbi, 2012). L'informatique décisionnelle propose des outils comme les entrepôts de données (*data warehousing*), les outils d'analyse en ligne et la fouille de données (*data mining*), qui permettent à leurs utilisateurs d'explorer de larges volumes de données, dans le but de découvrir des modèles et des connaissances au sein de ces données, et ainsi d'éventuellement confirmer leurs hypothèses.

Une architecture classique pour un système d'information décisionnel est l'architecture ROLAP (*Relational OLAP*), qui consiste en : (i) un système de gestion de base de données relationnel, qui stocke les données selon un paradigme multidimensionnel : il s'agit du stockage physique de l'entrepôt de données ; (ii) un serveur OLAP, qui implémente le modèle multidimensionnel et les opérateurs OLAP sur le système de gestion de base de données ; (iii) un client OLAP, qui combine et synchronise des vues tabulaires et graphiques des données, issues de requêtes lancées sur l'entrepôt de données ; (iv) un outil d'intégration de données (*Extract-Transform-Load : ETL*) qui extrait des données de sources multiples et hétérogènes, et les transforme pour les intégrer à l'entrepôt de données.

Ces outils présentent des caractéristiques avantageuses en termes de restitution de données, car ils permettent de faire apparaître très facilement les faits étudiés

en fonction de facteurs d'analyses et de réaliser très facilement des statistiques descriptives [1] sur les données (Alkharouf et al., 2005).

Par ailleurs, les requêtes sur les données étant simples à réaliser et ne nécessitant aucune connaissance préalable en informatique, ces outils permettent aisément de sélectionner et d'exporter des données en vue de réaliser des analyses statistiques plus poussées ou d'appliquer des méthodes de fouille de données (Codd et al., 1993).

Mais les systèmes OLAP sont des systèmes complexes, qui nécessitent, pour être conçus, un long travail de recueil des besoins auprès des futurs utilisateurs afin de s'assurer que les faits à analyser et les dimensions d'analyses nécessaires ont tous été identifiés.Par ailleurs, la mise en place d'un tel système nécessite une base de données, un serveur OLAP, un client OLAP, un outil d'intégration de données (ETL), ce qui constitue une architecture difficile à mettre en place sans l'appui d'informaticiens expérimentés (Inmon, 1996; Kimball, 1996), notamment dans le cas de données complexes générées par les sciences biologiques (Dubitzky et al., 2001).

1.2 Objectifs et problématique

Les chercheurs en écologie ont développé différentes approches pour décrire l'habitat des espèces. Dans ce cadre, la principale difficulté correspond à la gestion et le traitement des masses de données qu'ils intègrent. En effet, l'exploration de la problématique biologique implique la manipulation et le croisement de nombreuses données, hétérogènes, issues de sources variées et cela dans le but d'établir un modèle explicatif du phénomène biologique observé.

Pour éprouver notre démarche, nous avons ainsi disposé d'un jeu de données concernant l'abondance des oiseaux le long de la Loire. Ce jeu de données est structuré de la façon suivante : (1) nous disposons du recensement de 213 espèces d'oiseaux (décrites par un ensemble de facteurs qualitatifs, comme par exemple le régime alimentaire) en 198 points le long du fleuve pour 4 campagnes de recensement ; (2) chacun des 198 points est décrit par un ensemble de variables environnementales issues de différentes sources (relevés de terrain, images satellites, SIG [2]). Ce sont ces variables environnementales qui posent le plus de questions en termes de modélisation multidimensionnelle. Ces données sont issues de différentes sources, parfois indépendantes des campagnes de recensement des oiseaux, et sont

1. Les statistiques descriptives sont une synthèse, sous forme de tableaux, de graphiques et de résumé numériques, d'une variable numérique mesurée sur plusieurs individus (Saporta, 2011).
2. Système d'Information Géographique

donc inconsistantes dans le temps et l'espace. De plus, ces données sont hétérogènes : elles peuvent se présenter sous forme de facteurs qualitatifs, quantitatifs ou encore d'objets spatiaux. Pour finir, ces données environnementales intègrent un grand nombre de variables (158 variables retenues).

Le traitement de ces données est long, complexe, voire impossible, s'il est réalisé "à la main". Une automatisation dans la gestion et le traitement des données est nécessaire pour maximiser leur exploitation. Or, les écologues ne possèdent pas les compétences informatiques nécessaires à la mise en place de cette automatisation.

La gestion, l'exploitation et l'optimisation d'un jeu de données d'un volume important issu de sources hétérogènes à des fins d'analyse est un problème qui est apparu au sein des entreprises il y a de nombreuses années. Leur réponse à ce type de problème s'est traduite par la mise en place d'un ensemble de nouvelles méthodes et technologies regroupées sous le terme d'informatique décisionnelle (*business intelligence* en anglais). Plus récemment, ces technologies ont été adaptées pour répondre à des problématiques biologiques, dans les domaines de la foresterie (Miquel et al., 2002a), de la gestion de la pollution de l'eau (Bimonte, 2007) ou encore de la biologie moléculaire (Shah et al., 2005). Le programme STORI (Suivi Temporel des Oiseaux nicheurs en RIvière, voir Chapitre 3) ayant pour objectif de traiter et analyser les données récoltées, une solution de type *business intelligence* a été envisagée, au vu de leurs objectifs.

Un moyen de démocratiser les systèmes OLAP, ou du moins, de les rendre accessibles à des utilisateurs potentiels disposant de peu de ressources informatiques, est d'automatiser au maximum la conception et la mise en place de tels systèmes.

Dans la littérature, plusieurs travaux se sont penchés sur la conception automatique de schéma multidimensionnel, mais les exemples proposés par ces travaux concernaient des données classiques, issues de jeux de données usuels dans le domaine, c'est à dire, avec des données factuelles numériques, des données dimensionnelles catégorielles et sans inconsistance majeure. Par ailleurs, d'autres travaux traitent de la modélisation multidimensionnelle adaptée à des données complexes (inconsistance, données hétérogènes, intégration d'objets spatiaux, de textes, d'images au sein d'un entrepôt ...) mais les méthodes proposées par ces travaux sont rarement automatiques.

C'est pourquoi l'objectif de ce travail de thèse est de proposer une méthode de conception d'entrepôt de données et des cubes OLAP associés la plus automatique possible. Cette méthode doit être capable de prendre en compte la complexité des données inhérente aux sciences biologiques. La problématique peut donc être formulée comme :

Comment concevoir automatiquement un modèle multidimensionnel

et l'implémenter automatiquement pour des données écologiques ?

Pour cette étude, nous nous sommes essentiellement concentrés sur la modélisation des hiérarchies au sein des dimensions. En effet, l'identification des faits et des dimensions est relativement aisé pour les utilisateurs. En revanche la structuration des dimensions peut s'avérer beaucoup plus problématique. Ainsi nos contributions durant cette thèse se sont essentiellement centrées sur l'intégration des données environnementales à une dimension spatiale.

Nous avons donc, durant ce travail de thèse, abordé trois angles différents :

1. Tout d'abord, nous nous sommes intéressés à *la nature des données*. Notre premier objectif sera de proposer une méthode permettant de construire automatiquement une hiérarchie avec des membres décrits par des attributs quantitatifs, des attributs qualitatifs et sachant que certaines données peuvent être manquantes.
2. Ensuite, nous nous sommes intéressés à la prise en compte *des besoins analytiques et des connaissances des utilisateurs* concernant les données. Notre second objectif sera de proposer une méthode prenant en compte les spécifications des utilisateurs pour la construction automatique de hiérarchies, grâce au prototypage.
3. Pour finir, nous nous sommes intéressés *à la complexité des analyses* menées par les utilisateurs, qui impliquent de construire des requêtes avec des données interdépendantes. Notre troisième objectif sera de proposer une méthode capable de construire automatiquement une hiérarchie en prenant en compte la source des données utilisées pour construire cette hiérarchie et son contexte.

1.3 Plan de la thèse

Cette thèse inclut neuf chapitres, répartis en trois parties :

— Les chapitres 1 à 3, regroupés dans la Partie I, constituent l'introduction de cette thèse. Le Chapitre 1 présente le contexte de recherche dans lequel s'inscrit cette thèse, la problématique de cette thèse et définit les objectifs de nos travaux. Le Chapitre 2 propose un état de l'art des concepts généraux relatifs à ce travail de thèse. Pour finir, le cas d'étude qui sera le support à l'ensemble des tests réalisés durant ce travail de thèse est présenté dans le Chapitre 3.

— Les chapitres 4 à 6, regroupés dans la Partie II, sont les contributions que nous proposons dans cette thèse. Chaque chapitre est une réponse à l'un des trois objectifs définis dans la Section 1.2. Ainsi, le Chapitre 4 traite

particulièrement de la construction automatique de hiérarchies OLAP avec des données mixtes et manquantes. Le Chapitre 5 traite de l'intégration d'algorithmes de fouille de données à une méthode de prototypage de schéma OLAP pour la construction automatique de hiérarchies. Enfin, le Chapitre 6 propose une méthode d'enrichissement de dimension avec des hiérarchies contextuelles calculées à partir de données factuelles.

— Les chapitres 7 et 8, regroupés dans la Partie III, constituent la conclusion de cette thèse. Le Chapitre 7 est un bilan général des travaux de recherche présentés dans ce document, et le Chapitre 8 présente quelques perspectives de recherche relatives à ce travail de thèse.

— Pour finir, la Partie IV regroupe l'ensemble des annexes de ce document.

Chapitre 2

Concepts généraux

Dans cette partie, nous présentons un état de l'art relatif au domaine de recherche de ce travail de thèse. Nous définissons ici les concepts généraux sur lesquels nous nous sommes appuyés pour réaliser ce travail.

2.1 Analyse en ligne

Le terme "OLAP" signifie "*On Line Analytical Processing*". Ce terme à une signification large. Il désigne le processus d'analyse de données sur de larges collections de données historiques (c'est-à-dire un entrepôt de données) afin d'offrir un support à la prise de décision, en permettant aux décideurs de réaliser des analyses sur des données factuelles (Hurtado et al., 1999). Ainsi, le terme OLAP peut être associé à un processus, à un type de système, à un type d'analyses, à un type de données (Jerbi, 2012).

Le terme OLAP est en général opposé à l' "OLTP", qui signifie "*On Line Transactionnal Processing*". L'OLTP correspond aux bases de données relationnelles de production. Ces processus sont essentiellement basés sur la gestion des transactions : on gère de nombreuses opérations de mise à jour sur un petit nombre de données, lors d'une utilisation quotidienne des systèmes. Les systèmes OLTP sont basés sur le concept de transaction. Dans le domaine des bases de données, une transaction est une méthode permettant à un développeur d'applications d'empaqueter une séquence d'opérations sur une base de données, afin que le système de gestion de la base de données offrent un certains nombres de garanties, connues sous le nom de propriétés ACID [1] (Bernstein and Goodman, 1981; Deng et al.,

1. Atomicité, Consistance, Isolation, Durabilité

2003; O'Neil and O'Neil, 2000).

La comparaison entre OLAP et OLTP est présentée dans le tableau ci-dessous.

	OLTP	**OLAP**
Utilisation	Gestion des transactions	Prise de décision
Conception	Orientée applications	Orientée utilisateurs
Fréquence d'utilisation	Quotidienne	Sporadique
Données	Actuelles, détaillées	Historiques, multidimensionnelles, agrégées
Source	Base de données unique	Plusieurs bases de données
Nombre de lignes accédées	Dizaines	Milliers
Type d'utilisateurs	Opérateurs	Décideurs
Nombre d'utilisateurs	Milliers	Centaine
Dimension de la base de données	Gio	Tio

Tableau 2.1 – Comparaison de l'OLTP et de l'OLAP (Bimonte, 2007; Teste, 2006)

Les systèmes OLAP sont également appelés "Systèmes d'information décisionnels". Ces systèmes doivent, selon leur inventeur, respecter 12 règles (Codd et al., 1993) (voir le Chapitre 9, en annexe). Ces règles ayant été édictées dans le cadre d'un projet pour une entreprise privée, elles ont parfois été remises en question, ou du moins remaniées (Golli, 2009).

Un système OLAP est composé de trois éléments : la base de données multidimensionnelle, un serveur OLAP et le client OLAP qui permet aux usagers d'effectuer les différentes analyses via une interface spécialisée et des opérateurs adaptés (Proulx and Bédard, 2004). La base de données multidimensionnelle correspond à un entrepôts de données et ses magasins de données. L'interface OLAP permet à l'utilisateur de créer des requêtes multidimensionnelles via une interface graphique. Le serveur OLAP analyse et traduit les requêtes OLAP en requêtes pour la base de données, puis organise le résultat de la requête fourni par le système de gestion de base de données selon un format multidimensionnel, pour l'afficher à l'utilisateur (Jerbi, 2012).

L'interrogation d'une base de données multidimensionnelle peut être vue comme une succession d'opérations d'exploration, comme l'agrégation, la consolidation, l'application de formules mathématiques, la synthèse de données selon de mul-

tiples dimensions (Sarawagi et al., 1998). Cette interrogation est appelée "analyse OLAP" en référence à la technologie utilisée et peut être qualifiée de navigation, car elle est considérée comme la transition entre différents états d'analyse (Jerbi, 2012).

De nombreux travaux proposent des modèles pour les systèmes OLAP, s'appuyant sur des modèles existants (UML, Entité/Relation, Orienté Objet) ou proposant de nouvelles approches (Lehner, 1998; Nguyen et al., 2000; Pedersen and Jensen, 1999; Tsois et al., 2001). Quelques soient les modalités retenues par les auteurs pour définir les règles de leurs modèles, ceux ci s'appuient sur trois concepts de la modélisation multidimensionnelle : les *mesures*, les *dimensions* et les *hiérarchies* (Jerbi, 2012).

Les mesures sont définies comme des variables dynamiques et dépendantes (Nguyen et al., 2000). Les mesures correspondent à la quantification des objets sur lesquels porte l'analyse OLAP, appelés "faits". Un fait représente souvent un évènement qui a lieu au sein de l'organisation qui utilise le système d'aide à la décision, et que l'on souhaite expliquer, comme par exemple les ventes (Wehrle, 2009).

Les dimensions sont définies comme des variables statiques et indépendantes (Nguyen et al., 2000), qui correspondent à des axes d'analyse. Une dimension oriente les requêtes, ce qui permet d'obtenir différentes vue sur les données, suivant les critères d'analyse fournis par la dimension (Wehrle, 2009).

Les dimensions d'un modèle OLAP peuvent contenir une ou plusieurs hiérarchies au sein des données. Les hiérarchies permettent de structurer les dimensions : les données d'une dimension peuvent souvent être catégorisées selon plusieurs caractéristiques. Habituellement, les utilisateurs des systèmes OLAP ne s'intéressent pas uniquement aux simples mesures mais à des données agrégées (Par exemple, la moyenne des ventes dans une certaines aires géographiques) et les hiérarchies apportent une méthode pour décrire les différents niveaux d'agrégation existants au sein d'une dimension. Ainsi, les hiérarchies correspondent à différents niveaux d'agrégation des données (Bimonte, 2007; Markl et al., 1999; Sarawagi et al., 1998).

Il existe différents types de hiérarchies, celles-ci pouvant être très complexes dans les cas réels (Bimonte, 2007; Malinowski and Zimanyi, 2006; Pourabbas and Rafanelli, 1999; Wehrle, 2009) :

— Hiérarchies *multiples* : Une hiérarchie multiple comporte plusieurs niveaux alternatifs qui ne sont pas reliés par des liens de filiation directs, ce qui crée plusieurs chemins au sein du schéma de la hiérarchie.
— Hiérarchies *couvrantes* : A chaque niveau de ce type de hiérarchie, les membres couvrent l'ensemble des données de l'entrepôt. Il n'existe donc aucun lien de filiation direct qui « saute » un niveau.
— Hiérarchies *onto* : Il n'existe pas, dans la hiérarchie, de membres qui, sans se

trouver au niveau le plus détaillé de la hiérarchie, n'ont pas de descendants.
— Hiérarchies *strictes* : Au sein d'une hiérarchie stricte, il ne peut pas exister de relation de plusieurs à plusieurs entre les membres reliés par des liens de filiation.

Les différents types de hiérarchies sont illustrés sur la Figure 2.1., à partir d'exemples issus de notre cas d'étude.

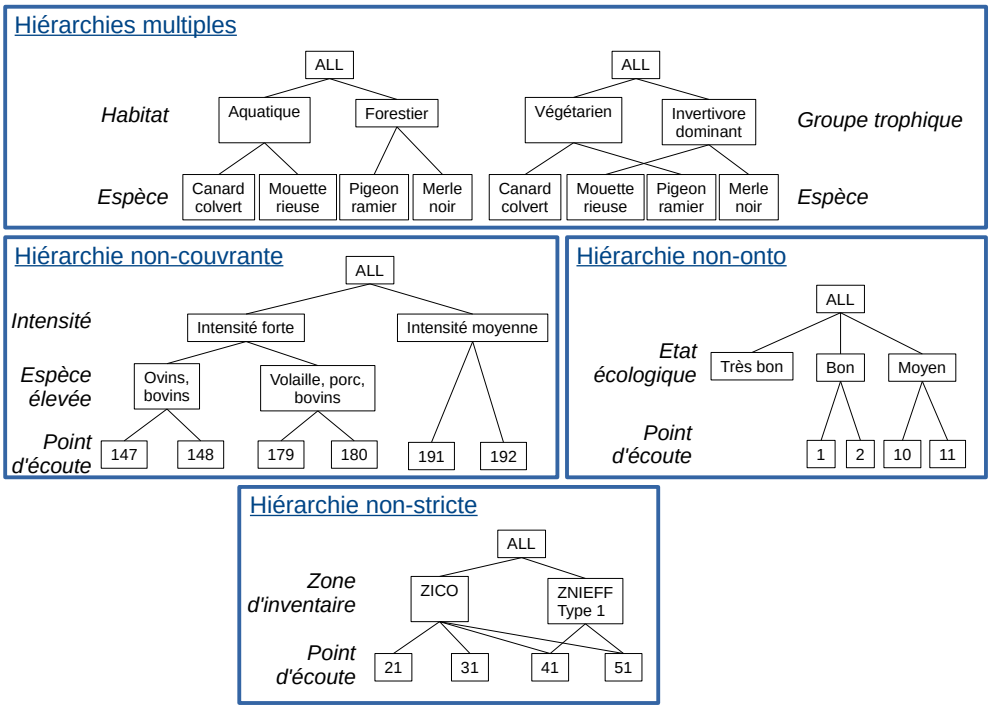


Figure 2.1 – Les différents types de hiérarchies

Les systèmes OLAP organisent les données en cubes. Un cube est une combinaison de plusieurs dimensions avec plusieurs niveaux hiérarchiques par dimension (Nguyen et al., 2000). Il représente les mesures détaillées et agrégées dans l'espace multidimensionnel formé par cette sélection de dimensions (Wehrle, 2009).

Il existe quelques standards pour représenter ces modèles :
— Les standards hérités de l'Entité/Relation : M E/R (Multidimensional Entity/Relationship) (Sapia et al., 1999), starER (Tryfona et al., 1999) et MultiDimEr (Malinowski and Zimanyi, 2006).
— Les standards hérités de l'UML : mUML (multidimensional UML) (Lujan-

Mora et al., 2006), OOMD (Object-Oriented Multidimensional Model) (Nguyen et al., 2000), YAM2 (Yet Another Multidimensional Model) (Abelló et al., 2006).
— Les modèles ad-hoc : DFM (Dimensional Fact Model) (Golfarelli and Rizzi, 2009) et son extension X-DFM.

Une analyse OLAP consiste à naviguer au sein d'un cube. Pour cela, les systèmes OLAP proposent différents opérateurs de navigation (Bimonte, 2007; Böhnlein and Ulbrich, 2000; Wehrle, 2009) :
— *Drill-down* : Descendre vers un niveau hiérarchique plus détaillé de la dimension.
— *Roll-up* : Monter vers un niveau hiérarchique plus résumé de la dimension.
— *Slice* : Sélectionner un sous-ensemble du cube réduit à un ensemble de membres sur une ou plusieurs dimensions. Il s'agit d'une sélection de données.
— *Dice* : Réduire le cube d'une ou plusieurs dimensions. Il s'agit d'une projection des données.
— *Rotate* : Pivoter le cube. Il s'agit d'afficher les données selon un angle de vue différent.
— *Drill-Accross* ou *OLAP Join* : Réunir, joindre deux cubes OLAP.

Ces opérateurs sont présentés dans la figure 2.2.

Un serveur OLAP se charge d'exécuter les requêtes transmises par l'interface proposée à l'utilisateur et de présenter le résultat de ces requêtes. Les données nécessaires aux analyses OLAP pouvant être stockées dans différents types de bases de données, il existe plusieurs types de serveurs OLAP, pouvant s'adapter aux différents types d'espaces de stockage choisis pour les données (Kimball, 2000).

Il existe trois implémentations physiques classiques de serveurs OLAP (Bimonte, 2007; Jerbi, 2012; Kimball, 2000; Proulx and Bédard, 2004; Teste, 2006; Vassiliadis and Sellis, 1999; Wehrle, 2009) :
— Le ROLAP (*Relational OLAP*) : Les données nécessaires à l'analyse sont alors stockées grâce à la technologie des bases de données relationnelles, dans un système de gestion de bases de données relationnel. Pour être performantes, ces solutions nécessitent une gestion avancée des index de la base de données et une modélisation particulière. Le serveur OLAP traduit la requête multidimensionnelle en requête SQL pour la base de données accueillant les données, récupère le résultat et le présente sous un format multidimensionnel. Ces opérations, qui nécessitent des jointures entre différentes tables de la base de données, peuvent prendre du temps. Ainsi, le serveur ROLAP peut être lent mais permet d'accéder aux données de bases, au niveau de granularité le plus fin au sein des données.
— Le MOLAP (*Multidimensional OLAP*) : Le serveur MOLAP s'appuie sur une base de données multidimensionnelle, spécialement conçue pour les sys-

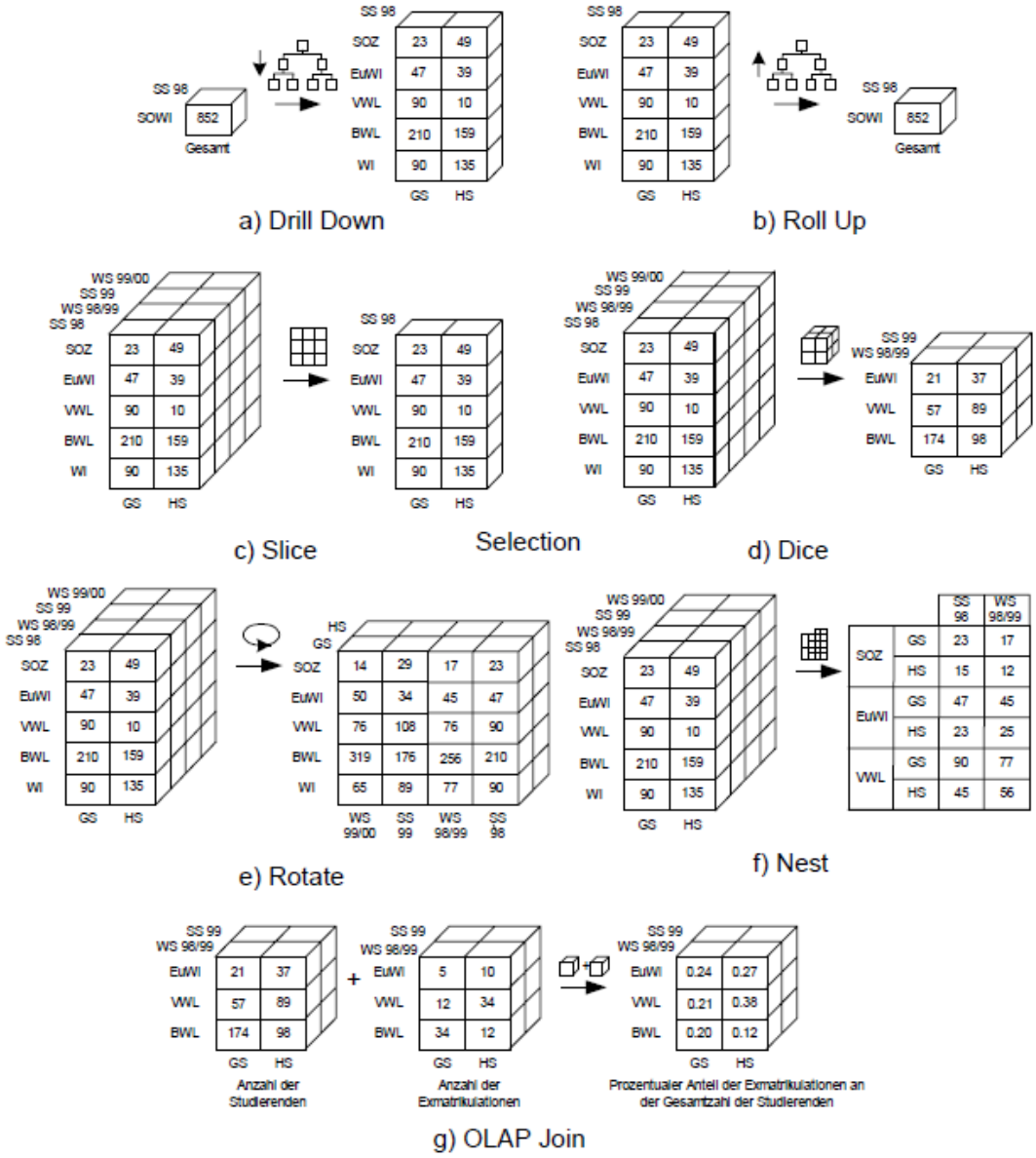


Figure 2.2 – Les opérateurs OLAP, d'après Böhnlein et Ulbrich-vom Ende (Böhnlein and Ulbrich, 2000)

tèmes d'aide à la décision. Contrairement aux serveurs ROLAP, les serveurs MOLAP ne recalculent pas les agrégations de données à chaque nouvelle requête, mais stockent un certain nombre de tableaux multidimensionnels, où les agrégations sont pré-calculées. Ce système permet aux serveurs MOLAP d'être très performants. Ainsi, le serveur MOLAP est généralement plus rapide qu'un serveur ROLAP mais ne permet pas d'accéder aux données atomiques.

— Le HOLAP (*Hybrid OLAP*) : Le serveur HOLAP est un hybride entre ROLAP et MOLAP, et tente de concilier les avantages des deux implémentations présentées ci-dessus. Un serveur HOLAP stocke les données dans une base de données relationnelles, ce qui permet de retrouver les données atomiques. Mais il offre également un espace de stockage multidimensionnel, ce qui permet de pré-calculer les agrégations des requêtes les plus classiques. Ainsi, un serveur HOLAP offre la possibilité d'accéder très rapidement aux données si elles ont été pré-calculées, mais donne également accès aux données les plus fines.

2.2 Entrepôts de données

Un entrepôt de données est donc une collection de données qui est le support d'un processus de prise de décision. Cette collection est conçue à des fins d'analyse. Les données qui composent cette collection doivent avoir plusieurs caractéristiques (Marksay and Pigneur, 2010) :

— Elles sont orientées "sujet" : elles sont organisées autour de différents sujets principaux (par exemple : les ventes, les consommateurs, les produits, etc ...). Elles se centrent sur les analyses nécessaires aux décideurs, et non sur des opérations quotidiennes ou des processus transactionnels. Elles proposent une vue sur un sujet en particulier : les données non nécessaires à l'analyse et à la prise de décision sont exclues.

— Elles sont intégrées : elles sont issues de sources de données multiples et hétérogènes (bases de données de production, fichiers plats, ERP, etc ...). Elles sont nettoyées et mises en forme avant d'être insérées dans l'entrepôt.

— Elles sont historiques : on conserve une trace au sein de l'entrepôt des valeurs prises par les différentes variables au cours du temps. Un entrepôt est conçu pour faire des analyses à une large échelle de temps (de l'ordre de la dizaine d'années), on conserve donc toutes les valeurs prises par une variable afin de pouvoir en analyser l'évolution.

— Elles ne sont pas volatiles : les données transformées et intégrées à l'entrepôt de données ne sont pas sensibles aux mises à jour qui ont lieu dans les

données “opérationnelles”, utilisées quotidiennement par l'organisation pour son fonctionnement et dont sont issues les données de l'entrepôt.

L'entrepôt de données est le lieu de stockage centralisé des données extraites des bases de production et pertinentes pour l'analyse. L'organisation des données au sein d'un tel espace est modélisée pour faciliter une gestion efficace des données et leur historisation.

Classiquement, il existe deux types de tables dans la modélisation d'un entrepôt de données (Bimonte, 2007; Inmon, 1996; Teste, 2006; Wehrle, 2009) :

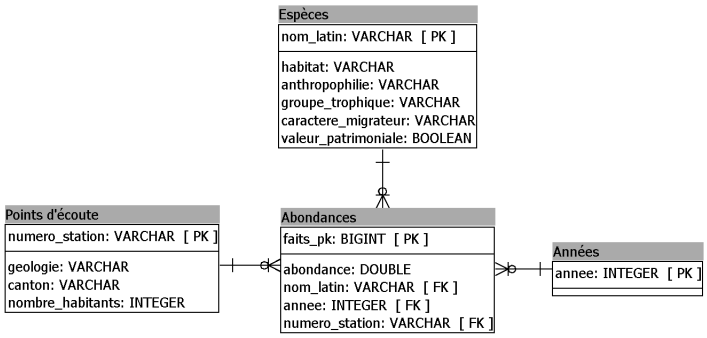
— **Les tables de faits** : une table de faits représente l'objet de l'analyse. Un fait est composé de plusieurs mesures. Une table de fait présente deux types de champs : les champs qui correspondent à des mesures et les champs qui correspondent à des clés renvoyant aux tables de dimension.

— **Les tables de dimension** : une table de dimension correspond à un axe d'analyse d'une (ou de plusieurs) table(s) de faits. Chacun de ces critères est un attribut de la table de dimension. Les attributs d'une dimension peuvent former une hiérarchie qui permettent alors à l'utilisateur de voir les données détaillées ou résumées, selon le niveau de la hiérarchie auquel il se place (par exemple, les attributs “ville”, “région” et “pays” peuvent former une hiérarchie). Une dimension peut avoir des attributs descriptifs, qui ne sont alors pas utilisés pour l'analyse (par exemple : le numéro de téléphone d'un fournisseur).

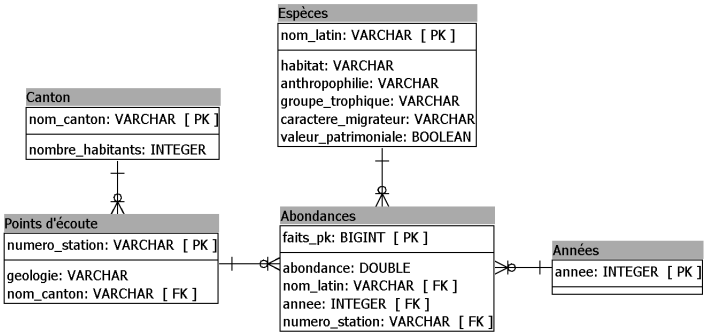
Les entrepôts de données ont une modélisation plus ou moins complexe selon la diversité des analyses menées au sein du système décisionnel et selon les besoins de normalisation (Teste, 2006). Le schéma le plus simple est le schéma dit **“en étoile”** : une table de faits centrale est connectée à un ensemble de tables de dimension. Un modèle plus complexe est le modèle dit **“en flocon de neige”**. Ce modèle est une extension du modèle en étoile : les tables de dimensions sont plus normalisées afin de représenter les hiérarchies, ce qui permet de maîtriser davantage la taille des tables de dimension. Enfin, le modèle le plus complexe d'entrepôt de données est le schéma dit **“en constellation”** : plusieurs tables de faits sont présentes dans ce modèle et partagent une ou plusieurs dimensions. Ce schéma peut être vu comme une collection de schémas en étoile. Des exemples de ces différents modèles, construits à partir de notre cas d'étude, sont présentés sur la Figure 2.3.

Il existe trois approches pour développer un entrepôt de données : l'approche **data-driven**, l'approche **user-driven** et l'approche **hybrid-driven** (Cravero and Sepúlveda, 2014; Tebourski et al., 2013). L'approche data-driven est centrée sur la structure des sources de données disponibles et construit l'entrepôt à partir de ces données disponibles. Un exemple de méthodologie de conception entièrement data-driven est présentée dans (Usman et al., 2010). L'approche user-driven en

a)



b)



c)

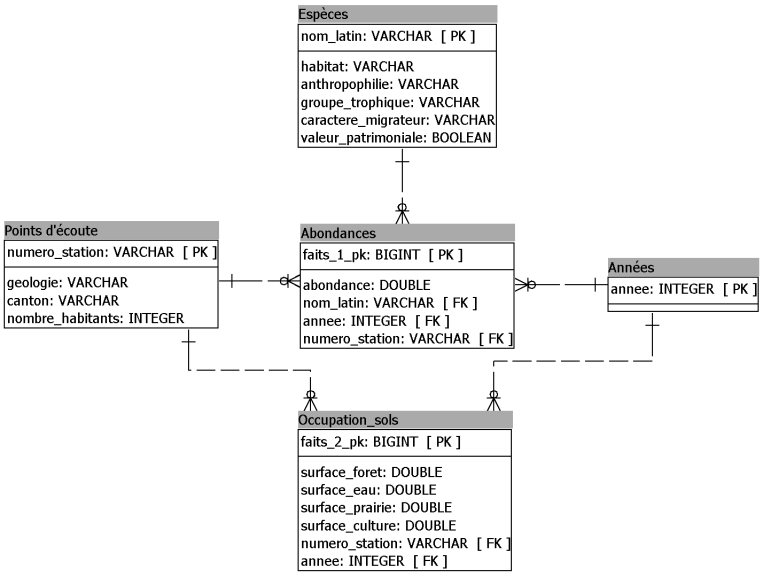


Figure 2.3 – Différents modèles d'entrepôt de données, a) en étoile, b) en flocon et c) en constellation

revanche s'appuie sur les spécifications des utilisateurs pour modéliser l'entrepôt de données. Un exemple de ce type de méthode est présentée dans (Jovanovic et al., 2012). Pour finir, l'hybrid-driven est une approche hybride, qui prend en compte aussi bien les spécifications des utilisateurs que la structure des données sources pour proposer un entrepôt de données. On peut trouver des exemples d'approches hybrid-driven dans (Romero and Abello, 2010) et dans (Abdelhedi et al., 2011).

2.3 Fouille de données

2.3.1 Définition générale

La data mining, encore appelé "fouille de données", est une des étapes du processus de découverte de connaissances au sein des bases de données de grandes dimensions (*Knowledge Discovery in Databases* en anglais, abrégé KDD). La KDD présente plusieurs étapes : sélection des données, pré-traitement des données, transformation des données, exploration des données, interprétation des résultats. L'étape d'exploration des données correspond à ce que l'on appelle le data mining (Fayyad et al., 1996; Frawley et al., 1992).

L'objectif principal de la KDD est la découverte, qui correspond à la construction de nouveaux modèles par le système autonome. Cet objectif de découverte peut être divisé en deux sous-objectifs :

— La prédiction, où le système trouve des modèles pour prédire le comportement futur de certaines entités.
— La description, où le système trouve des modèles pour présenter les données à un utilisateur dans un format humainement compréhensible.

Le data mining consiste à ajuster des modèles, ou à déterminer des profils au sein des données observées. Les modèles ajustés sont alors considérés comme des connaissances déduites. Savoir si les modèles reflètent des connaissances intéressantes et utiles fait partie de l'ensemble du processus, interactif, de KDD, et requiert généralement un jugement humain subjectif. Deux formalismes mathématiques classiques sont utilisés pour l'ajustement des modèles : ajustement statistique et ajustement logique (Fayyad et al., 1996).

Ainsi, le data mining est un ensemble de techniques descriptives et prédictives destinées à explorer les données, en mettant à jours des liens a priori inconnu entre les variables (Tuffery, 2011). Le data mining se situe à l'interface entre l'intelligence artificielle et les statistiques, avec des approches automatiques ou semi-automatiques.

2.3.2 Les différentes techniques de Data Mining

Comme nous l'avons vu dans la section 2.3.1, le data mining représente un ensemble de techniques ayant un objectif commun : explorer des données. Dans cette partie, nous passerons en revue ces techniques et nous les classerons selon différents critères.

Le data mining présente trois techniques principales (Tuffery, 2011) :

- Le regroupement (*clustering* en anglais) : cette technique correspond à l'organisation d'une collection de modèles (représentés par un vecteur de mesures ou un point dans un espace multidimensionnel) dans des classes basées sur la similarité entre ses membres (Jain et al., 1999). Il s'agit donc d'une approche non-supervisée. Les classes sont en nombre limité et ont deux caractéristiques :
  - Elles ne sont pas prédéfinies par l'analyste mais découvertes au cours de l'opération.
  - Elles regroupent les objets ayant des caractéristiques similaires et séparent les objets ayant des caractéristiques différentes.
- Le classement (*classification* en anglais) : cette technique permet de prédire de positionner un individu dans une classe, parmi un ensemble de classes prédéterminées par l'analyste. Il s'agit donc d'une approche supervisée.
- La recherche d'associations, qui visent à découvrir des règles au sein des données.

Ces techniques peuvent être (Fayyad et al., 1996; Tuffery, 2011) :

- Descriptives. Elles visent alors à représenter et à décrire les groupes au sein des données.
- Prédictives. Elles visent alors à prédire l'appartenance à un groupe.

2.3.2.1 Les méthodes descriptives

Il existe de nombreuses méthodes descriptives, parmi lesquelles ont peut citer la réduction de dimension (Jollife, 2002), les cartes de Kohonen (Jain et al., 1999; Jain et al., 2000; Kangas and Kohonen, 1996), la Classification par Agrégation des Similarités (Marcotorchino, 1982), ou encore la recherche d'association. Mais nous présenteron essentiellement les méthodes de clustering, que nous avons beaucoup utilisées durant ce travail de thèse.

Les méthodes à base de modèles géométriques se basent, pour fonctionner, sur des théories géométriques. Les données sont considérées comme des points au sein d'un espace multidimensionnel ou comme des vecteurs. Pour comparer deux individus, on se base sur la "distance" qui sépare les deux individus dans l'espace du jeu de données (Hammami, 2005). La distance peut être calculée selon plu-

sieurs modalités mais s'inspire en général de distances calculées en géométrie (par exemple, la distance euclidienne ou la distance de Manhattan) (Dumolard, 1999).

Un *clustering* correspond à un regroupement des données en classes homogènes : le but est de rechercher des types au sein du jeu de données (Vidal, 1955). Ces classes homogènes sont appelées *clusters* (Jain et al., 1999; Tan et al., 2006). Il existe deux types d'analyses typologiques (Jain et al., 1999) :

— Les méthodes de partitionnement : dans les méthodes de partitionnement, on sépare les objets en sous-ensembles non chevauchants (Tan et al., 2006).

— Les méthodes hiérarchiques : dans les méthodes hiérarchiques, on permet aux sous-ensembles d'avoir des sur-classes et on obtient ainsi un ensemble de classes incluses les unes dans les autres, organisées en arbre (Tan et al., 2006).

La méthode de partitionnement la plus connue et la plus simple est la méthode des K-means. La méthode K-means de base classe n individus, décrits par p variables quantitatives, en k classes, le nombre k étant donné par l'analyste. Le principe de cette méthode est le suivant (Tan et al., 2006; Tuffery, 2011) :

1. On part de k barycentres, donnés par l'analyste ou choisis aléatoirement.
2. On calcule pour chaque individu, la distance qui le sépare de chaque barycentre. La mesure de la distance est choisie par l'analyste. Une mesure de distance classique est la distance euclidienne.
3. On répartit les individus en k groupes : un individu est affecté au groupe dont il est le plus proche du barycentre.
4. On recalcule les barycentres des k groupes.
5. On répète les étapes 2, 3 et 4 jusqu'à ce que les barycentres n'évoluent plus.

Il existe cependant de nombreuses autres méthodes de partitionnement. Le choix d'une méthode dépend des caractéristiques du jeu de données auquel on souhaite appliquer la classification. Ces méthodes sont résumées dans la tableau 2.2.

Tableau 2.2 – Les méthodes de partitionnement, d'après Tufféry (Tuffery, 2011)

Nom de la méthode	**Paramètres d'entrée**	**Domaine privilégié**	**Forme des classes**
K-means	Nombre de classes	Classes séparées, grands effectifs	Sphérique
K-modes	Nombre de classes	Variables qualitatives, grands effectifs	
K-prototypes	Nombre de classes	Variables qualitatives et quantitatives, grands effectifs	
PAM	Nombre de classes	Classes séparées, petits effectifs, plus robustes que K-means.	Sphérique
CLARA	Nombre de classes	Assez grands effectifs	Sphérique
CLARANS	Nombre de classes Nombre maximum de voisins	Petits effectifs, classes de meilleure qualité que PAM et CLARA	Sphérique
Cartes de Kohonen	Nombre de classes	Classes séparées, grands effectifs	Sphérique

Un exemple classique de méthode hiérarchique est la Classification Ascendante Hiérarchique (CAH). Cette méthode se base sur des matrices des proximités : il s'agit d'un tableau contenant les distances entre tous les individus un à un. Les étapes de cette méthode sont les suivantes (Tan et al., 2006) :

1. Calcul de la matrice des proximités entre tous les individus.
2. Regroupement des deux individus les plus proches : le groupe ainsi formé sera considéré par la suite comme un individu, et ses coordonnées seront celles de son barycentre.
3. Répétition des étapes 1 et 2 jusqu'à la classification de tous les individus.

Cette méthode présente le résultat de la classification sous forme d'un arbre appelé *dendrogramme*.

Comme pour les méthodes de partitionnement, il existe différentes méthodes hiérarchiques, qui sont choisies selon le jeu de données dont on dispose. Ces méthodes

sont présentées dans la tableau 2.3.

Tableau 2.3 – Les méthodes hiérarchiques, d'après Tufféry (Tuffery, 2011)

Nom de la méthode	**Paramètres d'entrée**	**Domaine privilégié**	**Forme des classes**
CAH single linkage	Niveau de coupure dans le dendrogramme	Petits effectifs, classes de formes irrégulières	Allongée
CAH Ward	Niveau de coupure dans le dendrogramme	Petits effectifs, classes de meilleure qualité que K-means (voir 2.2, page 29)	Sphérique
CURE	Nombre de classes, nombre de représentants par classe	Forme quelconque de classes	Arbitraire
ROCK	Nombre de classes	Petits effectifs, variables qualitatives	
BIRCH	Nombre maximum de sous-classes d'un noeud intermédiaire, diamètre maximum des sous-classes des noeuds terminaux	Grands effectifs, variables qualitatives et quantitatives	Sphérique
CHAMELEON	Nombre des K plus proches voisins, taille des sous-classes construites, paramètre α	Forme quelquconque de classes	Arbitraire

2.3.2.2 Les méthodes prédictives

La prédiction vise à estimer la valeur d'une variable en fonction de la valeur d'autres variables, appelées "variables explicatives". Les techniques de prédiction se divisent en deux grandes opérations (Tuffery, 2011) :

— Le classement (ou discrimination), où la variable à prédire est qualitative.
— La prédiction (ou régression), où la variable à prédire est quantitative.

La prédiction ne doit pas être confondue avec la prévision, qui consiste à déduire la valeur d'une variable grâce aux valeurs précédentes prises par cette variable.

Il existe de types de techniques de classement et de prédiction :
— Les techniques *inductives*, dans lesquelles une phase d'apprentissage permet d'établir un modèle, qui sera ensuite appliqué aux nouvelles données.
— Les techniques *transductives*, dans lesquelles il n'y a pas de modèle : l'individu est classé par rapport aux individus déjà classés.

Certaines qualités sont attendues d'une technique de classement ou de prédiction, et peuvent être mesurées par différents paramètres :
— La précision : la proportion d'individus mal classés doit être la plus basse possible.
— La robustesse : le modèle construit doit dépendre le moins possible de l'échantillon d'apprentissage utilisé, avoir une durée de vie maximale et les variables sur lesquelles le modèle s'appuie doivent être solide.
— La concision : les règles du modèle doivent être aussi simples et aussi peu nombreuses que possible.
— Les résultats sont explicites : les règles du modèle sont accessibles et compréhensibles.
— La rapidité de calcul du modèle.
— Les possibilités de paramétrage.
— La diversité des types de données manipulées.

La mise en place d'un système prédictif suit généralement les étapes suivantes :

1. Identification des données en entrée et en sortie.
2. Normalisation des données.
3. Constitution d'un modèle avec la structure adaptée.
4. Apprentissage.
5. Test.
6. Application du modèle généré par l'apprentissage.
7. Dénormalisation des données en sortie.

Modèles à base de règles logiques

Un arbre de décision est un classifieur qui réalise une partition récursive de l'espace formé par les données. Dans un arbre de décision, chaque noeud coupe l'espace en deux (ou plus) sous-espaces selon la valeurs prise par la variable en entrée. Chaque noeud est donc un test. Dans les cas les plus fréquents, chaque

test prend en compte un seul attribut qui permet de scinder l'espace des données selon la valeur prise (Rokach et al., 2008). Il existe plusieurs méthodes pour construire des arbres de décision, dont les trois plus connues sont : CHAID, C5.0 et CART(Breiman, 1984; Murthy, 1997).

Modèles à base de fonctions mathématiques

Réseaux de neurones

Un réseau de neurones est un ensemble de *noeuds* (un noeud étant aussi appelé *neurone formel* ou *unité*) connectés entre eux (Bishop, 1995; Hastie et al., 2008; Jain et al., 2000; Tuffery, 2011). Chaque variable continue en entrée (ou chaque modalité d'une variable qualitative en entrée) correspond à un noeud de premier niveau appelé "couche d'entrée". Lorsque le réseau de neurones est prédictif, il y a une ou plusieurs variables à expliquer. Chaque variable continues (ou chaque modalité d'une variable qualitative) à expliquer correspond à un noeud de dernier niveau, appelé "couche de sortie". Les niveaux intermédiaires entre la couche d'entrée et la couche de sortie sont appelés "couches cachées". Dans les réseaux de neurones classiques, les noeuds sont constitués de deux fonctions :

— Une *fonction de combinaison* qui calcule une valeur à partir des noeuds connectés en entrée et des poids des connexions.
— Une *fonction de transfert* (ou *d'activation*) qui est appliquée à la valeur calculée par la fonction de combinaison puis transférée aux noeuds de la couche suivante.

L'apprentissage permet de fixer le poids des connexions, soit quand un optimal a été atteint, soit quand le réseau à effectuer un nombre d'itérations considérés comme suffisant. Les réseaux de neurones à apprentissage supervisé sont prédictifs tandis que les réseaux de neurones à apprentissage non-supervisés sont descriptifs.

Modèles paramétriques ou semi-paramétriques

En statistique, on dit qu'un problème est (Delyon, 2012) :

— *Paramétrique* si les données de départ sont paramétrées par un vecteur de dimension finie. Dans ce cas, la résolution d'un problème paramétrique se résume à l'identification de ce paramètre.
— *Non-paramétrique* s'il s'agit d'estimer un vecteur de dimension infinie.
— *Semi-paramétrique*, si on cherche à estimer un vecteur de dimension finie mais que les données de départ ne sont pas paramétrées par un vecteur de dimension finie.

Il est possible de se baser sur des modèles paramétriques (ou semi-paramétriques) pour faire de la prédiction. On estime alors que certaines lois statistiques classiques existent au sein de la population étudiée. Le choix d'une méthode dépend avant tout du type de données. Parmi les méthodes basées sur les modèles paramétriques ou semi-paramétriques, on trouve les régressions linéaires simples et multiples, les régressions PLS, les régressions logistiques, les modèles linéaires généralisés, les analyses de variances simples et multiples (Tuffery, 2011).

Utiliser un modèle paramétrique signifie que l'on fait des hypothèses fortes sur les données : il faudra s'assurer que ces hypothèses sont valables avant d'employer ce type de méthodes.

Prédictions sans modèle

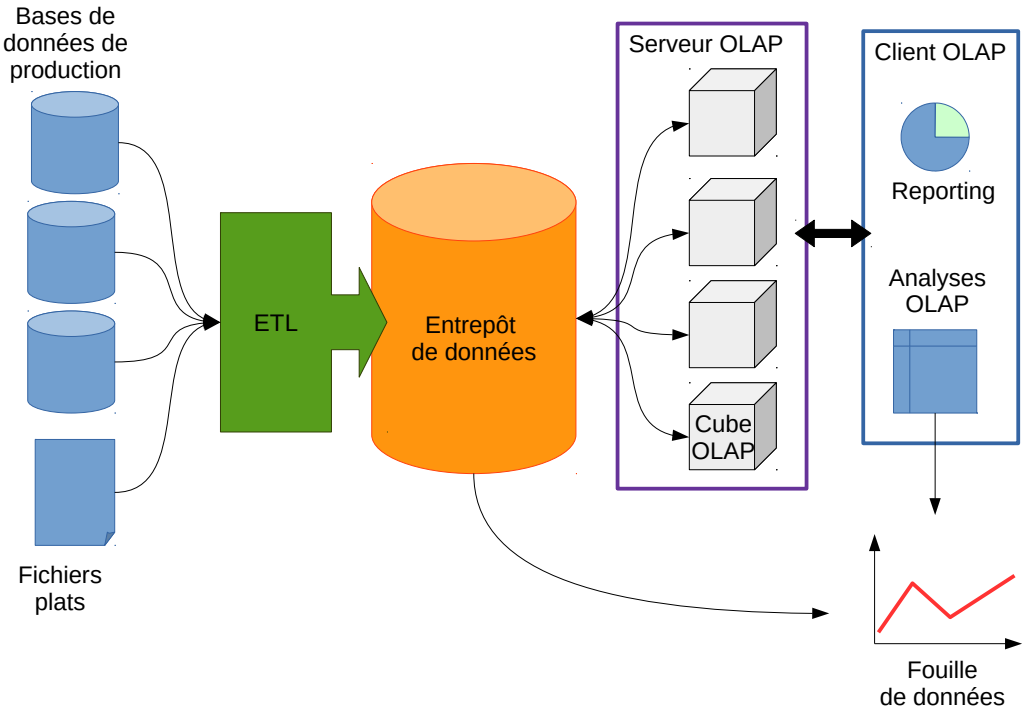
Les prédictions sans modèle se basent sur ce que l'on appelle une analyse probabiliste : le principe est de calculer la probabilité d'un évènement dans notre population, représentée par un jeu de données. Une de ces techniques est l'Analyse Discriminante Probabiliste. Le cadre d'une analyse discriminante est le suivant : les variables à expliquer sont qualitatives et les variables explicatives sont quantitatives. L'analyse discriminante se fonde sur le théorème de Bayes.

2.4 Conclusion

Dans ce chapitre, nous avons présenté les principaux concepts que nous avons manipulé pendant les travaux de recherche menés durant cette thèse. Nous avons proposé une définition des termes "Analyse en ligne", "Entrepôt de données" et "Fouille de données". En plus de ces définitions générales, nous avons détaillé plusieurs points techniques sur chacune des trois thèmes abordés dans cette partie.

Ainsi, nous avons détaillé la mise en place des systèmes d'analyse en ligne, ainsi que les opérateurs utilisés par ce type d'analyse. Nous avons également présenté la modélisation et la mise en place d'un entrepôt de données. Enfin, nous avons présenté quelques exemples illustrant les principes utilisés par les méthodes de fouille de données.

L'articulation entre analyse en ligne, entrepôt de données et fouille de données est présentée sur la Figure 2.4. L'entrepôt de données stocke les données nécessaires, après que ces données aient été extraites, transformées et chargées dans l'entrepôt grâce à un outil ETL (*Extract-Transform-Load*). Les données de cet entrepôt peuvent ensuite être transmises à l'utilisateur via des rapports,explorées par l'utilisateur via une interface d'analyse en ligne, ou analysées grâce à la fouille de



Sur le schéma ci-dessus, les flèches représentent les échanges de données.

Figure 2.4 – Schéma général d'un système d'information décisionnel

données. Cependant, comme nous le verrons dans la Partie II, la fouille de données peut également être utilisée pour concevoir le schéma de l'entrepôt de données et des cubes OLAP associés (Bentayeb, 2008; Choong et al., 2008; Lau et al., 2000; Usman et al., 2010).

Pour compléter cet état de l'art général, chaque contribution présentée dans la Partie II propose une revue de littérature relative au thème précis de la contribution.

Chapitre 3

Cas applicatif : Les données issues d'une étude de la biodiversité

Dans ce chapitre, nous décrirons le cas d'étude qui a été le support de cette thèse. Nous commencerons par décrire la problématique et les objectifs qui ont conduit au recueil de ce jeu de données. Puis, nous proposerons une description des données collectées.

3.1 Étudier les relations entre les espèces et leur environnement

Tous les êtres vivants évoluent dans des environnements complexes dynamiques appelés écosystèmes (Blew, 1996). Ces écosystèmes sont le résultat des interactions entre les facteurs biologiques (biotiques), géologiques, édaphiques, hydrologiques, climatiques (abiotiques), et anthropiques. Ces facteurs agissent à différentes échelles spatio-temporelles et à des intensités variées.

Un des enjeux de l'écologie est de décrire les phénomènes dynamiques spatialement et temporellement, et de comprendre les facteurs qui déterminent la distribution et l'abondance des organismes. Cela peut conduire à intégrer deux axes d'études fondamentaux complémentaires et intimement liés (Barbault, 1995) : la dynamique et le fonctionnement des populations. On cherche ici à décrire la nature des populations (animale, végétale, microbienne, ...), leur structure (organisation génétique, sociale, ...), leur dynamique (natalité, mortalité, ...), leur interaction avec d'autres populations, la dynamique et le fonctionnement des écosystèmes et des paysages. On ne s'intéresse plus ici uniquement à des objets biologiques mais

également à des objets physiques.

Ainsi, la recherche écologique actuelle, notamment du fait des questions posées par les gestionnaires des espaces naturels et anthropisés, s'oriente en partie vers l'étude du fonctionnement des systèmes écologiques hétérogènes. Cette considération de l'hétérogénéité s'accompagne nécessairement de l'analyse du contexte spatial et temporel des systèmes écologiques et des relations espèces-environnement. Analyser ces relations nécessite alors inévitablement de prendre en compte l'échelle spatiale à laquelle on l'étudie. Cette notion d'échelle est importante en écologie et plus particulièrement lorsque l'on s'intéresse à la dynamique paysagère.

L'intégration de la structure spatiale et de ses changements dans le temps constitue donc un élément incontournable pour la compréhension des processus écologiques, et plus particulièrement si les facteurs intègrent les activités anthropiques, qui altèrent les paysages, et surtout génèrent ses propres et nombreux changements qui repositionnent sans cesse les espaces dans des processus dynamiques plus ou moins stables. Ici, il ne s'agit plus de se limiter à une simple description des relations espèces-environnement mais de les modéliser, c'est-à-dire de proposer, à l'aide d'outils mathématiques, statistiques et informatiques, une généralisation de la description. On construit un système de relation espèces-environnement théorique, un modèle [1], dans le but de prévoir les évolutions du système réel. Ce genre d'étude est également très intéressante pour les gestionnaires de l'environnement. Ces études peuvent être menées à des échelles spatiales très variées, du très petit (étude de l'impact d'un vermifuge sur la biodiversité au sein d'une bouse de vache) au très grand (suivi des migrations au sein d'une population de thons rouges). Il s'agit donc d'un paramètre essentiel dans la réalisation d'étude écologique. Ainsi, l'étude des relations espèces-environnement est une étape incontournable de toute étude écologique, quel que soit l'objectif, le sujet et l'échelle de cette dernière.

3.2 Cas des suivis de population d'oiseaux dans les écosystèmes fluviaux

3.2.1 Quel est l'intérêt d'étudier les écosystèmes fluviaux ?

Les rivières sont des écosystèmes porteurs d'enjeux majeurs pour notre société actuelle. D'une part, elles représentent une manne économique. En effet, le trans-

1. Soit un observateur B, qui étudie un objet A. M est un modèle de A si B peut utiliser M pour répondre aux questions qui l'intéresse sur A (Minsky, 1965).

port de matériaux (via les voies navigables), la production d'énergie (via les systèmes hydroélectriques), l'agriculture (via la pisciculture et l'irrigation), la production de ressources en eau et le tourisme, s'appuient, dans notre pays, en partie sur ces milieux aquatiques. D'autre part, ils représentent une forte valeur écologique. L'ensemble de conditions uniques qui existent autour d'une rivière (lit mineur et lit majeur) en fond l'habitat de nombreuses espèces caractéristiques, dont certaines, par leur rareté, sont porteuses d'une forte valeur patrimoniale. Il n'est donc pas étonnant que ces milieux soient un enjeu national et international pour l'État, à travers différents échelons administratifs et instituts, qui organise leurs usages, leur protection, leur conservation et finance de nombreuses recherches sur les milieux aquatiques.

3.2.2 Le cadre de l'étude : la Loire

La présente étude s'inscrit dans le contexte d'une analyse à long terme avec parmi les perspectives d'alimenter les outils d'aide à la gestion des hydrosystèmes fluviaux. Dans ce cadre, le choix s'est porté sur le fleuve Loire (Figure 3.1) pour les raisons suivantes. D'une part, il est non seulement le principal cours d'eau français, mais c'est aussi l'un des plus grands fleuves d'Europe et, de surcroît, l'un des rares à être encore peu altérés par les aménagements lourds (Nabet, 2013). D'autre part, ce fleuve est un excellent modèle pour l'étude des gradients écologiques, en développant le long de son cours une grande variété de milieux.

Ce fleuve mesure un peu plus de 1000 km de long de sa source, à 1410 mètres d'altitude à son débouché dans l'océan Atlantique. Il nous offre donc une large échelle spatiale pour notre étude. Elle parcourt successivement une zone torrentueuse d'environ 60 km, une zone de moyenne montagne moins pentue sur environ 100 km, une très grande zone de plaine (800 km) à pente régulière et encore relativement forte et enfin un estuaire de 80 km de long. Son cours traverse des habitats très variés soumis dans le lit inondable à la dynamique fluviale (grèves, saulaies, ripisylves, berges érodées) et plus au large modelé par une ancienne occupation anthropique (alternance de forêts, prairies, cultures sur labour, agglomérations, falaises...). Ces habitats sont susceptibles d'être modifiés, parfois rapidement, sous l'effet des facteurs naturels (régime des crues) ou humain (changement des pratiques agricoles, ouvrages hydrauliques...). Cette distribution spatio-temporelle des habitats nécessite de pouvoir les décrire à l'échelle du cours entier.

3.2.3 Pourquoi le modèle ornithologique dans l'étude des écosystèmes fluviaux ?

De nombreux travaux s'intéressent à l'avifaune dans le cadre des suivis écologiques et d'analyses des relations espèces-environnement (par exemple : (Power et al., 1995; Manel et al., 1999; Faragó and Hangya, 2012)). Ils analysent les causes et les conséquences des variations temporelles et spatiales des effectifs des populations d'oiseaux (déclin, augmentation ou stabilité des espèces et des communautés). Ils discriminent l'influence de différents facteurs d'origines anthropiques ou non (changements climatiques, modification des pratiques agricoles, urbanisation et fragmentation des paysages). Finalement, il est important de noter que les oiseaux sont de bons indicateurs de l'état de santé des écosystèmes et particulièrement des écosystèmes forestiers ou fluviaux. Dans le cadre d'une étude écologique appliquée à un grand cours d'eau, les oiseaux présentent de nombreux intérêts (Roché and Frochot, 1993) :

- Ils sont présents sur l'ensemble du cours d'eau, des régions montagneuses proches de la source au bord de mer entourant l'estuaire. Leur échelle spatiale de vie et leur perception de l'environnement intègre le milieu aquatique mais également les berges et la vallée.
- Ils ne sont pas aussi dépendants de l'eau que les espèces purement aquatiques comme les poissons ou les invertébrés benthiques [2]. On n'enregistre pas, comme avec les insectes ou les petits mammifères, de fluctuations interannuelles importantes.
- Les populations ne sont pas génétiquement perturbées par des hybridations, comme c'est le cas avec les espèces végétales. La colonisation des milieux par les oiseaux est essentiellement "naturelle" : il y a peu de lâchers massifs (tels que ceux qui affectent les communautés de poissons) et peu d'espèces invasives.
- Les oiseaux sont facilement observables, et le nombre restreint d'espèces facilite les inventaires à grande échelle. L'étude intègre une large échelle spatiale, puisqu'elle recouvre le cours entier de la rivière (1000 km) et que l'on prend en compte le milieu aquatique mais également l'ensemble de la vallée fluviale.

Pour réaliser cette étude, on va donc s'intéresser aux relations entre les peuplements d'oiseaux et le paysage le long de Loire. C'est dans ce cadre qu'a été créé le programme STORI (Suivi Temporel des Oiseaux nicheurs en Rivière).

2. Le benthos est l'ensemble des organismes aquatiques qui vivent sur le fond des océans, des lacs et des cours d'eau.

3.3 Données liées au cas d'étude

3.3.1 Le programme STORI

À l'origine financé par le ministère de l'environnement (Direction de l'eau, Direction de la Nature et des paysages et DIREN Centre) dans les années 1990 puis par le FEDER Loire à travers le "Plan Loire Grandeur Nature" depuis les années 2000, le programme STORI (Suivi Temporel des Oiseaux nicheurs en Rivière) a été créé en 1989 par trois chercheurs : Bruno Faivre (Université de Bourgogne), Bernard Frochot (Université de Bourgogne), Jean Roché (consultant en environnement et spécialiste des oiseaux).

Ce programme s'intéresse particulièrement aux communautés d'oiseaux présentes le long de la Loire et de l'Allier, ainsi qu'aux paysages observés dans les vallées fluviales de ces rivières. Il a trois objectifs principaux. Le premier est de comprendre les relations espèces-environnement et plus particulièrement comment le paysage (diversité des habitats, occupation des sols, ...) impact la répartition des espèces d'oiseaux le long du cours d'eau. Le second objectif du programme STORI est le suivi communautaire, qui vise à apprécier les changements temporels des communautés d'oiseaux sur des pas de temps pluriannuels et à explorer les facteurs globaux et locaux pouvant expliquer ces changements. Le troisième et dernier objectif du programme STORI est l'élaboration d'outils de bio-indication : le but est de faire des communautés d'oiseaux des indicateurs de la qualité des milieux fluviaux en se basant sur les relations établies entre les espèces et l'environnement et la dynamique temporelle observée au sein des populations.

L'approche développée par le programme STORI est originale par rapport aux autres travaux répondant à des préoccupations semblables. En effet, les espèces bio-indicatrices classiques de l'état des rivières (les poissons ou les insectes) ne permettent pas la prise en compte de la vallée, car il s'agit d'espèces aquatiques, très inféodées à l'eau (Legube and Merlet, 2009). Les approches agro-environnementales actuelles sont généralement centrées sur une seule espèce, or ici, c'est l'échelon communautaire qui est le centre d'intérêt principal. L'approche communautaire a une pertinence écologique accrue, l'écosystème étant défini comme l'ensemble des interactions entre les espèces vivant dans un milieu physique donné. Cette étude originale par son approche communautaire et son échelle spatiale, a permis de récolter un jeu de données de vaste amplitude temporelle (de 1989 à 2011), spatiale (la Loire mesure plus de 1000 kilomètres de long) et biologique (environ 200 espèces recensées).

3.3.2 Les jeux de données à disposition

3.3.2.1 Les données ornithologiques

Les données ornithologiques sont récoltées sur le terrain grâce au programme STORI. Pour recenser ces espèces, un protocole précis a été mis en place basé sur la méthode des Indices Ponctuels d'Abondance (IPA) (I.B.C.C., 1977; Blondel et al., 1981). La méthode des IPA consiste à dénombrer, en un point précis appelé station de comptage, toutes les espèces d'oiseaux nicheurs (on exclut les espèces en migrations ou en hivernage) repérées par contact visuel ou auditif sans limitation de distance. Le long de la Loire, 198 stations de comptage, ou points d'écoute, ont été définies. Grâce à cette méthode, on peut obtenir, à chaque station de comptage, un indice semi-quantitatif non-paramétrique entre 0 et 5 décrivant l'abondance de chaque espèce. Au final, les données ornithologiques sont constituées d'un ensemble d'indices d'abondance, relevés pour 213 espèces d'oiseaux, en 198 points définis par leurs coordonnées géographiques, pour quatre campagnes de recensement (1990, 1996, 2002 et 2011). Quelques exemples de points d'écoute, ou stations, sont localisés sur la Figure 3.1.

Les données ornithologiques relevées grâce à la méthode IPA ont de nombreuses qualités. La méthode IPA a permis des relevés standards à une large échelle spatio-temporelle. Cependant, elle présente des caractéristiques qui peuvent perturber les algorithmes de calcul des modèles. Par exemple, établir un modèle linéaire généralisé (GLM) nécessite de vérifier que la variable à prédire présente une distribution classique, suivant par exemple la loi de Poisson. Tout d'abord, la distribution de nombreuses espèces, notamment les passereaux, ne sont pas des distributions classiques. On remarque en règle générale que les valeurs avec une décimale (0.5, 1.5, 2.5, etc. ...) sont sous-représentées dans les distributions. Ces distributions sont directement imputables à la méthode de détection. En effet, les passereaux sont généralement détectés grâce au chant (cotation 1 pour mâle chanteur) et peu détectés grâce à la vue (cotation 0.5 pour une simple observation d'un individu). A titre d'exemple, la Figure 3.2 présente une comparaison entre les distributions du Héron cendré (*Ardea cinerea*) et de la Fauvette grisette (*Sylvia communis*). Les valeurs décimales sont sous-représentées dans la deuxième distribution de la deuxième espèce qui est surtout détectée au chant, ce qui n'est pas cas pour la première où les individus sont simplement observés.

Pour se rapprocher d'une distribution statistique classique, comme une loi de Poisson, on peut arrondir les IPA à l'entier supérieur : on obtient ainsi une distribution très proche de la distribution observée et d'une distribution statistique classique.

Outre ces distributions qui peuvent être corrigées, les abondances observées

grâce à la méthode IPA présentent une grande variabilité spatiale. Cette variabilité peut être expliquée par la détectabilité des espèces. En effet, les abondances relevées sont comme « bruitées » par la chance qu'a l'opérateur de voir ou d'entendre l'espèce. Pour lisser les abondances et atténuer l'effet de la détectabilité des espèces, on peut appliquer un filtrage de Fourier (Duhamel and Vetterli, 1990) aux données ornithologiques. Un exemple de courbe obtenue grâce à cette méthode est présentée sur la Figure 3.3.

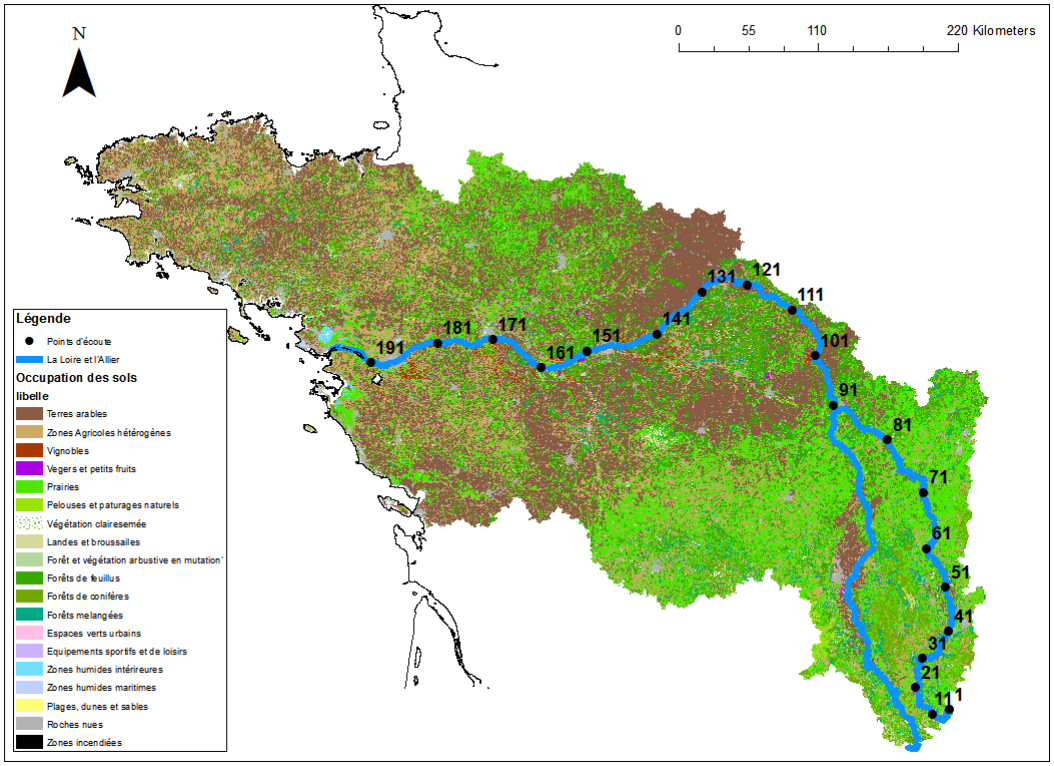


Figure 3.1 – Le bassin versant de la Loire, et quelques points d'écoute le long du fleuve (*Source des données : outil MAGDALNA, Agence de l'eau Loire-Bretagne*)

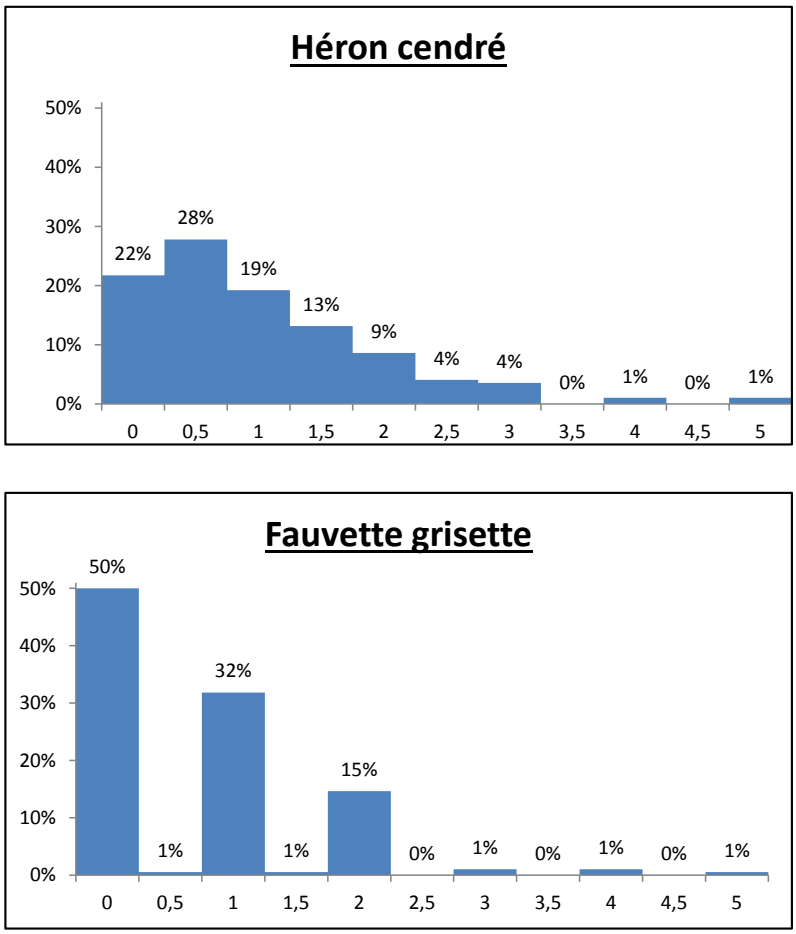


Figure 3.2 – Distributions des données IPA du Héron cendré (*Ardea cinerea*) et de la Fauvette grisette (*Sylvia communis*) en 1990

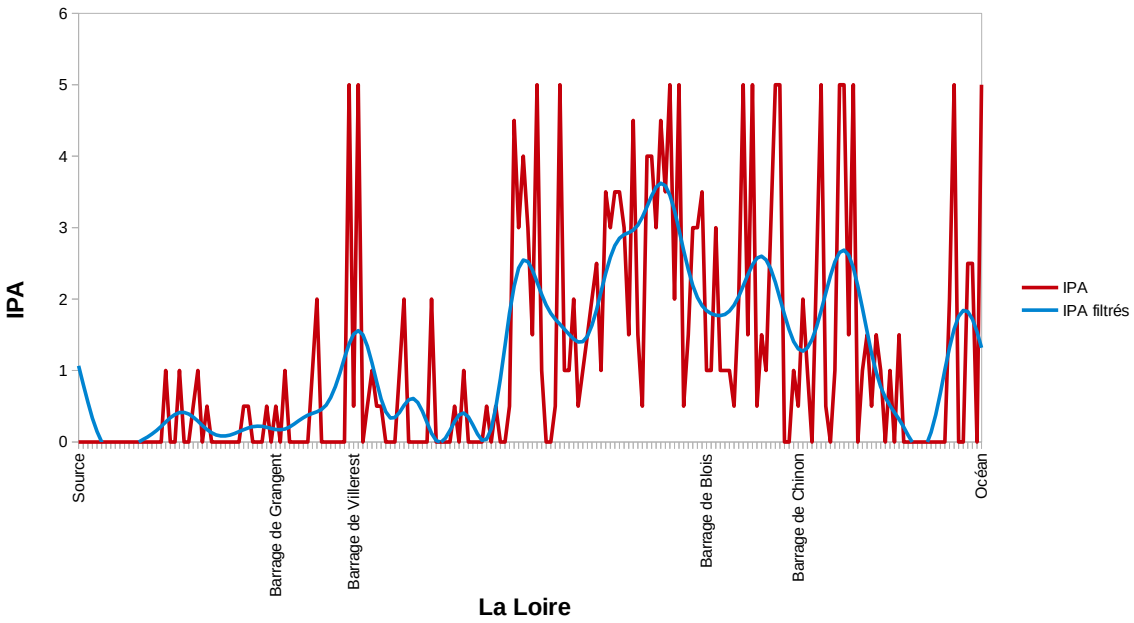


Figure 3.3 – Résultat du filtrage de l'abondance du Canard colvert (*Anas platyrhynchos*) le long de la Loire grâce à la transformation de Fourier rapide

3.3.2.2 Les données environnementales

Les données environnementales, décrivant la rivière et le paysage sur les berges, sont issues de deux sources principales :

— Des relevés de terrain effectués en même temps que le recensement des espèces (Roché, 2010).

— De l'analyse d'images satellites (Journaux et al., 2005).

— Des bases de données mise à disposition par les organismes publics. Des données permettant une description fine du milieu, ont été recherchées auprès des organismes partenaires du programme STORI. Ainsi, l'Agence de l'eau Loire-Bretagne a fourni les données issues des Systèmes d'Information Géographiques (SIG) Magdalena (Mise A Disposition Graphique des Données Alphanumériques Liées à l'Environnement Aquatique) et SYRAH (Système Relationnel d'Audit de l'Hydromorphologie des cours d'eau). De plus, la DREAL (Direction Régionale de l'Environnement, de l'Aménagement et du

Logement) de la région Centre met en ligne, à disposition du grand public, les données issues du SIEL (Système d'Information des Évolutions du lit de la Loire[3]). Ces différents systèmes proposent des cartes variées qui décrivent le réseau hydrographique (cours d'eau, masses d'eau, plan d'eau), les usages (pisciculture, navigation, barrage ...), le bassin versant (occupation des sols, agriculture, végétation, industrie, ...).

Les relevés de terrain décrivent le lit de la rivière (altitude, largeur de la vallée, courant, type de berges, substrat, etc.) avec des variables locales. Les analyses des images satellites décrivent la vallée à plus large échelle (pourcentage de la surface en forêt, en prairie, connectance, diversité de la végétation, etc.) avec différentes variables issues du traitement d'image ou de l'analyse de cartes. Chacune des variables est renseignées pour les 198 stations de comptage définies pour les recensements des espèces d'oiseaux.

La liste de l'ensemble des variables est présentée en Annexe (voir Chapitre 10).

Il faut noter que de nombreuses variables environnementales sont dépendantes du temps. D'une part, certaines d'entre elles peuvent avoir des valeurs qui varient avec le temps. D'autre part, de nombreuses variables environnementales ne sont pas disponibles pour l'ensemble des campagnes de recensement des oiseaux. Quelques exemples illustrant la dépendance temporelle des données environnementales :

— L'altitude d'un point d'écoute n'est pas une variable dépendante du temps, car on peut aisément faire l'hypothèse que cette données n'a pas évoluée entre 1990 et 2012. Ainsi, peu importe la date de collecte de cette données (voir le Tableau 10.2, Annexe 10.1).

— Le pourcentage de surface au sol occupée par les forêts de conifères autour d'un point d'écoute est en revanche une donnée susceptible d'évoluer entre 1990 et 2012. Cette données a été relevée sur une image satellite uniquement en 2001 et sera donc disponible uniquement pour la campagne de recensement de 2002 (voir Annexe 10.2).

3.4 Conclusion

Dans ce chapitre, nous avons présenté le cas d'étude qui a inspiré la question de recherche développée dans cette thèse (Chapitre 1) et qui a été le support de toutes nos contributions (Chapitres 4 à 6).

Notre cas d'étude est issu d'un travail de recherche en écologie. Ces travaux visent à étudier les relations entre peuplement d'oiseaux et environnement le long d'une vallée fluviale, celle de la Loire. Menés sur un temps long (plus de 20 ans),

3. http ://www.centre.developpement-durable.gouv.fr/le-siel-r104.html

ces travaux ont permis de récolter un jeu de données important, qui doit désormais être analysé.

Les données disponibles dans notre cas d'étude sont de deux types. D'une part, nous disposons de données ornithologiques, qui recensent et quantifient 213 espèces d'oiseaux, en 198 points géographiques définis le long du fleuve, lors de 4 campagnes de recensement (1990,1996, 2002 et 2011). D'autre part, la vallée fluviale est décrite par 158 variables, issues de sources de données variées. Ces données environnementales décrivent les 198 points spatiaux, définis pour recenser les oiseaux, à différentes échelles spatiales et temporelles.

Les données environnementales constituent un défi pour la modélisation multidimensionnelle de ce jeu de données, car elles sont :

— Complexes : la description des points spatiaux intègre de nombreux attributs.
— Hétérogènes : les sources de données étant variées, la nature des attributs l'est également. Certains attributs sont quantitatifs et d'autres qualitatifs.
— Inconsistantes : certaines sources de données étant indépendantes du programme de recensement des oiseaux, elles ne sont pas disponibles lors de toutes les campagnes de recensement.

Deuxième partie

Contributions

Chapitre 4

Construction automatique de hiérarchies au sein d'une dimension

Ce chapitre est consacré au second objectif, défini dans le Chapitre 1.2, intitulé "Proposer une méthode permettant de construire automatiquement une hiérarchie avec des membres décrits par des attributs quantitatifs, des attributs qualitatifs et sachant que certaines données peuvent être manquantes". **Le contenu de ce chapitre a été publié dans la revue *Ecological Informatics* en 2015. La référence complète de cet article est présente dans la bibliographie du présent document au numéro(Sautot et al., 2015).**

Ce chapitre est organisé en deux sections :

— La Section 4.1 propose une synthèse (en français) du contexte, de la problématique, de la méthodologie et des résultats proposés dans (Sautot et al., 2015). Cette synthèse proposera également une conclusion sur cette contribution et replacera les résultats obtenus dans le contexte de la thèse.

— La Section 4.2 correspond au texte publié dans la revue *Ecological Informatics* (en anglais).

4.1 Synthèse sur la construction automatique de hiérarchies avec des données mixtes et manquantes

Dans ce chapitre, nous nous sommes intéressés au type des données : comment créer automatiquement des hiérarchies pour des membres d'une dimension décrits par des données mixtes et qui peuvent être manquantes. Dans cette partie introductive, nous proposons une synthèse de l'article, reproduit dans la section suivante.

Dans cet article, nous commençons par présenter un état de l'art qui concerne la conception automatique d'entrepôts de données, et particulièrement, la conception de hiérarchies. Les travaux présentés dans cet état de l'art présentent deux aspects intéressants. Premièrement, ces travaux proposent une modélisation *a posteriori* du schéma OLAP, réalisée soit par l'utilisateur, soit par un algorithme. Par ailleurs, ces travaux offrent à l'utilisateur la possibilité de construire son propre schéma OLAP en fonction de la structure de ses propres données. L'article présenté dans la section suivante s'inspire de ces deux aspects, et propose un système automatique qui offre à l'utilisateur la possibilité de construire des hiérarchies au sein d'un schéma OLAP grâce à une méthode de fouille de données.

Lors d'une étude en biologie, les mesures et les dimensions sont, en général, clairement identifiées. Mais les données qui décrivent une dimension n'ont pas nécessairement de structure hiérarchique apparente :

— Une dimension peut être décrite par plusieurs attributs quantitatifs, et pas uniquement par des catégories.
— Les attributs d'une dimension peuvent être hétérogènes : les membres de la dimension peuvent être décrits par des attributs quantitatifs, catégoriels et binaires.
— Il peut y avoir des valeurs manquantes.

Les travaux présentés dans l'état de l'art proposent différentes méthodes pour construire de nouvelles hiérarchies au sein d'un schéma OLAP existant, parfois grâce à une algorithme de fouille de données, mais ces méthodes ne sont pas capables de prendre en compte l'hétérogénéité de nos données, ni les valeurs manquantes. Nous avons donc souhaité compléter ces travaux en proposant une méthode capable de prendre en compte ces caractéristiques de nos données.

Le système que nous avons proposé fonctionne en six étapes. Tout d'abord , le système récupère, au sein de l'entrepôt de données, les données et les métadonnées nécessaires au calcul de la nouvelle hiérarchie au sein d'une dimension (*étape 1*). Ensuite, le système identifie automatiquement le type de chaque attribut dimen-

sionnel, c'est à dire, si chaque attribut est qualitatif ou quantitatif (*étape 2*). Cette étape est détaillée ci-dessous. Après cela, le système calcule automatiquement une nouvelle hiérarchie à partir des attributs descriptifs des membres de la dimension considérée (*étape 3*), et enregistre le résultat dans l'entrepôt de données (*étape 4*). Le système utilise une classification ascendante hiérarchique. Cette étape sera également détaillée dans un des paragraphes suivants. L'étape suivante consiste à incorporer automatiquement la nouvelle hiérarchie calculée au fichier XML décrivant le cube OLAP (*étape 5*). Pour finir, le système publie le cube OLAP modifié sur le serveur OLAP (*étape 6*). A la fin de cette étape, le cube avec la nouvelle hiérarchie est disponible pour les utilisateurs du serveur OLAP.

Dans ce paragraphe, nous allons détailler l'étape 3 du fonctionnement général proposé dans le paragraphe précédent, car il s'agit de notre principal apport par rapport aux travaux existants. Pour calculer automatiquement une nouvelle hiérarchie, nous proposons d'utiliser une classification ascendante hiérarchique. En effet, le résultat de cet algorithme est un arbre binaire, dont la structure est compatible avec la définition d'une hiérarchie OLAP. Cependant, pour prendre en compte les particularité de nos données (données mixtes et manquantes), nous avons utilisé cet algorithme avec une métrique particulière : l'indice de similarité de Gower.

L'utilisation de l'indice de Gower implique de savoir quel attribut est qualitatif et quel attribut est quantitatif. Ainsi, pour utiliser la classification ascendante hiérarchique, le système doit déterminer quel attribut est qualitatif et quel attribut est quantitatif. C'est pourquoi nous avons proposé, à l'étape 2, une méthode pour déduire le type d'un attribut à partir du type de données (texte ou donnée numérique) et du nombre de valeurs observées, basée sur un arbre de décision. Ainsi, nous avons mis à jour des règles explicites pour reconnaitre automatiquement un attribut quantitatif d'un attribut qualitatif.

Enfin, nous avons évalué les performances du système que nous avons proposé en termes de temps de calcul et de mémoire nécessaire. Ces performances sont compatibles avec une utilisation de ce système dans la cadre de la conception d'un schéma OLAP.

En conclusion, nous avons proposé une méthode et un outil pour construire, au sein d'une dimension, de nouvelles hiérarchies basées sur les attributs descriptifs des membres de la dimension. Avec notre proposition, ces attributs peuvent être qualitatifs et quantitatifs, et intégrer des données manquantes. Cependant, à ce stade, nous ne pouvons pas construire complètement un entrepôt de données. En effet, cette contribution permet uniquement de construire des hiérarchies à partir de données déjà intégrées à la dimension considérée. Dans les chapitres suivants, nous intéresserons donc à la construction complète du schéma

du futur entrepôt grâce à une méthode de prototypage (Chapitre 5) et l'enrichissement d'une dimension avec des données issues d'autres cubes (Chapitre 6).

4.2 The Hierarchical Agglomerative Clustering with Gower index, a methodology for automatic design of OLAP cube in ecological data processing context

Abstract

The OLAP systems can be an improvement for ecological studies. In fact, ecology studies, follows and analyzes phenomenon across space and time and according to several parameters. OLAP systems can provide to ecologists browsing in a large dataset. One focus of current research on OLAP system is the automatic design of OLAP cubes and of data warehouse schemas. This kind of works makes accessible OLAP technology to non Information Technology experts. But to be efficient, the automatic OLAP building must take account into various cases.

Moreover the OLAP technology is based on the concept of hierarchy. Thereby the hierarchical clustering methods are often used by OLAP system designer.

In this article, we propose using hierarchical agglomerative clustering with a metric that comes from ecological studies (the Gower similarity index) to build automatically hierarchical dimensions in an OLAP cube. With this similarity index we can perform a hierarchical clustering on heterogeneous datasets that contains qualitative and quantitative variables.

We offer a prototypical automatic system which builds dimension for an OLAP cube and we measure the performances of this system according to the number of clustered individuals and according to the number of variables used for clustering. Thanks to these measures we can offer an approximation of performances with a large dataset.

Thereby the Gower index in a hierarchical agglomerative clustering permits the management of heterogeneous dataset with missing values in a context of automatic building of OLAP cube. With this methodology, we can build new dimensions based on hierarchies in the data, which are not evident. The data mining methods can complete the expert knowledge during the design of an OLAP cube, because these methods can explain the inherent structure of the data.

Keywords

OLAP ; Hierarchical Agglomerative Clustering ; Bird Population ; Automatic Design

4.2.1 Introduction : use data mining for OLAP cube design

Since 1993, OLAP (On Line Analytical Processing) systems have been proposed to improve decision making process due to analysis of large datasets (Codd et al., 1993). This kind of software is designed to explore easily and quickly multidimensional data (Rivest et al., 2005). The word OLAP can be associated with a process, a kind of system or a kind of data (Jerbi et al., 2009). A basic Relational OLAP (ROLAP) system architecture consists of (i) a relational Data Base Management System (DBMS), that stores data in accordance with data warehousing paradigm ; (ii) an OLAP server that implements the multidimensional model and OLAP operators on top of the DBMS ; (iii) an OLAP client, that combines and synchronizes tabular and graphical displays and allows query building ; (iv) an ETL tool that extracts data from heterogeneous sources, transforms them and loads them into a data warehouse.

In this paper, we are focused on design of OLAP schema, which is define by Usman as a collection of database objects, including tables, views, indexes and synonyms (Usman et al., 2010).

Several research works suggest modeling for OLAP schema, that either rely on existing models (Entity/Relationship, Object-Oriented, ...) or suggest new models (Lehner, 1998; Nguyen et al., 2000; Pedersen and Jensen, 1999; Tsois et al., 2001). Regardless of the methods chosen by the authors to define the rules of their models, these models are based on three concept of multidimensional modeling : *measures*, *dimensions* and *hierarchies*(Jerbi et al., 2009).

Measures are defined as dynamical and dependent variables (Nguyen et al., 2000). They quantify the objects covered by the analysis, called “facts”. A fact describes often an event (for example, the sales) that occurs within an organization which uses the decision making system. The organization wishes explain the fact (Wehrle et al., 2005).

Dimensions are defined as static and independent variables (Nguyen et al., 2000), that tally with analysis axes. A dimension guides the queries, which provides several views on data (Wehrle et al., 2005)

The dimensions of an OLAP schema can contain one or more hierarchies in data.

Hierarchies provide a structure to the dimensions : the data of a dimension can be categorized according to various characteristics. Users of OLAP system are usually interested in aggregated data (for example, the average of the sales for some geographical areas). Thus hierarchies are aggregation levels of data (Mahboubi et al., 2012; Markl et al., 1999; Sarawagi et al., 1998). Each level of a hierarchy contains descriptors, named "attributes" (Romero and Abelló, 2010). These attributes describe each member of each level.

To design an OLAP cube, we have to determine :

— What are the measures ? *i.e.* what is the phenomenon we want to study and how to measure it ? With a measure, we have to determine an aggregation function : do we use sum, average or count to join two values ?

— What are the dimensions ? *i.e.* what are the ways of our analysis ? What are the parameters we want consider explaining measure variations ? For each dimension, we have to determine hierarchies, *i.e.* data organization into the dimension, and attributes for the dimension members.

OLAP technology interests more and more fields and especially biology. An OLAP cube provides a very easy navigation into a data set, the possibility to build cross tabulation to analyze the data and the possibility to monitor a complex phenomenon, such as pollution of a bay (Boulil et al., 2013; Mahboubi et al., 2013; Radulescu and Radulescu, 2008) or growth of a forest (Miquel et al., 2002b). But biologists generally do not have skills to build and manage an OLAP system.

Thereby this needful high level of skills is an obstacle to democratizing of OLAP systems. Our objective in this article is to suggest an OLAP system that will be able to organize automatically hierarchies in a dimension. With this kind of system, OLAP design can be an automatic task and ultimately does not require specific IT skills.

To begin, we identified the type of automatic or semi-automatic approach, which are used to realize the design of a data warehouse or OLAP cube. Three types of approaches can be used to make the design of an data warehouse (Cravero and Sepúlveda, 2014; Tebourski et al., 2013) : (i) Methods based on user specifications, or demand-driven approach ; (ii) Methods based on available data, or data-driven approach ; (iii) Mixed methods, or hybrid approach.

For example, oriented to demand-driven methods, we cite the work of Jovanovic *et al.*, who developed a methodology for designing a data warehouse (**?**). This method is iterative : at each step, the system searches in the data that best correspond with information required by the user in terms of dimensions or facts. Data are modeled with an ontology.

Moreover, several other have proposed systems based on hybrid approach :

— Romero and Abello offer a hybrid methdology to build multidimensional

schema from a relational database (Romero and Abelló, 2010).
— Abdelhedi *et al.* have developed a prototype called CASE to build an OLAP cube with a hybrid method (Abdelhedi et al., 2011). The design is driven by both the data sources and the user specifications.
— As in many current works, Thenmozhi and Vivekanandan propose an automatic system to build the schema of a data warehouse from an ontology (Thenmozhi and Vivekanandan, 2013).

Finally, the following authors have worked on automatic data-driven systems and using data mining to build a data warehouse or an OLAP cube :

1. Eder *et al.* apply data mining algorithms such as auto-regression, auto-correlation, regression or fast Fourier transform on the data in a data warehouse (Eder et al., 2003). Their goal is to automatically detect the structural changes in a data warehouse, such as deleting, adding, merging member in a hierarchy.
2. Usman (Usman et al., 2010; Usman and Pears, 2010) provides a methodology to design automatically OLAP schema and data warehouses with hierarchical clustering. This author suggests a complete system to build OLAP systems with data sets. The system, which is proposed by Usman *et al.*, uses hierarchical agglomerative clustering to perform a pre-processing on the data. After that, the system identifies facts and dimensions into the clustered data. This system is able to build star schema, snowflake schema and constellation schema.
3. Rehman *et al.* propose a system to dynamically build hierarchies based on data from Twitter (Rehman et al., 2012). This paper has two Interests : a) The cube is built on original data, that are messages of users on a social network. b) Data mining is used to dynamically build hierarchies : thanks to data mining, the categories of network users described in hierarchies are updated automatically.

Moreover, the following authors use clustering algorithms to dynamically build or modify hierarchies in an OLAP cube :

1. Messaoud *et al.* propose a new OLAP operator named OPAC which allows to aggregate facts that refer to complex objects, such as images (Messaoud et al., 2004). This operator is based on hierarchical clustering algorithm. The prototype proposed by these authors incorporates a module to evaluate the quality of aggregations.
2. Favre, Bentayeb and Boussaid (Favre et al., 2006) suggest considering rules defined by the users during browsing in an OLAP system. These rules were used to change dynamically the data warehouse schema. The system, that Favre *et al.* have proposed, has a stable part and a dynamic part. The stable

part of the system corresponds of a basic OLAP schema with a star schema. From this basis, each user can define rules to build hierarchies in each dimension. These hierarchies, which depend of the user rules, constitute the dynamic part of the system.

3. In 2008, Bentayeb offers create new levels in a hierarchy with the K-means algorithm (Bentayeb, 2008). Thereafter, Bentayeb and Khemiri propose in 2013 (Bentayeb and Khemiri, 2013) an operator, called ProCK, which, as in the work of Hubert and Teste (Hubert and Teste, 2009), permits to the user to dynamically change the hierarchies during the navigation. This operator uses a K-means algorithm modified to take into account the constraints defined by the user. This operator allows to define new levels in a hierarchy.
4. Teste and Hubert propose in 2009 a new operator that allows the user to dynamically change the hierarchies within the cube OLAP during navigation (Hubert and Teste, 2009).
5. Leonhardi *et al.* offer the user to create new dimension during navigation (Leonhardi et al., 2010). These authors propose to increase the OLAP cube exploration functionalities by providing the user data mining algorithms applying on data, which are selected in the warehouse.

On the other hand, Ceci *et al.* use a hierarchical clustering to integrate continuous variables as dimensions in an OLAP schema (Ceci et al., 2011). Their tool uses a modified BIRCH algorithm. It discretizes a continuous dimension in order that the user can perform operations on conventional querying a cube : Roll-up and Drill-down. These authors use data mining to incorporate in a cube OLAP new data, whose the type lends itself poorly.

These works present several interesting aspects. First these works suggest the use of an *a posteriori* modeling of OLAP schema, perform by user or by an algorithm. Furthermore these works offer to the user the possibility to build his own OLAP schema or to build an OLAP schema according to the own structure of data. This article is inspired by these viewpoints, and we build a system that offers to user the possibility to build his own OLAP schema with a data mining method.

In a biological study, measures and dimensions are clearly identified. But the data which describe a dimension do not necessarily have an apparent hierarchical structure :

— The dimension can contain several quantitative variables and not only categories.
— The variables are heterogeneous : the data set can contain quantitative variables, nominal variables and binary variables.
— The data set can contain blank values.

The presented previous works offer to build automatically OLAP systems with

hierarchical use data set with binary and quantitative variables. We suggest to supplement these works with a similarity index comes from ecological analysis, the Gower index.

In this article we provide a methodology to build automatically a hierarchy with a biological data set that contains heterogeneous variables. Our approach is as follows :

— In the first part, we introduce foremost the data set that we use and the features of this data set.
— In a second part, we present several *a priori* OLAP schemas and their limitations.
— In a third part, we explain first how our system works. We present the hierarchical agglomerative clustering and we define what clustering parameters we need to perform the hierarchical agglomerative clustering with our data set. Next we explain what the Gower index is and what their interests are.
— In a fourth part, we suggest an evaluation of the needful memory and the needful calculation time according to the number of processed data.
— Finally we conclude on the system working and performances and we present our future work.

4.2.2 A data set from a large ecological study

Our data set comes from a census program for nesting birds along the Loire River (France) (Frochot et al., 2003). The STORI (*Suivi Temporel des Oiseaux nicheurs en Rivière* : Temporal Monitoring of Nesting Birds in River Valley) is a wide research program, which studies bird populations along the rivers. The objective of this program is the observation of temporal and spatial changes into bird populations. One hundred ninety eight points were defined along the river in the framework of this program. At each point the birds are identified with the IPA (*Indice Ponctuel d'Abondance* : Punctual Abundance Index) method (Blondel et al., 1981) during four census campaigns (1990, 1996, 2002 and 2011). Bird abundances were described by a semi-quantitative abundance index.

One of the main objectives of the STORI is studying global and local factors that explain these changes. In this context, the evolution of environments along the Loire River between 1990 and 2011 were described at each point in parallel with the IPA data, to find correlations between these populations and this environment.

In fact, the data set can be summarized by :

— A measure : bird abundances that can agglomerate with a sum or an average.
— Three dimensions to analyze the abundance : species, time and space.

In this context, we build an OLAP system to manage and store these data. The

working of our system was described in another section (section section §4.2.4). We build a data warehouse with a star schema and an OLAP schema with three dimensions. But the spatial dimension of the OLAP schema raises problems that were explained below. In this part,

To explain bird abundances we try to establish correlations between birds and landscapes. At each point, the river and the valley are described for several years. In fact many variables are defined only for one campaign. Moreover all kinds of variables are present : there are continuous variables, discrete variables, nominal variables and ordinal variables. The variables that describe landscapes are presented in the table below (tableau 4.1).

Tableau 4.1 – Number of variables used for landscape and river description according to the year

Variable types		1990	1996	2002	2011
Quantitative	Continuous	8	0	97	44
	Discrete	7	7	7	10
Qualitative	Ordinal	5	0	0	1
	Nominal	7	2	4	6
	Binary	5	0	0	3

This dimension has three interesting features :

— There is no intrinsic hierarchy into the description of environment along the river : except keys and station identifiers, only two station attributes (on 110) are linked by a functional dependency.
— Their attributes are heterogeneous.
— Their attributes are not defined for all campaigns.

As a consequence we suggest building automatically a hierarchy for this dimension because there is no explicit hierarchy in this dimension and we want offer to biologists the possibility of building their own OLAP schema.

In this article, we focus on this spatial dimension and our objective is generalizing the results that we obtain with these data. We do not discuss about spatial features of this dimension. Several works have proposed solutions to manage spatial data in Spatial OLAP (SOLAP) systems (Bimonte et al., 2010; Rivest et al., 2001). In this article, we focus on the complexity of attributes that describe the member of this dimension.

4.2.3 *A priori* OLAP schema design : what are the limitations ?

In the precedent section, we have presented the data set that we use in this study. The ideal OLAP schema to analyze these data is a three-dimensional schema with the abundance measurements as facts, a dimension that describes the species, a dimension that records the year of bird census and a dimension that describes the census stations (figure 4.1). With this structure we can perform the analysis that is interesting in this ecological study : ecology scientists want characterize spatio-temporal changes into bird populations along the Loire River.

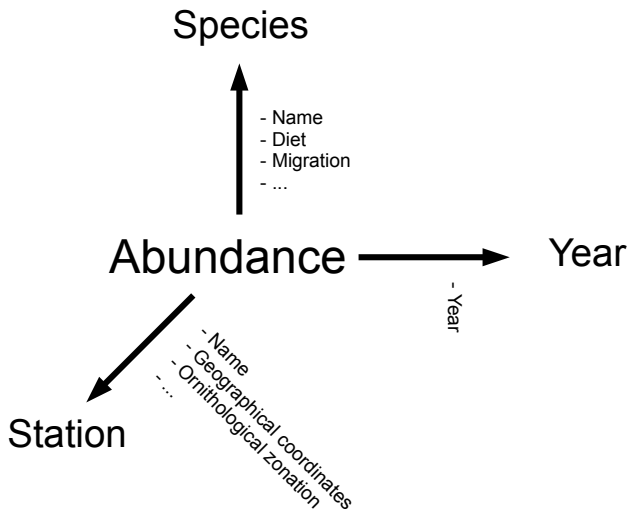


Figure 4.1 – An informal model of the biodiversity spatial multidimensional model

But we have described some features of the data set which ban a simple three-dimensional schema. The spatial dimension, that describes the environment along the Loire River, is strongly correlated to the time dimension. The description of the environment is time dependent because :

— The values of some attributes, that describe the stations, change according to the time.

— Many attributes are not measured for all years.

Several models of data warehouse may be proposed to consider this correlation between spatial dimension and time dimension. The following solutions are pre-

sented at the conceptual level, according to MultiDimER notations (Malinowski and Zimanyi, 2006). Details of these notations are summarized in Appendix.

The first solution is a fact constellation schema (figure 4.2). With this solution, there are two fact tables : a fact table for abundances according to species, stations and years and a fact table for environment descriptions according to stations and years. This solution is the more elegant solution. With this solution, the data storage is optimized. But the crossing between abundance data and environment data requires querying two independent cubes. Moreover qualitative variables cannot be stored in a fact table.

The second solution is a star schema (figure 4.3). With this solution, there are a fact table for abundances according to species, time and stations. But the data, that describe the spatial dimension, are related to time. Thus each station is duplicated for each census campaign. Thereby the station n°1 in 1990 and the same station n°1 in 1996 are not considered as the same object in the OLAP cube. With this solution, the spatial consistency of the dataset is lost.

The third solution is a fact constellation schema (figure 4.4). This kind of solution has been proposed by Miquel *et al.* in 2002 (Miquel et al., 2002b). With this solution, we build a fact table for each census campaign. Each yearly fact table is linked to the "species" dimension and to a yearly "stations" dimension. The main disadvantage of this solution is the loss of the temporal consistency of the data set.

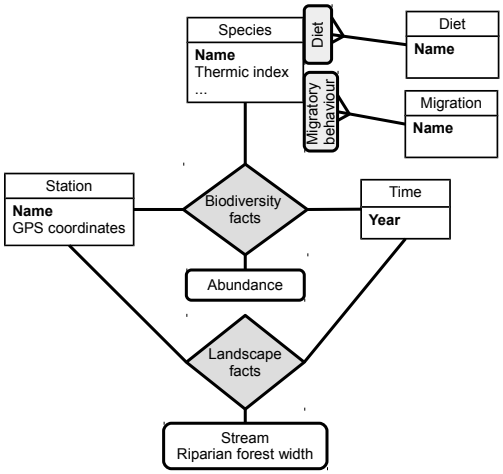


Figure 4.2 – A fact constellation schema with a fact table for abundances and a fact table for environment description

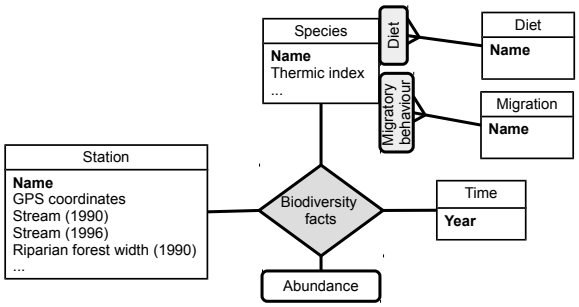


Figure 4.3 – A star schema with a time-dependent spatial dimension

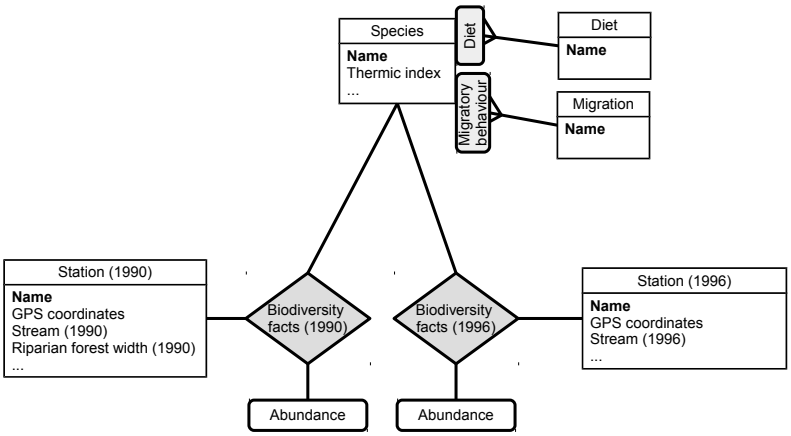


Figure 4.4 – A fact constellation schema with a fact table for each census year

Finally, none of these three solutions can provide a perfect schema (tableau 4.2). Thus we suggest in this article a solution to build a single spatial dimension. Thereby we obtain the three-dimensional cube that is shown in figure 4.1. To propose a spatial dimension, with a coherent hierarchy, we use a clustering method. This kind of method can detect a structure in a dataset. With a clustering method we can propose a prototype that builds automatically a dimension for an OLAP cube.

Tableau 4.2 – Summary of the limitations of each solution

	Solution 1	Solution 2	Solution 3
Solution description	Fact constellation schema with a fact table for abundances and a fact table for environment descriptions	Star schema with a time-dependent spatial dimension	Fact constellation schema with a fact table with abundances for each census year
Limitations of the solution	Crossing between abundance data and environment data requires querying two cubes. Qualitative environmental variables cannot be stored.	Spatial consistency of the dataset is lost.	Temporal consistency is lost.

4.2.4 Proposition : an automatic hierarchy design for OLAP schema based on clustering method

To ease understanding of sections 3 and 4, we offer to clarify some vocabulary. In a clustering context, “individuals” are items, which will be classified. Moreover “variables” are descriptors of individuals. Variables are used to perform the clustering algorithm, and to measure a distance between individuals. In this article, the clustering algortihm is perform in an OLAP context and is used to build a hierarchy. Thus, in the sections 3 and 4, “individual” is a synonym of “dimension member” and “variable” is a synonym of “attributes”.

4.2.4.1 Prototype working

General working of the prototype

We build a prototype which is able to extract the relevant data from a data warehouse and to design and publish a new hierarchy in a dimension. We suggest a system which performs a hierarchical clustering on a table in a database. This system deduces the organization of the hierarchy from the clustering process. Next it updates the OLAP schema, the dimension table in the data warehouse and the OLAP cube in XML.

The working of this system has several steps (the number of steps tallies with the number in the figure 4.5) :

1. The system recovers data and meta data from the database. The data that the system uses are : data that describe the dimension, data type (text or numeric) of each variable in the dimension and relationship between facts and processed dimension. In the figure 4.7, we present the four screenshots of our prototype, which are designed to ask to user the data that are used to build a hierarchy.
2. The system identifies the type of each variable. This identification is compulsory because the calculation of a hierarchical agglomerative clustering needs knowledges about type of each variable. The identification of a variable type can be performed by the user. In this case the variables types can be asked to the user or recorded as metadata in the data warehouse. Otherwise it is possible to determine automatically the type of a variable according to the type of data (text or numeric) and the number of values. This second point was explained in the 4.2.4.1.
3. The system performs the hierarchical agglomerative clustering with the Gower index (See 4.2.4.1 and 4.2.4.1).
4. According to the result of hierarchical clustering, the system creates a table in the data warehouse. The first column identifies the points and each other column is a level in the hierarchical clustering. In fact, the first column is the lower level of the hierarchy and a primary key. The values of this first column were used as foreign keys in the fact table. This step updates the OLAP schema. In our case each row is a census point along the river (section §4.2.2).
5. According to the result of hierarchical clustering, the system updates the XML file that describes the OLAP cube with the new hierarchy. This new hierarchy is the calculated hierarchy. The XML file specifies the data organization in the cube and the metadata. After the creation of the cube, this cube is published on the OLAP server.

6. After the creation of the new hierarchy in the data warehouse and after the publishing of the new cube, the users of the OLAP system can use the new cube thanks to the dedicated interface. A screenshot of OLAP client is presented in the figure 4.6. On this figure, we can see the new hierarchy in the dimension "LocationD". On this screenshot, the user has selected the sum of abundances as a measure, the "Year" level of the "Time" dimension and the "level11" level (the higher level) of the "LocationD" dimension.

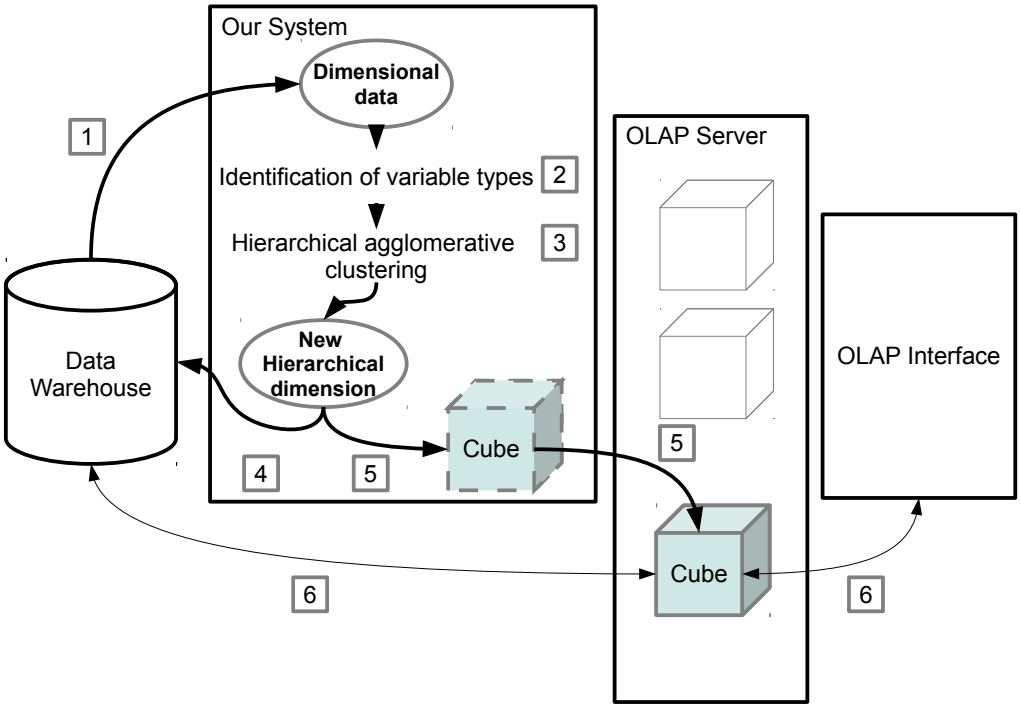
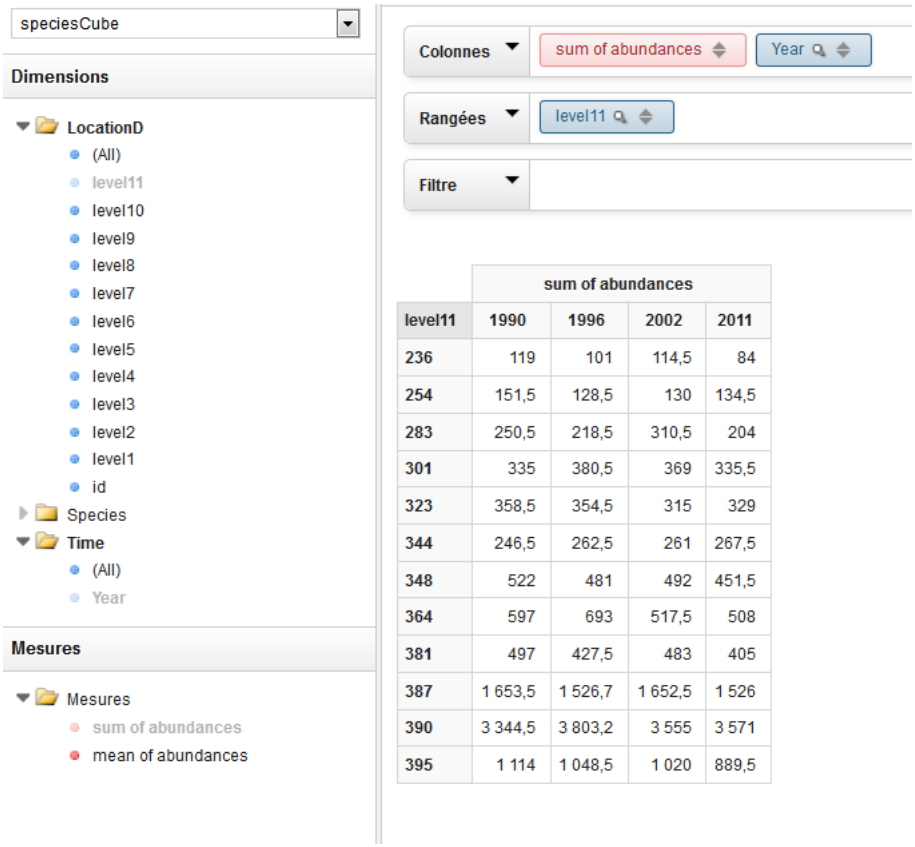
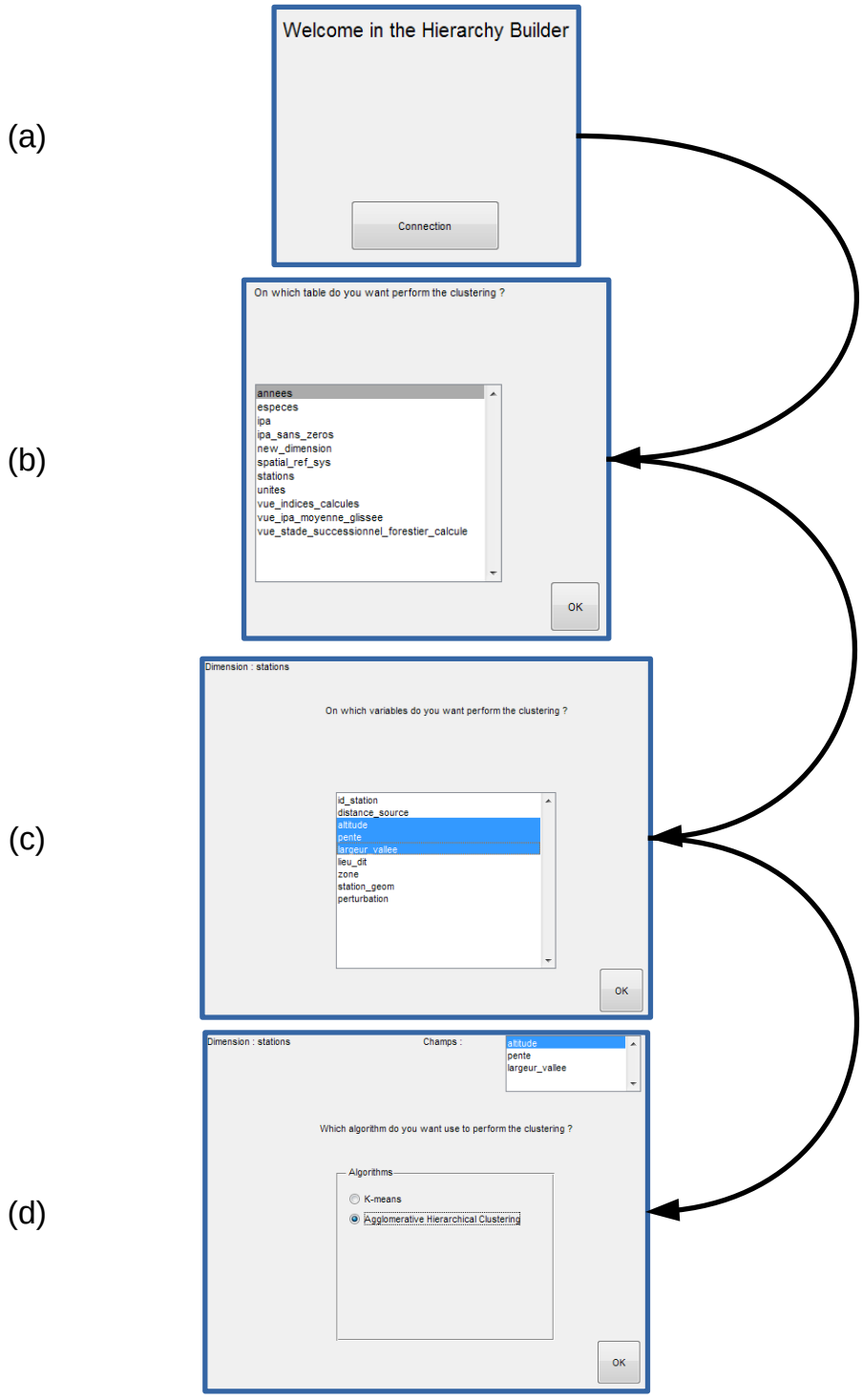


Figure 4.5 – The working of our prototype



	sum of abundances			
level11	1990	1996	2002	2011
236	119	101	114,5	84
254	151,5	128,5	130	134,5
283	250,5	218,5	310,5	204
301	335	380,5	369	335,5
323	358,5	354,5	315	329
344	246,5	262,5	261	267,5
348	522	481	492	451,5
364	597	693	517,5	508
381	497	427,5	483	405
387	1 653,5	1 526,7	1 652,5	1 526
390	3 344,5	3 803,2	3 555	3 571
395	1 114	1 048,5	1 020	889,5

Figure 4.6 – Screenshot of our OLAP client (Saiku)



The interface (a) is a connection with the data warehouse. The interface (b) permits selecting a table. The interface (c) permits selecting columns in the selected table. The interface (d) permits performing AHC.

Figure 4.7 – Screenshots of the hierarchy builder

Focus on clustering method : the hierarchical agglomerative clustering

During designing an OLAP schema, hierarchies are classically built by hand. For an automatic system, we need use an algorithm to build hierarchies. We suggest using hierarchical agglomerative clustering. Hierarchical clustering has been used in OLAP systems to improve performances of queries ((Markl et al., 1999), 1999) or to design OLAP schema (Usman et al., 2010).

The hierarchical agglomerative clustering is a clustering method. This method is an unsupervised method (*i.e.* no learning is needful). The aim of this method is the building of a hierarchy for find groups into the data. In a hierarchical agglomerative clustering, each branch of the built hierarchy is a cluster. This method has several steps (Tuffery, 2011) :

1. Calculation of distances between individuals.
2. Choice of the two nearest individuals.
3. Aggregation of the two nearest individuals in a cluster. The cluster is now considered an individual.
4. Go back to the step 1 and loop while there is more than one individual.

The results of a hierarchical agglomerative clustering can be showed as a tree which represents the distance between the individuals (Jain et al., 1999).

To perform a hierarchical agglomerative clustering, we have to define :
— A metric to measure the distance between individuals.
— A method to aggregate individuals in cluster.
The problem with our data set is qualitative variables. With qualitative variables we cannot define a cluster like the centroid of these members. To measure the distance between two clusters, we calculate the average of all distances between all individuals in each cluster. We use unweighted average linkage. Several linkage methods can be used : unweighted average distance (UPGMA), furthest distance, shortest distance and weighted average distance (WPGMA). We use UPGMA, because, with no knowledge on the data structure, this linkage appears like the best summary of the distance between two clusters (Kojadinovic, 2004).

The distance between two individuals must mix quantitative and qualitative variables. The traditional metrics like Manhattan distance, Euclidian distance or Minkowski distance are not relevant in the case of a mixed data set. Thereby we suggest measuring the distances between individuals with an similarity index that comes from biology : the Gower similarity index (4.2.4.1).

Focus on distance measurement : the Gower index

The Gower index is designed to measure similarity between two individuals that are defined by heterogeneous variables (Gower, 1971). The Gower index is a classical similarity index, which is often used in an ecological study or in a modeling work (Segurado and Araujo, 2004; Westphal et al., 2007). The Gower index is calculated as follow :

- $I_1$and I_2 are two individuals.
- N is the number of variables used to define the individuals.
- w_i is a weight. If the variable n°i is not define for I_1 or I_2, then $w_i = 0$. Else $w_i = 1$.
- $S_i(I_1, I_2)$ depends of the type of the variable n°i called V_i :
  - If variable n°i is qualitative then :
    - If $V_i(I_1) = V_i(I_2)$ then $S_i(I_1, I_2) = 1$,
    - Else $S_i(I_1, I_2) = 0$
  - If variable n°i is quantitative then : $S_i(I_1, I_2) = 1 - \frac{|V_i(I_1) - V_i(I_2)|}{Max(V_i) - Min(V_i)}$

in the following equation

$$S_G(I_1, I_2) = \frac{\sum_{i=1}^{N}[w_i S_i(I_1, I_2)]}{\sum_{i=1}^{N}[w_i]}$$

Some features of the Gower index can be detailed. First, the Gower index is a similarity index. Thus if a Gower index value between two individuals is close to 1, it means that the two individuals are very similar.

Secondly we explain the building of the Gower index. The calculation of Gower index corresponds to a weighted average. In fact, we calculate a similarity value between two individuals for each variable. The Gower index is the weighted average of these similarities according to variables. The Gower index distinguishes qualitative variables and quantitative variables. On the one hand this similarity index treats a qualitative variable with a boolean. If the individuals are in the same class, the boolean is equal to 1. Else the boolean is equal to 0. On the other hand this similarity index treats the quantitative variables as follow : we calculate a distance between two individuals with the absolute value of the difference. This absolute difference is divided by the range (the difference between maximum and minimum) of the variable. With this division, the difference between two individuals according to a variable is independent of the range of the variable. Finally, the fraction is subtracted to 1. Thereby we obtain the similarity between two individuals according to one variable.

Now we can calculate the similarity between two individuals according to each variable. But we need define weights for each variable. The weights permit to

manage the missing values. When we calculate the Gower index between two individuals, sometimes a variable is undefined for an individual. In this calculation, the undefined variable is weighted to 0 : this variable is excluded of the Gower index calculation. Thereby, we manage missing values with variable weights. Moreover, with the weights, we can manage the importance of each variable. If the user want give more importance to a variable, he can fix accordingly the weight of each variable.

We propose to calculate the Gower index for an example (tableau 4.3, tableau 4.4 and tableau 4.5).

Tableau 4.3 – Variables used for the example

The following table is the description of the variables that we use in this example :

Variable name	Variable type	Minimum value	Maximum value
Altitude	Quantitative	0	1410
Confluence	Qualitative	-	-
Bank	Qualitative	-	-
Current	Qualitative	-	-
Substratum	Qualitative	-	-
Aquatic vegetation	Qualitative	-	-
Salinity	Quantitative	0	35
Slope	Quantitative	0	120
Valley width	Quantitative	0	2950

Tableau 4.4 – Individuals used for the example

The following table is the description of two stations, which are described with the previous variables :

Variable name	Station n°1	Station n°11
Altitude	1410	899
Confluence	No	No
Bank	0	1-15
Current	<10	10-25
Substratum	mud and silt	blocks
Aquatic vegetation	0	1-15
Salinity	0	0
Slope	120	3.6
Valley width	0.2	11

Tableau 4.5 – Calculation of Gower index of similarity between two stations
The following table shows the members of the formula for calculation of the similarity index :

VARIABLE NAME	w_i	S_i
Altitude	1	0.64
Confluence	1	1
Bank	1	0
Current	1	0
Substratum	1	0
Aquatic vegetation	1	0
Salinity	1	1
Slope	1	0.03
Valley width	1	0.99
Sum	*9*	*3.66*

The following formula is the calculation of the similarity between station n°1 and station n° 11 :

$$S_G = \frac{\sum w_i S_i}{\sum w_i} = \frac{3.66}{9} \simeq 0.41$$

Focus on the determination of a variable type

In our system, the user tells if the variable is quantitative or qualitative. But if the number of variable is very important or if the information is missing, we can imagine that the system find the type of variable itself. Type of a variable depends of type of data (text or number) and the number of appearance of each values (tableau 4.6). Two cases are very easy to solve :

1. If data are numbers and if the number of values is approximately equal to the number of individuals, then the variable is quantitative.
2. If data are texts and if the number of values is very smaller than the number of individuals, then the variable is qualitative.

Two cases are more problematic :

1. If data are texts and if the number of values is approximately equal to the number of individuals. In this case, the question is : does the comparison between two character strings make sense ? If the comparison between two character sequences makes sense, this comparison is possible and a similarity between two value can be calculated. Else the variable is probably a primary

key, a unique name for each individual. If this variable is a primary key, it does not provide benefit for the clustering process. Thereby this type of variables will be excluded to the clustering process.

2. If data are numbers and if the number of values is smaller than the number of individuals, then the variable can be a qualitative variable recorded with numbers or a discrete quantitative variable.

In these two problematic cases, the system can asks the user what the type of the variable is.

Tableau 4.6 – How to determine the type of a variable ?

		Number of values	
		Number of values ≈ Number of individuals	Number of values << Number of individuals
Data type	Text	Primary key	Qualitative
	Number	Quantitative	?

The problem is : what is the limit of the number of values for a qualitative variable encoded with numeric data ? To solve this problem we use several data sets to build a decision tree. Thus, to find the threshold for our data set, we have to consider a learning variable set, which has the same characteristics as our variable set.

Therefore, we have built a data set that contains qualitative and quantitative variables. This dataset should contain 198 individuals (as our data set). We have built this dataset with external datasets, which come from the UCI Machine Learning Repository (Bache and Lichman, 2013).We choose multivariate datasets *i.e.* datasets which contains qualitative and quantitative variables. These datasets contain data about :

— Physical measurements of Abalone [1]
— Census income [2]

1. Warnick J. Nash and Tracy L. Sellers and Simon R. Talbot and Andrew J. Cawthorn and Wes B. Ford, "The Population Biology of Abalone (Haliotis species) in Tasmania - Blacklip Abalone (H. rubra) from the North Coast and Islands of Bass Strait.", Marine Resources Division, Marine Research Laboratories - Taroona, Departement of Primary Industry and Fisheries - Tasmania (1994).

2. Ron Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifiers : a Decision-Tree Hybrid", in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (1996).

— Steel annealing data [3]
— Ward's Automotive Yearbook [4]
— Cylinder bands in rotogravure printing [5]
— Horse disease [6]
— Housing [7].

In our data set, we have 198 individuals. So we choose 198 individuals in each dataset from UCI Machine Learning Repository. Each item used for the learning is a variable. And, for the learning phase, we want consider variables, which are not in our environmental and ornithological data set. Thus the building of the learning variable set is very time consuming. We have limited the learning variable set so that the number of variables has an order of magnitude near of our data set. With 129 variables, we have a learning variable set quite similar to our data.

We make a decision tree with 129 variables from the external datasets (Rokach et al., 2008). A decision tree is a classification method, which has the advantage of providing automatically explicit rules. The rules of our decision tree are presented in the figure 4.8.

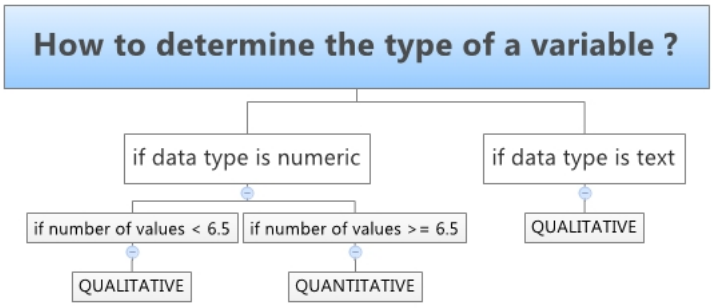


Figure 4.8 – Decision tree to decide if a variable is quantitative or qualitative

If we apply this decision tree (figure 4.8) to our data set, 10 variables on 110 are badly classified. These ten variables are quantitative variables with a very small number of values, and with the decision tree we consider that these ten variables are qualitative. This kind of error (a quantitative variable considered like a qualitative

3. No reference is associated to this dataset.
4. D. Kibler and D.W. Aha and M. Albert, "Instance-based prediction of real-valued attributes", Computational Intelligence 5 (1989), pp. 51-57.
5. B. Evans and D. Fisher, "Overcoming process delays with decision tree induction", IEEE Expert 9, 1 (1994), pp. 60-66.
6. No reference is associated to this dataset.
7. D. Harrison and D.L. Rubinfeld, "Hedonic prices and the demand for clean air", J. Environ. Economics & Management 5 (1978), pp. 81-102.

variable) is not a serious problem because in this situation, similar values are well processed and the algorithm neglects the similarity between two near values. On the other hand, a qualitative variable considered like a quantitative variable is a serious problem because the calculations performed by the algorithm have no meaning.

In conclusion we can determine automatically if a variable is qualitative or quantitative with metadata like data type and number of values. But the classification is not totally reliable. Thereby we recommend fixing a confidence interval :

- If the data type is text then the variable is qualitative.
- If the data type is numeric :
  - If the number of values is higher as 6 values then the variable is quantitative.
  - If the number of values is lower as 6 or equal to 6 values then the type of variable is problematic and the system must ask this type to the user.

4.2.4.2 Comparison between *a priori* schema and calculated schema

We detail several *a priori* OLAP schemas and their limitations in the section §4.2.3. The schema that we obtain with the prototype is presented in the figure 4.9. The structure of the new schema is a star schema. The structure is like of the structure, that is showed on the Figure figure 4.3. The fact table contains the bird abundances. The fact table is linked to three dimensions : the species dimension, which described the bird species, the temporal dimension and the new dimension. The new dimension is, for our example, a spatial dimension. This new dimension contains a hierarchy and this hierarchy is the result of the hierarchical agglomerative clustering. The new schema has the same structure as the natural dimensionality of the data set.

A calculated hierarchy is presented in the figure 4.10.

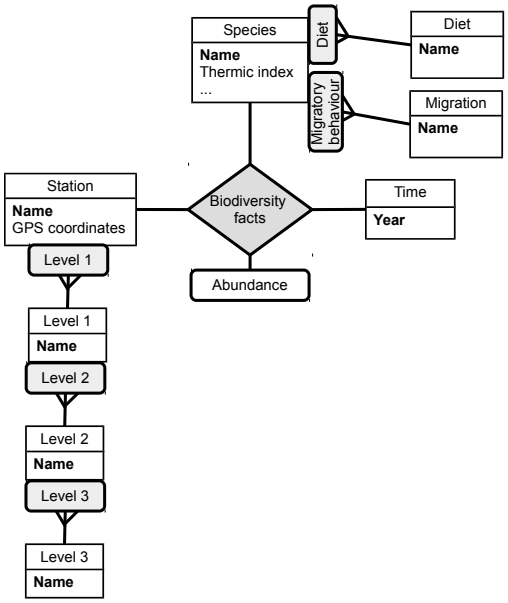


Figure 4.9 – A star schema with the new hierarchical dimension

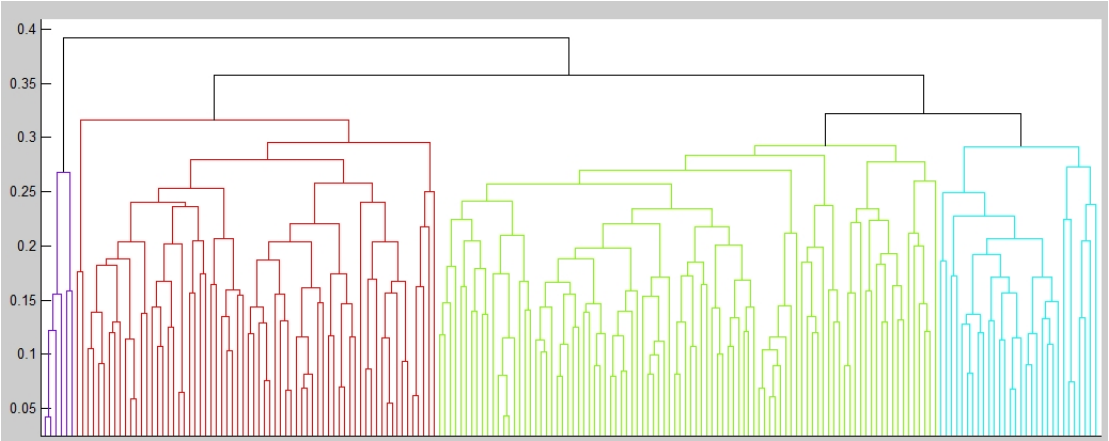


Figure 4.10 – One hierarchy built by the system

4.2.5 System performances

In the context of this study we work with a dimension that contains approximately 200 objects (the census points along the Loire River. See section §4.2.2). But OLAP systems are designed to manage large quantities of data. Thus we suggest

measuring performances of our system in order to predict calculation time and needful memory with a larger data set.

The system performances can be measured by two ways :
- The needful time for calculation of the hierarchy with Gower index.
- The number of levels of the obtained hierarchy. This number of levels tallies with the number of columns of the table which represent the new calculated hierarchy in the database. Thus the number of levels is an estimation of the needful memory to save the hierarchy.

The calculation time and the number of levels were measured according to the number of individuals and the number of variables used to build the hierarchy. The number of input data is reflected in these two parameters and we can expect that the impact of these parameters is independent to the computer configuration.

In the figure 4.11 we show the number of levels according to the number of individuals and the number of levels according to the number of variables. About these graphs, we note that :
- The theoretical minimum of levels according to the number of individuals obeys to a logarithmic function (Devroye, 1986).
- The number of levels according to the number of individuals is near to this minimum : an asymptotic behavior.
- By contrast, the number of variables has no effect on the number of levels.

To model the number of level according to the number of individuals, the two best models are a power function or a logarithmic function. Despite the fact that the power function has a correlation coefficient higher ($R^2 = 0.54$) than the correlation coefficient of the logarithmic function ($R^2 = 0.47$), we believe that the logarithmic function is more relevant, because we know that the minimum follows a logarithmic function.

Moreover the best model for the number of levels according to the number of variables is a quadratic function. But the x^2 coefficient and the x coefficient are very near to 0. We can except that the number of variables has a very little impact on the number of levels. The correlation coefficient for this model is very low ($R^2 = 0.02$).

We note that the correlation coefficients are low for each estimation of number of levels.

Thus the hierarchical agglomerative clustering performed with a Gower index as distance measurement produces binary trees whose height depends of the number of individuals. The average height of these binary trees is very near the minimum height. The needful memory used to record the hierarchy is so near the minimum.

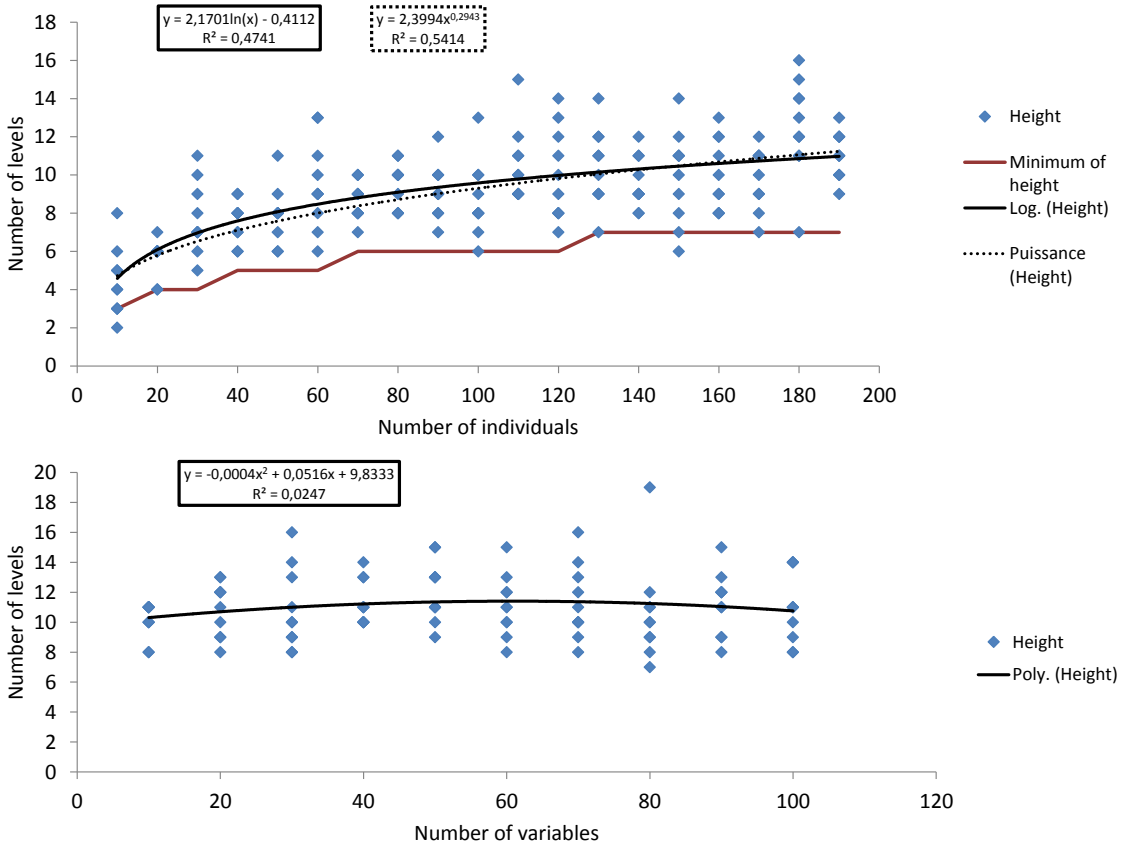


Figure 4.11 – Height of the hierarchy according to number of individuals and according to number of variables

In the figure 4.12 we show the calculation time according to the number of individuals and the number of variables. We note that :

— The calculation time according to the number of variables obeys to a linear function.

— The calculation time according to the number of individuals obeys to a quadratic function.

The complete model, which can express the calculation time according to a linear function of the number of variables and a quadratic function of the number of individuals, is :

$$t(v,M) = b_1M^2 + b_2M + b_3M^2v + b_4Mv + b_5v + b_6$$

In this formula, t is the estimated calculation time, M is the number of individuals, v is the number of variables and b_i with i in $\{1,2,3,4,5,6\}$ are coefficients

that depend on the configuration of the computer which perform the hierarchy calculation.

We perform a stepwise linear regression to fix the coefficients. The coefficients, which can be statiscally considered equal to zero, are removed. We obtain a formula like :

$$t(v, M) = (b_1 + b_3 v)M^2 + b_2 M + b_6$$

With the computer, that we use for the performances tests, we obtain $b_1 = 1.83\text{x}10^{-3}$, $b_2 = -1.06\text{x}10^{-6}$, $b_3 = 1.51\text{x}10^{-5}$and $b_6 = 1.15$. The correlation coefficient between this model and the measured calculation time is equal to 99.7%. In the figure 4.13 we show the measured calculation time and the model that we suggest above. The estimation shows well the changes of calculation time according to the number of individuals and the number of variables.

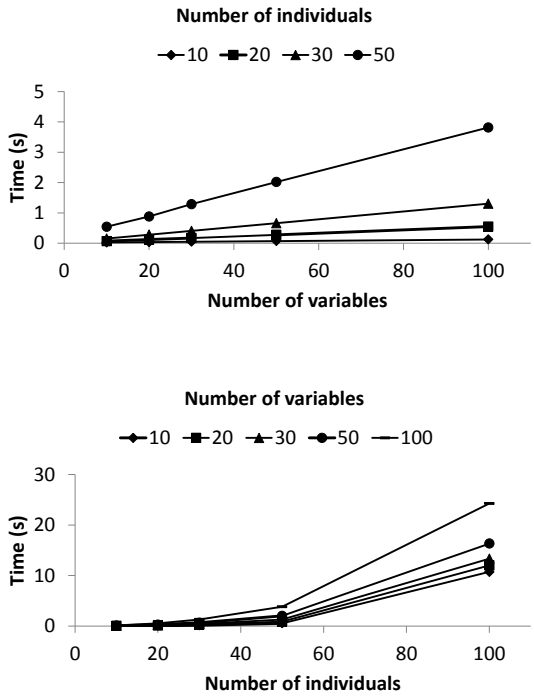


Figure 4.12 – Calculation time according to the number of individuals and the number of variables

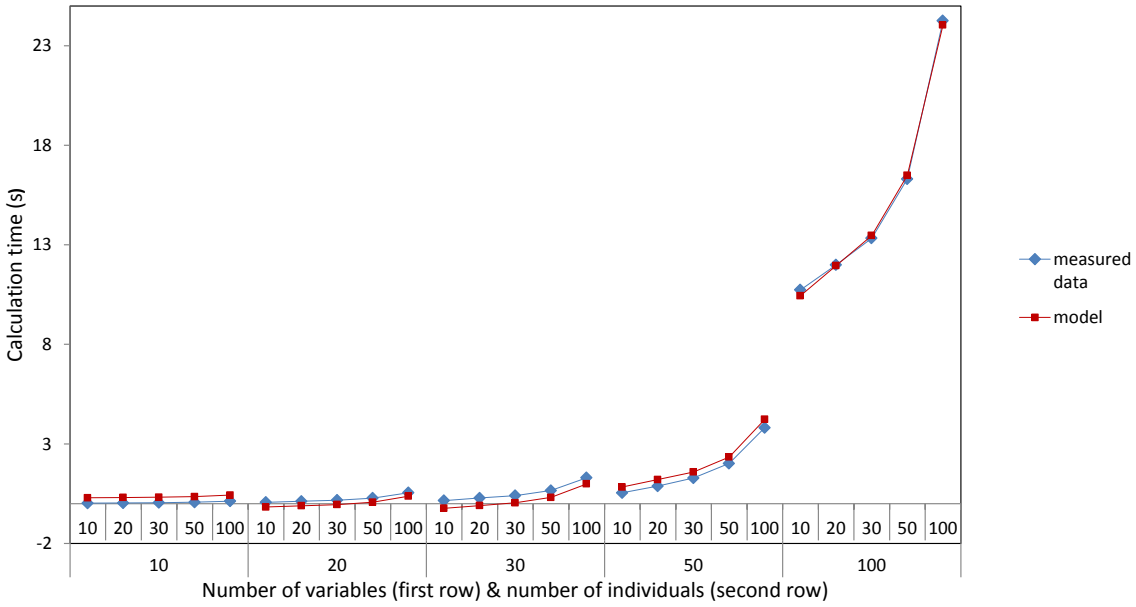


Figure 4.13 – Calculation time according to number of variables (first row of X axis) and to number of individuals (second row of X axis) and an estimation of calculation time

These performance tests have been performed on the following configuration :
— The computer has a Intel® Core™ 2Duo processor and 4Go RAM.
— The Operating System (OS) is a Windows 7, 32-bit (© Microsoft Corporation).
— The prototype runs on the software MATLAB® 2011 (© MathWorks).

4.2.6 Discussion

4.2.6.1 Discussion about the system that we have proposed

In this part, we discuss about the system that is proposed and we suggest perspectives to improve the prototype. First, we discuss about the clustering method.

Secondly, we discuss about the use of the Gower index. Thirdly, we discuss about a perspective of cluster characterization.

The use of agglomerative hierarchical clustering

We use a hierarchical agglomerative clustering, that provides a complete hierarchy of the data. But the prototype works perfectly with another clustering algorithm, like the K-means algorithm. Thereby our prototype can work with several clustering algorithm. It will be interesting to compare hierarchical and simple clustering algorithm. Thereby we know which type of clustering method is more efficient to build a new hierarchy in an OLAP schema.

Secondly, we use an unweighted average distance as a linkage method. But there are several linkage methods. The use of a linkage method could be chosen by the user if he has knowledge about his data set. Else, we could propose to user several hierarchies, which are obtained with several linkage methods. The user could choose his favorite hierarchy. There are two ways to show the hierarchies at the user : the system can present the result of hierarchical agglomerative clustering with different parameters or the system can provide to the user the possibility to test the new cube (Bimonte et al., 2013b).

The use of the Gower index

The using of the Gower index to perform a hierarchical agglomerative clustering asks some questions.

First, to perform a hierarchical agglomerative clustering with the Gower index, we need to know what the type of each variable is. In 4.2.4.1, we suggest a way to determine automatically the type of a variable. But this method is not perfect and there is an error risk. In our case we obtain approximately 10% error. However we identify two types of error and with our data set we obtain the less problematic errors. Thus the type of a variable should be determined by an algorithm or directly by the user, and the database must save the metadata that indicate the type of the variable.

Secondly, we can question the calculation of the Gower index. A hierarchical agglomerative clustering with the Gower index permits building a hierarchy with a multitype data set. But this Gower index poses two problems :

— Foremost, the processing of a variable depends on the type of the variable. Thus we are not sure that all the variables have the same weight in the calculation process of the Gower index.

— Otherwise, the presence of qualitative variables bans the calculation of a

centroid or an average individual. Thus the comparison between two clusters can be problematic.

Thus the Gower index permits the integration of qualitative variables in a clustering methodology. But these qualitative variables must be used cautiously.

Finally, the calculation of Gower index requires knowledge about the type of variables (qualitative or quantitative). But there is a third variable type : ordinal variables. Ordinal variables are qualitative variables but there is an order relationship between the classes of the variables. For example, an ordinal variable is a variable that can take the values {very low, low, medium, high, very high}. This variable is qualitative. But we know that the value 'very low' is closer to 'low' than 'very high'. A calculation of distance is therefore possible between two values of this variable. For the moment, the Gower index is not defined for the ordinal variables and the ordinal variables are treated as qualitative variables. It would be interesting to define the Gower index for ordinal variables. But the automatic detection of ordinal variables would be difficult.

How can the calculated clusters be characterized ?

The final point of this discussion, which is focused on our prototype, is about cluster characterization. With a data mining method, we determine a hierarchy in the data. But after this calculation, the clusters should be characterized. Thereby the system could find a label for each cluster. We can expect that a statistical method could find a label for each cluster. We develop now an opinion to find label for each cluster.

We define four main clusters in our data with the hierarchy in the figure 4.10. We perform statistical test to determine which variables are related to clusters. We perform Chi² test for qualitative variable and ANOVA test for quantitative variables. With these tests, we know which variables are significantly related to the clusters. In the figure 4.14, the variables significantly related to the clusters have a p-value under the significance level of 5%. We can see on this figure, that the land cover of aquatic environment (MIAQ) and the land cover of urban area (URBA) are not significantly related (with a significance level of 5%) to the clusters. All other variables are significantly related to the clusters.

If we consider a significant related variable, we can characterize each cluster. For example, the maximum height of riparian forest is near to 0 m for the stations of the cluster n°1 and between 10 and 35 m for the stations of the cluster n°4 (figure 4.15). According to the figure 4.15, the cluster n°1 is characterized by low values of maximum height of riparian forest, the cluster n°2 and the cluster n°3 is characterized by medium values of maximum height of riparian forest and the

cluster n°4 is characterized by high values of maximum height of riparian forest. On this figure, the red line represents the median.

If this kind of methodology is developed and automatized, the system could be find label for each data clusters. There is a notch around the median. If the notches of two boxplot do not overlap, we can conclude that the medians differ with 95% confidence.

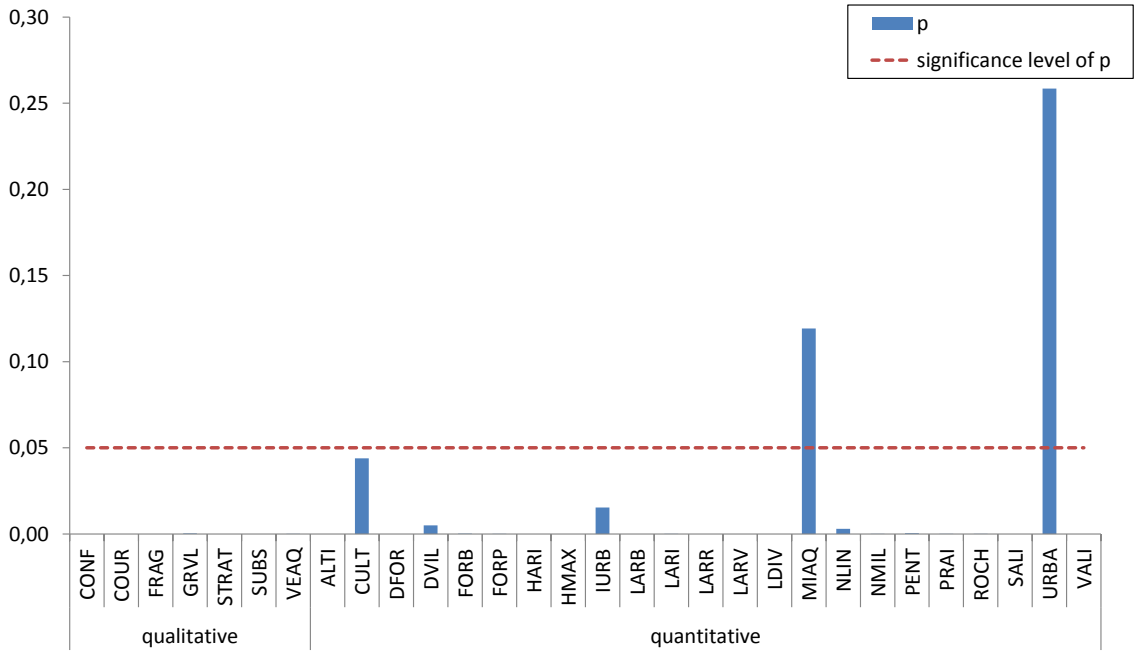


Figure 4.14 – p-values of statistical tests for each variable, which are used to build the hierarchy

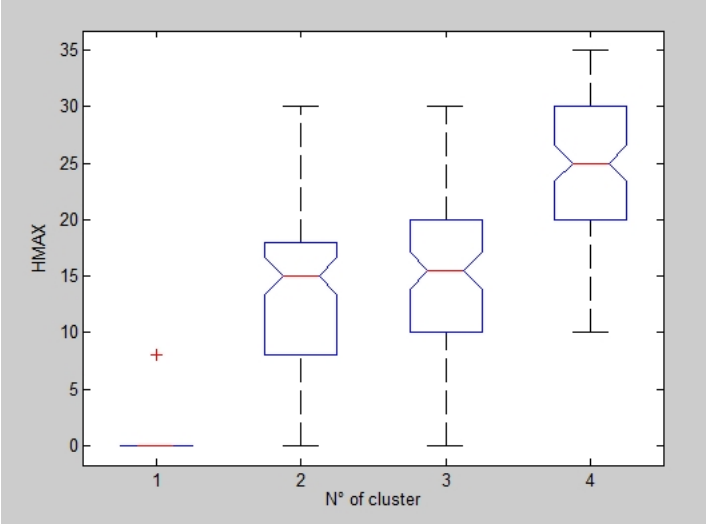


Figure 4.15 – Values of the maximum height of riparian forest (HMAX, in meters) according to the clustering results

4.2.6.2 Discussion about the system performances

In this part, we discuss about the performances of the system that is proposed and we suggest perspectives to improve the prototype performances. In fact, we have made choices about the data mining method, which is used to calculate the new hierarchy. But these choices have a strong impact on the calculation time of a new hierarchy.

First, the hierarchical agglomerative clustering permits to obtain a complete hierarchy of the data. But we can think that the system can work with another clustering method, like the K-means clustering algorithm. A more simple clustering method may offer better calculation performances. But we know that with an algorithm, like K-means algorithm, the calculated hierarchy will be simple, with only a level. Thus, improving performances with a simpler algorithm produces a simpler hierarchy. The question is : when the hierarchical agglomerative clustering is gainful ? *i.e.* when the hierarchical agglomerative does provide an interesting hierarchy (no more simple and no more complex), which warrants the high calculation time ?

Secondly, our clustering algorithm is not optimized. But we think that the performances of our prototype can be improved, because several steps of the calculation can be parallelized.

Thereby, the calculation time performances can be widely improved.

4.2.7 Conclusion

In this article, we presented a method to build automatically new hierarchies in a dimension with a clustering algorithm. The prototype that we have built is able to design and publish a new OLAP schema and a new OLAP cube from a table of a data warehouse.

Our system loads the data from a data warehouse. Next the system calculates a hierarchy with a hierarchical agglomerative clustering. But, the data sets, which are used in ecology, contain often qualitative variables and quantitative variables. Moreover a data set can contain missing values. To manage this data set and perform a hierarchical agglomerative clustering, we use a similarity index to characterize the distance between two records. This similarity index is the Gower index, an index comes from the ecology. The Gower index permits to mix qualitative and quantitative variables and so this similarity index permits the comparison between individuals that are described by heterogeneous variables. Moreover the Gower index manages missing values. To compare two individuals, this similarity index calculates a weighted average of similarities. Similarities are calculated for each variable and the formula depends on the type of variable (qualitative or quantitative). The weights concern the variables and permit to manage missing values.

Using the Gower index entails the identification of the type of variables. This identification can be entrusted to the user. But the type of a variable can be also determined by an algorithm according the data type (text or numeric) and the number of values. To automatize the decision process about the type of variable, we construct a decision tree with external data sets. The decision tree classifies the variable according to the data type (text or numeric) and the number of values. We point the threshold of the number of values : if the data type is numeric and is the number of values if lower than 6 then the variable is qualitative. Else, if the data type is numeric and is the number of values if higher than 6 then the variable is quantitative.

After the calculation of the new hierarchy, the system builds a new dimension in the data warehouse and publishes the cube on the OLAP server with a XML file.

Thus with this kind of method we can build a hierarchy based on the structure of the data, when the dimension contains heterogeneous data or when the data are not hierarchical.

We have measured the performances of our prototype. We have measured the needful calculation time and the needful memory to perform a hierarchical agglomerative clustering with the Gower index. We approximate the needful memory

with the height of the binary tree which is the result of a hierarchical clustering algorithm. These performance measurements show that :

— The height of the calculated tree is follows a logarithmic function according to the number of individuals and is a constant according to the number of variables.

— The calculation time follows a quadratic function according to the number of individuals and a linear function according to the number of variables.

The calculation time performances are not very satisfactory. Indeed a good performance for an algorithm is a time function under the linear function, like logarithmic function. The algorithm, that we have written to calculate hierarchy with the Gower index, has a calculation time function equal to a quadratic function according to the number of hierarchy members. But this algorithm is not optimized and we expect that some calculations can be parallelized. Thereby the calculation time performances can be improved.

In conclusion, the data mining, and in particular the clustering methods, permits to analyze the structure of the data. This structure can be used to build dimensions automatically in an OLAP cube. This type of analysis can resolve problems of OLAP cubes modeling, in particular if the data set contains missing values, or inconsistency according to space or time.

Appendix : MultiDimER notations

As a reminder, we provide the notations defined by Malinowski and Zimanyi in (Malinowski and Zimanyi, 2006) to describe a data warehouse at the conceptual level. The following figure summarizes the notations :

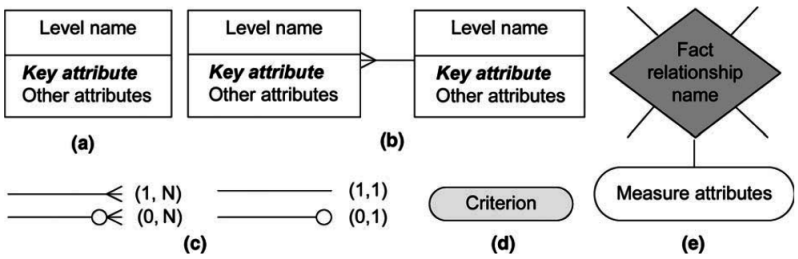


Figure 4.16 – Notations for multidimensional model : (a) level, (b) hierarchy, (c) cardinalities, (d) analysis criterion, and (e) fact relationship

Chapitre 5

Utilisation du data mining au sein d'une méthode de prototypage

Ce chapitre est consacré au deuxième objectif, défini dans le Chapitre 1.2, intitulé "Proposer une méthode prenant en compte les spécifications des utilisateurs pour la construction automatique de hiérarchies". **Le contenu de ce chapitre a été publié en partie dans la revue *Lecture Notes in Computer Science* (n°8748), suite à la conférence *Model and Data Engineering* (MEDI) en 2014. Ce premier article a été complété et soumis à la revue *Data and Knowledge Engineering* en 2015.** C'est cette version complétée qui est présentée dans ce chapitre.

Ce chapitre est organisé en deux sections :

— La Section 5.1 propose une synthèse (en français) du contexte, de la problématique, de la méthodologie et des résultats proposés dans la section suivante. Cette synthèse proposera également une conclusion sur cette contribution et replacera les résultats obtenus dans le contexte de la thèse.

— La Section 5.2 correspond au texte soumis à la revue *Data and Knowledge Engineering* (en anglais).

5.1 Synthèse sur l'utilisation du data mining au sein d'une méthode de prototypage

Dans ce chapitre, nous sommes intéressés à la prise en compte des besoins analytiques et des connaissances des utilisateurs concernant les données.

Les formalismes classiques utilisés par les méthodes classiques de conception

d'un entrepôt de données sont, dans notre contexte, difficilement utilisables, car ils sont basés sur des formalismes complexes de système d'information (par exemple, UML [1]). En effet, ces formalismes sont souvent inconnus des futurs utilisateurs de l'entrepôt de données en conception, qui n'ont généralement pas de connaissances approfondies en système d'information ou en système OLAP. Ainsi, ces utilisateurs peuvent trouver très difficile d'exprimer leurs besoins analytiques en termes de mesures et de dimensions sur un schéma conceptuel, c'est à dire, sans visualiser les résultats possibles de requêtes OLAP.

Les méthodes de conception par le prototypage permettent de contourner cet écueil en proposant aux futurs utilisateurs un (ou plusieurs) prototype(s) du futur système. Ainsi, les utilisateurs peuvent "jouer" avec le futur système, et ainsi valider ou non la définition de leurs besoins analytiques en termes de faits et de dimensions, sans passer par la lecture d'un schéma conceptuel abstrait.

Plusieurs auteurs se sont intéressés aux méthodologies de prototypage d'entrepôt de données. Parmi les méthodologies de prototypage d'entrepôt de données, les méthodes de prototypage rapide sont basées sur la définition interactive et itérative de schémas multidimensionnels par les utilisateurs (Bimonte et al., 2013b). Les méthodes statistiques, en revanche, permettent uniquement de sélectionner un sous-ensemble de données à intégrer comme faits ou comme dimensions (Huynh and Schiefer, 2001). Pour finir, certaines études évaluent les schémas OLAP a posteriori pour sélectionner la meilleure solution du point de vue des besoins utilisateurs (Phipps and Davis, 2002).

Dans ce chapitre, nous proposons une extension de la méthode développée dans (Bimonte et al., 2013b). Cette méthode, et l'outil qui y est associé, sont basés sur la définition de schéma conceptuel et l'implémentation automatique d'entrepôts de données et de modèles OLAP. Mais l'utilisation de cette méthode implique que les utilisateurs intègrent des échantillons de données au sein du prototype d'entrepôt de données, dimension par dimension, et niveau par niveau pour chaque hiérarchie, afin de simuler un processus ETL [2].

Or, dans notre cas d'étude, la définition des hiérarchies au sein de la dimension spatiale n'est pas une tâche simple. En effet, comme nous l'avons vu dans la Section 3.3.2.2 (page 43), nos données environnementales sont nombreuses et complexes. Ainsi, dans les cas similaires à notre cas d'étude, les données dimensionnelles n'ont pas (ou peu) de structure hiérarchique prédéfinie, ce qui rend les hiérarchies difficiles à définir "à la main".

C'est pourquoi nous avons proposé une nouvelle méthode de prototypage, qui est

1. Unified Modeling Language (Rumbaugh et al., 2004)
2. Extract-Transform-Load

une extension de(Bimonte et al., 2013b). Dans notre proposition, des algorithmes de data mining sont utilisées pour définir des hiérarchies au sein des dimensions. Ces algorithmes peuvent être un clustering ou une classification supervisée. L'utilisation d'un type d'algorithme ou de l'autre dépend des besoins et des spécifications des utilisateurs concernant la hiérarchie, ainsi que de leurs connaissances sur la structure de la future hiérarchie.

Dans un tel contexte, la méthodologie que nous proposons doit avoir les caractéristiques suivantes :

1. Elle doit créer automatiquement des schémas et des instances de hiérarchies, en prenant en compte les connaissances des utilisateurs grâce à :
   a) L'intégration de la fouille de données au niveau conceptuel pour créer des schémas et des instances de hiérarchies.
   b) La prise en compte de la disponibilité des données et des connaissances des utilisateurs pour choisir l'algorithme de fouille de données le plus adapté.
   c) L'implémentation d'algorithmes de fouille de données qui permettent de créer :
      i. Des hiérarchies non-complexes, c'est à dire des hiérarchies strictes, onto et couvrantes (Pedersen et al., 2001), qui peuvent être aisément intégrées dans un système OLAP classique.
      ii. Des hiérarchies avec un nombre de niveaux contrôlé : on peut choisir le nombre de niveau dans la hiérarchie résultant des calculs de l'algorithme.
      iii. Des hiérarchies dont les membres et les niveaux ont des noms qui ont un sens pour l'utilisateur.
2. Elle doit respecter le paradigme du prototypage rapide (Bimonte et al., 2013b; Martin, 2003).
3. Elle doit être mixte, c'est à dire prendre en compte à la fois les spécifications des utilisateurs et la structure des sources de données (Romero and Abello, 2009).

Pour couvrir l'ensemble de ces objectifs, nous avons proposé :

1. Une nouvelle méthode de prototypage rapide, qui intègre deux algorithmes de fouille de données (une classification ascendante hiérarchique et une machine à vecteurs supports) permettant de définir automatiquement des hiérarchies, en prenant en compte les connaissances de l'utilisateur.
   Cette méthode est réalisable en douze étapes. Pour commencer, les utilisateurs définissent informellement leurs spécifications concernant les axes

d'analyse et les sujets d'analyse (*étape 1*). A partir de ces premières spécifications, les utilisateurs peuvent définir informellement leurs spécifications concernant les algorithmes de fouille de données (*étape 2*). Ensuite, à partir de ces spécifications, l'algorithme de fouille de données est choisi grâce à un framework particulier (*étape 3*), défini dans la Section 5.2.4.2, qui peut être résumé par : si les utilisateurs ont des connaissances sur la structure et la signification de la future hiérarchie, le framework recommandera une classification supervisée, sinon, il recommandera un clustering. Il faut cependant noter que, dans certains cas, aucun algorithme n'est recommandé. Dans ce cas là, il faudra construire la hiérarchie manuellement. Après cela, les concepteurs de l'entrepôt de données utilisent les spécifications informelles de l'étape 1 pour proposer un schéma conceptuel de l'entrepôt de données, intégrant les paramètres des algorithmes choisis à l'étape 3 (*étape 4*). Suite à cela, le schéma conceptuel de l'entrepôt de données, est transformé grâce aux algorithmes de fouille de données choisis, qui créent des hiérarchies (*étape 5*). En se basant sur ce schéma conceptuel modifié, le schéma logique de l'entrepôt de données est conçu et le prototype est déployé (*étape 6*). Les utilisateurs peuvent ensuite intégrer des données de test au prototype (*étape 7*). A partir de ce point, il est question de la validation du prototype. Le prototype est donc proposé aux utilisateurs pour un test (*étape 8*). Les utilisateurs évaluent d'abord les hiérarchies calculées (*étape 9*). Puis, les dimensions et les mesures sont évaluées par les utilisateurs (*étape 10*). Si le modèle ne convient pas, la méthode prévoit de boucler à l'étape 1. Sinon, elle continue à l'étape suivante. Une fois le modèle validé, les concepteurs de l'entrepôt de données construisent les processus ETL nécessaires et intègrent le jeu de données complet au prototype (*étape 11*). Pour finir, l'entrepôt de données validé est déployé et mis à disposition des utilisateurs (*étape 12*).

2. Un profil UML complet, permettant de définir un schéma conceptuel d'entrepôt de données intégrant les deux algorithmes de fouille de données.
3. Un processus de mapping permettant de transformer le schéma multidimensionnel selon le résultat obtenu avec les algorithmes de fouille de données.
4. Un outil implémentant la méthodologie de prototypage proposée.
5. Une validation complète de la méthodologie et de l'outil, basée sur notre cas d'étude. Cette validation a été réalisée grâce à la méthode *Goal-Question-Metric*.

En conclusion, nous avons intégré le système développé dans le Chapitre 4 à une méthode de prototypage rapide d'entrepôt de données. En plus de l'algorithme de clustering proposé dans le chapitre précédent, nous avons également intégré à la méthode de prototypage la possibilité de construire des hiérarchies grâce à une classification supervisée.

En effet, nous avons constaté que, quand les membres d'une dimension sont décrits par de très nombreux attributs, la définition manuelle de hiérarchie peut être une tâche longue et complexe pour l'utilisateur. Cependant, le choix entre un clustering, une classification supervisée ou une construction classique "à la main" dépend des connaissances et des exigences des utilisateurs vis à vis des futures hiérarchies. Nous avons donc intégré à la méthode de prototypage la prise en compte de ces exigences et de ces connaissances.

A cette étape, nous avons donc proposé une méthode complète permettant de concevoir facilement et d'implémenter automatiquement un entrepôt de données et ses cubes OLAP associés. Nous avons proposé notamment de construire automatiquement des hiérarchies grâce à des algorithmes de fouille de données. Nous pouvons donc gérer des données nombreuses, mixtes voire manquantes. Cependant, une caractéristique de nos données n'a pas encore été prise en compte : l'inconsistance temporelle, qui sera traitée dans le chapitre suivant (Chapitre 6).

5.2 Multidimensional Model Design Using Data Mining, A Rapid Prototyping Methodology

Abstract

Designing and building a Data Warehouse (DW), and associated OLAP cubes, are long processes, during which decision-maker requirements play an important role. But decision-makers are not OLAP experts and can find it difficult to deal with the concepts behind DW and OLAP. In order to simplify user participation in the design process, we provide a methodology to build a prototype of the future DW that decision-makers can "play" with. In this way, these users can more easily express their design requirements.

In some cases, OLAP systems may possess useful features for data analysis but, unfortunately, these data may have a structure incompatible with a multidimensional schema. In our case study, the dimensional data had no hierarchical structure. We therefore integrated a DM process into our methodology, to discover hierarchies in dimension member attributes.

To support DW design in this context, we propose : (i) a new rapid prototyping methodology, integrating two different DM algorithms, to define dimension hierarchies according to decision-maker knowledge ; (ii) a complete UML Profile, to define a DW schema that integrates both the DM algorithms ; (iii) a mapping

process to transform multidimensional schemata according to the results of the DM algorithms ; (iv) a tool implementing the proposed methodology ; (v) a full validation, based on a real case study concerning bird biodiversity.

In conclusion, we confirm the rapidity and efficacy of our methodology and tool in providing a multidimensional schema to satisfy decision-maker analytical needs.

Keywords

Data Warehouse, OLAP, Data Mining, Methodologies and Tools

5.2.1 Introduction

Business Intelligence technology provides tools, such as Data Warehouses (DWs), On-Line Analytical Processing (OLAP), and Data Mining (DM), that allow decision-makers to explore huge volumes of data, in order to discover patterns and knowledge, and thus confirm their hypotheses.

DWs are large data repositories that support the decision-making process through flexible, interactive data analysis (Kimball, 1996). Warehoused data are built according to a multidimensional model that defines concepts of facts and dimensions. Facts represent objects and are described by numerical attributes, called measures. Facts are analyzed along dimensions representing the axes of analysis. Dimensions are organized in hierarchies. Measures are aggregated with classical SQL aggregation functions (e.g. SUM, MIN, MAX, etc.) along hierarchical levels, using OLAP operators (Inmon, 1996). These OLAP systems allow decision-makers to visualize and explore facts during query sessions by applying OLAP operators : Slice selects a subset of warehoused data ; Roll-Up aggregates measures by moving up through the hierarchy ; Drill-Down is the opposite of Roll-Up, etc. A basic Relational OLAP (ROLAP) system architecture consists of : (i) a relational Data Base Management System (DBMS), which stores data in accordance with a multidimensional paradigm ; (ii) an OLAP server, which implements the multidimensional model and OLAP operators on top of the DBMS ; (iii) an OLAP client, which combines and synchronizes tabular and graphical displays, and allows DW queries ; (iv) an ETL tool, which extracts data from multiple heterogeneous sources, then transforms and loads them into the DW. The classic development cycle of DWs includes several steps, among which ETL design is typically the most time-consuming (Bimonte et al., 2013b). Several DW design methodologies can be characterized by the relative importance of user requirements (Romero and Abello, 2009; Kimball, 1996) : in requirement-driven approaches, the conceptual DW schema is based primarily on user requirements ; in source-driven approaches, the conceptual DW schema is

(semi-automatically) derived from the schemata of the data sources ; in mixed approaches, these two processes are carried out in parallel. Rapid DW prototyping is crucial when dealing with complex applications, and has therefore been the object of several studies (Bimonte et al., 2013b; Golfarelli and Rizzi, 2011; Huynh and Schiefer, 2001). The Bimonte et al. study presented a rapid, requirement-driven design methodology and tool, called ProtOLAP. Their methodology is based on conceptual DW models, which are then implemented automatically. After DW implementation, decision-makers must manually feed sample data into the prototype, dimension by dimension and level by level, for each hierarchy, to simulate an ETL process in the context of a requirement-driven methodology. However, feeding DWs with sample data is not always easy and, in some cases, dimensional data lack the hierarchical structure necessary to fit the user's requirements.

Data Mining (DM) is a data exploration phase of a Knowledge Discovery in Databases (KDD) process (Fayyad et al., 1996). DM is a set of descriptive and predictive methods that aim to explore data by discovering a priori unknown links between data attributes (Tuffery, 2011). DM is at the interface between machine learning and statistics, and includes automatic and semi-automatic approaches. DM offers three main techniques :

1. Clustering, or unsupervised classification : this approach corresponds to organizing a data collection (represented by a vector or a point in a multidimensional space) into classes (groups or clusters), based on similarity between group members according to a mathematical indicator (Jain et al., 1999). Classes are not defined by analysts but discovered during the clustering process.
2. Supervised classification : this approach includes an item in a class, within a set of classes predetermined by analysts.
3. Association rule learning, which discovers rules from data.

The integration of OLAP and DM can be achieved by enhancing OLAP operators with DM algorithms (i.e. DM over OLAP (Han, 1997)), but DM can be also used in physical and conceptual phases of DW design (i.e. OLAP design by DM (Liu and Luo, 2005)). In the field of conceptual modeling, Abello et al. (2006) focused on DW design (Abelló et al., 2006), while Torlone (2003) sought to facilitate interaction between decision-makers and DW experts (Torlone, 2003). Only Zubcoff et al. (2009) have presented an integrated framework, based on UML, to define conceptual models for DM algorithms on warehoused data using the DM over OLAP approach (Zubcoff et al., 2009).

As yet, however, no rapid prototyping methodology has integrated DM into DW design.

Therefore, in a preliminary study (Sautot et al., 2014), we briefly presented a

new prototyping methodology for DWs, using clustering methods to define the DW schema. Building upon our previous study, we now include more advanced DM methods, thus proposing the following improvements :

1. A new rapid prototyping methodology, integrating two different DM algorithms, to define dimension hierarchies according to decision-maker knowledge.
2. A complete UML Profile, to define a DW schema that integrates both DM algorithms.
3. A mapping process to transform multidimensional schemata according to the results of the DM algorithms.
4. A tool implementing the proposed methodology.
5. A full validation, based on a real case study, concerning bird biodiversity.

The paper is organized in the following way. In Section 2, we outline our rationale for developing a rapid prototyping methodology for data warehouses, using a bird biodiversity case study. In Section 3, the literature review therefore addresses the following three topics : (i) DW design methodologies, (ii) DW prototyping methodologies and (iii) automatic design of hierarchies in an OLAP schema. In Section 4, we describe our prototyping methodology, while Section 5 describes the UML profile associated with our prototyping methodology. In Section 6, we briefly explain how our prototyping software works. Section 7 presents a case study and assessment of our methodology and software, followed by our concluding remarks.

5.2.2 Rationale

In this section, using an ecological case study concerning bird biodiversity, in the context of the French STORI project, we present the rationale for our work and the main constraints affecting our proposed methodology.

The STORI (Suivi Temporel des Oiseaux nicheurs en Rivière : Temporal Monitoring of Nesting Birds in River Valleys) is a bird census program along the Loire River, France (Frochot et al., 2003). This program aims to detect temporal and spatial changes in bird communities. Along the river valley, 198 census points were chosen, one every 5 Km. At each point, birds were counted using a point count census method : the IPA (Indice Ponctuel d'Abondance : Punctual Abundance Index (Blondel et al., 1981), over a 21-year period, from 1990 to 2011, as illustrated in figure 3.1 (page 41).

Using a classical Data-Driven DW methodology, such as (Jensen et al., 2004) on this data set, it is possible to identify a numerical value as a measure representing abundance, and the three dimensions that characterize it : time, space and species

Tableau 5.1 – Census points : examples of environmental variables

(figure 4.1, page 59). The description of each species is a set of qualitative attributes, including diet, migratory behavior and the preferred environment of the species. The dimensions that describe species and time are easy to design, while the spatial dimension is more complex.

The model depicted in figure 4.1 (page 59) can answer OLAP questions, such as : "*What is the sum of abundance per census point*[3] *per year ?*" or "*What is the sum of abundance per census point per species ?*"

In general, applications require contextual information, such as the environmental factors influencing bird communities over space and time, to explain abundance results (Pérez-Martínez et al., 2008). In our case study, the environment around each census point was described in the years chosen for bird census, thus theoretically allowing abundances to be correlated with environmental variables (such as altitude, or river width, as shown in tableau 5.1). However, these environmental variables belong to different categories : continuous, discrete, ordinal and qualitative. It is therefore difficult to establish a spatial hierarchy, because the description of each point along the river consists of a mixed data set, with no obvious hierarchical structure (French administrative divisions are not based on ecological principles).

To summarize, the spatial dimension obtained from the data-driven methodology has two important limitations :

— There is no hierarchical structure in the environmental description of census points.
— Census point attributes are heterogeneous : the manual integration of these data into a dimension can be difficult, due to the number of attributes, and the heterogeneity of data.

Therefore a data-driven methodology is not sufficient for complex OLAP applications, and a mixed methodology is needed.

It is also important to note that, as the decision-makers in this project are ecological experts, their opinion could be useful for the definition of the multidimensional schema. For example, they could classify census points according to environmental disturbance (reference point without disturbance, point under the influence of a dam, etc.) or according to geomorphological type (Mountain, Plain, Estuary).

However, the formalisms used by existing mixed methodologies are not exploitable in such a context, as they are based on complex Information System (IS)

3. In this paper, « census point » and « station » are strict synonyms.

/OLAP formalisms (UML, ER, etc.), which are often unknown to ecologists, who tend to be unskilled IS and OLAP users. Ecologists may therefore find it very hard to express their analytical needs in terms of measures and dimensions on a conceptual schema, i.e., without visualizing sample query results (Sautot et al., 2014). These decision-makers need DW prototypes to validate their analytical needs in terms of dimensions and measures.

In this context, the design methodology of a DW should be based on a mixed methodology with the following specifications :

1. automatically created hierarchy schemata and instances, taking into account decision-makers' knowledge, using :
   a) data mining, included at the conceptual level to create hierarchy schemata and instances
      It is widely recognized that conceptual models are useful in complex applications to provide a bridge between decision-makers and information technology experts (Abelló et al., 2006).
   b) DM methods, chosen to fit data and user knowledge
   c) a data mining algorithm, to provide :
      i. non-complex hierarchies : Strict, onto and covering hierarchies (Pedersen et al., 2001) are easily handled by all existing OLAP servers.
      ii. a controlled number of levels : It is necessary to generate a hierarchy with levels, and then control the number of levels in the calculated hierarchy, since too many levels are not readily usable in classical OLAP exploration sessions.
      iii. semantic names : It is necessary to generate level names that have meaning for the user.
2. a rapid prototyping paradigm (Bimonte et al., 2013b; Martin, 2003)
   It must remain possible to go back over some of the key steps of the design in order to revise the choices made and refine the DW modeling and DM setting.
3. a mixed methodology (Romero and Abello, 2009)
   The methodology should allow decision-makers to define their functional requirements and, at the same time, analyze existing data sources to be mined during the hierarchy creation process.

5.2.3 Related Work

In this section, we offer a literature review. Several topics covered by research on decision systems can be interesting with regard to the requirements defined for

our methodology. First, we provide a summary of the current state of knowledge about DW design methodologies. In the second and third paragraphs, we focus on automatic design methodologies driven by data mining, and particularly on automatic hierarchy design. In both these paragraphs, we shall see that parts of automatic DW design methodologies are based on data mining. In the fourth paragraph, we focus on prototyping methodologies. Finally, we focus on UML extensions for DW design, because conceptual schema design is often performed with a UML profile.

First, we identified possible types of automatic or semi-automatic approaches, which are used to design a data warehouse or OLAP cube. Three types of approaches can be used to design a data warehouse (Cravero and Sepúlveda, 2014; Tebourski et al., 2013) : (i) Methods based on user specifications : the demand-driven approach ; (ii) Methods based on available data : the data-driven approach ; (iii) Mixed methods : the mixed approach. One example of a demand-driven method is the work by Jovanovic et al., who developed a methodology for designing a data warehouse (Jovanovic et al., 2012). This method is iterative : at each step, the system selects the data that best correspond to the information required by the user in terms of dimensions or facts. Data are modeled with an ontology. One example of data-driven method is the work by Usman et al., who provide a methodology to design automatically OLAP schema and data warehouses with hierarchical clustering (Usman et al., 2010). Many authors have proposed systems based on the mixed approach, often focusing on automatic methodologies. Romero and Abello offer a mixed methodology to build multidimensional schema from a relational database (Romero and Abello, 2010). Abdelhedi et al. have developed a prototype called CASE to build an OLAP cube with a mixed method (Abdelhedi et al., 2011). The design is driven by both the data sources and the user specifications. Finally, as in many recent works, Thenmozhi and Vivekanandan have proposed an automatic system to build the schema of a data warehouse from an ontology (Thenmozhi and Vivekanandan, 2013).

Many authors have proposed automatic systems to design or refine a DW or part of a DW using data mining methods. The following authors have worked on automatic data-driven systems, using data mining to build a data warehouse or an OLAP cube. Eder et al. apply data mining algorithms, such as auto-regression, auto-correlation, regression or fast Fourier transform, to the data in a data warehouse (Eder et al., 2003). Their goal is to detect automatically the structural changes in a data warehouse, such as deleting, adding, or merging members in a hierarchy. But this work is more a refinement methodology than a design methodology. In the same way, Lau et al. use an artificial neural network to improve an OLAP schema dynamically (Lau et al., 2000). Concerning complete design methodologies, we can cite Usman and collaborators (Usman et al., 2010; Us-

man and Pears, 2010), who provide a methodology to design automatically OLAP schema and data warehouses with hierarchical clustering. Usman proposes a complete system to build OLAP systems with data sets, using hierarchical agglomerative clustering to pre-process the data. After that step, the system identifies facts and dimensions in the clustered data. This system is able to build star schemata, snowflake schemata and constellation schemata. Furthermore, some authors use data mining to design part of a DW, including Rehman et al., who propose a system to dynamically build hierarchies based on data from Twitter (Rehman et al., 2012). This paper is of interest for two reasons : (i) the cube is built on original data that are messages of users in a social network, and (ii) data mining is used to dynamically build hierarchies : through data mining, the categories of network users described in hierarchies are updated automatically. The following authors use clustering algorithms to dynamically build or modify hierarchies in an OLAP cube. Ben Messaoud et al. propose a new OLAP operator, named OPAC, which aggregates facts that refer to complex objects, such as images (Messaoud et al., 2004). This operator is based on a hierarchical clustering algorithm. The prototype proposed by these authors incorporates a module to evaluate the quality of the aggregations. Bentayeb creates new levels in a hierarchy with the K-means algorithm (Bentayeb, 2008). Bentayeb and Khemiri also propose an operator, called ProCK (Bentayeb and Khemiri, 2013), which, as in the work of Hubert and Teste (Hubert and Teste, 2009), allows the user to dynamically change the hierarchies during navigation. This operator uses a K-means algorithm, modified to take into account user-defined constraints. This operator defines new levels in a hierarchy. Hubert and Teste also propose a new operator that allows the user to dynamically change the hierarchies within the OLAP cube during navigation (Hubert and Teste, 2009). Several other authors define methodologies to enrich OLAP schemata without defining OLAP operators, such as Favre, Bentayeb and Boussaid (Favre et al., 2006) suggest taking into account the rules defined by users when browsing in an OLAP system. These rules are used to change dynamically the data warehouse schema. This system has a stable and a dynamic part. The stable part of the system corresponds to a basic OLAP schema, with a star schema. From this basis, each user can define rules to build hierarchies in each dimension. These hierarchies, which depend on user-defined rules, constitute the dynamic part of the system. In the same way, Leonhardi et al. allow the user to create new dimensions during navigation, by applying data mining algorithms to the warehoused data, thus increasing potential OLAP cube exploration (Leonhardi et al., 2010). Furthermore, at a logical level, Zhang and Huang propose a new SQL operator to perform clustering on spatial data (Zhang and Huang, 2007). Ceci et al. use hierarchical clustering to integrate continuous variables as dimensions in an OLAP schema (Ceci et al., 2011). Their tool uses a modified BIRCH algorithm. It discretizes a continuous

dimension in order for the user to perform conventional querying operations on a cube : Roll-up and Drill-down. These authors use data mining to incorporate in an OLAP cube new data which are poorly adapted in type.

Regarding DW design, several authors have worked on DW prototyping methodologies. Rapid prototyping DW methodologies are based on interactive and iterative multidimensional schemata defined by users (Bimonte et al., 2013b), where statistical methods (Huynh and Schiefer, 2001) are used only to select a subset of data to feed fact and dimension data. Some studies evaluate OLAP schemata a posteriori to select the best solution from the point of view of the decision-makers (Phipps and Davis, 2002). However, to the best of our knowledge, no rapid prototyping methodology for OLAP design by DM has yet been proposed.

Finally, DW design methodologies are often supported by using UML at the conceptual level. Extending UML for DW design has been explored in several studies, e.g. (Abelló et al., 2006) and (Lujan-Mora et al., 2006), or more recently (Boulil et al., 2015). Concerning the integration of DM in a conceptual model, only Zubcoff, Pardillo, and Trujillo have proposed a UML extension for DW that integrates DM (Zubcoff and Trujillo, 2007; Zubcoff et al., 2009). However, these papers only use data mining to analyze data in a DW, and do not investigate DW design. Rizzi offers a complete UML Profile to design pattern bases, which specifically integrates UML stereotypes for clustering (Rizzi, 2004).

In conclusion, some studies address DW design in terms of design methodology, automatic design, DM use in a multidimensional context, or UML extensions for DW (for a summary, see tableau 5.2). Nevertheless, among existing DW prototyping methodologies, no work proposes the use of data mining. Among the methodologies and tools that use DM to design or refine DWs, few articles integrate several DM methods. In fact, many studies integrating DM to design a multidimensional schema use only one type of DM method. Finally, among UML extensions for DW, none has been defined to integrate DM in order to use DM methods at the conceptual level during the design phase. Therefore, in this paper, we propose a rapid prototyping DW methodology, using several DM methods (to define hierarchies in dimensional data) and a UML extension (to integrate DM methods and parameters at the conceptual level, at each step of our methodology).

5.2.4 Prototyping methodology

In this section we present our methodology (section 5.2.4.1), and the DM algorithms used for hierarchy design (section 5.2.4.2).

Tableau 5.2 – Summary of the proposed literature review

	DW designed by DM			Rapid DW prototyping	
	DM algorithms	Framework to choose the DM algorithm	Conceptual model	Conceptual model	Automatic implementation in a ROLAP architecture
(Abdelhedi et al., 2011)					X
(Abelló et al., 2006)				X	
(Messaoud et al., 2004)	Hierarchical clustering				
(Bentayeb, 2008)	K-means				
(Bimonte et al., 2013a)				X	X
(Boulil et al., 2015)				X	X
(Ceci et al., 2011)	An extension of the BIRCH algorithm				
(Eder et al., 2003)	Autoregression, autocorrelation, discrete Fourier transform, ...	X			
(Favre et al., 2006)			X	X	X
(Hubert and Teste, 2009)					
(Huynh and Schiefer, 2001)					X
(Jovanovic et al., 2012)					X
(Lau et al., 2000)	Artificial Neural Network				X
(Leonhardi et al., 2010)	Clustering				X
(Lujan-Mora et al., 2006)				X	
(Phipps and Davis, 2002)				X	X
(Rehman et al., 2012)			X		X
(Rizzi, 2004)	Several algorithms (Association rule discovery and clustering as examples)		X		
(Romero and Abello, 2010)					X
(Thenmozhi					X

5.2.4.1 Rapid Prototyping Methodology

With the aim of building a DW prototype adapted to decision-maker requirements and data sources, based on the rapid prototyping paradigm, our methodology uses iterative, incremental and semi-automated processes. Rapid prototyping ensures decision-maker validation of design choices, and the tool associated to our methodology simplifies DW design tasks for decision-makers and OLAP designers.

Our methodology consists of the following steps (figure 5.1) :

1. Decision-makers informally define their multidimensional functional requirements (i.e. analysis axes and subjects). In our case study, the decision-makers sought to analyze bird abundance according to three axes : year, census point, and bird species.
2. For some previously defined analysis axes, decision-makers informally define the functional DM requirements.
3. The best-adapted DM method is chosen using the framework presented in detail in Section 5.2.4.2. This framework can be summarized in the following way : If decision-makers have knowledge about the structure and signification of the future hierarchy (although this hierarchy does not appear in dimension member attributes), then a supervised classification must be used. Otherwise, an unsupervised clustering algorithm can be used. Note that the framework may recommend not using a DM algorithm, but building the hierarchy manually. In fact, DM algorithms have specific features and limitations, which imply that, in some cases, no algorithm can deal with decision-makers' requirements.
4. Starting from the decision-makers' requirements defined in the previous steps, DW designers create a multidimensional conceptual schema, which integrates DM parameters. We define a ***conceptual multidimensional-DM schema*** : a classical conceptual multidimensional schema enriched with DM methods for hierarchy creation. In our case study, the DW designers create two classical dimensions (time and species) and a dimension with an automatically generated hierarchy (the spatial dimension).
5. The conceptual multidimensional-DM schema is transformed by the DM algorithm, which creates hierarchies in the DW.
6. Based on the conceptual multidimensional schema obtained from the previous step, the logical schema is designed and the prototype is deployed.
7. Decision-makers feed the prototype with domain data for classical hierarchies.
8. The prototype will then be available for "beta testing", to validate dimensions and measures.

9. The hierarchies calculated by DM are presented to users. After this step, users validate (or not) the hierarchies (see figure 5.4 and 5.2.4.2 for details).
10. If the prototype is satisfactory, the methodology continues on to step 11 but, when the multidimensional model (measures or dimensions) is unsatisfactory, in the user's opinion, the methodology loops back to step 1.
11. For classical hierarchies and facts, DW designers build ETL processes from data sources, but for DM built hierarchies, real data are mined, and the DM algorithm is applied to the complete data set.
12. The final DW is deployed by designers and can be used by decision-makers.

5.2.4.2 Data mining methods for hierarchy design

In this section, we present the two types of DM methods used to define hierarchies, according to the requirements defined in section 5.2.2(supervised classification in 5.2.4.2, and clustering in 5.2.4.2). The framework used to choose the best adapted algorithm is also described (5.2.4.2).

Supervised classification

In data mining, classification corresponds to a supervised learning technique used to assign predefined classes to each instance of the data set. Thus, a classification algorithm requires training data. The classification model is created from the training data and is then used to classify new instances. In this context, many approaches have been proposed, such as the connectionist approach (Bishop, 1995) or metric-based methods, k-nearest neighbors (Cover and Hart, 1967), and kernel-based methods, e.g. Support Vector Machines or SVM (Cortes and Vapnik, 1995).

In our case, we use supervised classification to create a new hierarchy in an OLAP dimension. Classified instances are consequently dimension members.

The following section describes how supervised classification algorithms can satisfy the requirements described in Section 2.

Although the first requirement, "non-complex hierarchies", can be obtained with any supervised classification method, very few classification methods can satisfy the second requirement, "a controlled number of levels". Conventional supervised classification methods can generate only two levels : classified dimension members and groups. There are some hierarchical supervised classification methods (Adami et al., 2003), chiefly developed for text mining, but they are not yet sufficiently well documented to be successfully integrated into a prototyping methodology, which

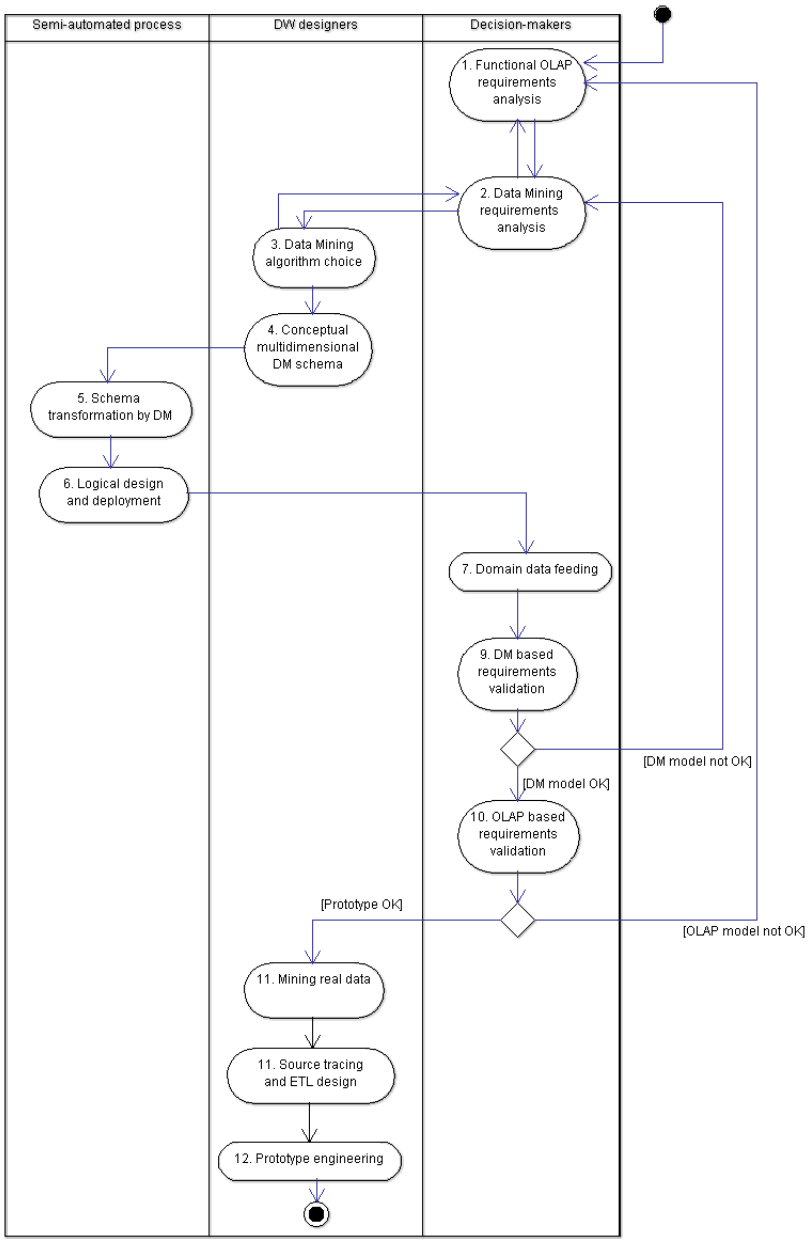


Figure 5.1 – Activity diagram of our methodology

Tableau 5.3 – Classification results for our case study

Number of dimension members in each class		Predicted class	
		Mountain	Plain
Expected class	Mountain	22	0
	Plain	1	75

must be directly usable. Regarding the third requirement, "semantic names", supervised classification is already able to generate semantic names for each predicted group.

Here we focus on the support vector machine (SVM), first developed by Vapnik for pattern recognition and function regression (Cortes and Vapnik, 1995). Over the past two decades, it has frequently been demonstrated that this technique outperforms all other classification methods to solve various real world problems, such as handwritten digit recognition, image and natural texture classification, face detection, object detection, text classification, etc. (Shih and Liu, 2006; Song et al., 2002; Vapnik, 1999).

To provide an example using our case study, we present the classification of census points in two groups (Mountain points and Plain points), according to altitude, river width and river slope. To perform this classification, we used an SVM with a Radial Basis Function (RBF) kernel. As previously described, supervised classification needs a learning step. To demonstrate SVM classification on our data, we separated our data set into two groups : from the 196 census points, 98 were included in the training data set (used for the learning phase) and 98 were included in the test data set (used to evaluate classification performance). The decision-makers had already identified the correct classification for each of the 196 census points. The class of each census point in the training data set was used to train the SVM to recognize the features of each group (Mountain census points versus Plain census points). The class of each census point in the test data set was used to compare the classification by the decision-makers with the automatic classification. In tableau 5.3 and figure 5.2a, we present the SVM results in terms of good classification performance and the hierarchy obtained for the spatial dimension. In figure 5.2a, each census point has been classified as "Mountain" or "Plain" at the higher level, named "station_type". The classifier can be considered efficient because only one census point (out of 98) is wrongly classified (tableau 5.3).

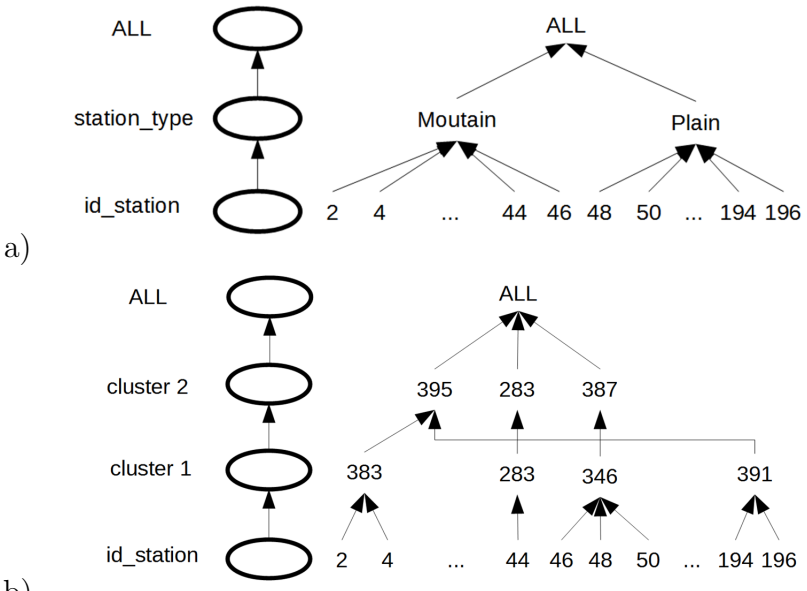


Figure 5.2 – a) Hierarchy obtained with SVM classifier b) Hierarchy obtained with AHC

Hierarchical Clustering

Clustering is a statistical method that aggregates dimension members into several groups (or clusters). Clusters have two properties. First, they are not defined by decision-makers, but discovered during the clustering process, unlike supervised classification (see 5.2.4.2). Secondly, clusters aggregate similar items, while separating items that have different characteristics.

Three types of clustering methods exist (Jain et al., 1999; Tuffery, 2011) : partitioning methods, hierarchical methods, and mixed methods, which combine the advantages of the two previous types.

The following section describes the clustering algorithms that satisfy the requirements listed in section 5.2.2.

The first requirement, “non-complex hierarchies”, can be obtained with any clustering method, but only hierarchical methods can satisfy the second requirement, “a controlled number of levels”, because this type of method generates a strict, onto, covering hierarchy that can have more than two levels. Regarding the third requirement, “semantic names", hierarchical clustering is not able to generate semantic names for each group predicted.

In the implementation of our methodology, we chose Agglomerative Hierarchical Clustering (AHC) as an example of a DM algorithm, but our methodology can deal with any hierarchical clustering algorithm. To run this algorithm, we need to define a metric, in order to measure the distance between individuals (distance), and a method to aggregate individuals into different clusters (linkage). Unfortunately, our data set contains qualitative and quantitative variables, so we chose the Unweighted Pair-Group Method with Arithmetic mean (UPGMA) for linkage (Kojadinovic, 2004) and the Gower similarity index for distance (Gower, 1971). The hierarchy thus calculated contains numerous levels with numerous clusters. But users of AHC do not traditionally use the full result of this algorithm. The traditional method to select a cut-off point for a hierarchy is based on the distance between two levels, on the desired number of clusters or on the desired minimum number of members in a cluster. In an OLAP context, we used desired number of levels to select the cut-off point (see (Sautot et al., 2015) for more details).

In figure 5.2b, we present the AHC hierarchy obtained for the spatial dimension, where each census point has been classified according to its similarity with other census points. Note that levels and level members have no semantic name.

Framework choice

In this section, we explain in detail step 3 of our methodology, which chooses the best strategy to build a hierarchy, as described in figure 5.3 (5.2.4.2), and step 9, which validates this choice (5.2.4.2).

Step 3 : Choosing a Data Mining Algorithm

Table 5.4 outlines the features of hierarchy building algorithms. These criteria are used within the framework to choose the appropriate DM algorithm. Unsupervised clustering and supervised classification do not generate the same type of hierarchy. Neither algorithm generates a complex hierarchy, but supervised classification generates hierarchies with only two levels ("dimension members" and "classes"), while unsupervised clustering can generate hierarchies with several levels. Supervised classification provides semantic names for levels and groups whereas unsupervised clustering cannot. Therefore, using supervised classification implies that decision-makers must provide information about future levels and groups, while unsupervised clustering can build a hierarchy without semantic information.

Concerning framework choice (figure 5.3), the first criterion used to choose an algorithm is user knowledge. If decision-makers have information about the future hierarchical structure, they can use supervised classification. If not, they must

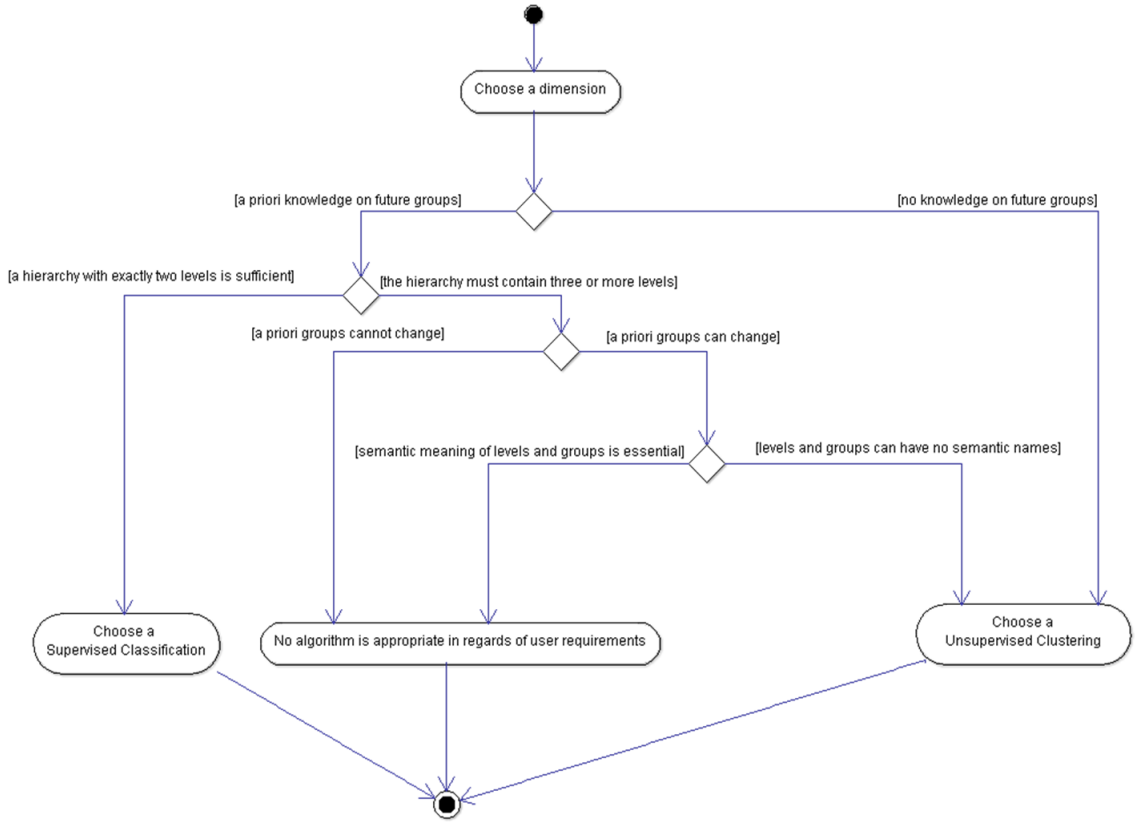


Figure 5.3 – Choosing a hierarchy building strategy

use unsupervised clustering. But supervised classification algorithms provide a hierarchy with only two levels. Therefore, the framework choice process must check whether the features of the future hierarchy are adequate to meet decision-maker requirements.

If the future hierarchy requires more than two levels, the framework choice process asks decision-makers whether the groups that they have defined a priori are rigidly fixed or if they can be modified. After that, the process verifies the importance of semantic names for groups and levels in the opinion of the decision-makers. If the pre-defined groups must not be modified or if semantic level names are essential for decision-makers, then no DM algorithm can be used and the hierarchy should be defined manually. If no such restrictions apply, the framework choice process proposes an unsupervised clustering algorithm.

Tableau 5.4 – Features of hierarchy building algorithms

		Supervised Classification	Clustering
Hierarchy type	Non-complex hierarchies	Yes	Yes
	A controlled number of levels	No	Yes
	Semantic level meaning	Yes	No
Decision-makers have knowledge about the future hierarchy		Yes	No

Step 9 : Validating Data Mining requirements

In step 9 of our methodology (see figure 5.1), the choice of a DM algorithm to build a hierarchy is validated (or not) by decision-makers according to two main criteria : semantic performance and time performance. The process of hierarchy validation is described below and in figure 5.4. If supervised classification has been chosen, step 9 aims to evaluate its semantic performance (i.e. to verify that the classifier is really able to assign the right class to a new dimension member) and its time performance (i.e. algorithm execution is fast and efficient) :

— If both semantic and time performances are satisfactory, the choice of supervised classification is confirmed (go on to step 10).

— If semantic performance is not satisfactory, but time performance is good, and decision-makers want to define groups with semantic meaning, they may revise their DM requirements (go back to step 2). If decision-makers accept another grouping structure than their pre-defined groups, they can build a hierarchy with unsupervised clustering (go back to step 3). Otherwise they must go back to step 1 to define a dimension with only one level (i.e. without hierarchies).

— If semantic performance is satisfactory, but time performance is not good, designers can perform the learning phase again, and limit the number of iterations. Semantic performance must then of course be retested, as a low number of iterations may impact semantic performance. For supervised classification, semantic performance has priority over time performance. If, despite parameter changes, either time performance or semantic performance is unsatisfactory, go back to step 2.

If unsupervised clustering has been used, step 9 evaluates time performance only :

— If time performance is efficient, the choice of unsupervised clustering is confirmed (go on to step 10).

— If time performance is not efficient, designers can run the algorithm again, changing distance and/or linkage parameters. But, with unsupervised clus-

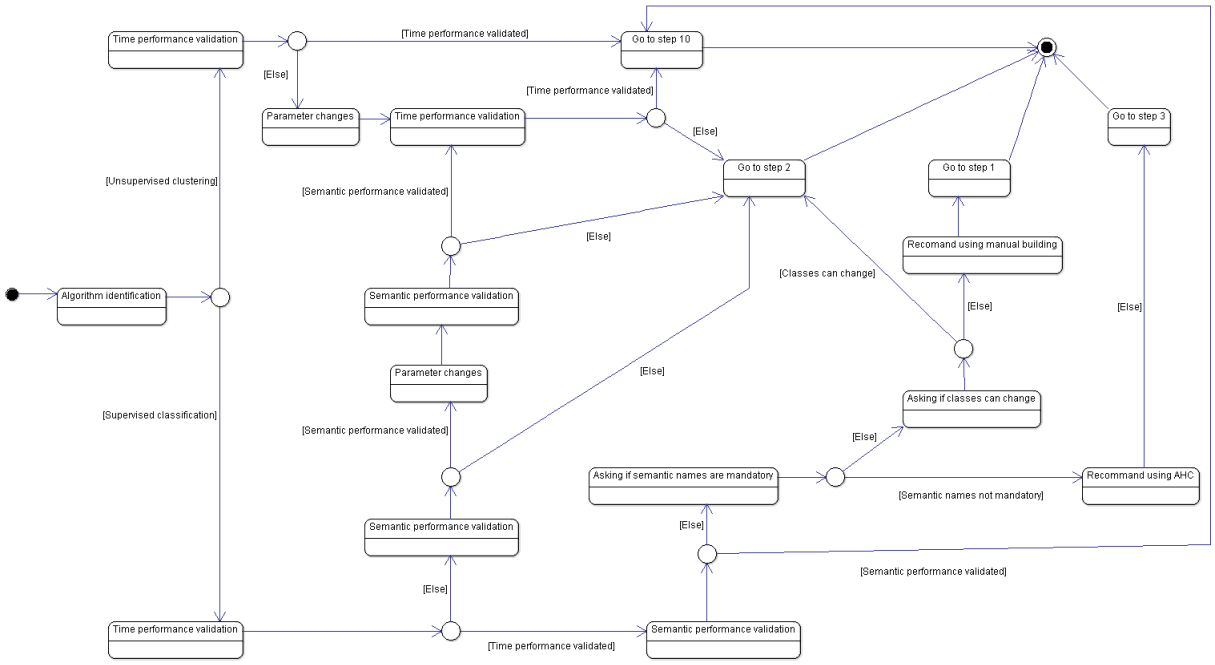


Figure 5.4 – DM hierarchy validation process

tering, it is not possible to limit the number of iterations, if we seek to obtain a complete hierarchy. If the time performance remains unsatisfactory despite parameter changes, go back to step 2.

5.2.5 The UML Profile associated with our methodology

As previously described, our methodology is based on the formalization of data mining and multidimensional requirements, using a conceptual multidimensional-DM model. We then extend the ICSOLAP UML profile defined by Bimonte, Boulil, Pinet, & Kang in 2013 (section 5.2.5.1) to include DM parameters in a new DMICSOLAP UML Profile (Section 5.2).

5.2.5.1 Preliminaries : ICSOLAP UML Profile

In this section, we present the main concepts of the ICSOLAP UML Profile, which defines a stereotype for each spatial multidimensional model (see (Boulil et al., 2015) for more details). The "*Hypercube*" [4] package stereotype defines an

4. In this section, we use quotes and italic font to name elements from the meta-model presented. Thus, "*Hypercube*" names a stereotype in the UML profile, whereas the word, hypercube,

OLAP hypercube. A hypercube contains at least one "*Dimension*" but only one *"Fact"*. The "*Fact*" class stereotype describes the fact, and it can present "*Measures*" and one or more "*DimRelationship*" associations. The "*Measure*" property stereotype defines a measure. The "*Dimension*" package stereotype represents a dimension. A dimension contains one or more hierarchies ("*Hierarchy*" package stereotype). Different dimension types have been defined, based on the type of data represented by the dimension. Thus, a dimension can be thematic ("*ThematicDimension*"), temporal ("*TemporalDimension*") or spatial ("*SpatialDimension*"). A dimension owns at least one "*Hierarchy*". In particular, the "*Hierarchy*" package stereotype contains several "*AggLevel*" levels, associated with the "*AggRelationship*" association. Like dimensions, hierarchies can be thematic ("*ThematicHierarchy*"), temporal ("*TemporalHierarchy*") or spatial ("*SpatialHierarchy*"). The "*AggLevel*" is a class stereotype, representing a level in a hierarchy. Each level presents particular attributes. The "DimensionalAttribute" property stereotype represents the descriptive attributes of each level. A level can own several attributes. There are three types of levels : the "*ThematicAggLevel*" class stereotype is used for classic levels ; The "*SpatialAggLevel*" class stereotype represents a spatial level, containing spatial data ; and the "*TemporalAggLevel*" class stereotype represents levels containing temporal data. The entire meta-model is presented in figure 5.5.

The spatial multidimensional model in figure **??** can be represented using the ICSOLAP UML profile as shown in figure 5.6.

Using this SOLAP model, decision-makers can answer queries like Q1 or Q2 in tableau 5.5. Decision-makers cannot aggregate abundance values on the spatial dimension from this SOLAP model, since it presents only one spatial level.

5.2.5.2 DM-ICSOLAP UML Profile for Data Mining

In this section, we describe the extension of the ICSOLAP profile to integrate supervised classification and hierarchical clustering (see figure 5.5 for the general meta-model of the DM-ICSOLAP UML Profile).

Supervised Classification

In this section, we describe the extension of the ICSOLAP UML to integrate the supervised classification algorithm during step 4 of our methodology (see section 5.2.4), and the transformation (step 8 of our methodology) of the conceptual multidimensional-MD model into a conceptual multidimensional model (see section 5.2.4).

without quotes refers to the concept of hypercube in general.

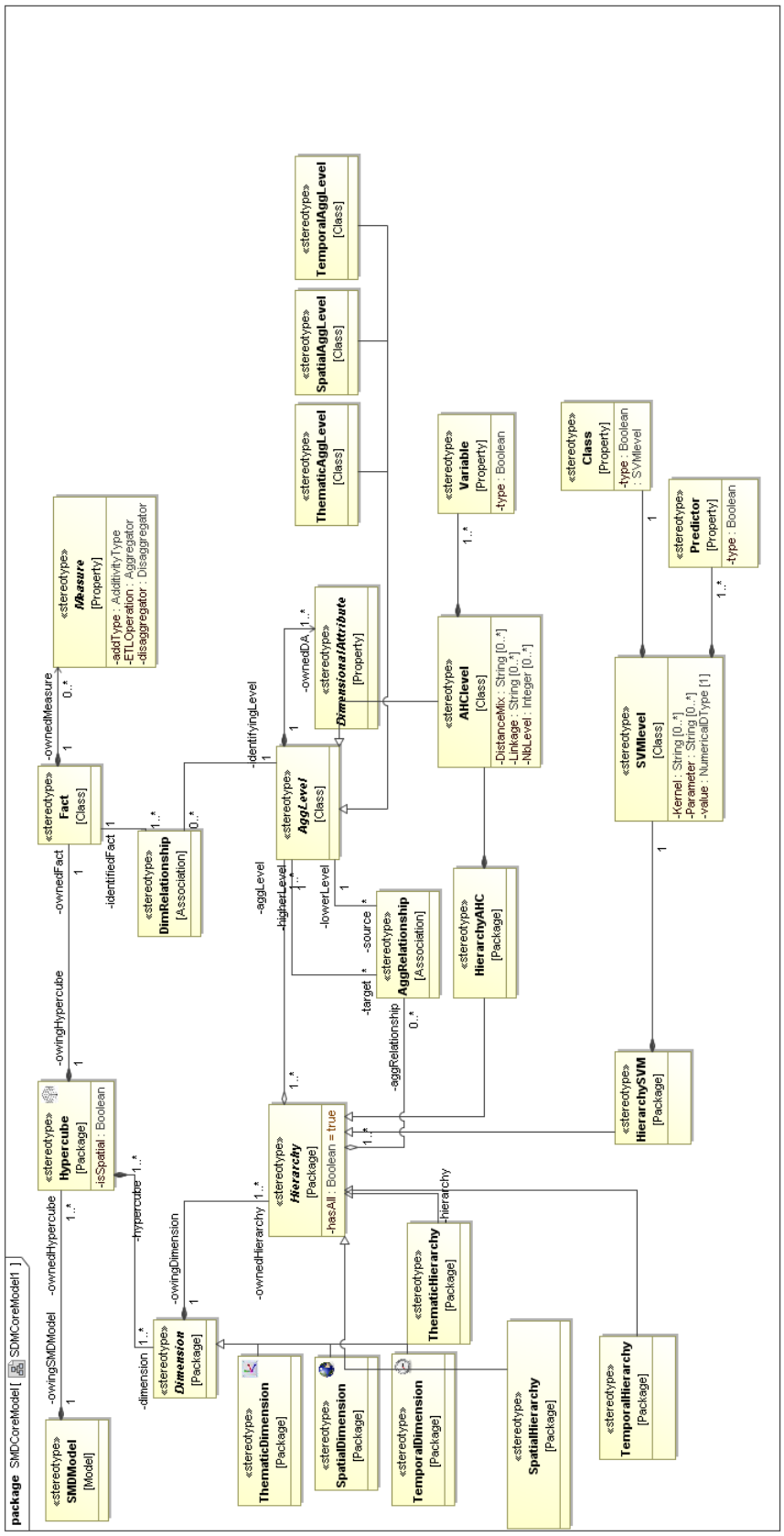


Figure 5.5 – the ICSOLAP UML meta-model

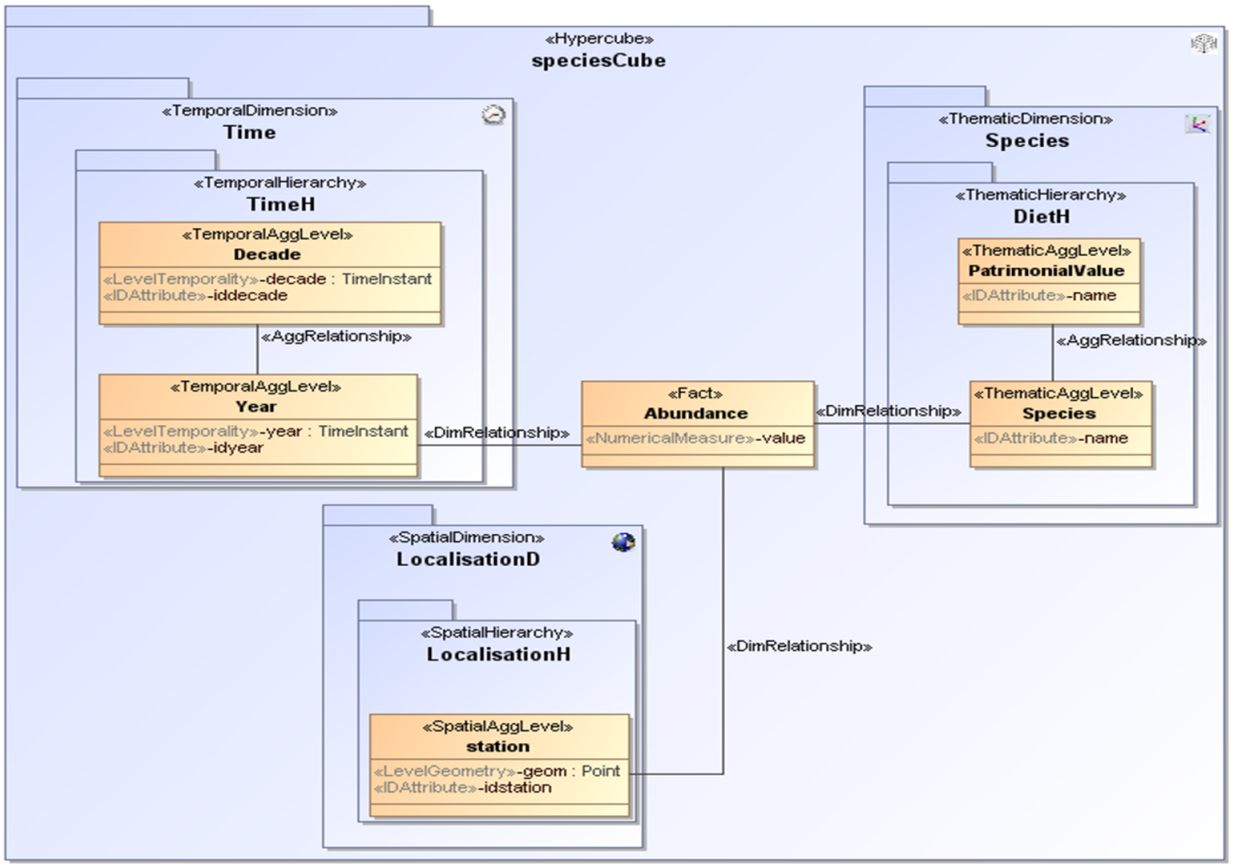


Figure 5.6 – UML Profile : bird biodiversity spatial multidimensional model from figure 4.1

Tableau 5.5 – Analysis needs related to each hierarchy type

Hierarchy type	Query 1	Query 2	Analysis needs
No hierarchy (figure 5.6)	Q1 : "*What is the total abundance value per census point, year and patrimonial value ?*"	Q2 : "*What is the total abundance value per census point, decade and species ?*"	Observing spatial and temporal changes in bird communities.
SVM hierarchy (figure 5.7)	Q1SVM : "*What is the total abundance value per class of stations, year and patrimonial value ?*"	Q2SVM : "*What is the total abundance value per class of stations, decade and species ?*"	Observing the impact of climate on bird communities in order to define if climate has an impact on bird abundances, and should therefore be taken into account during future analyses.
AHC hierarchy (figure 5.10)	Q1AHC : "*What is the total abundance value for the first group of stations, per year and patrimonial value ?*"	Q2AHC : "*What is the total abundance value for the second group of stations, per decade and species ?*"	Observing the impact of agriculture on bird abundances.

UML Profile

We extended the ICSOLAP profile to allow decision-makers to define the parameters of the SVM algorithm. In particular, we defined the "*HierarchySVM*" package stereotype that extends "*Hierarchy*". It is used for hierarchies that will be built by an SVM algorithm. A "*HierarchySVM*" hierarchy must contain only one "*SVMLevel*" level. The "*SVMLevel*" class stereotype extends the "*AggLevel*" class stereotype. This class has three tagged values : "*Kernel*" describing the kernel function used by the SVM, "*Parameter*", representing the name of the SVM parameter, and "*Value*" the value of the SVM parameter. Finally, an "*SVMLevel*" level owns one "*Class*" property, and one or more "*Predictor*" properties. In other terms, the "*SVMLevel*" class stereotype represents the inputs of the SVM algorithm. The "*Class*" property stereotype represents groups predicted during classification by SVM, and the "*Predictor*" property stereotype represents attributes used for prediction by an SVM classifier. Each "*Predictor*" has a tagged value "*type*", which specifies whether the attribute is quantitative or qualitative.

Finally, we completed our profile with constraints formulated with Object Constraint Language (OCL). An example of an OCL constraint is : "SVMLevel can only be present in a HierarchySVM" :

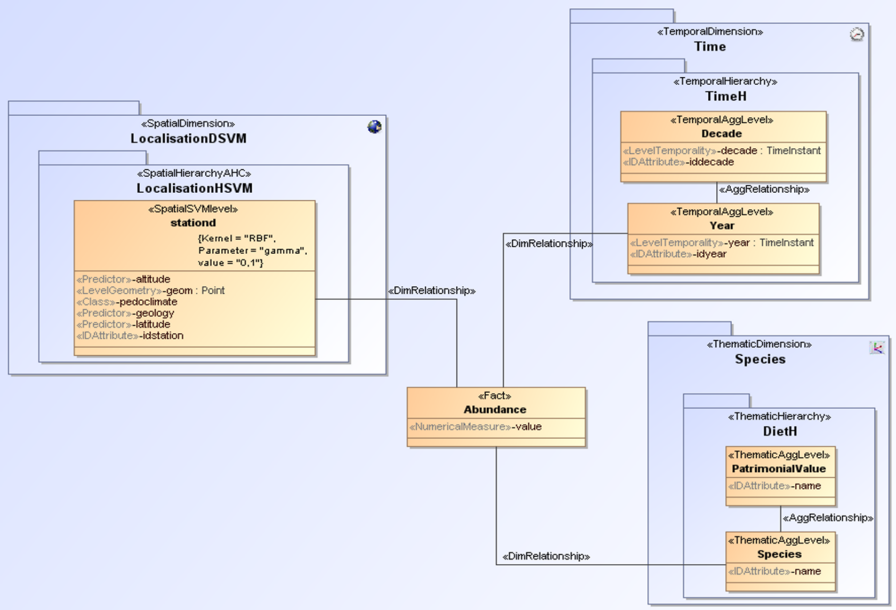


Figure 5.7 – Conceptual multidimensional-DM schema for supervised classification

```
Owner <<ThematicHierarchy>>
and <<SpatialHierarchy>>
and <<TemporalHierarchy>>
self.ownedMember
->select (m |m.oclIsTypeOf(SVMLevel)
->size()=0
```

Figure 5.7 shows the conceptual multidimensional-DM model of our case study. We can note three dimensions, two of which are classical dimensions (temporal, "Time", and thematic, "Species"), and one spatial dimension, composed of one hierarchy "LocationH". The fact represents the abundance.

The "LocationH" hierarchy is stereotyped as a "*HierarchySVM*" hierarchy, since it represents SVM parameters in the conceptual schema. This hierarchy contains only one level ("stationd").

The "stationd" level is a spatial level presenting a geometry ("geom"). It also presents a stereotyped "*Class*" attribute, "pedoclimate", representing the class values used by the SVM algorithm. This value will be predicted using the following stereotyped "*Predictor*" attributes : "latitude", "altitude" and "geology". In the general parameters of the algorithm, the kernel function is "RBF", the parameter is "gamma", and the value is "0.1".

Concerning SVM parameters, the kernel trick corresponds to a function used to find boundaries between classes during the learning phase (Tuffery, 2011). In our case, we used a Radial Basis Function (RBF) as the Gaussian kernel function. Depending on the kernel function used, certain parameters should be defined. Using a RBF kernel requires a "gamma" parameter, which represents the variation amplitude of the RBF kernel.

With this SVM algorithm, decision-makers aim to classify census points according to their attributes, in predefined classes, which represent census point type in terms of climate. Here, the objective is to automatically attribute a climate type to new census points (according to their features), i.e. classifying new census points without soliciting decision-makers.

Transformation

The transformation of a "*HierarchySVM*" into a "*Hierarchy*" is shown in figure 5.8 for the case of a spatial hierarchy. The same mapping is provided for the other types of "*HierarchySVM*".

After the learning phase of an SVM classifier, the transformation creates a new "*SpatialHierarchy*" for each "*SpatialHierarchySVM*". This new "*SpatialHierarchy*" has only two levels (described by the stereotype "*SpatialAgglevel*") :

1. The lowest level has the same "*LevelGeometry*" (our example is based on a spatial hierarchy : we classify spatial members), and a "*DescriptiveAttribute*" for each "*Predictor*" attribute.
2. The highest level has the same "*LevelGeometry*", and it represents predicted classes ; it contains only one descriptive attribute obtained for the "*Class*" attribute.

The other spatial multidimensional elements do not change.

The result of this transformation for the schema of figure 5.7 is shown in figure 5.9 : "LocationHRSVM" becomes a classic spatial hierarchy, including two "*SpatialAgglevel*" levels : "stationRSVM", which represents the lowest level (each predictor has become a descriptive attribute), and "*Class*", which represents the class predicted by the SVM.

Thus, this new spatial multidimensional model extends the model described in Figure 6, and decision-makers can enrich SOLAP queries previously described by grouping abundance values on the spatial dimension : (Q1SVM) "*What is the total abundance value per class of stations, year and patrimonial value ?*" or (Q2SVM) "*What is the total abundance value per class of stations, decade and species ?*" (tableau 5.5).

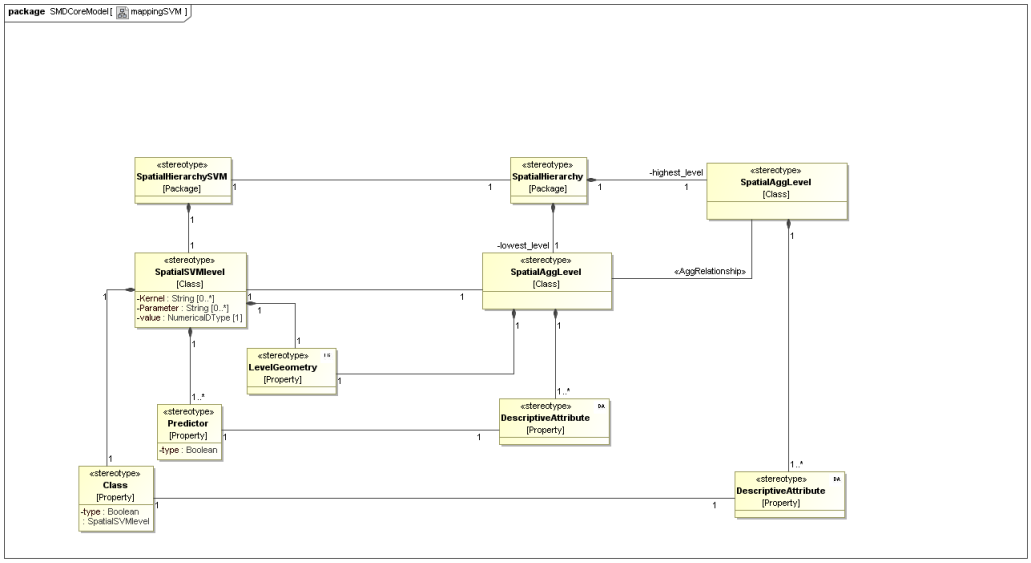


Figure 5.8 – Mapping to transform a "*HierarchySVM*" into a classical hierarchy

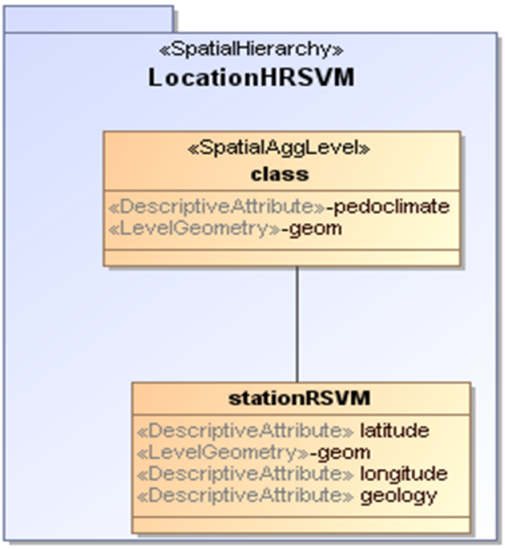


Figure 5.9 – Result of transformation mapping

Hierarchical Clustering

In this section, we describe the extension of the ICSOLAP UML to integrate the hierarchical algorithm during step 4 of our methodology, and the transformation (step 8 of our methodology) of the conceptual multidimensional-MD model into a conceptual multidimensional model (see section 5.2.4).

UML Profile

The "*HierarchyAHC*" package stereotype extends "*Hierarchy*". The "*HierarchyAHC*" hierarchy must contain only one "*AHCLevel*" level. Our extension of ICSOLAP defines a new stereotype, "*AHCLevel*", extending a level using a set of attributes with the "*Variable*" stereotype. The "*Variable*" stereotype represents attributes used by the AHC algorithm, for example the substratum. A "*Variable*" can be "Quantitative" or "Qualitative". We also define a tagged value, "*Linkage*", representing the linkage parameter of the algorithm, e.g. UPGMA, as in our case study, WPGMA, etc. In the same way, we defined three tagged values, representing the distance calculated when quantitative variables are used ("*DistanceQuantitative*"), when qualitative variables are used ("*DistanceQualitative*"), and when both qualitative and quantitative variables are used together, "*DistanceMix*". In our case study, "DistanceMix" has the value "Gower". Finally, the number of levels needed by decision-makers is represented by the "*LevelsNb*" tagged value. An example is shown in figure 5.9.

Decision-makers wanted to study the impact of agriculture on bird communities. To this end, they would group census points into homogeneous clusters, in terms of agricultural landscape. To describe agricultural profiles, they chose four attributes, measured in an area around each census point : the percentage of forest ("percentage_forest"), the percentage of cultivated area ("percentage_crop"), the most frequent type of livestock ("livestock_farming_type") and the most frequent type of crop ("crop_type"). In summary, decision-makers want to group census points according to agricultural profile, without predefining classes.

As explained in section 5.2.2, census point attributes can be qualitative or quantitative. Decision-makers chose distance and linkage parameters, based on environmental data features : using these parameters, it is possible to cluster mixed data. Decision-makers can choose the number of levels according to their analytical needs.

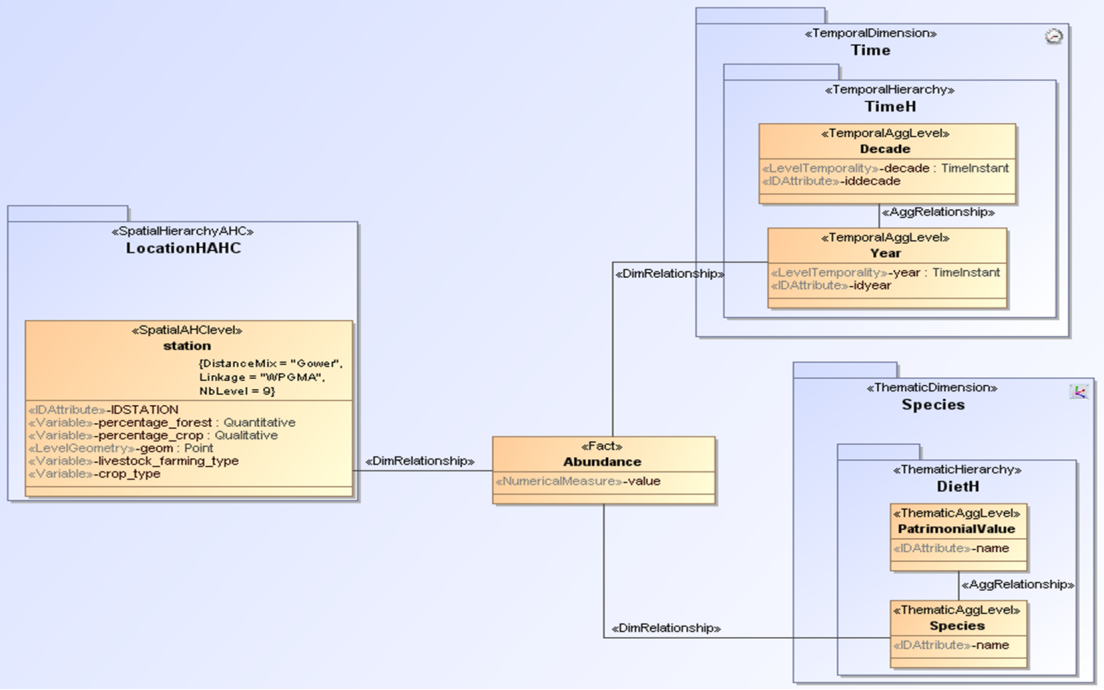


Figure 5.10 – Conceptual multidimensional-DM schema for hierarchical clustering

Transformation

As in the previous subsection, we need to define the mapping to transform a "*HierarchyAHC*" hierarchy into a classical hierarchy (figure 5.11). This mapping shows how a "*SpatialHierarchyAHC*" hierarchy becomes a classical "*SpatialHierarchy*" hierarchy. This new spatial hierarchy contains "*LevelsNb*" levels (a tagged value in the "*SpatialAHClevel*" level). The lowest level of this new hierarchy is a "*SpatialAgglevel*" level, and each descriptive attribute ("*DescriptiveAttribute*" property stereotype) of this new level corresponds to a "*Variable*" property in the source "*SpatialAHClevel*" level. All other levels in the new hierarchy are classical "*SpatialAggLevel*" levels. Relationships between members of different levels are defined by the AHC algorithm, run using parameters specified in the conceptual schema.

The result of this transformation is shown in figure 5.12 : "LocationH" becomes a classical spatial hierarchy and includes three "*SpatialAgglevel*" levels, named "stationR", representing the lowest level (each variable has become a descriptive attribute), "cluster1", and "cluster2", representing levels calculated by the AHC.

Thus, this new spatial multidimensional model extends the model described in figure 5.6, and decision-makers can enrich SOLAP queries previously described

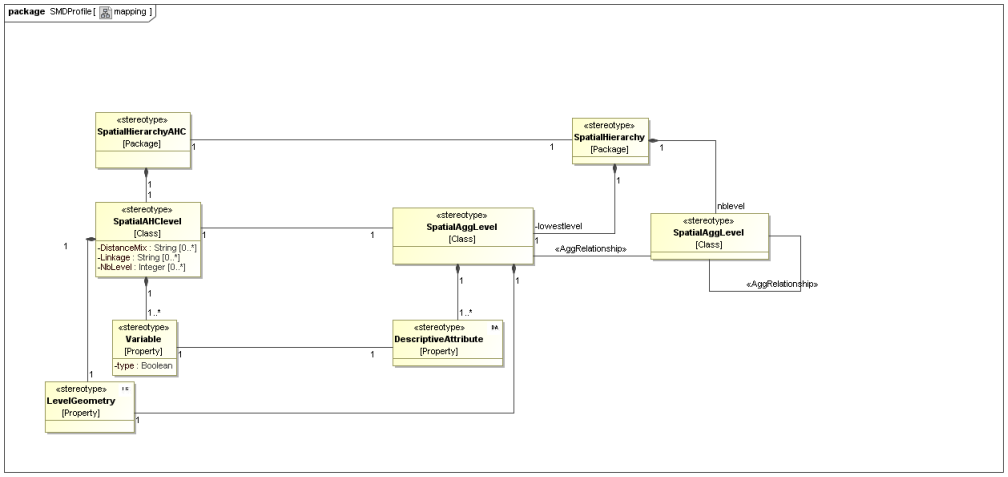


Figure 5.11 – Mapping to transform a "*HierarchyAHC*" into a classical hierarchy

by grouping abundance values with the spatial dimension : (Q1AHC) "*What is the total abundance value for the first group of stations per year per patrimonial value ?*" or (Q2AHC) "*What is the total abundance value for the second group of stations per decade per species ?*" (tableau 5.5).

5.2.6 The ProtOLAPMining tool

ProtOLAPMining is the system implementing our methodology. It extends ProtOLAP with the DM deployment tier (figure 5.13). ProtOLAP is based on a relational architecture with PostgreSQL as the DBMS for storing warehoused data, and Mondrian as the OLAP server. ProtOLAP takes as input the UML file representing the multidimensional model and automatically generates the SQL and Mondrian schemata (Design tier). The Design tier is implemented using the CASE tool Magic Draw, as described in section 5.2.5. The DW can be fed with the same sample data, using the Feeding tier.

The new DM deployment tier implements both AHC and SVM algorithms. With this tier, it is possible to set the input parameters and the location from which the DM algorithm should take the data. The tier runs the algorithm, and then creates the SQL and Mondrian schemata for the newly created hierarchy. Finally, other dimensions and measures are also automatically created, and decision-makers can analyze data with the OLAP client, SAIKU.

In order to illustrate the working of the newly developed tool, we provide some displays. First, figure 5.14 shows part of the DM Deployment Tier, which sets

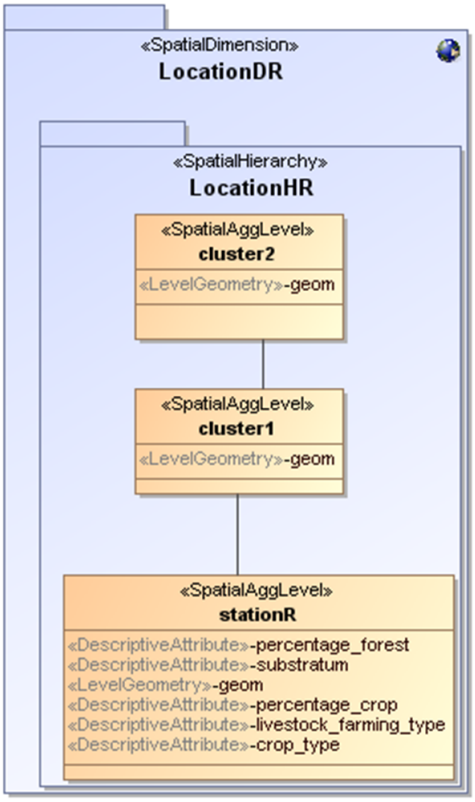


Figure 5.12 – Result of transformation mapping

the location of data used during the clustering process. Via this tier, OLAP designers and decision-makers set the tagged values and properties associated with the DM algorithms (see figure 5.5). As an example, concerning the AHC, displays in figure 5.14 can be used to set "Variable" and "IDAttribute" properties (see figure 5.10). Secondly, the DM Deployment Tier makes the DW Deployment Tier implement the new levels calculated by AHC or SVM algorithms.

Figure 5.15 presents the star schema obtained in PostgreSQL after rapid prototyping of our case study. Note that only AHC has been used to classify census points ("location_D" table in the star schema). The star schema presented is the schema obtained after the model transformation by the DM algorithm. This star schema is the implementation of the conceptual model shown in figure 5.10 and (concerning the spatial dimension) in figure 5.12 : the "abundances" table is a fact table, while "location_D", "species" and "years" are dimensions. The columns "level1" and "level2" in the "location_D" table come from the AHC calculation.

Finally, we provide in figure 5.16 an example of analysis performed with the Analysis Tier. In this figure, sums of bird abundances were presented by year and by member of the second level in the AHC hierarchy. In accordance with the conceptual model and its implementation in PostgreSQL, the Analysis Tier presents bird abundances as facts, with three dimensions : "LocationD", "Species" and "Time". In the spatial dimension, two hierarchies are available : the "geology" hierarchy, which comes from the native attributes, and the "LocationH" hierarchy, which is calculated by AHC.

5.2.7 Evaluation

To evaluate the performances of our methodology, we define an experimental framework based on the Goal Question Metric approach (section 5.2.7.1) describing semantic (section 5.2.7.3) and time performance (section 5.2.7.2). In section 5.2.7.4, we sum up performance evaluation.

5.2.7.1 Experimental framework

To evaluate our methodology, we need to verify that : i) time performance is suitable in a rapid prototyping approach ; and ii) the methodology defines good multidimensional models (facts, dimensions, and hierarchies) that correspond to decision-makers' analytical needs.

Therefore, to organize our experiments, we used the Goal Question Metric (GQM) approach (van Solingen et al., 2002). The GQM is a practical method for quality improvement of software development. It is based on the formal definition

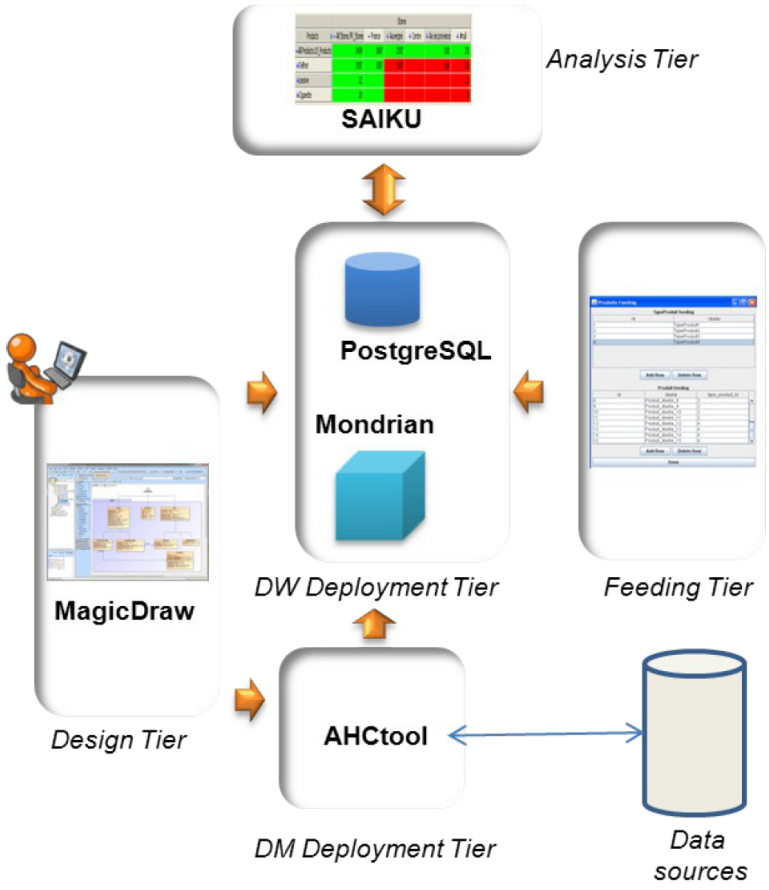
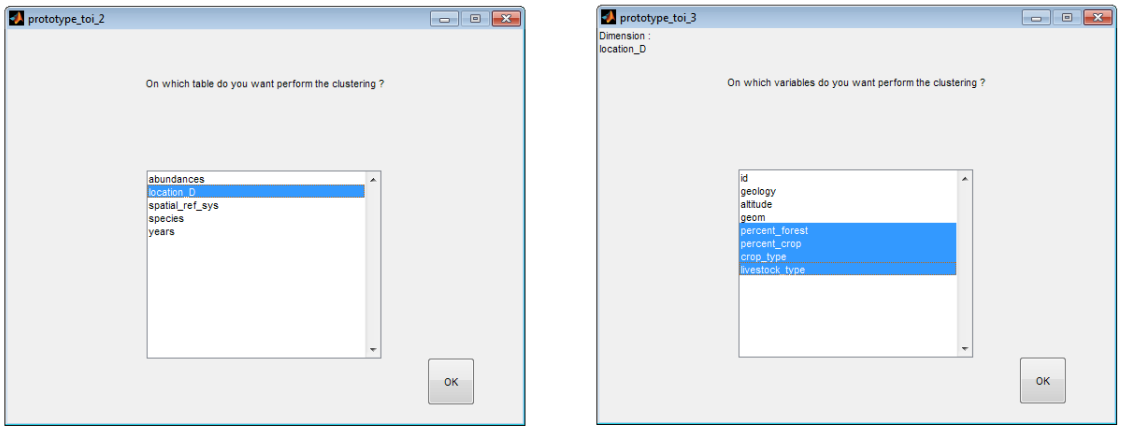


Figure 5.13 – ProtOLAPMining architecture



This module allows data to be chosen for a clustering process.

Figure 5.14 – Displays from the DM Deployment Tier

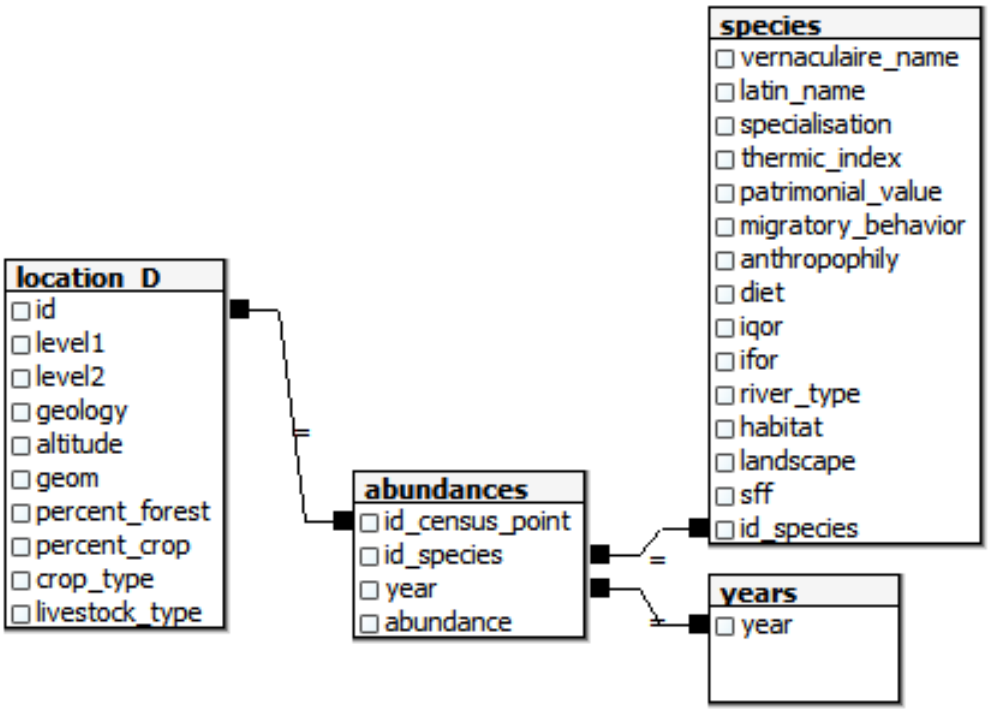


Figure 5.15 – Display of the DW deployment Tier from PostgreSQL

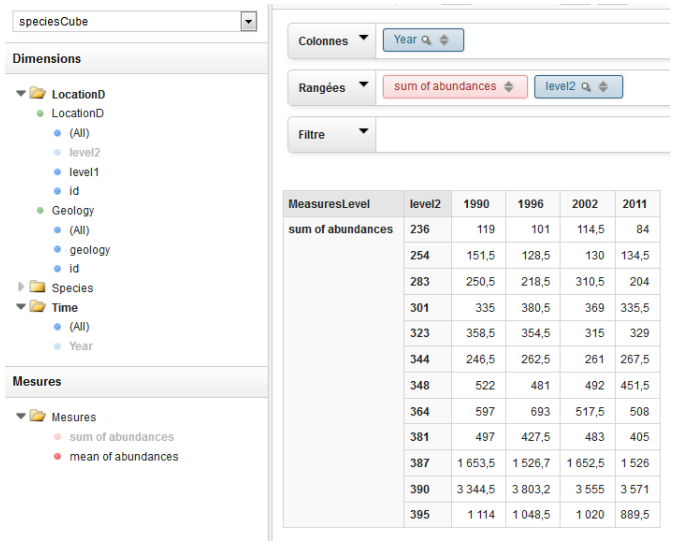


Figure 5.16 – Display of the Analysis Tier (Saiku)

of goals, evaluated through a set of questions, which are described by measurable values (metrics). The GQM has been successfully used in other domains.

To evaluate our methodology, we define a global purpose, which we describe using GQM parameters.

The purpose is to characterize the efficiency of the DMprotolap methodology in terms of design effort and design quality, from the point of view of the decision makers.

To achieve our purpose we define two subgoals, as described in tableau 5.6.

The first subgoal, SB1, aims to verify that the proposed methodology is well suited to the rapid prototyping context. In order to evaluate this point, we measured its rapidity using two quantitative metrics : the number of iterations needed to obtain a good multidimensional model ; the duration of each iteration (feed time + schema generation time + hierarchy generation time).

The second subgoal, SB2, aims to confirm that the methodology produces good models. For this purpose, we measured the correctness of the multidimensional model obtained, in terms of analysis axes (dimensions) and subjects (measures), with question Q2a and its metric M2a. We also checked that the hierarchies created are useful for decision-makers, by defining question Q2b, which is measured using a Boolean metric (M2b) that represents the satisfaction of the decision-makers, and some other quantitative metrics : M2c1, M2c2 and M2c3. These metrics, M2c1, M2c2 and M2c3 do not rely on decision-makers, but only on the DM algorithm

used, thus allowing some objective metrics to be included in our GQM framework. The first metric, M2c1, is the good classification rate for the SVM classifier, while M2c2 represents intra-cluster distances and M2c3 inter-cluster distances (Ward, 1963). Both M2c2 and M2c3 aim to evaluate the clustering performance of AHC.

To check whether the goal is achieved or not, we want to take account of all metrics, and interpret the answers to all questions. The interpretation model offers a synthesis of all metrics, in a formula validating (or not) whether the proposed methodology implies low effort to be complete while producing high-quality results (tableau 5.6).

In order to evaluate the metrics of our framework, we conducted a real-life experiment on our case study data, in collaboration with an ecologist. The ecologist was seeking to understand the impact of agriculture along the river on bird populations. First, she wanted to define agricultural profiles for each census point along the river. Secondly, she needed bird data : abundances for each species, and specific richness. Finally, she was aware that climate may have a strong impact on bird populations.

5.2.7.2 Subgoal 1 : Evaluation of time performances

In this section, we describe the evaluation of the metrics for Subgoal SB1.

The metric M1a corresponds to the number of iterations. Using a data-driven approach, we presented the decision-maker with the multidimensional model in Figure 8. This model was not validated, as it does not present richness (i.e. the number of bird species present) as a measure. In one further iteration, after the richness measure had been added, the decision-maker pointed out that census points had not been classified according to agricultural profile and pedo-climate. We therefore created hierarchies of census points using AHC and SVM. Agricultural profiles were obtained with AHC and pedo-climates were deduced with an SVM. However, as both hierarchies exist in one single dimension, decision-makers cannot directly visualize bird abundances according to both pedo-climate and agricultural profile in the same OLAP client. Therefore, in another iteration we suggested creating a new "census point" dimension, which contained the pedo-climate hierarchy, which can therefore be crossed with any other hierarchy of the initial "census point" dimension. *To conclude, 4 iterations were needed to achieve a good multidimensional model in our case study application : M1a : number of iterations= 4.*

The metric M1b corresponds to the sum of feed time + schema generation time + hierarchy generation time. To run the performance tests, we used a computer with an Intel® Xeon® processor and 16Go RAM ; the Operating System (OS) was Windows 7, 64-bits ; the prototype runs on MATLAB® 2013 software.

Tableau 5.6 – Our evaluation strategy formalized with the GQM method

Goal	***Subgoal***	***Question***	***Metric***
Purpose : characterize *Issue :* methodology *Object :* efficiency *Viewpoint :* from the decision-maker's viewpoint	SB1 : *Purpose :* characterize *Issue :* design *Object :* efforts *Viewpoint :* from the decision-maker's viewpoint	Q1 : Is the methodology rapid ?	M1a : number of iterations
			M1b : M1bi + M1bii+ M1biii where M1bi : feed time M1bii : schema generation time M1biii : hierarchy generation time
	SB2 : *Purpose :* characterize *Issue :* multidimensional design *Object :* quality *Viewpoint :* from the decision-maker's viewpoint	Q2a : Do the multidimensional model facts and dimensions correspond to analytical needs ?	M2a : feedback : true or false
		Q2b : Do the hierarchies built with DM correspond to analytical needs ?	M2b : feedback : true or false
			M2c1 : For SVM hierarchies : good classification rate
			M2c2 : For AHC hierarchies : intra-cluster distance
			M2c3 : For AHC hierarchies : inter-cluster distance

Interpretation model
IF M1a * M1b < "planned duration" AND M2a = "true" AND M2b = "true" AND M2c1 > "minimal rate accepted by decision-maker" AND M2c2 < M2c3 THEN the prototyping methodology is efficient.

First, we measured feed time. Our methodology is a prototyping methodology, so decision-makers feed the prototype with sample data only : not every member of each dimension is loaded during the design phase. Facts and some dimension members are automatically randomly generated by the design tool. Thus, the feed time is quite short. In our case study, the decision-maker needed only 30 minutes to feed the prototype.

Next, we measured schema generation time. Schema generation is performed by OLAP designers, with the support of the design tier and the deployment tier of the proposed tool. In our case study, the OLAP designers needed only one hour to design and deploy the prototype.

Thus, in our case study, feed time, M1bi = 30 minutes, schema generation time, M1bii = 60 minutes.

Finally, we evaluated hierarchy generation time, M1biii.

Concerning the AHC algorithm, we note that neither the number of clustered dimension members nor the number of attributes used to perform the clustering process had any real effect on calculation time. We observed a median calculation time of 80 sec., with a minimum calculation time of 60 sec. and a maximum calculation time of 84 sec. These results were obtained from 342 hierarchies (the number of clustered dimension members varied from 20 to 190, and the number of attributes used for classification varied from 20 to 100). In fact, no strong correlation between calculation time and data volume was found, possibly due to the low volume of data used for this test.

Concerning the SVM algorithm, the entire calculation process (which included a cross-validation with ten partitions, the sampling of training data and test data, the classifier learning phase and the classifier test phase. See Section 7.3 for more details about setting up the SVM classifier) took 1.113 seconds. This included 0.872 seconds for SVM classification and 0.128 seconds for SVM training. Thus, the SVM algorithm is very time efficient and totally compatible with a rapid prototyping methodology.

Thus, in our case study, hierarchy generation time, M1biii was less than two minutes.

We assume that descriptions of a dimension member containing several decades of variables will be uncommon. In fact, few DW models contain hundreds of attributes describing members of a dimension. We therefore consider that the performances of our data mining algorithms are adequate for a rapid prototyping method, because these algorithms work quickly with a large set of dimensional attributes.

In conclusion, the metric M1b, which evaluates the duration of an iteration du-

ring the prototyping process, is approximately equal to 91 minutes for our case study.

5.2.7.3 Subgoal 2 : Evaluation of design quality

In this section, we evaluated the metrics related to subgoal 2, which concerns design quality.

First, concerning the quality of the prototyped multidimensional model, after the iterations described in the previous section, the decision-maker considered that the multidimensional model was satisfactory in terms of facts and dimensions. Therefore, the value of M2a is "TRUE".

Secondly, concerning the mined hierarchy quality, we estimated user performances of data mining algorithms : we checked if the hierarchical structures deduced by the SVM algorithm and the AHC algorithm were appropriate according to the decision-maker. As the mined hierarchies were judged satisfactory, the value of M2b is "TRUE".

To complete our evaluation of hierarchies, we also used less subjective performance metrics.

Concerning SVM, we noted that semantic performance was not very satisfactory in the first instance : the decision-maker sought a classification rate higher than 95%, (M2c1 metric in tableau 5.6), but this condition was not fulfilled. To improve the semantic performance of our supervised classification, we proposed a cross-validation with ten partitions (Tuffery, 2011). Thus, ten SVM classifiers were trained. After that, to decide into which class a new census point should be integrated, each classifier classified the new dimension member, which was then integrated into the most frequently chosen class. With this method, only 3 dimension members out of 198 were wrongly classified (a classification rate equal to 98.5%). Such cross-validation can only be used when the SVM algorithm is very efficient, in terms of time performance. *In conclusion, the M2c1 metric performs above the minimum rate accepted by the decision-maker.*

Concerning the AHC hierarchy, we briefly evaluated clustering performances by comparing intra-cluster distances (M2c2 metric in Table 6) and inter-cluster distances (M2c3 metric in tableau 5.6). Intra-cluster distances represent distances between dimension members in the same cluster after the AHC. In contrast, inter-cluster distances represent distances between dimension members in different clusters after the AHC (Ward, 1963). We calculated the average distances between dimension members according to the clusters containing these dimension members. In the following table, the cell [Cluster 1, Cluster 1] contains the average

distance between two dimension members (i.e. census points) in cluster 1, while the cell [Cluster 1, Cluster 2] contains the average distance between two dimension members, one in cluster 1 and the other in cluster 2. In this table, intra-cluster distances (cells [Cluster 1, Cluster 1], [Cluster 2, Cluster 2] and [Cluster 3, Cluster 3]) are lower on average than inter-cluster distances (cells [Cluster 1, Cluster 2], [Cluster 1, Cluster 3], [Cluster 2, Cluster 3]). To verify the significance of this result, we performed a one-way ANOVA on these distances, and thus confirmed that these averages are significantly different ($F = 914.37$, $p << 5\%$). *In conclusion, in each case, we observe that M2c2 < M2c3.*

Tableau 5.7 – Average distance between dimension members, calculated by cluster

	Cluster 1	**Cluster 2**	**Cluster 3**
Cluster 1	0.36	0.55	0.56
Cluster 2	-	0.44	0.59
Cluster 3	-	-	0.39

5.2.7.4 Interpretation model and final methodology evaluation

To conclude this performance evaluation, we provide a summary of the metrics necessary to obtain a fully satisfactory OLAP prototype, according to decision-maker opinion. In this section, we will refer to the goals, questions and metrics defined in tableau 5.6, section 5.2.7.1. All metrics measured are summarized in tableau 5.8.

First, we characterized the design effort necessary from the decision-maker's point of view (SB1 in tableau 5.6). The number of iterations, M1a, is equal to 4 and the time required to perform an iteration, M1B, is 91 minutes. Thus, the time necessary to define a satisfying prototype is 364 minutes (i.e. approximately 6 hours). Therefore, we consider that the methodology is rapid (Q1 in tableau 5.6).

Secondly, we characterized the multidimensional design quality from the decision-maker's point of view (SB2 in tableau 5.6). Decision-maker feedback on the multidimensional model (M2a in tableau 5.6) was positive. We can therefore consider that the multidimensional model facts and dimensions correspond to analytical needs (Q2a in tableau 5.6). Decision-maker feedback on the calculated hierarchies, M2b, was positive, the successful classification rate of the proposed SVM, M2c1, was 98.5% and, for the AHC hierarchy, the intra-cluster distance, M2c2, was lower than the inter-cluster distance, M2c3. Therefore we consider that the hierarchies built with DM correspond to analytical needs (Q2b in tableau 5.6).

*In conclusion, since M1a*M1b is brief, M2a and M2b are equal to "TRUE",*

Tableau 5.8 – Summary of metrics measured during evaluation

Metrics
M1a : number of iterations **= 4**
M1b : M1bi + M1bii+ M1biii where M1bi : feed time = **30 min** M1bii : schema generation time = **60 min** M1biii : hierarchy generation time < **2 min**
M2a : feedback : **TRUE**
M2b : feedback : **TRUE**
M2c1 : For SVM hierarchies : good classification rate = **98.5 %**
M2c2 : For AHC hierarchies : intra-cluster distance = **0.397** (on average)
M2c3 : For AHC hierarchies : inter-cluster distance = **0.567** (on average)
Interpretation model
IF M1a * M1b < "planned duration" AND M2a = "true" AND M2b = "true" AND M2c1 > "minimal rate accepted by decision-makers" AND M2c2 < M2c3, THEN the prototyping methodology is efficient.

M2c1 is higher than 95%, and M2c2 is lower than M2c3, our prototyping methodology is efficient.

It is important to note that we used our case study to evaluate the performances of the DM algorithms and of our rapid prototyping methodology. To generalize these performances, this methodology and this tool must be applied to several case studies. Therefore, although we observed satisfying performances for our case study, we cannot attest that such performances are valid for all possible cases.

5.2.8 Conclusion and Future Works

In this paper, we presented a rapid prototyping methodology to design data warehouses (DW).

The main idea of this methodology is that decision-makers are not OLAP experts and can find it difficult to deal with the concepts associated with DW and OLAP. In order to simplify user participation in the design process, our methodology builds a prototype of the future DW that decision-makers can "play" with. In this way, decision-makers can easily express their design requirements.

In some cases, OLAP systems may possess useful features for data analysis but, unfortunately, our data have a structure incompatible with a multidimensional schema. In our case study, the dimensional data had no hierarchical structure. Therefore, we integrated a DM process into our methodology, to discover hierarchies in dimension member attributes. We propose a framework that will choose a DM algorithm (supervised classification or unsupervised clustering) according to decision-maker requirements and knowledge of the future hierarchy.

To support DW design with the proposed methodology, we have developed : (i) a UML Profile for multidimensional schemata, integrating DM parameters and results, (ii) a prototyping tool, to maximize the automation of our methodology.

Finally, we evaluated the time and semantic performances of our methodology and tool on a case study from avian ecology.

Long-term assessment of DM algorithm performances has already begun. For the moment, clustering and supervised classification efficiency is evaluated according to decision-maker requirements during the prototyping process. But we also seek to evaluate the performance of supervised or unsupervised classifiers in an operational DW after many years of use. Our future work will explore the following questions : (i) Have DM results and parameters a life cycle compatible with the long life of the DW ? (ii) How can the DM algorithms integrated into the prototyped DW be updated in evolving requirement contexts ?

Acknowledgments

Data acquisition received financial support from FEDER Loire, Etablissement Public Loire, DREAL Bassin Centre, the Région Bourgogne (PARI, Projet Agrale 5) and the French Ministry of Agriculture.

We also heartily thank Carmela Chateau, from UFR-SVTE, Université de Bourgogne, France, for her help in rereading.

Chapitre 6

Enrichissement de dimension avec des données factuelles

Ce chapitre est consacré au troisième objectif, défini dans le Chapitre 1.2, intitulé "Proposer une méthode capable de construire automatiquement une hiérarchie en prenant en compte la source des données utilisées pour construire cette hiérarchie et son contexte". **Le contenu de ce chapitre a été publié dans la *Revue des Nouvelles Technologies de l'Information* (n°B.11), suite aux *Journées francophones sur les Entrepôts de Données et l'Analyse en ligne* (EDA) en 2015. Ces travaux ont également été présentés pendant la conférence *International Conference on Enterprise Information Systems* (ICEIS) en 2015. La version de l'article envoyé à ICEIS 2015, et qui est reproduite dans ce chapitre, a été sélectionnée pour être publiée dans la série *Lecture Notes in Business Information Processing*.**

Ce chapitre est organisé en deux sections :

— La Section 6.1 propose une synthèse (en français) du contexte, de la problématique, de la méthodologie et des résultats proposés dans la section suivante. Cette synthèse proposera également une conclusion sur cette contribution et replacera les résultats obtenus dans le contexte de la thèse.

— La Section 6.2 correspond au texte figurant dans les actes de la conférence ICEIS 2015 (en anglais).

6.1 Synthèse sur l'enrichissement de dimension avec des données factuelles

Les hiérarchies sont des structures cruciales dans un entrepôt de données puisqu'elles permettent l'agrégation de mesures dans le but de proposer une vue analytique plus ou moins globale sur les données entreposées, selon le niveau hiérarchique auquel on se place. Pour ces raisons, plusieurs travaux se penchent sur la définition de hiérarchies grâce a des algorithmes de fouille de données (Favre et al., 2006; Sautot et al., 2015). Cependant, cette phase de conception est appliquée une fois que le modèle multidimensionnel a été défini et elle prend en compte uniquement les membres d'une dimension, et les faits et les autres dimensions du modèle en constellation ne sont pas impactés.

De notre point de vue, ces méthodologies présentent une limitation importante car, dans les projets réels d'entrepôt de données, les données qui décrivent les membres d'une dimension, et qui peuvent donc être utilisées pour créer une nouvelle hiérarchie, sont souvent issues de différents faits et dimensions préexistants dans le schéma multidimensionnel.

Dans notre cas d'étude, les décideurs ont besoin d'une nouvelle méthode de conception qui groupe les points d'écoute (données dimensionnelles) selon les paramètres environnementaux (données factuelles) et les années (données dimensionnelles).

Cependant, une requête OLAP utilisant les données environnementales de 2002 pour décrire les membres de la dimension spatiale et les abondances des oiseaux récoltées en 2011 n'est pas cohérente car elle associe le nombre d'oiseaux en 2011 avec la configuration géographique et environnementale de 2002, pouvant ainsi induire des interprétations erronées. C'est pourquoi, il est important de prendre en compte ces différents contextes lors d'une session d'analyse OLAP impliquant une navigation dans la dimension temporelle. Par exemple, la requête "*quel est le nombre total d'oiseaux en 2002 et dans les points d'écoutes ayants les même paramètres environnementaux ?*" doit utiliser uniquement des données environnementales de 2002 pour décrire les points d'écoute.

Dans cette contribution, nous introduirons notre méthodologie pour l'enrichissement d'un schéma multidimensionnel grâce à une approche mixte. L'idée principale est d'utiliser une méthodologie existante centrée sur les données comme première étape, afin d'obtenir un schéma multidimensionnel en constellation. Après cela, nous enrichirons le modèle multidimensionnel obtenu : nous collecterons les besoins des utilisateurs à propos des hiérarchies qui ne peuvent pas être déduites par l'analyse de dépendances fonctionnelles puis, ces besoins utilisateur seront expri-

més sous la forme de faits existant dans le modèle multidimensionnel à intégrer à une dimension. En fait, notre idée principale est de fournir un algorithme qui transforme le schéma multidimensionnel en constellation en éliminant un noeud factuel et en intégrant les données factuelles dans une dimension associée, où elle seront utilisées pour créer de nouveaux niveaux. Pour réaliser cela, nous formaliserons le modèle multidimensionnel sous forme de graphe multidimensionnel.

Notre approche permet donc d'enrichir une dimension avec une nouvelle hiérarchie, qui intègre différentes versions, car le contexte multidimensionnel du fait source et de la dimension cible doivent être pris en compte. Cela implique une transformation du modèle en constellation. En d'autre termes, l'enrichissement d'une hiérarchie correspond à une nouvelle phase de conception, qui impacte tout le modèle en constellation.

En conclusion, la conception d'un entrepôt de données est une tâche complexe et cruciale, qui dépend des sources de données disponibles et des besoins en termes d'analyses décisionnelles. Une des étapes de cette démarche de conception est la définition de hiérarchies. Les travaux existant exploitent peu l'environnement factuel de la dimension considérée pour créer automatiquement des hiérarchies complexes. Ainsi, dans cet article, nous avons présenté une méthodologie mixte d'enrichissement d'un schéma multidimensionnel, qui transforme un schéma en constellation, en définissant de nouvelles hiérarchies grâce à la Classification Ascendante Hiérarchique. De plus, nous avons présenté une implémentation de cet algorithme sur une architecture ROLAP. Nous avons testé la méthodologie que nous proposons sur un cas applicatif réel, issu de l'étude de la biodiversité au sein des peuplements d'oiseaux. En fait, les méthodologies automatiques actuelles de conception multidimensionnelle ne peuvent pas produire un schéma multidimensionnel qui couvre les besoins des décideurs, en raison de la complexité des données. Notre méthodologie propose d'enrichir une dimension avec des données factuelles, et par ce moyen, transforme le schéma multidimensionnel afin de rendre possible de nouvelles analyses des données.

6.2 Mixed driven Refinement design of Multidimensional models based on Agglomerative Hierarchical Clustering

Abstract

Data warehouses (DW) and OLAP systems are business intelligence technologies allowing the on-line analysis of huge volume of data according to users' needs. The success of DW projects essentially depends on the design phase where functional requirements meet data sources (mixed design methodology) (Phipps and Davis, 2002). However, when dealing with complex applications existing design methodologies seem inefficient since decision-makers define functional requirements that cannot be deduced from data sources (data driven approach) and/or they have not sufficient application domain knowledge (user driven approach) (Sautot et al., 2014b). Therefore, in this paper we propose a new mixed refinement design methodology where the classical data-driven approach is enhanced with data mining to create new dimensions hierarchies. A tool implementing our approach is also presented to validate our theoretical proposal.

Keywords

Multidimensional design, Data Warehouse, OLAP, Data Mining

6.2.1 Introduction

Data warehouses (DW) and OLAP systems are business intelligence technologies allowing the online analysis of huge volume of data. Warehoused data is organized according to the multidimensional model that defines the concepts of dimensions and facts. Dimensions represent analysis axes and they are organized in hierarchies. Facts are the analysis subjects and they are described by numerical indicators called measures. Warehoused data are then explored and aggregated using OLAP operators (e.g. Roll-up, Slice, etc.) (Kimball, 1996).

The success of DW projects essentially depends on the design phase where functional requirements meet data sources (Phipps and Davis, 2002). Three main methodologies have been developed : userdriven, datadriven and mixed (Romero and Abello, 2009). User-driven approach puts decision-makers at the center of the design phase by providing them tools to define the multidimensional model exclusively according to their analysis needs. Usually, data driven methodology

proposals deduce the multidimensional model from structured and semistructured (Mahboubi et al., 2009; Jensen et al., 2004) data sources exploiting metadata (e.g. foreign keys) and some empirical values. Finally, mixed approaches fusion the two previous described methods.

Hierarchies are crucial structures in DW since they allow aggregation of measures in order to provide a global and general analytic view of warehoused data. For that reasons, some works investigate definition of hierarchies by means of Data Mining (DM) algorithms (Favre et al., 2006; Sautot et al., 2015). However, this design step is applied once the multidimensional model has been defined, and it takes into account only members of one dimension.

From our point of view, these methodologies present an important limitation since in real DW projects often those DM algorithms need data of different dimensions and facts. Thus, in this paper we present a framework for a mixed design of multidimensional models by integrating DM algorithms in a classical data driven-approach. This allows defining hierarchical structures, according to decisional users' requirements, that cannot be deduced by classical datadriven methods. This hierarchical organization of dimensional data is translated in a complex multifactual multidimensional model in order to represent as well as possible semantic of data sources.

The paper is organized in the following way : Section 6.2.2 introduces related work ; a retail case study and the motivation are presented in Section 6.2.3 ; our design method is detailed in Section 6.2.4 and its implementation is shown on Section 6.2.5.

6.2.2 Related Work

Three types of approaches can be used to design a data warehouse : (i) Methods based on user specifications, or demand-driven approaches ; (ii) Methods based on available data, or data-driven approaches ; (iii) Mixed methods, or hybrid approaches. For example, (Jovanovic et al., 2012) is an iterative demanddriven method where at each iteration, the system searches for the best data corresponding with the information required by the user in terms of dimensions or facts. Moreover, several other have proposed systems based on hybrid approach such as (Romero and Abello, 2010) that propose to express functional requirements using SQL queries.

Relational data driven approaches deduce multidimensional structures (facts and dimensions) from conceptual (Phipps and Davis, 2002) and/or logical models (Carme et al., 2010; Jensen et al., 2004). In particular some works investigate automatic discovering facts using some heuristics (Carme et al., 2010). About

Tableau 6.1 – Summary of literature review related to automatic hierarchy building

		Data Sources		
		Star Schema		Constellation Schema
		One Dimension	Facts	Facts and dimensions
Algorithm	K-means	(Bentayeb, 2008)	(Bentayeb, 2008)	
	Hierarchical Classification	(Ceci et al., 2011)	(Messaoud et al., 2004)	Our proposal
	Other	(Favre et al., 2006; Nguyen et al., 2000)	(Leonhardi et al., 2010)	

dimensions some works propose using logical database metadata such as foreign keys (Jensen et al., 2004) or some heuristics.

Other works use more complex algorithm to identify dimensions hierarchies. Nguyen and Tjoa propose a system to dynamically build hierarchies based on data from Twitter (Nguyen et al., 2000). Ben Messaoud *et al.* (Messaoud et al., 2004) present a new OLAP operator named OPAC that allows to aggregate facts that refer to complex objects, such as images. This operator is based on hierarchical clustering algorithm. Favre *et al.* (Favre et al., 2006) provide a framework for automatic defining hierarchies according to user rules. In order to personalize the multidimensional schema, Bentayeb (Bentayeb, 2008) propose to create new levels in a hierarchy with the K-means algorithm. Leonhardi *et al.* (Leonhardi et al., 2010) propose to increase the OLAP cube exploration functionalities by providing the user data mining algorithms to analyze data. Ceci *et al.* (Ceci et al., 2011) use a hierarchical clustering to integrate continuous variables as dimensions in an OLAP schema. In the same line, (Sautot et al., 2015) propose using Agglomerative Clustering for designing hierarchies, and the integration in a rapid prototyping methodology is presented in (Sautot et al., 2014). However, all existing works define hierarchies using only either dimensional data (i.e. attributes of dimension members) or factual data (i.e. measures) (see Table *6.1*). But, in a constellation schema, a dimension can be enriched with a hierarchy created by using other dimensions and facts. It means that the creation of a new hierarchy can involved a refinement of facts and dimensions in the entier constellation schema. We detail this issue in the following section, using a real application case from bird biodiversity.

6.2.3 Motivation

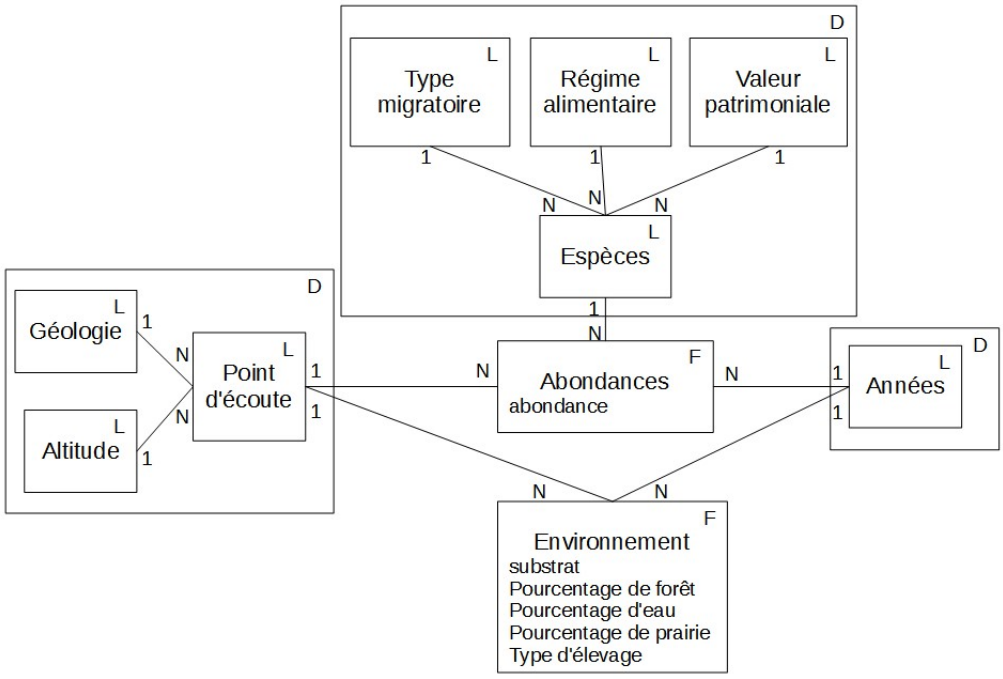
In order to describe motivation of our new DW design methodology we present in this section a real case study concerning the bird biodiversity analysis (Sautot et al., 2015). This dataset has been collected to analyze spatio-temporal changes

in bird populations along the Loire River (France) and to identify local and global environmental factors that can explain these changes. Data sources are stored in a relational database (PostGIS). Applying the data driven algorithm proposed in (Romero and Abello, 2010), we obtain the constellation schema depicted in Figure 6.1, which presents two facts as described in the following. Abundances is one fact, and can be analyzed according to three dimensions (an instance is shown on Table 6.3) : (i) the species dimension, which stores species names and attributes, (ii) the time dimension, which corresponds to the census years and (iii) the spatial dimension, which describes census points along the river. Using this model decisionmaker can answer to queries like : "*What is the total of birds per year and census point ?*" or "*What is the total of birds per year and altitude ?*". To complete bird census, the landscape and the river are described around each census point. Environment descriptions are represented by another fact, which is associated to the time dimension and the spatial dimension. With this model, it is possible to describe census points, for example a possible OLAP query is "*What is the percentage of forest per census point in 2012 ?*".

Note that descriptions of census points that are not dependent from time, such as altitude and geology, are used as spatial dimension levels, while other attributes are represented as measures of another fact (e.g. percentage of forest). Unfortunately, abundances for a specie have not meaning if not related to environmental data of census points. In this situation a drill-across operation is not adequate since it will hide the species dimension. Indeed, with the drill-across operators facts are joined only on common dimensions. Moreover, the multidimensional model of Figure 6.1 does not make possible to provide the decision-makers with OLAP queries aggregating abundance by classes of environmental variable (30% of forest, 50% of water, etc.), for example "*What is the total of birds per year and group of census point with 30% of forest ?*" or "*What is the total of birds per year and group of census point with 50% of water ?*", since environmental parameters do not appear as levels, but as measures, prohibiting group-by queries.

Therefore, in our case study, decision-makers need for a new design method that group census points (dimensional data) by environmental parameters (factual data) and year (dimensional data).

The multidimensional model allowing correct OLAP analysis should be the one shown on Figure 6.2 (Miquel et al., 2002b). This multidimensional schema presents only one fact and the spatial dimension is enriched with some levels representing group of environmental parameters for each year. Indeed, environmental parameters for census points in 2001 can be different from ones of 2002 implying that the same census point is not grouped in the same level on two different years as shown on Table 6.2. For example, data describing agricultural activities around



This formalism is proposed by (Golfarelli and Rizzi, 2009).

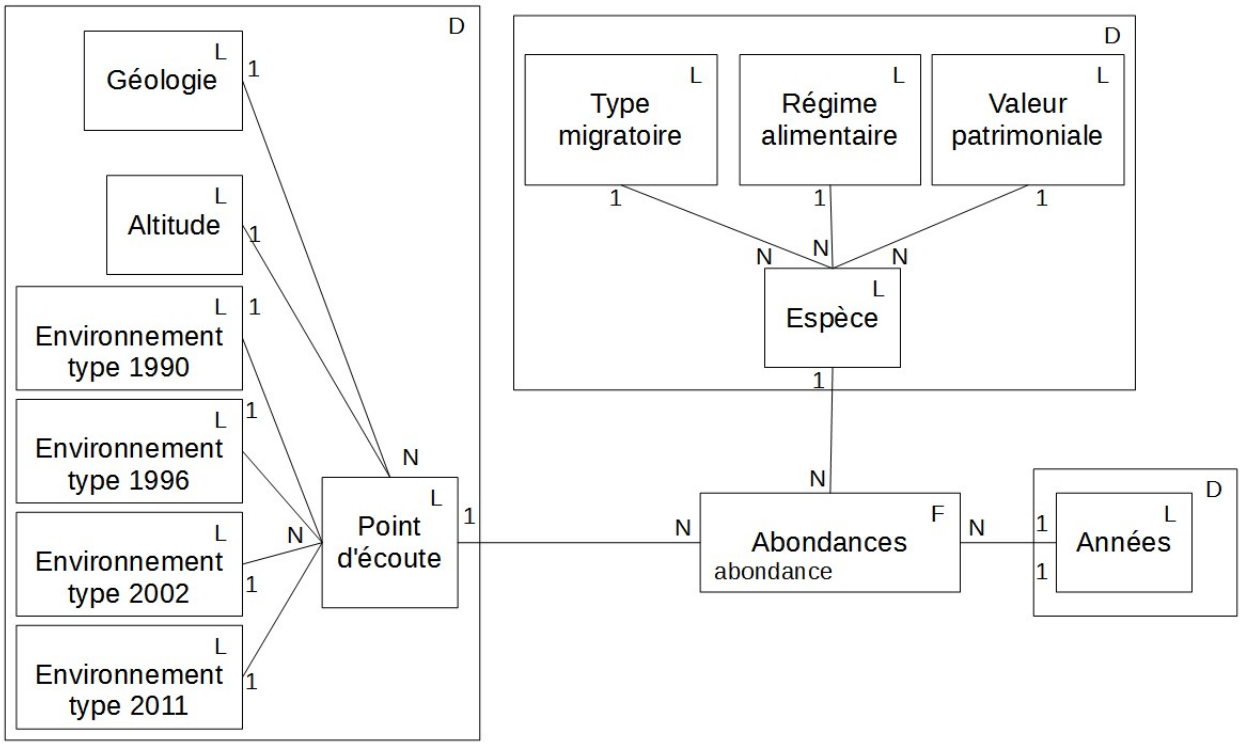
Figure 6.1 – Bird biodiversity case study : Data-driven constellation schema

the census points, are available only for the 2002 census campaign. Therefore, it is important to take into this different classification when navigating on the temporal dimension during an OLAP analysis session.

For example, the query "*What is the total of birds in 2002 and in census points with the same environmental parameters ?*" has to use the environment type 2002 level, and "*What is the total of birds in 2011 and in census points with the same environmental parameters ?*" has to use the environment type 2011 level. For example an OLAP query using the environment type 2002 level and the temporal member 2011 is not coherent since it associates the number of birds on 2011 in the past geographical-environmental configuration of 2002, leading to erroneous interpretation.

6.2.4 Our Proposal

In this section we introduce our framework for the refinement of multidimensional in a mixed approach. The main idea of our proposal is using an existing data driven methodology in a first step. Then, in our new design step, we collect user needs about hierarchies that are not been deduced in the multidimensional schema



This formalism is proposed by (Golfarelli and Rizzi, 2009).

Figure 6.2 – Bird biodiversity case study : manually driven multi-version schema

Tableau 6.2 – Factual data of "Environments" node

Years	Census Points	Agencies	Percent of Forest	Percent of Grassland
2002	1	LE2I	0,176	0,250
2002	1	ONEMA	0,356	0,261
2002	2	LE2I	0,311	0,420
2002	2	ONEMA	0,255	0,574
2011	1	LE2I	0,189	0,278
2011	1	ONEMA	0,241	0,385
2011	2	LE2I	0,322	0,568
2011	2	ONEMA	0,257	0,575

Tableau 6.3 – Factual data of "Abundances" node

Years	Census Points	Species	Abundance
2002	1	Bruant jaune	1,5
2002	1	Mésange noire	0,5
2002	2	Bruant jaune	1,5
2002	2	Mésange noire	0
2011	1	Bruant jaune	1
2011	1	Mésange noire	3
2011	2	Bruant jaune	1
2011	2	Mésange noire	2

by means of the functional dependencies. These users' needs are expressed in the form of facts existing in the constellation multidimensional model. In particular, the main idea is to provide an algorithm that transforms the constellation multidimensional schema by eliminating a fact node and integrating factual data in an associated dimension used for creating new levels.

To perform this algorithm, we translate the multidimensional model in a multidimensional graph.

In the following section we describe the multidimensional graph definitions (Section 6.2.4.1), the main algorithm is detailed in Section 6.2.4.2 and the calculation of new versioned hierarchies is explained in Section 6.2.4.3.

6.2.4.1 Preliminaries

In this subsection, we present some preliminary definitions. We represent a multidimensional model using a graph.

Definition 1. **Multidimensional Graph.**

A multidimensional graph is a directed graph $M_G =< D, F, A >$with :

$D = \{d_1, ..., d_m\}$, dimensional nodes, which represent dimensions.

$F = \{f_1, ..., f_n\}$, fact nodes representing facts.

$A = \{a_1, ..., a_p\} \mid \forall i \in [1,p], a_i = (f_j, d_k)$ with $j \in [1,n]$ and $k \in [1,m]$, are arcs [1], meaning that arcs are only directed from a fact node to a dimensional node. Moreover, M_G contains no alone node, isolated of another node, but can contain

1. In this paper, the notation (f_i, d_j) represents the arc from fact node f_i to dimensional node d_j.

possibly disconnected sets of nodes if each sub-graph must contain at least one fact node.

Example. An example of multidimensional graph is shown on Figure 6.3.. "Species" dimension, "Census points" dimension, "Years" dimension, "Abundances" fact and "Environments" fact are described in previous sections. "Sources" dimension represents agencies, which collect data. "Budget" fact represents the funds allowed by each agency for each year to collecting data.

In our approach decision-maker want to enrich a dimension with some new hierarchies using some factual data. That dimension is called Target dimension

Definition 2. **Target Dimension.**

The target dimension d_t of a multidimensional graph M_G is a dimension such as : $d_t \in D \mid \exists (f_1, d_t), ..., (f_u, d_t)\ avec u \in [2, n]$. This means that d_t is associated at least to two facts since one has to be removed and used to create its new levels.

Example. An example of possible target dimension is the "census point" dimension (Figure 6.3).

Let us now formalize the fact node that is used to create levels.

Definition 3. **Source Node.**

The source node of a M_G with a target dimension d_t is a fact node $f_s \mid \exists a \in A \mid a = (f_s, d_t)$.

Example. With "census point" dimension as target node, an example of possible source node is the fact node "Environments".

As we have said before our algorithm removes the source node from the graph. Therefore, a part of the structure of the graph is changed. Note that only nodes related to the source nodes are affected. We define this sub-graph in the following way :

Definition 4. **Source-target multidimensional sub-graph.**

Let M_G a multidimensional graph with a target dimension d_t and a source node f_s, then the Source-target multidimensional sub-graph M'_G is a multidimensional graph such as : $M'_G =< D', F', A' >$ with :

$F' = \{f_i \in F \mid \exists (f_i, d_t)\}$

$D' = \{d_i \in D \mid \exists (f_s, d_i)\}$

$A' = \{(f_i, d_j) \mid f_i \in F', d_j \in D'\}$

M'_G contains thereby only fact nodes linked to d_t and dimensional nodes linked

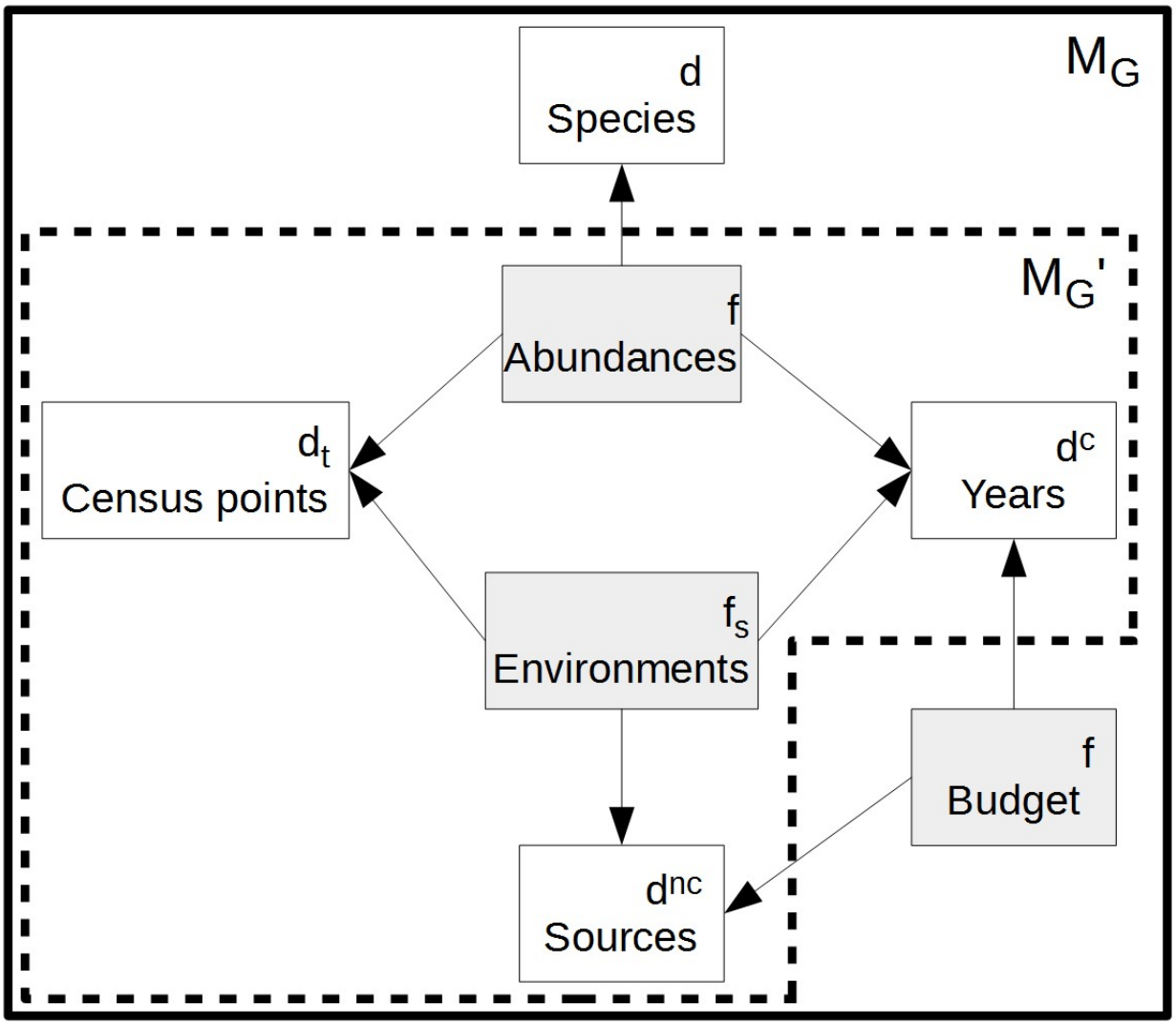


Figure 6.3 – Multidimensional graph M_G

Tableau 6.4 – Factual data of "Environments" node aggregated on "Agencies"

Years	Census Points	Percent of Forest	Percent of Grassland
2002	1	0,266	0,256
2002	2	0,283	0,497
2011	1	0,215	0,332
2011	2	0,290	0,572

to f_s. In M'_G, all fact nodes are so linked to at least one dimensional node and all dimensional nodes are so linked to at least one fact node. There is no isolated node in this sub-graph. M'_G is so a well-formed multidimensional graph.

Example. An example of Source-target multidimensional sub-graph using the previous example is shown on Figure figure 6.3.

In order to formalize inputs of the agglomerative hierarchical clustering algorithm used for the creation of levels of the target dimension, we formalize factual data aggregated to a set of dimensions levels using the definition of instance fact node.

Definition 5. **Fact Node Instance.**

Let M_G a multidimensional graph. Let m_i a member of the dimension d_i. Then the fact node instance $I(f, d_1.m_1, ..., d_n.m_n)$ is the set of tuples representing facts of f aggregated to the dimensions members $d_1.m_1$, ..., $d_n.m_n$.

Example. Let, Table 6.2 representing the instance fact node for the node "Environments", then Table 6.4 represents facts aggregated to the All member of the "Agencies" dimension :

$I(\text{“}Environments\text{”}, \text{“}Agencies.ALL\text{”}, \text{“}Years.1990\text{”}, \text{“}Censuspoints.*\text{”})$ [2].

6.2.4.2 Algorithm

In this section we provide details and formalize our approach.

Removing a fact node from the multidimensional graph implies its redefinition. Thus, the main idea is in a first step to work on the source-target multidimensional graph exclusively, transform this subgraph adding levels to the target dimension and removing the source node, and then finally re-integrate the new sub-graph in the rest of original multidimensional graph.

2. '*' means 'all members of the dimension'

Removing the source node implies to handle its associated dimensions. It is possible to distinguish three types of dimensions :

— The target dimension d_t that will rest in the transformed sub-graph,
— the Non Context dimensions D_{nc}, and
— the Context dimensions D_c.

The Non context dimensions D_{nc} are dimensions that are only associated to the source node fact. In order to remove one dimension it is possible to provide a classical Dice operator, which consists in aggregating fact data to the top dimension member. Let us note that in order to avoid summarazability problems (aggregation cannot be reused) (Lenz and Thalheim, 2009), in our approach we allow using only distributive and algebraic aggregation functions for the Dice operator.

Example. An example of Non contextual dimension is the "Agencies" node. In Table 6.4 is shown an example of the Dice operator on the "Agencies" dimension, which is a Non contextual dimension.

Formally,

Definition 6. **Non contextual dimension.**

Let Source-target multidimensional sub-graph $M'_G =< D', F', A' >$, then the set of non contextual dimension D_{nc} is

$$D_{nc} = \{d_1^{nc}, ..., d_v^{nc}\} \subset D' \mid \forall i \in [\![1, v]\!] \exists !(d_i^{nc}, f_j) \mid f_j \in F'$$

Note that in the previous formula, all dimensional nodes in D_{nc} are only linked to f_s. Indeed, all dimensional nodes in M'_G are linked to f_s and dimensional nodes in D_{nc} are linked to one (and only one) dimensional node.

The Context dimensions D_c are dimensions in M'_G that are associated to f_s and another fact node f. With the future refined graph, users analyze facts in f according to d_t. But, data used for calculating new hierarchies in d_t come from f_s and are thereby dependent of dimensions in D_c. Therefore, we need to ensure that data used to create the hierarchy are coherent with data consulted by the user during their OLAP analysis. With this in mind, we offer a system that calculates hierarchies according a context, this context defining with D_c. Formally,

Definition 7. **Contextual Dimension.**

Let Source-target multidimensional sub-graph M'_G, then the set of contextual dimension D_c is

$$D_c \subset D' \mid D_c = D' - (D_{nc} \cup \{d_t\})$$

with d_t, the target dimension.

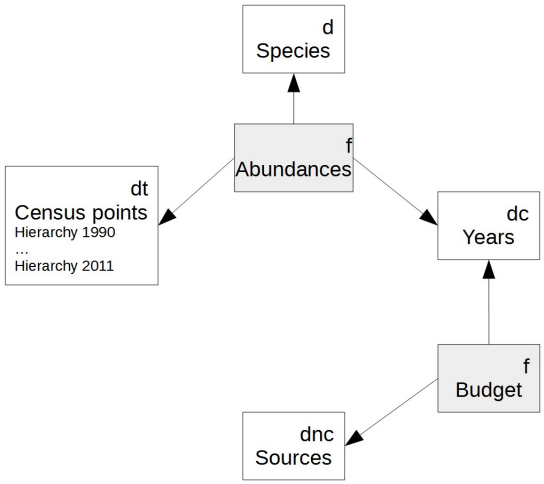


Figure 6.4 – Refined multidimensional graph M_G

Example. An example of contextual dimension is the "Years" node. On Table 6.3, we present data from "Abundances" node : data are dependent of "Years" dimensional node.

Once we have defined non context and contex dimensions let us provide our algorithm (see Algorithm 6.1) supposing that we have only one context dimension. The input of this algorithm is the multidimensional graph M_G presented on Figure 6.3.

Algorithme 6.1 Main algorithm

Inputs : M_G is a multidimensional sub-graph, d_t is a target dimension and f_s is a source node

$M'_G \leftarrow GetSubGraph(M_G, d_t, f_s)$;
$d_t \leftarrow BuildHierarchies(M'_G, d_t, f_s)$;
$M_G \leftarrow Delete(f_s)$;
$M_G \leftarrow Clean(M_G)$;
return M_G

The output of this algorithm is a multidimensional graph, presented on Figure 6.4. We note that f_s has been removed and there are new hierarchies in the "census points" node. Moreover, M_G remains a well-formed multidimensional graph and can be also implmented in a ROLAP architecture.

6.2.4.3 Automatic creation of hierarchies

In this section we describe how the is applied to create new levels of the target dimension.

A complete methodology to create new hierarchies in a multidimensional model with Hierarchical Agglomerative Clustering is presented in (Sautot et al., 2015). The main idea of this methodology is to build a new hierarchy into a dimension by using data, which describe items at the lowest level of the hierarchy. In our case, items are census points and description data are factual data. We suggest to use the Hierarchical Agglomerative Clustering, due to the similarity between the output of the Hierarchical Agglomerative Clustering and a hierarchy into an OLAP dimension (Messaoud et al., 2004).

Main steps of this algorithm are : (1) Calculation of distances between individuals ; (2) Choice of the two nearest individuals. (3) Aggregation of the two nearest individuals in a cluster. The cluster is considered an individual. (4) Go back to the step 1 and loop while there is more than one individual.

In our approach the clustering (AHC) takes as inputs the instance of the source node f_s evaluated on each member of the context dimension and dicing it non context dimensions.

Formally, the step 2 of our algorithm is the following :

Algorithme 6.2 Hierarchy builder algorithm

Inputs : M'_G is a Source-Target multidimensional sub-graph, d_t is a target dimension, f_s is a source node

$d_c \leftarrow GetContext(M'_G, d_t, f_s)$;
$d_{nc} \leftarrow GetNonContext(M'_G, d_t, f_s)$;
for each member m of d_c
  $I \leftarrow GetInstance(f_s, d_{nc}.ALL, d_c.m, d_t.*)$;
  $H \leftarrow CAH(I)$;
  $d_t \leftarrow SetNewHierarchy(d_t, H)$;
end
return d_t

An example is presented on Figure 6.5. We note that two hierarchies for the spatial dimension have been created for years 2002 and 2011.

Years	Census Points	Forest	Grassland
2002	1	17.55%	30.29%
2002	2	31.12%	47.41%
2002	3	41.29%	58.71%
2002	4	0.00%	0.00%
2002	5	15.31%	84.69%
2011	1	35.62%	62.16%
2011	2	22.27%	74.85%
2011	3	58.50%	13.72%
2011	4	47.88%	21.51%
2011	5	37.06%	37.96%

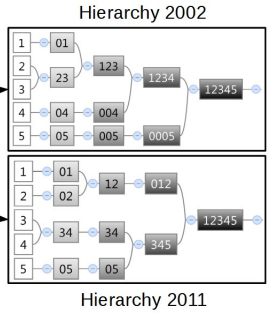


Figure 6.5 – Contextual hierarchies of census points

6.2.5 Validation and Experiments

In this section we present the implementation our proposal. A semantic and performance evaluations are detailed in Sections 6.2.5.1 and 6.2.5.2 respectively.

The refinement tool implements our algorithm using Matlab®. It allows defining graph using a simple visual interface as shown on Figure 6.6. The considered multidimensional graph is presented on the top part of the visual interface. On the bottom one, the algorithm ask inputs to users in a command window.

6.2.5.1 Semantic Evaluation

In this section, we describe the added-value of our methodology from a design point of view (i.e. does the refinement methodology corresponds to decisionmakers needs ?). For that goal two we have investigated two aspects : 1) Do dimensions and facts created using our methodology correspond to decisionmakers analysis needs ? ; 2) Do hierarchies created using our methodology improve analysis capabilities ?

Therefore have decided to compare the result of our methodology with with one proposed in(Miquel et al., 2002a). Indeed, Miquel *et al.* propose a manually method to obtain a multi-version multidimensional schema,and when the time dimension is chosen as the context dimension our approach results a multi-version multidimensional schema. The result of this validation shows that the multidimensional schema produced with the manual methodology and our automatic methodology are equal.

Moreover, in order to validate the semantic correctness of using AHC for hierarchies definition, we have asked to ecologists of the project to choice between a spatial dimension with only one level, and a spatial dimension with a hierarchy created using AHC. When the number of created levels is not superior to 5,

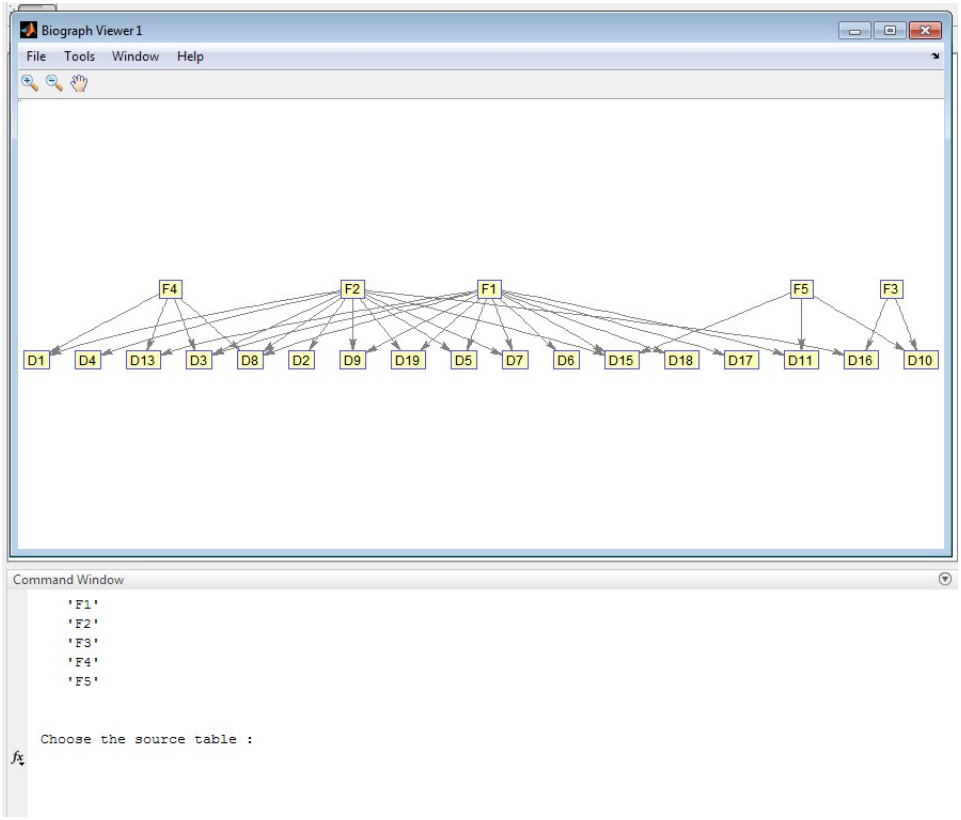


Figure 6.6 – Visual interface of the refinement tool

decision-makers prefer having hierarchies, since they can reveal interesting pattern such as agricultural profiles of census points. For example, data in the "Environments" fact table contains data that describe agriculture policies around each census point at each year. The data clustering according to these data can classify census points and allows decisionmakers analyzing impact of agricultural practices on bird biodiversity. For example, decision-makers can analyze biodiversity according to agricultural forest and grassland parameters of census points, by using this simple OLAP query : "*What is the biodiversity value per group of census points (first level of the hierarchy obtained with clustering) in 2002 and 2003 ?*". This query can reveal that for the same year, for example 2002, biodiversity is very affected by agricultural parameters since the aggregated biodiversity value for each group of census point is different.

6.2.5.2 Performance Evaluation

In this section, we test time performance of our methodology in order to validate its feasibility from a project deployment process point of view.

In particular we study time performance related to : 1) refinement algorithm for facts and dimension design, and 2) hierarchy creation using AHC. In order to test the first point, we have created a set of 200 simulated constellation schema using from 2 to 100 dimensions, since real usable multidimensional schema presents maximum between 3 and 10 dimensions (Kimball, 1996). Finally, the worst time execution is 15.23 s. The average execution time is equal to 11.7 s with a standard deviation equal to 1.17 s. These performances are satisfactory for are good for an off-line design phase. In this paragraph, we study time performances of the AHC algorithm.

In this paragraph, "classified items" are census points (which are members of the "census points" dimension, the target dimension) and "attributes" are aggregated facts from the "Environments" fact node (which is the source fact node). The AHC algorithm has been also implemented in Matlab and its performance has been also tested. Using our case study data, we perform 2090 tests, with a number of classified items (source node instances- Environments facts) between 10 and 190, and a number of attributes (source node attributes-Environments fact measures) between 10 and 100, and the average calculation time is equal to 0.072 s, with a standard deviation equal to 0.002 s. To complete our evaluation, we simulate a data set with 10,000 classified items and 150 attributes. In this case, the AHC calculates a hierarchy in 147.36 s, with a standard deviation equal to 4.03 , with a maximal calculation time equal to 214 s. All time performances are shown on Figure 6.7. This calculation time (approximately four minutes) is efficient for an

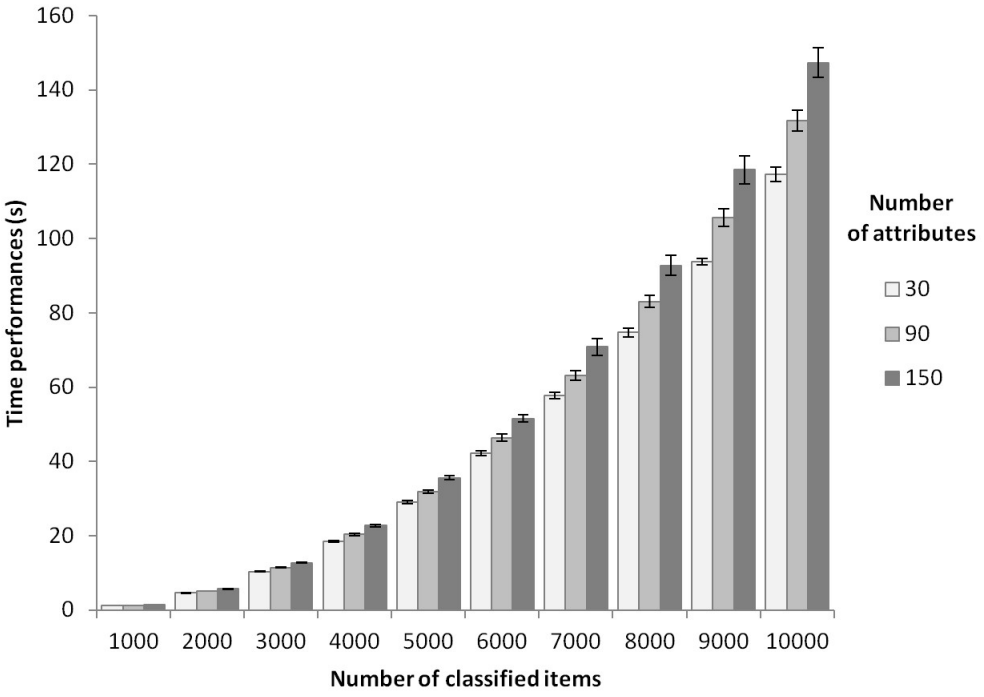


Figure 6.7 – Execution times according the number of attributes and classified items

off-line design phase.

6.2.6 Conclusion and Future Work

Design data warehouses system is a complex and crucial task depending on available data sources and decisional requirements. Existing work do not exploit the semantics of data to automatically create complex hierarchies. Thus in this paper, we present a mixed multidimensional refinement methodology, that transform constellation schema to define hierarchy level using a hierarchical clustering algorithm. Our refinement methodology enriches a dimension with factual data, and considers the context of factual data. We present also the implementation of our method in a ROLAP architecture.

We perform the proposed methodology on a real application case from bird biodiversity. We have noted that actual automatic multidimensional design methodologies cannot produce a multidimensional schema, which covers all decision-maker needs due to the data complexity. Our methodology offers a solution to enrich dimensions with factual data and, by this way, to refine the multidimensional

schema.

Our ongoing work is the extension of our methodology to simplify and reduce the number of created levels, using other DM algorithms such as SVM, etc., in order to provide decision-makers with easy OLAP exploration analysis and its implementation in a ROLAP architecture.

Moreover, we are also working to integrate our approach in the rapid prototyping methodology proposed in (Sautot et al., 2014), and extending to help decision-makers and DW experts choose the right DM algorithms and parameters of the refinement algorithm (source node, contextual dimensions, etc.). Future work concerns the usage of the formal evaluation framework Goal Question Metric (Briand et al., 2002) to evaluate our methodology.

Acknowledgments

Data acquisition received financial support from the FEDER Loire, Etablissement Public Loire, DREAL de Bassin Centre, the Région Bourgogne (PARI, Projet Agrale 5) and the French Ministry of Agriculture. We also thank heartily Pr. John Aldo Lee, from the Catholic University of Leuven, for his help.

Troisième partie

Conclusions

Chapitre 7

Bilan général

Dans ce chapitre, nous présenterons le bilan général de nos contributions, la conclusion des travaux de recherche menés durant cette thèse.

A l'origine de ces travaux, nous avons constaté que les systèmes OLAP présentent un intérêt pour la gestion, le stockage et l'analyse de données écologiques, mais que la complexité de ces systèmes les rend difficiles à mettre en place dans ce contexte particulier. C'est pourquoi nous nous sommes intéressés aux méthodes de conception automatique d'entrepôts de données. Par ailleurs, nous avons constaté que les données produites par les sciences du vivant peuvent être très complexes. Ces données peuvent provenir de sources hétérogènes, être nombreuses, voir être inconsistantes. La littérature scientifique propose différentes méthodes pour gérer ces difficultés.

Mais, on peut constater que les méthodes de conception automatique ne sont pas prévues pour des données très complexes et que les méthodes qui permettraient de gérer ce type de données ne sont pas automatiques. Nous avons donc proposé un ensemble de méthodes et d'outils permettant d'automatiser la conception d'un entrepôt de données à partir de données complexes. Comme support de nos travaux de recherche, nous avons eu à disposition un cas d'étude concernant la biodiversité des oiseaux. Ce jeu de données est constitué du recensement de 213 espèces d'oiseaux en 198 points spatiaux le long de la Loire. Ce recensement a été effectué quatre fois en 21 ans : en 1990, 1996, 2002 et 2011. De plus, les points spatiaux ont été décrits par 158 variables, issues de différentes sources, et permettant de décrire la vallée fluviale à différentes échelles.

Durant ces travaux, nous nous sommes essentiellement intéressés à la conception de hiérarchies. En effet, l'identification des faits et des dimensions est relativement aisé pour les utilisateurs. En revanche la structuration des dimensions, et dans

notre cas notamment de la dimension spatiale, peut s'avérer beaucoup plus problématique.

En effet, nous avons constaté les caractéristiques suivantes au sein des données environnementales décrivant les points d'écoute :

- Ces données sont hétérogènes : les données décrivant chaque points peuvent être qualitatives ou quantitatives.
- Ces données peuvent être manquantes : il arrive que, ponctuellement, un attribut n'est pas été renseigné pour quelques points.
- Ces données sont nombreuses : en tout, 158 attributs environnementaux décrivent chaque point.
- Ces données sont inconsistantes : les sources de données étant hétérogènes, et certaines étant indépendantes du programme de recensement des oiseaux, certains attributs ne sont disponibles que pour une seule campagne de recensement. Ainsi, la structure de la description de la vallée fluviale est fortement dépendante du temps.

Pour ***concevoir automatiquement un modèle conceptuel multidimensionnel et l'implémenter automatiquement pour des données écologiques complexes***, nous avons travaillé selon trois angles différents :

1. Tout d'abord, nous nous sommes intéressés à la nature des données. Notre premier objectif a été de proposer une méthode permettant de construire automatiquement une hiérarchie avec des membres décrits par des attributs quantitatifs, des attributs qualitatifs et sachant que certaines données peuvent être manquantes.
2. Ensuite, nous nous sommes intéressés à la prise en compte des besoins analytiques et des connaissances des utilisateurs concernant les données. Notre second objectif a été de proposer une méthode prenant en compte les spécifications des utilisateurs pour la construction automatique de hiérarchies grâce au prototypage.
3. Pour finir, nous nous sommes intéressés à la complexité des analyses menées par les utilisateurs, qui impliquent de construire des requêtes avec des données interdépendantes. Notre troisième objectif sera de proposer une méthode capable de construire automatiquement une hiérarchie en prenant en compte la source des données utilisées pour construire cette hiérarchie et son contexte.

Pour répondre à ces objectifs, nous avons proposé différentes méthodes et mis en place plusieurs outils.

Pour répondre au premier objectif, nous avons proposé une méthode qui consiste à construire de nouvelles hiérarchies grâce à la Classification Ascendante Hiérarchique. En effet, cet algorithme produit des arbres binaires dont la structure est

compatible avec une hiérarchie au sein d'une dimension OLAP. Cependant, comme nous disposons de données mixtes et qui peuvent être manquantes, nous proposons d'utiliser une métrique particulière, issue de l'écologie : l'indice de similarité de Gower. Cette méthode permet donc de gérer la nature de nos données. Mais elle insuffisante : cette méthode ne permet pas de concevoir l'ensemble de l'entrepôt de données, ni de gérer les inconsistances temporelles de nos données. Elle constitue cependant une première étape, et sera utilisée dans les autres parties de la thèse.

Pour répondre au second objectif, nous avons proposé d'utiliser une méthode de prototypage. Ce type de méthode présente l'avantage d'offrir aux utilisateurs la possibilité de spécifier leurs besoins analytiques sans l'intermédiaire d'un schéma conceptuel dont leur formalisme peut leur être étranger. Nous avons proposé une extension d'une méthode de prototypage existante afin d'y intégrer la construction automatique de hiérarchie, grâce à des algorithmes de fouille de données. En effet, la définition manuelle des hiérarchies, même via une méthode de prototypage, peut être une tâche difficile quand les membres d'une dimension sont décrits par 158 attributs. Afin de prendre en compte au mieux les besoins des futurs utilisateurs, nous avons intégré à cette extension la possibilité de choisir de construire des hiérarchies selon trois méthodes différentes : (i) la construction manuelle, classique, présente dans la méthode de prototypage originale ; (ii) le clustering, qui permet de répartir les membres de la future dimension en groupes homogènes ; (iii) la classification supervisée, qui apprend à reconnaître plusieurs types de membres dimensionnels selon les spécifications des utilisateurs. Le clustering utilisé est une Classification Ascendante Hiérarchique basée sur l'indice de Gower, décrite dans le paragraphe précédent. La classification supervisée en revanche, est une Machine à Vecteurs Supports. L'extension que nous avons proposée intègre à la méthodologie de prototypage originale deux algorithmes de fouille de données, ainsi qu'un protocole de choix entre les différentes méthodes de construction de hiérarchie. Cette extension propose également de compléter le méta-modèle multidimensionnel utilisé pour y intégrer les différents paramètres des algorithmes de fouille.

Enfin, pour répondre au troisième objectif, nous avons proposé une méthode permettant d'intégrer les données factuelles d'un cube OLAP à une dimension partagée par plusieurs autres cubes. Ces données factuelles sont intégrées sous forme de hiérarchies grâce à une Classification Ascendante Hiérarchique. Notre méthode prend en compte les dépendances entre les données factuelles intégrées à la dimension cible et les faits des autres cubes partageant la dimension cible. En effet, les nouvelles hiérarchies intégrées dans la dimension cible vont être utilisée pour construire des requêtes sur d'autres faits, qui peuvent partager des dimensions avec le fait source. Notre méthode prend donc en compte le contexte du fait source et de la dimension cible, et construit des hiérarchies contextuelles, qui sont disponibles en fonction de la session de requêtage en cours. Cette méthode permet

notamment de gérer l'inconsistance temporelle de nos données environnementales, en permettant de construire des hiérarchies contextuelles avec les années qui jouent le rôle d'instances de contexte.

Les différentes méthodes présentées ont été implémentées au sein de deux outils :

— Un système qui construit des hiérarchies à partir des attributs des membres dimensionnels, et qui a été intégré à l'outil de prototypage rapide d'entrepôts de données.

— Un système qui construit des hiérarchies contextuelles à partir de données factuelles.

Ainsi, les apports les plus marquants de cette thèse sont la construction automatique de hiérarchies sur données mixtes, l'intégration d'un module de construction automatique de hiérarchies basé sur la fouille de données dans une méthode de prototypage d'entrepôt de données et enfin la construction automatique de hiérarchies à partir de données factuelles. Ce dernier apport nous a permis d'étendre le concept de versionning : les hiérarchies construites à partir de données factuelles sont dépendantes de contextes qui n'incluent pas uniquement des méta-données temporelles mais une requête OLAP complète.

Ces apports présentent cependant plusieurs limites. Tout d'abord, il convient de rappeler nos hypothèses de départ. Nous avions en effet postulé que, dans le cadre d'une étude en écologie, le caractère factuel ou dimensionnel des données, était facile à identifier par l'utilisateur. Cependant, comme nous l'avons noté dans le Chapitre 6, les données environnementales, bien que clairement identifiées comme attributs dimensionnels par les besoins analytiques associés à notre cas d'étude, peuvent être identifiées comme des faits par certaines méthodes (Romero and Abelló, 2010). Ainsi, nous avons automatisé essentiellement la construction de hiérarchies, et bien que l'utilisation d'une méthode prototypage permette de construire un entrepôt complet, l'identification automatique de faits et de dimensions n'est pas proposée par nos contributions.

Outre cette limite générale, issue de notre hypothèse de départ, une limite intrinsèque des contributions que nous avons proposées est l'utilisation d'un algorithme de clustering. En effet, les algorithmes de clustering regroupent des objets en fonction des attributs qui décrivent ces objets en se basant sur des critères mathématiques. Ainsi, la structure de la hiérarchie construite par un algorithme de clustering peut changer selon la métrique choisie pour calculer les distances entre objets. Le choix de la métrique et des attributs utilisés pour construire les hiérarchies sont laissés à la discrétion de l'utilisateur, et on doit donc questionner la signification analytique des regroupements.

Une autre piste à explorer lors de travaux futurs peut se centrer sur la sélection de

variables. En effet, avec 158 paramètres susceptibles de décrire les points d'écoute, nous disposons d'un jeu de données conséquent. Or, parmi ces variables, nous ne sommes pas tous que toutes soient pertinentes pour analyse les changements spatio-temporels qui apparaissent au sein des peuplements d'oiseaux. Il pourrait donc être intéressant d'intégrer une méthode de sélection de paramètres parmi les algorithmes de fouille de données que nous avons proposé d'intégrer à une méthode de conception d'entrepôt de données.

Pour finir sur les limites, il faut interroger la méthode de prototypage que nous avons utilisée. Essentiellement basée sur des boucles, on peut penser que, dans des cas d'entrepôts très complexes et/ou impliquant de nombreux acteurs, d'une part le processus de conception du prototype soit très long et implique de réaliser de nombreux cycles, et que d'autre part, certains cycles conduisent à des régressions fonctionnelles du future prototype.

En conclusion, pour concevoir et implémenter automatiquement un entrepôt de données, à partir de données complexes, nous avons proposé d'utiliser une méthode de prototypage rapide pour identifier les faits et les dimensions. La structuration des dimensions en hiérarchies peut se faire automatiquement, même dans le cas de données mixtes et/ou manquantes, en utilisant des algorithmes de fouille de données avec les métriques adaptées. Ces algorithmes peuvent être supervisés ou non selon les besoins des utilisateurs. Pour finir, il est possible, pour construire ces hiérarchies, d'utiliser des données internes à la dimension (attributs des membres de la dimension) ou bien des données factuelles issues d'un cube utilisant la dimension. On doit alors prendre en compte le contexte multidimensionnel du fait source et de la dimension cible. Ce dernier point permet notamment de gérer les inconsistances éventuelles au sein du jeu de données.

Chapitre 8

Perspectives

8.1 Cycle de vie

Une première perspective de recherche que nous avons identifiée est la prise en compte du cycle de vie de l'entrepôt de données. En effet, notre cas d'étude nous a permis de nous placer dans un contexte confortable : nous nous sommes placés lors d'une phase de conception de l'entrepôt de données, mais en disposant des données nécessaires pour construire des hiérarchies via un algorithme de fouille de données. Or, les entrepôts de données, et plus généralement les systèmes OLAP, doivent être conçus pour fonctionner sur un temps long. On peut citer, concernant les travaux portant sur l'évolution de schéma d'entrepôt de données, l'état de l'art proposé par Arora et Gosain en 2011 (Arora and Gosain, 2011).

La question de l'évolution des spécifications se pose donc concernant les méthodes de conception de hiérarchies que nous avons proposées. Ainsi, il serait intéressant de savoir quelle est la durée de vie du paramétrage des algorithmes de construction automatique de hiérarchies, qu'ils soient supervisés ou non, quand l'entrepôt de données est mis en place et qu'il est utilisé. On peut en effet penser que, lorsque de nouvelles données sont intégrées pendant une longue durée, les paramètres choisis pour les algorithmes de fouille au moment de la conception ne soient plus pertinents. Par exemple, dans le cas d'une classification supervisée, on peut imaginer qu'au fil du temps, de nouveau type de membres soient apparus et qu'un nouvel apprentissage, avec un nouvel échantillon d'apprentissage, soit nécessaire. D'autant plus que la question de l'évolution des algorithmes de fouille de données est rarement posée et peu automatisée (Ganti et al., 2001).

Une étude approfondie du comportement des algorithmes de fouille à long terme est donc nécessaire pour valider complètement les propositions formulées dans cette

thèse. Pour mener cette étude, on peut imaginer de développer un tableau de bord spécifiquement conçu pour suivre les performances des algorithmes de fouille de données. Les mesures de performances des algorithmes de fouille peuvent être assez simples, comme celles définies dans la Section 5.2.7.1 (page 121), à savoir un taux de bonne classification pour une classification supervisée et un ratio distance inter-cluster sur distance inter-cluster pour un clustering. Ainsi, on pourra suivre, au fil des intégrations de données, l'évolution des performances des algorithmes de fouille. On pourra ainsi comparer le cycle de vie du paramétrage des algorithmes de fouille et le cycle de vie de l'entrepôt de données, ainsi faire des recommandations de mise à jour du paramétrage des différentes algorithmes utilisables pour construire automatiquement des hiérarchies.

Toujours concernant le cycle de vie de l'entrepôt de données, on peut également s'interroger sur la mise à jour des algorithmes de fouille quand les spécifications des utilisateurs évoluent. Dans les paragraphes précédents, nous avons posé la question du comportement des algorithmes de fouille lors de l'utilisation d'entrepôt de données et de l'intégration d'un volume de données important. Mais on peut aussi poser le problème d'une éventuelle évolution du schéma, due à une évolution des spécifications utilisateurs. Il s'agit alors de faire également évoluer les algorithmes de fouille de données.

8.2 Entrepôts de domaine

Comme nous l'avons vu dans le chapitre précédent, les contributions que nous avons proposées permettent d'automatiser la construction de hiérarchies, mais l'identification des faits et des dimensions est manuelle.

Les méthodes actuelles de construction d'entrepôt de données qui sont capables d'identifier les dimensions et les faits déduisent le schéma multidimensionnel à partir d'informations structurées, comme par exemple du schéma d'une ontologie (Thenmozhi and Vivekanandan, 2013) ou d'une base de données relationnelle et de requêtes SQL associées à cette base (Romero and Abello, 2010). Or, dans notre cas, les données étaient présentées sous forme de différents fichiers plats (issus d'un tableur ou d'un SIG).

De plus, les faits et les dimensions peuvent être interprétés comme des variables1 respectivement à expliquer et explicatives, des concepts connus et maîtrisés par les utilisateurs dans notre cas d'étude. Il paraît donc difficile et peu productif d'essayer de déduire complètement l'identification des faits et des dimensions à partir des données sources.

Cependant, de même qu'il existe des ontologies de domaine, notamment pour

l'écologie (Madin et al., 2007; Pundt and Bishr, 2002), on pourrait imaginer proposer des "entrepôts de domaine" ou des "cubes de domaine". Il s'agirait alors de proposer un schéma multidimensionnel typique d'un champ disciplinaire, qui puisse servir de base à la mise en place d'un système OLAP spécifique de ce champ disciplinaire. Ces schémas multidimensionnels typiques, mis à disposition de la communauté scientifique, pourraient un gain de temps considérable dans le développement de systèmes OLAP pour l'écologie et les sciences du vivant en général.

Il faut cependant noter que utilisabilité des ontologies de domaine est parfois remise en question (Soldatova and King, 2005), et c'est un point qui devra être largement pris en compte pour développer un "entrepôt de domaine" pour l'écologie.

Quatrième partie

Annexes

Chapitre 9

Annexe : Les 12 règles de Codd

Les systèmes OLAP doivent, selon leur inventeur, respecter 12 règles (Codd et al., 1993) :

1. *Vue conceptuelle multidimensionnelle des données* : Les outils OLAP permettent aux utilisateurs d'avoir une vue multidimensionnelle des données, et de manipuler facilement les données.
2. *Transparence* : Les données manipulées par un système OLAP peuvent provenir de sources variées et hétérogènes (fichiers plats, bases de données diverses, ...), mais cette hétérogénéité est transparente pour l'utilisateur. Il ne voit pas la source des données, qui constituent pour lui un ensemble homogène.
3. *Accessibilité* : Les outils OLAP doivent être capables d'accéder à des données issues de sources hétérogènes et d'opérer les transformations nécessaires sur les données pour présenter une vue cohérente à l'utilisateur. C'est l'outil (et non l'utilisateur) qui se charge de savoir où sont physiquement stockées les données.
4. *Constance des performances* : Les performances d'un outils OLAP ne souffrent d'une augmentation du nombres de dimensions d'analyse. Le temps de réponse est fonction uniquement de la taille des réponses retournées et non de la taille de la base de données.
5. *Architecture Client-Serveur* : La composante serveur d'un outil OLAP permet d'intégrer facilement différents clients.
6. *Indépendance des dimensions* : Au sein d'un système OLAP, chaque dimension doit être équivalente aux autres en termes de structure et de capacités opérationnelles.
7. *Gestion dynamiques des matrices creuses* : Un serveur OLAP doit avoir une structure physique permettant une manipulation optimal des matrices

creuses.

8. *Accès multi-utilisateurs* : Un outil OLAP doit permettre un accès concurrent aux données, tout en garantissant l'intégrité et la sécurité.
9. *Opérations de croisements inter et intra-dimensions illimitées* : Le croisement des données doit être possibles selon n'importe quelles dimensions, sans limitation du nombre de dimensions impliquées dans le calcul. L'agrégation des données doit donc être définies pour toutes les dimensions.
10. *Manipulation intuitive des données* : La manipulation des données, notamment les calculs sur les dimensions, doivent être possibles directement sur les cellules d'une feuille de calcul, sans passer par des menus ou des opérations multiples dans l'interface proposée à l'utilisateur.
11. *Reporting flexible* : Les modules qui permettent de faire du reporting (création de rapports) doivent présenter les informations comme l'utilisateur souhaite les voir.
12. *Dimensions et niveaux d'agrégation illimités* : Le nombre de dimensions supporté doit être illimités. Chaque dimension peut avoir un nombre illimité de niveaux d'agrégation définis par l'utilisateur.

Chapitre 10

Annexe : Variables environnementales

10.1 Liste des variables issues des relevés de terrain et des études cartographiques du programme STORI

Tableau 10.1 – Variables relevées sur Google Earth

Variable	**Type**	**Indépendance au temps**	**Disponible en**	
			1990	**2012**
Latitude (degré décimaux)	Localisation	Oui	-	
Longitude (degré décimaux)	Localisation	Oui	-	
Largeur de la rivière (m)	Quantitative	Non	x	x
Largeur de la ripisylve (m)	Quantitative	Non		x
Largeur de la bande de divagation (m)	Quantitative	Non		x
Distance au massif forestier le plus proche (m)	Quantitative	Non		x
Distance au hameau, village ou ville le plus proche (m)	Quantitative	Non	x	x
Largeur de la bande active (m)	Quantitative	Non		x

Tableau 10.2 – Variables relevées sur une carte IGN 1/25000

Variable	**Type**	**Indépendance au temps**	**Année de collecte**
Distance à la source (km)	Quantitative	Oui	1990
Altitude (m)	Quantitative	Oui	1990
Pente de la rivière (m/km)	Quantitative	Oui	2012
Confluence	Qualitative	Oui	2012
Largeur de la vallée (m)	Quantitative	Oui	1990

Tableau 10.3 – Variables relevées sur le terrain

Variable	**Type**	**Indépendance au temps**	**Disponible en**		
			1990	**2011**	**2012**
Etendues des grèves	Qualitative	Non	x		x
Dynamique aquatique	Qualitative	Non	x		
Présence de cascade	Qualitative	Non	x		
Présence d'îles	Qualitative	Non	x		
Pente de la berge	Qualitative	Non	x		
Escarpement abrupt	Qualitative	Non	x		
Ourlets végétaux	Qualitative	Non	x		
Végétation aquatique	Qualitative	Non	x		
Physionomie dominante	Qualitative	Non	x		
Vitesse du Courant	Qualitative	Non	x		x
Substrat dominant	Qualitative	Non	x		x
Etendue de la végétation aquatique	Qualitative	Non	x		x
Salinité (g/l)	Quantitative	Non	x		x
Hauteur maximale de la ripisylve (m)	Quantitative	Non			x
Hauteur moyenne de la ripisylve (m)	Quantitative	Non			x
Largeur de la ripisylve (m)	Quantitative	Non			x
Nombre de strates de la ripisylves	Qualitative	Non			x
Fragmentation de la ripisylve	Qualitative	Non		x	
Etendue des cultures (%)	Quantitative	Non	x	x	
Etendue des forêts pures (%)	Quantitative	Non	x	x	
Etendue des buissons, taillis, landes (%)	Quantitative	Non	x	x	
Etendue des prairies (%)	Quantitative	Non	x	x	
Etendue des milieux aquatiques annexes (%)	Quantitative	Non	x	x	
Etendue des milieux bâtis (%)	Quantitative	Non	x	x	
Etendue des milieux rocheux (%)	Quantitative	Non	x	x	
Nombre de milieux	Quantitative	Non		x	
Nombre d'éléments linéaires	Quantitative	Non		x	

Tableau 10.4 – Autres variables

Variable	Type	Indépendance au temps	Disponible en		
			1990	2002	2012
Indice de tressage	Quantitative	Non	x		x
Largeur du val inondable (m) (*issue de l'Atlas des ZI et Cartorisque*)	Quantitative	Non		x	
Indice d'urbanisation	Quantitative	Non			x

10.2 Liste des variables issues des images satellites

Toutes ces données sont disponibles pour l'année 2001 et ont été obtenue à partir du logiciel FRAGSTAT

Indice d'agrégation

Moyenne des aires des patches

Aire occupée par les cultures basses

Aire occupée par l'eau et les zones humides

Aire occupée par les forêts hautes

Aire occupée par les forêts taille basse et les friches

Aire occupée par les forêts taille moyenne

Aire occupée par les cultures sans végétation

Aire occupée par les grèves caillouteuses

Aire occupée par les sols nus artificiels

Moyenne des aires des cercles circonscrits dans les patches

Mesure de la connexion physique du type de patch correspondant

Indice de connectance : nombre de jointures fonctionnelles entre tous les patches de même type divisé par le nombre total de jointures possibles entre les patches de même type (%)

Mesure de contagion (nombre de patches de même type à côté d'un patch par rapport au nombre de voisins possibles)

Mesure moyenne de contiguïté

Probabilité que deux pixels choisis de façon aléatoire dans le paysage ne soient pas situés dans le même type de patch

Densité des contours (m par ha)

Densité des contours des zones en cultures et en prairies (m par ha)

Densité des contours des zones en eau et zones humides (m par ha)

Densité des contours des zones en forêt haute (m par ha)

Densité des contours des zones en forêt basse et en friche (m par ha)

Densité des contours des zones en forêt moyenne (m par ha)

Densité des contours des zones en culture sans végétation (m par ha)

Densité des contours des zones en grèves caillouteuses (m par ha)

Densité des contours des zones en sols nus artificiels (m par ha)

Moyenne des distances euclidiennes au plus proche voisin (mesure de l'isolation des patches)

Indice de complexité du contour

Moyenne de la distance (m) entre chaque patch et le barycentre des patches

Entremêlement observé par rapport à l'entremêlement possible pour le nombre de types de raccordement

Pourcentage du paysage occupé par la plus large zone

Pourcentage du paysage occupé par la plus large zone en culture et en prairie

Pourcentage du paysage occupé par la plus large zone en eau et zones humides

Pourcentage du paysage occupé par la plus large zone en forêt haute

Pourcentage du paysage occupé par la plus large zone en forêt basse et en friche

Pourcentage du paysage occupé par la plus large zone en forêt moyenne

Pourcentage du paysage occupé par la plus large zone en culture sans végétation

Pourcentage du paysage occupé par la plus large zone en grèves caillouteuses

Pourcentage du paysage occupé par la plus large zone en sol nu artificiel

Densité standardisée des contours

Densité standardisée des contours des zones en cultures et en prairies

Densité standardisée des contours des zones en eau et zones humides

Densité standardisée des contours des zones en forêt haute

Densité standardisée des contours des zones en forêt basse et en friche

Densité standardisée des contours des zones en forêt moyenne

Densité standardisée des contours des zones en culture sans végétation

Densité standardisée des contours des zones en grèves caillouteuses

Densité standardisée des contours des zones en sols nus artificiels

Mesure de la taille des pastilles lorsque le type de patch correspondant est subdivisé en S patches, où S est la valeur de l'indice de fractionnement

Indice de diversité de Simpson modifié appliqué au paysage

Indice d'uniformité de Simpson modifié

Nombre de zones distinctes (nombre de patches)

Nombre de zones en culture ou en prairie

Nombre de zones en eau et en zones humides

Nombre de zones en forêt haute

Nombre de zones en forêt basse ou en friche

Nombre de zones en forêt moyenne

Nombre de zones en culture sans végétation

Nombre de zones en grève caillouteuse

Nombre de zones en sol nu artificiel

Mesure de la complexité des formes des patches

Moyenne du rapport entre le périmètre et l'aire des patches

Densité de zones distinctes (densité de patches) (nb pour 100 ha)

Densité des zones en culture et en prairie (nb de zones pour 100 ha)

Densité des zones en eau et en zone humide (nb de zones pour 100 ha)

Densité des zones en forêt haute (nb de zones pour 100 ha)

Densité des zones en forêt basse et en friche (nb de zones pour 100 ha)

Densité des zones en forêt moyenne (nb de zones pour 100 ha)

Densité des zones en culture sans végétation (nb de zones pour 100 ha)

Densité des zones en grèves caillouteuses (nb de zones pour 100 ha)

Densité des zones en sols nus artificiels (nb de zones pour 100 ha)

Proportions de cellules adjacentes impliquant la même classe

Pourcentage du paysage en culture et en prairie

Pourcentage du paysage en eau et zones humides

Pourcentage du paysage en forêt haute

Pourcentage du paysage en forêt taille basse et en friche

Pourcentage du paysage en forêt taille moyenne

Pourcentage du paysage en culture sans végétation

Pourcentage du paysage en grèves caillouteuses

Pourcentage du paysage en sol nu artificiel

Richesse des types de patch (nombre de types de patch présents)

Richesse standardisée des types de patch (nombre de types de patch présents standardisé)

Moyenne du rapport ajusté entre le périmètre et l'aire des patches

Indice de diversité de Shannon appliqué au paysage

Indice d'uniformité de Shannon

Indice de diversité de Simpson appliqué au paysage

Indice d'uniformité de Simpson

Répartition : le nombre de taches avec une taille constante lorsque le correctif du type de patch correspondant est subdivisé en parcelles S, où S est la valeur de la Indice de fractionnement

Aire totale de l'image analysée (ha)

Somme des contours

Somme des contours des zones en cultures et en prairies

Somme des contours des zones en eau et zones humides

Somme des contours des zones en forêt haute

Somme des contours des zones en forêt basse et en friche

Somme des contours des zones en forêt moyenne

Somme des contours des zones en culture sans végétation

Somme des contours des zones en grèves caillouteuses

Somme des contours des zones en sols nus artificiels

10.3 Liste des variables issues de l'outil MAGDALENA

Nom et emplacement des principales villes

Différentes classes de villes selon leurs tailles

Emprise des réseaux urbains

Pourcentage d'urbanisation dans les bassins versants

Type de massif géologique

Type de sol

Grandes régions écologiques

Zones de protection spéciale des oiseaux (Natura 2000 ZPS)

Zone d'intêret pour la conservation des oiseaux (ZICO)

Zone naturelle d'intérêt écologique, faunistique et floristique, type 1 (ZNIEFF type 1)

Zone naturelle d'intérêt écologique, faunistique et floristique, type 2 (ZNIEFF type 2)

Artificialisation en lit majeur (2000)

Pression agricole en lit majeur (2000)

Type de cultures pratiquées dans la région d'après le recensement agricole (2000)

Type d'élevage pratiqué dans la région d'après le recensement agricole (2000)

Etat écologique des principaux cours d'eau (2009)

Parcelles agricoles (2009)

Bâtiments (2009)

Bibliographie

Abdelhedi, F., Pujolle, G., Teste, O., and Zurfluh, G. (2011). Computer-aided data-mart design. In *13th International Conference on Entreprise Information Systems (ICEIS 2011)*. *4 citations pages 26, 55, 97 et 100*

Abelló, A., Samos, J., and Saltor, F. (2006). Yam2 : a multidimensional conceptual model extending uml. *Information Systems*, 31 :541–567. *5 citations pages 21, 93, 96, 99 et 100*

Adami, G., Avesani, P., and Sona, D. (2003). Bootstrapping for hierarchical document classification. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 295–302. ACM. *Cité page 102*

Alkharouf, N. W., Jamison, D. C., and Matthews, B. F. (2005). Online analytical processing (olap) : A fast and effective datamining tool for gene expression databases. *Journal of Biomedicine and Biotechnology*, 2 :181–188. *Cité page 13*

Arora, M. and Gosain, A. (2011). Schema evolution for data warehouse : A survey. *International Journal of Computer Applications*, 22(6) :6–14. *Cité page 163*

Bache, K. and Lichman, M. (2013). UCI machine learning repository. *Cité page 72*

Barbault, R. (1995). *Écologie générale : structure et fonctionnement de la biosphère*. Masson, Paris. *Cité page 35*

Bentayeb, F. (2008). K-means based approach for olap dimension updates. In *10th International Conference on Enterprise Information Systems (ICEIS)*, pages 531–534. *5 citations pages 34, 56, 98, 100 et 138*

Bentayeb, F. and Khemiri, R. (2013). Adapting olap analysis to user's constraints through semantic hierarchies. In *Proceedings of the 15th International Conference on Enterprise Information Systems (ICEIS 2013)*, volume 1, pages 160–167. *2 citations pages 56 et 98*

Bernstein, P. A. and Goodman, N. (1981). Concurrency control in distributed database systems. *ACM Computing Surveys*, 13 :185–221. *Cité page 18*

Böhnlein, M. and Ulbrich, A. (2000). *Grundlagen des Data Warehousing*. *2 citations pages 21 et 22*

Bimonte, S. (2007). *Intégration de l'information géographique dans les entrepôts de données et l'analyse en ligne : de la modélisation à la visualisation.* PhD thesis, Institut National des Sciences Appliquées de Lyon. *5 citations pages 14, 18, 19, 21 et 24*

Bimonte, S., Boulil, K., Pinet, F., and Kang, M.-A. (2013a). Design of complex spatio-multidimensional models with the icsolap uml profile. *ICEIS 2013*, pages 3–19. *Cité page 100*

Bimonte, S., Edoh-alove, E., Nazih, H., Kang, M.-A., and Rizzi, S. (2013b). Protolap : Rapid olap prototyping with on-demand data supply. In *Proceedings of the Sixteenth International Workshop on Data Warehousing and OLAP*, DOLAP '13, pages 61–66, New York, NY, USA. ACM. *7 citations pages 80, 88, 89, 92, 93, 96 et 99*

Bimonte, S., Tchounikine, A., Miquel, M., and Pinet, F. (2010). When spatial analysis meets olap : Multidimensional model and operators. *International Journal of Data Warehousing and Mining (IJDWM)*, 6(4) :33–60. *Cité page 58*

Bishop, C. (1995). *Neural Networks for Pattern Recognition.* Oxford University Press. *2 citations pages 32 et 102*

Blew, R. D. (1996). On the definition of ecosystem. *Bulletin of the Ecological Society of America*, 77(3) :171–173. *Cité page 35*

Blondel, J., Ferry, C., and Frochot, B. (1981). *Estimating Numbers of Terrestrial Birds. Studies in avian biology.*, volume 6, chapter Point counts with unlimited distance, pages 414–420. RALPH and SCOTT Eds. *3 citations pages 40, 57 et 94*

Boulil, K., Bimonte, S., and Pinet, F. (2015). Conceptual model for spatial data cubes : A uml profile and its automatic implementation. *Computer Standards & Interfaces*, 38 :113–132. *3 citations pages 99, 100 et 109*

Boulil, K., Pinet, F., Bimonte, S., Carluer, N., Lauvernet, C., Cheviron, B., Miralles, A., and Chanet, J.-P. (2013). Guaranteeing the quality of multidimensional analysis in data warehouses of simulation results : Application to pesticide transfer data produced by the macro model. *Ecological Informatics*, 16 :41–52. *Cité page 54*

Breiman, L. (1984). *Classification and regression trees.* Wadsworth International Group, Belmont, Calif. *Cité page 32*

Briand, L., Morasca, S., and Basili, V. (2002). An operational process for goal-driven definition of measures. *IEEE Transactions on Software Engineering*, 28(12) :1106–1125. *Cité page 153*

Carme, A., Mazon, J.-N., and Rizzi, S. (2010). A model-driven heuristic approach for detecting multidimensional facts in relational data sources. In Pedersen,

T., Mohania, M., and Tjoa, A. M., editors, *Proceedings of 12th International Conference on Data Warehousing and Knowledge Discovery (DaWaK)*, volume LNCS 6263, pages 13–24. *Cité page 137*

Ceci, M., Cuzzocrea, A., and Malerba, D. (2011). Olap over continuous domains via density-based hierarchical clustering. In *15th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES 2011)*, volume 2, pages 559–570. *4 citations pages 56, 98, 100 et 138*

Choong, Y. W., Laurent, A., and Laurent, D. (2008). Mining multiple-level fuzzy blocks from multidimensional data. *Fuzzy Sets and Systems*, 159 :1535–1553. *Cité page 34*

Codd, E., Codd, S., and Salley, C. (1993). Providing olap (on-line analytical processing) to user-analysts : An it mandate. *Codd and Dat, Inc*, 32 :31. *4 citations pages 13, 18, 53 et 169*

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20 :273–297. *2 citations pages 102 et 104*

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1) :21–27. *Cité page 102*

Cravero, A. and Sepúlveda, S. (2014). Multidimensional design paradigms for data warehouses : A systematic mapping study. *Journal of Software Engineering and Applications (JSEA)*, 7 :53–61. *3 citations pages 24, 54 et 97*

Delyon, B. (2012). *Estimation paramétrique*. IRMAR - Université Rennes 1. *Cité page 32*

Deng, Y., Frankl, P., and Chen, Z. (2003). Testing database transaction concurrency. In *In Proceedings of the 18th International Conference on Automated Software Engineering*, pages 184–195. Society Press. *Cité page 18*

Devroye, L. (1986). A note on the height of binary search trees. *Journal of the ACM (JACM)*, 33(3) :489–498. *Cité page 76*

Dubitzky, W., Krebs, O., and Eils, R. (2001). Minding, olaping, and mining biological data : Towards a data warehousing concept in biology. In *Proc. Network Tools and Applications in Biology (NETTAB), CORBA and XML : Towards a Bioinformatics Integrated Network Environment*, pages 78–82. *Cité page 13*

Duhamel, P. and Vetterli, M. (1990). Fast fourier transforms : a tutorial review and a state of the art. *Signal Processing*, 19 :259–299. *Cité page 41*

Dumolard, P. (1999). Accessibilité et diffusion spatiale. *Espace géographique*, 28 :205 – 214. *Cité page 28*

Eder, J., Koncilia, C., and Mitsche, D. (2003). Automatic detection of structural changes in data warehouses. In *Proceedings of the 5th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2003)*, pages 119–128. *3 citations pages 55, 97 et 100*

Faragó, S. and Hangya, K. (2012). Effects of water level on waterbird abundance and diversity along the middle section of the danube river. *Hydrobiologia*, 697(1) :15–21. *Cité page 38*

Favre, C., Bentayeb, F., and Boussaid, O. (2006). A knowledge-driven data warehouse model for analysis evolution. *Frontiers in Artificial Intelligence and Applications*, 143 :271. *6 citations pages 55, 98, 100, 134, 137 et 138*

Fayyad, U., Piatetsky-shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, 17 :37–54. *3 citations pages 26, 27 et 93*

Frawley, W. J., Piatetsky-shapiro, G., and Matheus, C. J. (1992). Knowledge discovery in databases : An overview. *AI Magazine*, pages 57 – 70. *Cité page 26*

Frochot, B., Eybert, M., Journaux, L., Roché, J., and Faivre, B. (2003). Nesting birds assemblages along the river loire : result from a 12 years-study. *Alauda*, 71(2) :179–190. tiré à part. *2 citations pages 57 et 94*

Ganti, V., Gehrke, J., and Ramakrishnan, R. (2001). Demon : Mining and monitoring evolving data. *IEEE Transactions on Knowledge and Data Engineering*, 13(1). *Cité page 163*

Golfarelli, M. and Rizzi, S. (2009). *Data Warehouse Design : Modern Principles and Methodologies*. McGraw-Hill, Inc., New York, NY, USA, 1 edition. *3 citations pages 21, 140 et 141*

Golfarelli, M. and Rizzi, S. (2011). Data warehouse testing : A prototype-based methodology. *Information and Software Technology*, 53(11) :1183–1198. *Cité page 93*

Golli, I. G. E. (2009). *Ingénierie des Exigences pour les Systèmes d'Information Décisionnels : Concepts, Modèles et Processus - La méthode CADWE*. PhD thesis, Université Paris 1 - Panthéon - Sorbonne. *Cité page 18*

Gower, J. (1971). A general coefficient of similarity and some of its properties. *Biometrics*, 27(4) :857–871. *2 citations pages 69 et 106*

Hammami, M. (2005). *Modèle de peau et application à la classification des images et au filtrage des sites Web*. PhD thesis, Ecole centrale de Lyon. *Cité page 27*

Han, J. (1997). Olap mining : An integration of olap with data mining. In *Proceedings of the 7th IFIP*, volume 2, pages 1–9. Citeseer. *Cité page 93*

Hastie, T., Tibshirani, R., and Friedman, J. (2008). *The Elements of Statistical Learning*. Springer-Verlag, 2nd edition. *Cité page 32*

Hubert, G. and Teste, O. (2009). Analyse multigraduelle olap. In *EGC 2009*, volume RNTI-E-15, pages 241–252. *3 citations pages 56, 98 et 100*

Hurtado, C. A., Mendelzon, A. O., and Vaisman, A. A. (1999). Updating olap dimensions. In *Proceedings of the 2nd ACM international workshop on Data warehousing and OLAP*, DOLAP'99, pages 60–66, New York, NY, USA. ACM. *2 citations pages 12 et 17*

Huynh, T. N. and Schiefer, J. (2001). Prototyping data warehouse systems. In *Proceedings of the Third International Conference on Data Warehousing and Knowledge Discovery (DaWaK)*, Lecture Notes in Computer Science, pages 195–207. Springer Berlin Heidelberg. *4 citations pages 88, 93, 99 et 100*

I.B.C.C. (1977). Censuring breeding bird by the i.p.a. method. *Polish Ecological Studies*, 3 :15–17. *Cité page 40*

Inmon, W. (1996). *Building the Data Warehouse*. Wiley, New York (U.S.A.), 2nd edition. *3 citations pages 13, 24 et 92*

Jain, A. K., Duin, R. P. W., and Mao, J. (2000). Statistical pattern recognition : A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1) :4–37. *2 citations pages 27 et 32*

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering : A review. *ACM Computing Survey*, 31(3) :264–322. *5 citations pages 27, 28, 68, 93 et 105*

Jensen, M. R., Holmgren, T., and Torben (2004). Discovering multidimensional structure in relational data. In *Data Warehousing and Knowledge Discovery : 6th International Conference (DaWaK)*. *3 citations pages 94, 137 et 138*

Jerbi, H. (2012). *Personnalisation d'analyses décisionnelles sur des données multidimensionnelles*. PhD thesis, Université de Toulouse. *5 citations pages 12, 17, 18, 19 et 21*

Jerbi, H., Ravat, F., Teste, O., and Zurfluh, G. (2009). Applying recommendation technology in olap systems. In *Enterprise Information Systems*, pages 220–233. Springer. *Cité page 53*

Jollife, I. (2002). *Principal component analysis*. Springer, 2 nd edition. *Cité page 27*

Journaux, L., Foucherot, I., and Gouton, P. (2005). Reduction of the number of spectral bands in landsat images with projection methods : pertinence of the resulting information. In *Proceedings of the 1st International Conference on Signal-Image Technology and Internet-Based Systems (SITIS)*, Yaounde, Cameroon. *Cité page 43*

Jovanovic, P., Romero, O., Simitsis, A., and Abelló, A. (2012). Ore : An iterative approach to the design and evolution of multi-dimensional schemas. In *Proceedings of the Fifteenth International Workshop on Data Warehousing and OLAP*, DOLAP '12, pages 1–8, New York, NY, USA. ACM. *4 citations pages 24, 97, 100 et 137*

Kangas, J. and Kohonen, T. (1996). Developments and applications of the self-organizing map and related algorithms. *Math Comput Simulat*, 41 :3–12. *Cité page 27*

Kimball, R. (1996). *The Data Warehouse Toolkit : Practical Techniques for Building Dimensional Data Warehouses.* Wiley. *4 citations pages 13, 92, 136 et 151*

Kimball, R. (2000). *Concevoir et déployer un data warehouse*, chapter Infrastructure et métadonnées. Eyrolles. *Cité page 21*

Kojadinovic, I. (2004). Agglomerative hierarchical clustering of continuous variables based on mutual information. *Computational Statistics & Data Analysis*, 46(2) :269 – 294. *2 citations pages 68 et 106*

Lau, H., Chin, K., Pun, K., and Ning, A. (2000). Decision supporting functionality in a virtual enterprise network. *Expert Systems with Applications*, 19 :261–270. *3 citations pages 34, 97 et 100*

Legube, B. and Merlet, N. (2009). *L'analyse de l'eau*, chapter Les indicateurs biologiques de la qualité de l'eau, pages 865–962. Dunod, 9e edition. *Cité page 39*

Lehner, W. (1998). Modeling large scale olap scenarios. In *In Advances in Database Technology - EDBT'98, volume 1377 of LNCS*, pages 153–167. Springer. *2 citations pages 19 et 53*

Lenz, H.-J. and Thalheim, B. (2009). A formal framework of aggregation for the olap-oltp model. *Journal of Universal Computer Science*, 15(1) :273–303. *Cité page 146*

Leonhardi, B., Mitschang, B., Pulido, R., Sieb, C., and Wurst, M. (2010). Augmenting olap exploration with dynamic advanced analytics. In *13th International Conference on Extending Database Technology (EDBT 2010).* *4 citations pages 56, 98, 100 et 138*

Liu, W. and Luo, Y. (2005). Applications of clustering data mining in customer analysis in department store. In *International Conference on Services Systems and Services Management, 2005. Proceedings of ICSSSM'05*, volume 2, pages 1042–1046. IEEE. *Cité page 93*

Lujan-Mora, S., Trujillo, J., and Song, I.-Y. (2006). A uml profile for multidimensional modeling in data warehouses. *Data and Knowledge Engineering*, pages 725–769. *3 citations pages 21, 99 et 100*

Madin, J., Bowers, S., Schildhauer, M., Krivov, S., Pennington, D., and Villa, F. (2007). An ontology for describing and synthesizing ecological observation data. *Ecological informatics*, 2(3) :279–296. *Cité page 165*

Mahboubi, H., Bimonte, S., Deffuant, G., Chanet, J.-P., , and Pinet, F. (2013). Semi-automatic design of spatial data cubes from simulation model results. *International Journal of Data Warehousing and Mining*, 9 :70–95. *Cité page 54*

Mahboubi, H., Faure, T., Bimonte, S., Deffuant, G., Chanet, J.-P., , and Pinet, F. (2012). *New Technologies for Constructing Complex Agricultural and Environmental Systems*, chapter A Multidimensional Model for Data Warehouses of Simulation Results, pages 1–18. P. Papajorgji and F. Pinet. *Cité page 54*

Mahboubi, H., Ralaivao, J.-C., Loudcher, S., Boussaïd, O., Bentayeb, F., Darmont, J., et al. (2009). X-wacoda : an xml-based approach for warehousing and analyzing complex data. *Data Warehousing Design and Advanced Engineering Applications : Methods for Complex Construction*, pages 38–54. *Cité page 137*

Malinowski, E. and Zimanyi, E. (2006). Hierarchies in a multidimensional model : From conceptual modeling to logical representation. *Data and Knowledge Engineering*, 59 :348–377. *4 citations pages 19, 20, 60 et 85*

Manel, S., Dias, J.-M., and Ormerod, S. J. (1999). Comparing discriminant analysis, neural networks and logistic regression for predicting species distributions : a case study with a himalayan river bird. *Ecological Modelling*, 120 :337–347. *Cité page 38*

Marcotorchino, F., M. P. (1982). Agrégation de similarités en classification automatique. *Revue de Statistique Appliquée*, 30(2) :21–44. *Cité page 27*

Markl, V., Ramsak, F., and Bayer, R. (1999). Improving olap performance by multidimensional hierarchical clustering. In *Proc. of IDEAS 99*, pages 165–177. *3 citations pages 19, 54 et 68*

Marksay, G. and Pigneur, Y. (2010). *Modéliser par l'exemple, pratique des tableurs et des bases de données*. Presses polytechniques et universitaires romandes. *Cité page 23*

Martin, R. (2003). *Agile Software Development : Principles, Patterns, and Practices*. Prentice Hall PTR. *2 citations pages 89 et 96*

Messaoud, R. B., Boussaid, O., and Rabaséda, S. (2004). A new olap aggregation based on the ahc technique. In *DOLAP 2004, ACM Seventh International Workshop on Data Warehousing and OLAP*, pages 65–72. *5 citations pages 55, 98, 100, 138 et 148*

Minsky, M. (1965). Matter, mind and models. *Artificial Intelligence Memo*, 77. *Cité page 36*

Miquel, M., Bédard, Y., and Brisebois, A. (2002a). Conception d'entrepôts de données géospatiales à partir de sources hétérogènes. exemple d'application en foresterie. *Ingénieries des Systèmes d'information*, 7(3) :89–111. *2 citations pages 14 et 149*

Miquel, M., Bédard, Y., Brisebois, A., Pouliot, J., Marchand, P., and Brodeur, J. (2002b). Modeling multi-dimensional spatio-temporal data werehouses in

a context of evolving specifications. *International Archives Of Photogrammetry Remote Sensing And Spatial Information Sciences*, 34(4) :142–147. *3 citations pages 54, 60 et 139*

Murthy, S. K. (1997). Automatic construction of decision trees from data : A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2 :345–389. *Cité page 32*

Nabet, F. (2013). *Etude du réajustement du lit actif en Loire moyenne, bilan géomorphologique et diagnostic du fonctionnement des chenaux secondaires en vue d'une gestion raisonnée*. PhD thesis, Université Paris 1 Panthéon Sorbonne. *Cité page 37*

Nguyen, T. B., Tjoa, A. M., and Wagner, R. (2000). An object oriented multidimensional data model for olap. In *Proceedings of the 1st International Conference on Web-Age Information Management (WAIM)*, number 1846 in Lecture Notes in Computer Science, pages 69–82. Springer. *5 citations pages 19, 20, 21, 53 et 138*

O'Neil, P. and O'Neil, E. (2000). *Database : Principles, Programming and Performance*. Morgan Kaufmann Publishers. *Cité page 18*

Pedersen, T. B. and Jensen, C. S. (1999). Multidimensional data modeling for complex data. In *Proceedings of ICDE '99*. *2 citations pages 19 et 53*

Pedersen, T. B., Jensen, C. S., and Dyreson, C. E. (2001). A foundation for capturing and querying complex multidimensional data. *Information Systems*, 26 :383–423. *2 citations pages 89 et 96*

Phipps, C. and Davis, K. C. (2002). Automating data warehouse conceptual schema design and evaluation. In *Proceedings of the 4th International Workshop on Design and Management of Data Warehouses (DMDW)*, volume 2. *5 citations pages 88, 99, 100, 136 et 137*

Pourabbas, E. and Rafanelli, M. (1999). Characterization of hierarchies and some operators in olap environment. In *DOLAP '99 ACM Second International Workshop on Data Warehousing and OLAP*, pages 54–59. *Cité page 19*

Power, M. E., Parker, G., Dietrich, W. E., and Sun, A. (1995). How does floodplain width affect floodplain river ecology ? a preliminary exploration using simulations. *Geomorphology*, 13 :301–317. *Cité page 38*

Pérez-Martínez, J., Llavori, R. B., Cabo, M. A., and Pedersen, T. (2008). Contextualizing data warehouses with documents. *Decision Support Systems*, pages 77–94. *Cité page 95*

Proulx, M. and Bédard, Y. (2004). Le potentiel de l'approche multidimensionnelle pour l'analyse de données géospatiales en comparaison avec l'approche transactionnelle des sig. In *Colloque Géomatique Montréal - Un choix stratégique !* *2 citations pages 18 et 21*

Pundt, H. and Bishr, Y. (2002). Domain ontologies for data sharing–an example from environmental monitoring using field gis. *Computers and Geosciences*, 28 :95–102. *Cité page 165*

Radulescu, C. Z. and Radulescu, M. (2008). A multidimensional data model for environment protection. In *Proceedings of the 12th WSEAS international conference on Computers (ICCOMP'08)*, pages 1101–1106. *Cité page 54*

Rehman, N. U., Mansmann, S., Weiler, A., and Scholl, M. H. (2012). Discovering dynamic classification hierarchies in olap dimensions. In *ISMIS 2012 : 20th International Symposium on Methodologies for Intelligent System*, pages 425–434. *3 citations pages 55, 98 et 100*

Rivest, S., Bédard, Y., and Marchand, P. (2001). Toward better support for spatial decision making : defining the characteristics of spatial on-line analytical processing (solap). *Geomatica*, 55(4) :539 – 555. *Cité page 58*

Rivest, S., Bédard, Y., Proulx, M.-J., Nadeau, M., Hubert, F., and Pastor, J. (2005). Solap technology : Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data. *ISPRS journal of photogrammetry and remote sensing*, 60(1) :17–33. *Cité page 53*

Rizzi, S. (2004). Uml-based conceptual modeling of pattern-bases. In *Proceedings of the International Workshop on Pattern Representation and Management (PaRMa)*. *2 citations pages 99 et 100*

Roché, J. (2010). Suivi temporel des oiseaux nicheurs en rivière (programme "stori") : Le cas de l'évolution sur 16 années (1991-2006) des communautés de l'allier. *Alauda*, 78(4) :253–268. *Cité page 43*

Roché, J. and Frochot, B. (1993). Ornithological contribution to river zonation. *Acta Oecologica*, 14(3) :415–434. 6. *Cité page 38*

Rokach, L., Maimon, O., and Miamon, O. Z. (2008). *Data Mining with Decision Trees : Theory and Applications*, volume 69 of *Machine Percpetion and Artificial Intelligence*. World Sctientific Publishing Co. *2 citations pages 32 et 73*

Romero, O. and Abelló, A. (2010). A framework for multidimensional design of data warehouses from ontologies. *Data & Knowledge Engineering*, 69 :1138–1157. *3 citations pages 54, 55 et 160*

Romero, O. and Abello, A. (2009). A survey of multidimensional modeling methodologies. *International Journal of Data Warehousing and Mining*, 5(2) :1–23. *4 citations pages 89, 92, 96 et 136*

Romero, O. and Abello, A. (2010). Automatic validation of requirements to support multidimensional design. *Data and Knowledge Engineering*, 69 :917–942. *6 citations pages 26, 97, 100, 137, 139 et 164*

Rumbaugh, J., Jacobson, I., and Booch, G. (2004). *The Unified Modeling Language Reference Manual*. Pearson Higher Education, 2nd edition edition. *Cité page 88*

Sapia, C., Blaschka, M., Höfling, G., and Dinter, B. (1999). Extending the e/r model for the multidimensional paradigm. *Advances in Database Technologies*, pages 105–166. *Cité page 20*

Saporta, G. (2011). *Probabilités, Analyse des données et Statistique*. TECHNIP, 3e edition. *Cité page 13*

Sarawagi, S., Agrawal, R., and Megiddo, N. (1998). Discovery-driven exploration of olap data cubes. In *In Proc. Int. Conf. of Extending Database Technology (EDBT'98*, pages 168–182. Springer-Verlag. *2 citations pages 19 et 54*

Sautot, L., Bimonte, S., Journaux, L., and Faivre, B. (2014). A methodology and tool for rapid prototyping of data warehouses using data mining : Application to birds biodiversity. In *Proceedings of 4th International Conference on Model & Data Engineering (MEDI)*. In Press. *4 citations pages 93, 96, 138 et 152*

Sautot, L., Faivre, B., Journaux, L., and Molin, P. (2015). The hierarchical agglomerative clustering with gower index : a methodology for automatic design of olap cube in ecological data processing context. *Ecological Informatics*, 26 :217–230. In Press. *6 citations pages 49, 106, 134, 137, 138 et 148*

Segurado, P. and Araujo, M. B. (2004). An evaluation of methods for modelling species distributions. *Journal of Biogeography*, 31 :1555–1568. *Cité page 69*

Shah, S., Huang, Y., Xu, T., Yuen, M., Ling, J., and Ouellette, F. (2005). Atlas – a data warehouse for integrative bioinformatics. *BMC Bioinformatics*, 34(6). *Cité page 14*

Shih, P. and Liu, C. (2006). Face detection using discriminating feature analysis and support vector machine. *Pattern Recognition*, 39 :260–276. *Cité page 104*

Soldatova, L. N. and King, R. D. (2005). Are the current ontologies in biology good ontologies ? *Nature Biotechnology*, 23(9) :1095–1098. *Cité page 165*

Song, Q., Hu, W., and Xie, W. (2002). Robust support vector machine with bullet hole image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 32(4) :440–448. *Cité page 104*

Tan, P.-N., Steinbach, M., and Kumar, V. (2006). *Introduction to Data Mining*, chapter 8 : Cluster Analysis : Basic Concepts and Algorithms, pages 487 – 568. Addison-Wesley. *2 citations pages 28 et 29*

Tebourski, W., Karâa, W. B. A., and Ghezala, H. B. (2013). Semi-automatic data warehouse design methodologies : a survey. *International Journal of Computer Science Issues (IJCSI)*, 10(2) :48–54. *3 citations pages 24, 54 et 97*

Teste, O. (2006). *Modélisation et manipulation d'entrepôts de données complexes et historisés*. PhD thesis, Université Paul Sabatier de Toulouse. *3 citations pages 18, 21 et 24*

Thenmozhi, M. and Vivekanandan, K. (2013). A tool for data warehouse multidimensional schema design using ontology. *International Journal of Computer Science Issues (IJCSI)*, 10(3) :161–168. *4 citations pages 55, 97, 100 et 164*

Torlone, R. (2003). *Multidimensional Databases*, chapter Conceptual multidimensional models, pages 69–90. *Cité page 93*

Triplet, T. and Butler, G. (2011). Systems biology warehousing : Challenges and strategies toward effective data integration. In *Proceedings of the Third International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA)*, pages 34–40. *Cité page 12*

Tryfona, N., Busborg, F., and Borch Christiansen, J. G. (1999). starer : a conceptual model for data warehouse design. In *Proceedings of the 2nd ACM international workshop on Data warehousing and OLAP*, pages 3–8. ACM. *Cité page 20*

Tsois, A., Karayannidis, N., and Sellis, T. (2001). Mac : Conceptual data modeling for olap. In *3rd International Workshop on Design and Management of Data Warehouses (DMDW 2001*, page 2001. *2 citations pages 19 et 53*

Tuffery, S. (2011). *Data mining and statistics for decision making*. John Wiley & Sons. *12 citations pages 26, 27, 28, 29, 30, 32, 33, 68, 93, 105, 115 et 128*

Usman, M., Asghar, S., and Fong, S. (2010). Data mining and automatic olap schema generation. In *Fifth International Conference on Digital Information Management (ICDIM)*, pages 35–43. IEEE. *8 citations pages 24, 34, 53, 55, 68, 97, 98 et 100*

Usman, M. and Pears, R. (2010). A methodology for integrating and exploiting data mining techniques in the design of data warehouses. In *6th International Conference on Advanced Information Management and Service (IMS)*, pages 361–367. IEEE. *2 citations pages 55 et 98*

van Solingen, R., Basili, V., Caldiera, G., and Rombach, H. D. (2002). *Encyclopedia of Software Engineering*, chapter Goal Question Metric (GQM) Approach. John Wiley & Sons. *Cité page 121*

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5) :988–999. *Cité page 104*

Vassiliadis, P. and Sellis, T. (1999). A survey on logical models for olap databases. *SIGMOD Record*, 28 :64–69. *Cité page 21*

Vidal, A. (1955). L'analyse de groupes à variables multiples ou analyse typologique. *Revue de statistique appliquée*, 3(4) :87 – 94. *Cité page 28*

Ward, J. (1963). Hierarchical grouping to optimize an objective function. *Journal of American Statistical Association*, 48 :236–244. *2 citations pages 125 et 128*

Wehrle, P. (2009). *Modèle multidimensionnel et OLAP sur architecture de grille*. PhD thesis, Institut National des Sciences Appliquées de Lyon. *4 citations pages 19, 20, 21 et 24*

Wehrle, P., Miquel, M., and Tchounikine, A. (2005). A model for distributing and querying a data warehouse on a computing grid. In *Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on*, volume 1, pages 203–209. IEEE. *Cité page 53*

Westphal, M. I., Field, S. A., and Possingham, H. P. (2007). Optimizing landscape configuration : A case study of woodland birds in the mount lofty ranges, south australia. *Landscape and Urban Planning*, 81 :56–66. *Cité page 69*

Zhang, C. and Huang, Y. (2007). Cluster by : a new sql extension for spatial data aggregation. In *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*. *2 citations pages 98 et 100*

Zubcoff, J., Pardillo, J., and Trujillo, J. (2007). Integrating clustering data mining into the multidimensional modeling of data warehouses with uml profiles. In *Data Warehousing and Knowledge Discovery*, pages 199–208. Springer. *Cité page 100*

Zubcoff, J., Pardillo, J., and Trujillo, J. (2009). A uml profile for the conceptual modelling of data-mining with time-series in data warehouses. *Information and Software Technology*, 51 :977–922. *3 citations pages 93, 99 et 100*

Zubcoff, J. and Trujillo, J. (2007). A uml 2.0 profile to design association rule mining models in the multidimensional conceptual modeling of data warehouses. *Data and Knowledge Engineering*, 63 :44–62. *Cité page 99*