N° D'ORDRE : 03 / 2007 – M / IN

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي و البحث العلمي
جامعة هواري بومدين للعلوم و التكنولوجيا

FACULTE D'ELECTRONIQUE ET INFORMATIQUE

جامعة هواري بومدين
للعلوم والتكنولوجيا
U S T H B

MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTER
EN : INFORMATIQUE

Spécialité : Intelligence Artificielle et Bases de Données Avancées

Par

M^r *FANTAZI Abdelouaheb*

Thème

ETUDE COMPARATIVE DES METHODOLOGIES DE CONCEPTION DES ENTREPÔTS DE DONNEES

Soutenu publiquement le : 11 / 04 / 2007, devant la commission d'examen :

N. BADACHE	Professeur à l'U.S.T.H.B.	Président
Z. ALIMAZIGHI	Maître de Conférence à l'U.S.T.H.B.	Directeur de thèse
M. AHMED NACER	Maître de Conférence à l'U.S.T.H.B.	Examinateur
S. LARABI	Maître de Conférence à l'U.S.T.H.B.	Examinateur
N. SELMOUNE	Maître Assistant à l'U.S.T.H.B.	Invité

Remerciements

Je tiens à remercier très sincèrement l'ensemble des membres du jury qui me font le grand honneur d'avoir accepté de juger mon travail.

Je remercie Monsieur ***Badache Nadjib***, *Professeur à l'USTHB, pour avoir accepté d'être président de jury et examinateur de mon travail. Je tiens à lui exprimer mes remerciements pour l'honneur qu'il me fait en participant à ce jury.*

Je remercie Monsieur **Ahmed Nacer Mohamed**, *Professeur à l'USTHB, pour avoir accepté d'être examinateur de mon travail. Je tiens à lui exprimer mes remerciements pour l'honneur qu'il me fait en participant à ce jury.*

Je remercie Monsieur **Larabi Slimane**, *Maître de conférence à l'USTHB, pour avoir accepté d'être examinateur de mon travail. Je tiens à lui exprimer mes remerciements pour l'honneur qu'il me fait en participant à ce jury.*

Je remercie Madame **Alimazighi Zaia**, *Maître de conférence à l'USTHB, et directeur de ma thèse, pour toute la confiance qu'elle m'a témoignée tout au long de ces années et sa constante disponibilité. Ses remarques constructives ont contribué à améliorer les travaux de recherche présentés dans ce mémoire. Qu'elle soit ici assurée de ma profonde gratitude et de mon très grand respect.*

Je remercie Monsieur **Selmoune Nazih**, *Chargé de cours à l'USTHB, pour son soutien et sa collaboration de tous les instants. Son aide et sa disponibilité ainsi que ses précieuses remarques ont grandement contribué à améliorer la qualité de ce mémoire. Qu'il trouve donc ici l'assurance de ma profonde gratitude. Je tiens à souligner également ses qualités humaines qui ont contribué à tisser des liens d'amitié entre nous.*

Je voudrais également exprimer mes remerciements aux personnes extérieures au monde universitaire qui m'ont soutenu. En particulier, je remercie tous mes amis avec lesquels j'ai passé des moments inoubliables. J'exprime en particulier ma gratitude à MESKINE kamel, MOSTEFAI Khemissi, et CHEURFA Elkhemissi pour toute l'aide qu'ils m'ont accordé.

Je remercie tout particulièrement mes parents qui m'ont toujours soutenu et qui m'ont permis de mener à bien mes études. Je tiens à remercier également mes sœurs et frères qui m'ont supporté de nombreuses années. Enfin, je remercier ma femme qui me supporte encore.

Table des matières

FIGURES

Tableaux

Lexique

Datacube

Le datacube est une représentation conceptuelle de données multidimensionnelles. Le datacube a autant de dimensions que la donnée a d'axes d'analyse.

Datamart

Le terme de Datamart (littéralement *magasin de données*) désigne un sous-ensemble du datawarehouse contenant les données du datawarehouse pour un secteur particulier de l'entreprise (département, direction, service, gamme de produit, ...). On parle ainsi par exemple de *DataMart Marketing*, *DataMart Commercial*, ...

Datamining

Les outils de datamining, ou *fouille de données*, s'appuyant sur des techniques d'intelligence artificielle, permettent d'extraire de la connaissance des données en découvrant des modèles, des règles dans le volume d'information présent dans les entreprises.

Datawarehouse

C'est un entrepôt de données d'une entreprise centralisées regroupant des informations structurées nécessaires à la prise de décision. D'après Bill Inmon, un Datawarehouse est intégré, orienté sujet et contient des données non volatiles et historisées.

EIS

Un EIS (Executive Information System) est un outil de visualisation et de navigation dans les données permettant de constituer des tableaux de bord. Il est constitué d'outils qui permettent aux différents niveaux de management d'accéder aux informations essentielles de leur organisation, de les analyser et de les présenter de façon élaborée. Ces outils sont dotés d'une interface graphique très conviviale et très esthétique.

ETL

Un outil d'ETL (Extract Transform Load) est un outil qui permet de chercher des données, de les extraire, de les transformer puis de les charger dans une base de données.

OLAP

On Line Analytical Processing. Caractérise l'architecture nécessaire à la mise en place d'un système d'information décisionnel. S'oppose à OLTP (On Line Transaction Processing),

s’adressant les systèmes d’information transactionnels. OLAP est souvent utilisé pour faire référence exclusivement aux bases de données multidimensionnelles. De plus en plus, le terme est souvent utilisé pour désigner plus généralement le décisionnel dans ses aspects techniques.

Reporting

Consiste à analyser des données décisionnelles et les présenter sous forme de rapports prédéfinis ou de tableaux de bords afin d’aider à la prise de décision.

Système d’Information Décisionnel (SID)

Le système d'information décisionnel est un ensemble de données organisées de façon spécifique, facilement accessible et appropriées à la prise de décision ou encore une représentation intelligente de ces données au travers d'outils spécialisés. La finalité d'un système décisionnel est le pilotage de l'entreprise.

Contexte général de l'étude

Le concept de Data Warehouse a été formalisé pour la première fois en 1990. L'idée de constituer une base de données orientée sujet, intégrée, contenant des informations datées, non volatiles et exclusivement destinées aux processus d'aide à la décision fut dans un premier temps accueillie avec une certaine perplexité. Beaucoup n'y voyaient que l'habillage d'un concept déjà ancien : l'infocentre.

Mais l'économie actuelle en a décidé autrement. Les entreprises sont confrontées à une concurrence de plus en plus forte, des clients de plus en plus exigeants, dans un contexte organisationnel de plus en plus complexe et mouvant. Pour faire face aux nouveaux enjeux économiques, l'entreprise doit anticiper. L'anticipation ne peut être efficace qu'en s'appuyant sur de l'information pertinente. Cette information est à la portée de toute entreprise qui dispose d'un capital de données gérées par ses systèmes opérationnels et qui peut en acquérir d'autres auprès de fournisseurs externes.

Actuellement, les données sont surabondantes, non organisées dans une perspective décisionnelle et éparpillées dans de multiples systèmes hétérogènes. Pourtant, les données représentent une mine d'informations. Il devient fondamental de rassembler et d'homogénéiser les données afin de permettre d'analyser les indicateurs pertinents pour faciliter les prises de décisions. Pour répondre à ces besoins, le nouveau rôle de l'informatique est de définir et d'intégrer une architecture qui serve de fondation aux applications décisionnelles : le Datawarehouse (Entrepôt de Données).

Problématique

Les entrepôts de données sont nés d'un besoin utilisateur qui n'était pas satisfait par les Systèmes de Gestion de Bases de Données traditionnels. Les premières réponses ont été pragmatiques et à l'initiative de petites sociétés qui ont ainsi occupé un marché vacant. Les chercheurs et les grands constructeurs s'y sont ensuite intéressés. Les chercheurs ont alors commencé à isoler et étudier les nouveaux problèmes posés par la conception et la mise en œuvre d'entrepôts de données. Souvent en lien étroit avec le monde de la recherche, les constructeurs ont intégré dans leurs outils des techniques pour répondre à ces nouveaux besoins.

L'industrie, ayant emboîté le pas à la recherche, des résultats sont apparus assez rapidement, et plusieurs produits opérationnels, se disputent un marché des plus juteux. En parallèle, plusieurs travaux sont consacrés, à la modélisation, et implémentation des entrepôts et magasins de données.

Objectifs

L'objet de ce mémoire concerne plus particulièrement la modélisation des entrepôts et des magasins de données. Issus originellement de l'industrie, les entrepôts et magasins de données sont devenus aujourd'hui un thème de recherche à part entière.

Le but de **la modélisation des entrepôts et des magasins** de données est de fournir des abstractions permettant de détacher la manière de représenter les données de leur implantation physique. L'utilisation qui est faite des systèmes décisionnels (analyses décisionnelles au travers de processus OLAP) nécessite des représentations des données différentes de celles qui sont proposées dans les bases de données classiques [Codd 93] [Kimball 96]. Il faut organiser les données décisionnelles en fonction des analyses multidimensionnelles effectuées (analyses des données suivants plusieurs axes). D'autre part, la modélisation des données doit prendre en compte leur aspect évolutif (historisation des données) ; en effet, par nature, l'entrepôt de données conserve l'historique des informations décisionnelles [Inmon 94].

Le but de notre travail est de :

- Etudier les concepts de base des entrepôts de données, ainsi que les modèles associés.
- Rendre compte de la variété des techniques et des approches de conception des entrepôts de données, qui ont commencé à voir le jour tout en essayant de dégager les principaux problèmes sur lesquels un effort de recherche est encore nécessaire.
- Définir des critères de comparaison, entre ces approches, et déceler leurs avantages et leurs lacunes.
- Proposer une démarche de conception pour les entrepôts de données. Notre démarche devra se baser sur des méthodologies proposées dans la littérature.

Organisation du mémoire

Pour présenter notre travail et le domaine dans lequel il s'inscrit, nous avons retenu pour ce mémoire une organisation en huit chapitres.

Dans le premier chapitre, nous présentons les concepts de base des systèmes d'aide à la décision. Nous présentons les principes de la modélisation multidimensionnelle et nous distinguons les concepts d'entrepôt et de magasin de données. Nous effectuons un état de l'art concernant la recherche actuelle dans le domaine des entrepôts.

Dans le deuxième chapitre, nous présentons une méthode proposée par S.Rizzi et M.Golfarelli. L'objectif de cette méthode est d'ébaucher les phases essentielles dans la conception des entrepôts de données qui motive la séquence des étapes, et qui discute les

relations entre les différentes étapes et les difficultés dans les entrepôts de données. Cette méthode semi automatique est basée sur un modèle conceptuel de l'entrepôt appelé le Modèle Dimensionnel de Fait « Dimensional Fact Model ».

Dans le troisième chapitre, nous présentons une méthodologie de conception des entrepôts et magasins de données proposée par Daniel L. Moody et Mark A.R. Kortink ; les deux auteurs proposent de partir d'un modèle existant, le modèle d'entreprise, puis de le transformer en modèle dimensionnel. La première section est une présentation du modèle de données utilisé, alors que la deuxième section est une présentation de la méthode.

Dans le quatrième chapitre, nous présentons une méthode de modélisation des entrepôts et magasins de données proposée par Frank Ravat, Olivier Teste et Gilles Zurfluh. Ces auteurs exigent que ; la conception d'un système décisionnel doit passer par la séparation de l'entrepôt de données et des magasins de données. Ces différences de fonction et d'objectif se répercutent dans la modélisation de ces deux espaces de stockage. Pour la partie entrepôt les auteurs [Ravat, Teste, Zurfluh 00] proposent un modèle permettant de décrire l'entrepôt comme un référentiel centralisé de données complexes, temporelles et extraites d'une source d'information. Ce modèle intègre trois concepts : l'objet entrepôt, la classe entrepôt et l'environnement. Ensuite, les auteurs [Ravat, Teste, Zurfluh 00] définissent un processus d'élaboration d'entrepôt à partir d'une source globale. Les magasins de données sont dédiés aux analyses décisionnelles de type OLAP. La modélisation multidimensionnelle est utilisée à ce niveau puisqu'elle s'avère adaptée à ce type d'activité.

Dans le cinquième chapitre, nous présentons une quatrième approche de conception d'entrepôt de données proposée par Michael Böhnlein, et Achim Ulbrich-vom Ende [M.Boh, A.Ulb 00]. C'est une approche basée sur la dérivation des structures de l'entrepôt de données à partir des modèles de processus d'affaires.

Dans le sixième chapitre, nous présentons une méthode de modélisation dimensionnelle des entrepôts et magasins de données proposées par Ralph Kimball ; dans son livre « Concevoir et déployer un data warehouse ».

Dans le septième chapitre, nous allons faire l'évaluation de chacune des cinq méthodologies de développement présentées précédemment. Nous définissons un ensemble de critères de comparaisons entre ces cinq méthodologies de conception d'entrepôt de données.

Dans le dernier chapitre, nous définissons un modèle d'élaboration d'un système décisionnel basé sur la dualité entre les deux méthodologies ; « définition des étapes essentielles pour la conception d'un entrepôt de données [M.Gol, S.Riz 00] », Et « une méthode de construction : du modèle relationnel vers le modèle dimensionnel [Kortink et al 99] ».

I.1. Introduction

Avec la généralisation de l'informatique dans tous les secteurs d'activité, les entreprises produisent et manipulent de très importants volumes de données électroniques. Ces données sont stockées dans les systèmes opérationnels de l'entreprise au sein de bases de données, de fichiers… L'exploitation de ces données dans un but d'analyse et de support à la prise de décision s'avère difficile et fastidieuse ; elle est réalisée le plus souvent de manière imparfaite par les décideurs grâce à des moyens classiques (requêtes SQL, vues, outils graphiques d'interrogation…).

Ces systèmes paraissent peu adaptés pour servir de support à la prise de décision. Ces bases opérationnelles utilisent le modèle relationnel ; celui-ci convient bien aux applications gérant l'activité quotidienne de l'entreprise, mais s'avère inadapté au décisionnel [Codd 93] [Kimball 96]. Face à cette inadéquation, les entreprises ont recours à des systèmes d'aide à la décision spécifiques, basés sur l'approche des entrepôts de données. Cependant, de tels systèmes restent difficiles à élaborer et sont souvent réalisés de manière empirique, rendant l'évolution du système décisionnel délicate. Actuellement, les entreprises ont besoin d'outils et de modèles pour la mise en place de systèmes décisionnels comportant des données évolutives.

L'objet de ce chapitre est de présenter les concepts inhérents aux systèmes décisionnels et de présenter les différents axes de recherche dans les domaines des entrepôts de données.

I.2. Infocentre et data warehouse

Certaines caractéristiques sont identiques. Mais il existe de nombreux éléments permettant de différencier les deux notions.

L'infocentre est une collection de données orientées sujet, intégrées, volatiles, actuelles, organisées pour le support d'un processus de décision ponctuel.

Le Data Warehouse est une collection de données orientées sujet, intégrées, non volatiles, historisées, organisées pour le support d'un processus d'aide à la décision.

Dans un infocentre, chaque nouvelle valeur remplace l'ancienne valeur. Il est donc impossible de retrouver une valeur calculée dans une session préalable aux dernières alimentations. La non volatilité est une caractéristique essentielle du Data Warehouse.

De même, l'historisation des données dans un infocentre, il n'y a pas de gestion d'historique des valeurs.

L'infocentre sert à prendre des décisions opérationnelles basées sur des valeurs courantes.

Au niveau d'un Data Warehouse, l'utilisateur travaille sur les historiques pour des prises de décisions à long terme, des positionnements stratégiques et pour analyser des tendances.

Dans un infocentre, l'intégration des données est plus ou moins poussée. Le processus d'alimentation est simple.

La finalité d'un infocentre est de permettre aux utilisateurs d'accéder à leurs données dans leurs propres termes.

Donc : L'infocentre est un outil alors que le Data warehouse est une architecture.

I.3. Définition de l'entrepôt de données

De nombreuses définitions ont été proposées, soit académiques, soit par des éditeurs d'outils, de bases de données ou par des constructeurs, cherchant à orienter ces définitions dans un sens mettant en valeur leur produit.

La définition que l'on retrouve le plus souvent :

Un entrepôt de données se définit comme « une collection de données intégrées, orientées sujet, non volatiles, historisées, résumées et disponibles pour l'intégration et l'analyse » [Inmon 96].

- **Données intégrées**

Un Data Warehouse est un projet d'entreprise. Par exemple dans la distribution, le même indicateur de chiffre d'affaires intéressera autant les forces de vente que le département financier ou les acheteurs. Pour y parvenir, les données doivent êtres intégrées.

Avant d'être intégrées dans le Data Warehouse, les données doivent êtres mises en forme et unifiées afin d'avoir un état cohérent. Par exemple, la consolidation de l'ensemble des informations concernant un client donné est nécessaire pour donner une vue homogène de ce client. Une donnée doit avoir une description et un codage unique.

- **Orientées sujet**

Le Data Warehouse est organisé autour des sujets majeurs de l'entreprise, contrairement aux données des systèmes de production. Ceux-ci sont généralement organisés par processus fonctionnels. Les données sont structurées par thème.

L'intérêt de cette organisation est de disposer de l'ensemble des informations utiles sur un sujet le plus souvent transversal aux structures fonctionnelles et organisationnelles de l'entreprise.

Cette orientation « sujet » va également permettre de développer son système décisionnel via une approche par itérations successives, sujet après sujet.

L'intégration dans une structure unique est indispensable car les informations communes à plusieurs sujets ne doivent pas être dupliquées. Dans la pratique, une structure supplémentaire appelée Data Mart (magasin de données) peut être créée pour supporter l'orientation « sujet ».

- **Données non volatiles**

La non volatilité des données est en quelque sorte une conséquence de l'historisation. Une même requête effectuée à quelques mois d'intervalle en précisant la date de référence de l'information recherchée donnera le même résultat.

- **Données historisées**

Dans un système de production ; la donnée est mise à jour à chaque nouvelle transaction. Dans un Data Warehouse, la donnée ne doit jamais être mise à jour. Un référentiel temps doit être associé à la donnée afin d'être capable d'identifier une valeur particulière dans le temps.

- **Données résumées**

Les informations issues des sources de données doivent être agrégées et réorganisées afin de faciliter le processus de prise de décision.

- **Disponibles pour l'interrogation et l'analyse**

Les utilisateurs doivent pouvoir consulter les données réorganisées de l'entrepôt en fonction de leurs droits d'accès.

I.4. Les objectifs du data warehouse

Voici les objectifs du data warehouse tels que définis par Ralph Kimball, dans son livre « Entrepôts de données, Guide pratique du concepteur de data warehouse »,

Accès aux informations de l'entreprise

L'entrepôt de données assure l'accès aux informations de l'entreprise et de l'organisation.

Les informations d'un entrepôt de données sont cohérentes.

Cela signifie que les requêtes faites à des moments différents doivent fournir les mêmes résultats. La cohérence veut aussi dire que lorsque les personnes demandent la définition de l'élément « contrat », elles obtiennent une réponse leur permettant de savoir ce qu'elles obtiendront de la base de données. La cohérence implique que les données sont chargées dans leur totalité. Les données d'un entrepôt doivent pouvoir être séparées et combinées au moyen de toutes les mesures possibles de l'activité.

Les outils de présentation d'informations font partie du data warehouse

L'entrepôt de données ne comporte pas seulement des données, mais aussi un ensemble de requêtes, d'analyses et de présentations des informations.

Les données publiées sont stockées dans le data warehouse

L'entrepôt de données est le lieu où sont publiées des données qui ont déjà servi. Les données sont soigneusement rassemblées à partir de sources d'informations situées à différents endroits de l'organisation. Elles sont nettoyées, leur qualité est vérifiée et elles ne sont diffusées que si elles sont prêtes à être utilisées. Si l'information est peu fiable ou incomplète, les données ne peuvent être publiées à destination de la communauté des utilisateurs.

Qualité de l'information d'un data warehouse

La qualité de l'information d'un entrepôt de données est très importante. En effet, comment obtenir des analyses fiables si les données brutes sont de mauvaise qualité ?

L'entrepôt de données ne peut remédier à la mauvaise qualité des données ou à l'absence d'une donnée. La seule façon de remédier à la médiocre qualité des données consiste, pour les personnes concernées par la saisie des données et pour le management, à retrouver la source des informations et à mettre en place de meilleurs systèmes ou à mieux faire comprendre l'importance de la qualité des données.

II. Les concepts de bases

II.1. Systèmes OLTP versus systèmes OLAP

Les bases de données sont utilisées dans les entreprises pour gérer les importants volumes d'informations contenus dans leurs systèmes opérationnels. Ces données sont gérées selon des processus transactionnels en ligne (OLTP : "*On-Line Transactional Processing"* [Cod93]) qui se caractérisent de la manière suivante [Codd 93] [Inmon 94] [Kimball 96] [Cha, Day 97] :

- ils sont nombreux au sein d'une entreprise,
- ils concernent essentiellement la mise à jour des données,
- ils traitent un nombre d'enregistrements réduit,
- ils sont définis et exécutés par de nombreux utilisateurs.

L'exploitation de l'information contenue dans ces systèmes opérationnels est devenue une préoccupation essentielle pour les dirigeants des entreprises qui désirent améliorer leur prise de décision par une meilleure connaissance de leur propre activité, de celle de la concurrence, des employés, des clients et des fournisseurs. Les entreprises sont donc à la recherche de systèmes supportant efficacement les applications d'aide à la décision. Ces applications décisionnelles utilisent des processus d'analyse en ligne de données (OLAP : "*On-Line Analytical Processing"* [Codd 93]). Ces processus répondent aux besoins spécifiques des analyses d'information. Dans [Codd 93] E.F. Codd définit un cahier des charges comprenant douze règles que doivent satisfaire les systèmes décisionnels : l'analyse des données doit se faire de manière interactive et rapide, pour des données quelconques et historisées. Ces processus OLAP se caractérisent de la manière suivante [Codd 93] [Inmon 94] [Kimball 96] [Cha, Day 97] :

- ils sont peu nombreux, mais leurs données et traitements sont complexes,
- il s'agit uniquement de traitements semi-automatiques visant à interroger, visualiser et synthétiser les données,
- ils concernent un nombre d'enregistrements importants aux structures hétérogènes,
- ils sont définis et mis en oeuvre par un nombre réduit d'utilisateurs qui sont les décideurs.

II.2. Systèmes décisionnels

Cette nouvelle utilisation de l'information contenue dans les bases opérationnelles des entreprises a donné lieu à l'élaboration de nouveaux systèmes dédiés à l'analyse et à la prise de décision. Ces systèmes sont désignés par le terme de systèmes décisionnels. Ils regroupent un ensemble d'informations et d'outils mis à la disposition des décideurs pour supporter de manière efficace la prise de décision [Codd 93] [Inmon 94] [Cha, Day 97].

L'architecture des systèmes décisionnels met en jeu quatre éléments essentiels : les sources de données, l'entrepôt de données, les magasins de données et les outils d'analyse et d'interrogation.

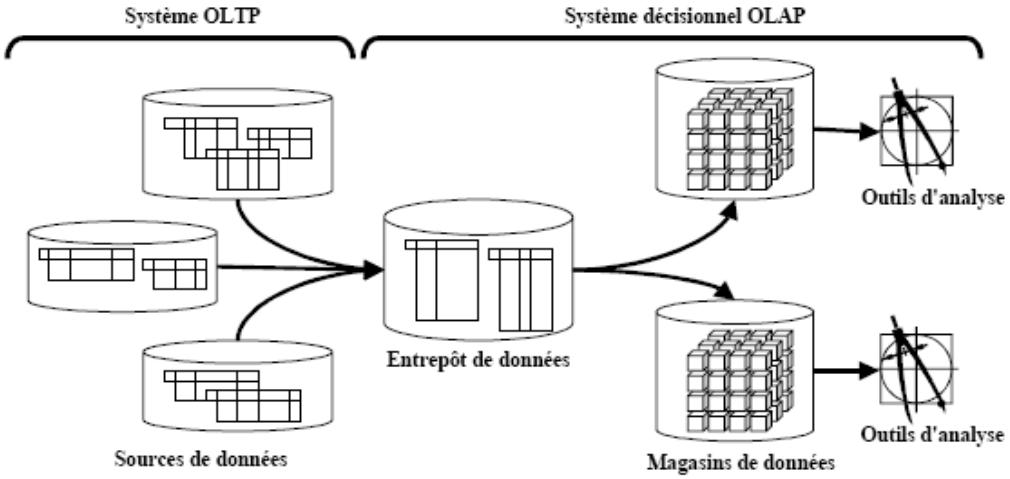


Figure 1. Architecture des systèmes décisionnels.

Les **sources de données** sont nombreuses, variées, distribuées et autonomes. Elles peuvent être internes (bases de production) ou externes (Internet, bases des partenaires) à l'entreprise.

L'**entrepôt de données** est le lieu de stockage centralisé des informations utiles pour les décideurs. Il met en commun les données provenant des différentes sources et conserve leurs évolutions.

Les **magasins de données** sont des extraits de l'entrepôt orientés sujet. Les données sont organisées de manière adéquate pour permettre des analyses rapides à des fins de prise de décision.

Les **outils d'analyse** permettent de manipuler les données suivant des axes d'analyses. L'information est visualisée au travers d'interfaces interactives et fonctionnelles dédiées à des décideurs souvent non informaticiens (directeurs, chefs de services,…).

II.3. Le modèle dimensionnel

L'entrepôt de données est aujourd'hui l'application la plus importante dans la technologie des bases de données. La modélisation d'un entrepôt de données est une étape cruciale car il faut

concevoir un modèle qui permettra d'historiser des données et de répondre à des questions que les décideurs se posent. Comment doit on procéder ? Doit-on faire évoluer un modèle existant ? Doit-on tout repenser à zéro ?

Dans les années 1990, l'entrepôt fut proposé comme une solution aux problèmes de satisfaction de décisions organisationnelles. Un entrepôt de données est une base de données qui contient une collection de données consistantes, ce qui permet leur analyse. Contrairement aux bases de données où les requêtes sont prédéfinies et figées par les informaticiens, les modèles d'entrepôts sont conçus pour être utilisés en libre-service par les utilisateurs appropriés. L'utilisateur doit pouvoir interroger l'entrepôt simplement, avec ses mots, sans faire appel au service informatique.

Ainsi, le modèle OLTP, concept des bases de données classiques, est inadapté à l'entrepôt car celui-ci n'est pas basé sur les transactions. Kimball [Kimball 96], en 1996 et 1997, propose le modèle dimensionnel qui est basé sur l'observation et l'utilisation des données dans le temps.

D'une manière très générale, le modèle dimensionnel possède une table de faits centrale et des tables de dimensions situées autour. Chaque enregistrement de la table de faits stocke les clés des tables de dimensions et les mesures faites à un instant précis. La taille de la table de faits est de plusieurs millions d'enregistrements, et peut nécessiter plusieurs giga octets d'occupation sur disque.

II.3.1. Définition

La **modélisation multidimensionnelle** consiste à considérer un sujet analysé comme un point dans un espace à plusieurs dimensions. Les données sont organisées de manière à mettre en évidence le sujet analysé et les différentes perspectives de l'analyse [Test 00].

II.3.2. Concepts de modèle dimensionnel

Les objectifs

Le modèle dimensionnel est plus simple qu'un modèle relationnel normalisé, en vue de faciliter la compréhension et l'interrogation de l'utilisateur final. L'objectif des bases normalisées est d'éviter les redondances afin de maximiser les performances et de garantir la cohérence des données lors des mises à jour. Poser une question non prédéfinie à une base de données normalisée relève du service informatique, à cause de la complexité du schéma de la base. L'entrepôt de données est mis à jour par des programmes d'alimentation à partir des bases sur lesquelles l'entrepôt s'appuie, et non par des utilisateurs. Il n'y a donc pas de problème lors des mises à jour. Par contre, il doit être conçu pour répondre aux requêtes des utilisateurs, qu'elles soient nombreuses ou non. Ainsi, les objectifs d'un entrepôt de données

sont de produire une structure compréhensible par les utilisateurs finaux et d'augmenter la qualité et les performances des requêtes.

Le schéma en étoile et ses dérivés

La première structure utilisée en modèle dimensionnel est l'étoile. Elle est composée d'une table centrale, appelée la table des faits, qui est reliée à des tables dimensionnelles de taille plus petite, les points de l'étoile. La table des faits contient les mesures (prix, nombre d'unités vendues….). Les tables dimensionnelles contiennent les conditions de regroupement des mesures. La table des faits est liée aux tables dimensionnelles par des associations, à tout élément d'une table de dimensions sont associés plusieurs éléments de la table centrale, éventuellement aucun ; à tout élément de la table centrale est associé un élément d'une table de dimension et un seul.

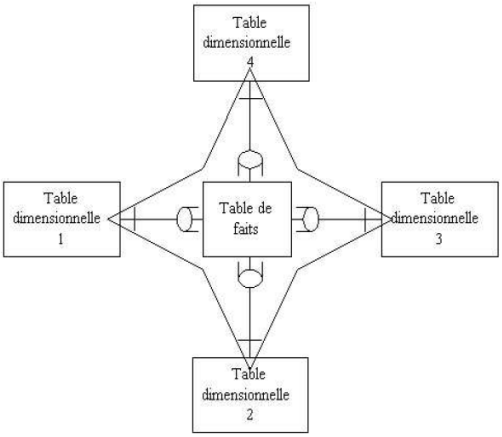


Figure 2. Schéma générique de l'étoile.

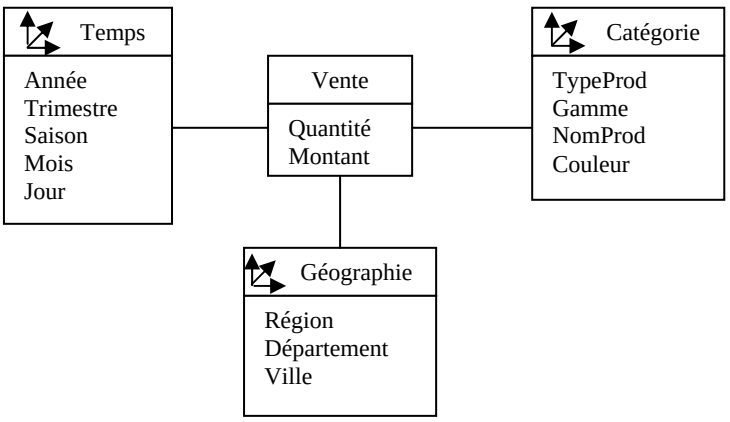


Figure 3. Exemple d'une modélisation en étoile.

Une autre technique de modélisation, issue du modèle en étoile, est la **modélisation en constellation**. Il s'agit de fusionner plusieurs modèles en étoile qui utilisent des dimensions communes. Un modèle en constellation comprend donc plusieurs faits et des dimensions communes ou non.

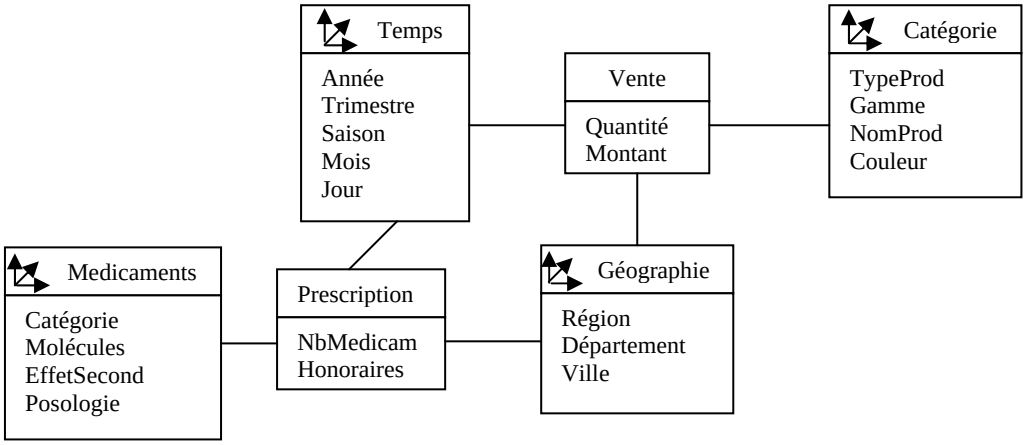


Figure 4. Exemple d'une modélisation en constellation.

Plusieurs schémas en étoile peuvent être combinés en galaxie. La galaxie est un ensemble d'étoiles avec des tables de dimension partagées. Contrairement à la constellation, les tables de faits d'une galaxie n'ont pas forcément de liens entre elles.

Le schéma en flocon de neige

Il existe d'autres techniques de modélisation dimensionnelle, notamment la **modélisation en flocon**. Une modélisation en flocon consiste à décomposer les dimensions du modèle en étoile en sous hiérarchies. La modélisation en flocon est donc une émanation de la modélisation en étoile ; le fait est conservé et les dimensions sont éclatées conformément à sa hiérarchie des paramètres. L'avantage de cette modélisation est de formaliser une hiérarchie au sein d'une dimension. Par contre, la modélisation en flocon induit une dénormalisation des dimensions générant une plus grande complexité en termes de lisibilité et de gestion.

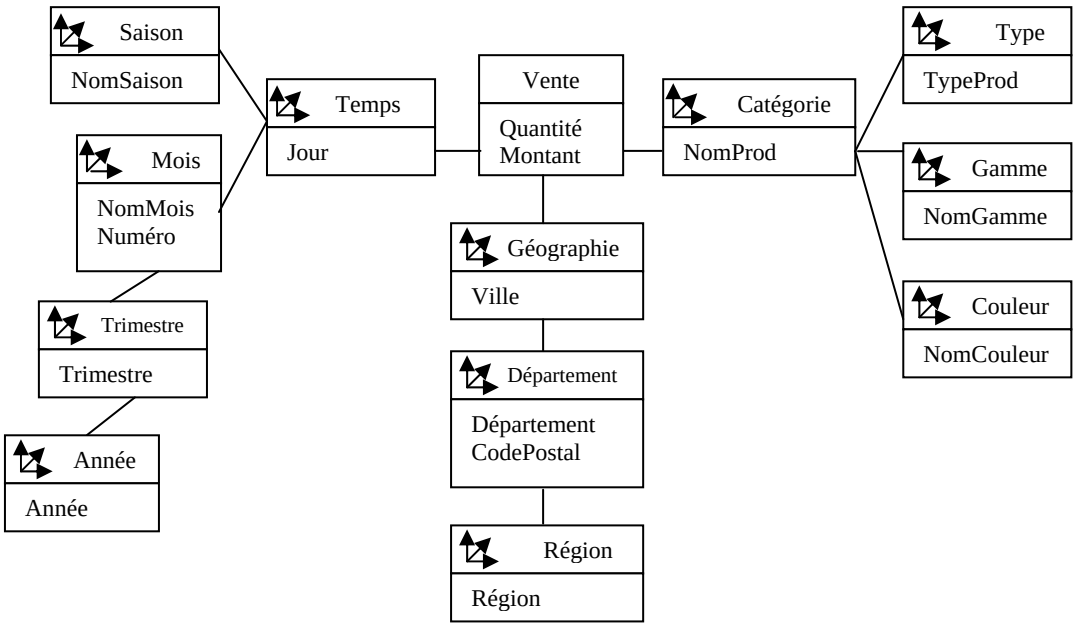


Figure 5. Exemple d'une modélisation en flocon.

Le schéma en grappe

Ce schéma est apparu car il n'existe pas de schéma en étoile ou de schéma en flocon parfait. Le schéma en grappe est une dérivation de ces deux schémas pour en former un troisième. Kimball [Kimball 96] déclare qu'un schéma en flocon n'est pas optimal, car il est trop complexe. Toujours pour les mêmes raisons de simplifications des tables, afin de pouvoir trouver facilement les informations dans l'entrepôt, le schéma en grappe apparaît alors comme un compromis entre le schéma en étoile et le schéma en flocon.

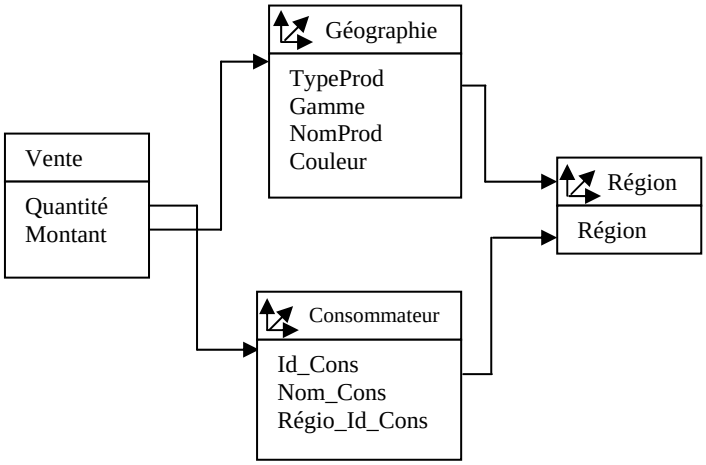


Figure 6. Exemple d'une modélisation en grappe.

III. Différents axes de recherches

Issus originellement de l'industrie, les entrepôts et magasins de données sont devenus aujourd'hui un thème de recherche à part entière.

Il existe deux axes de recherche principaux :

- l'extraction et le stockage des données,
- la modélisation des entrepôts et des magasins ainsi que leur langage d'interrogation.

L'**extraction des données** repose essentiellement sur la technique des vues matérialisées [Gupta, Mumick 95] [Widom 95]. Une vue matérialisée consiste à calculer une vue exprimée sur une source de données et à stocker physiquement les données obtenues dans l'entrepôt. Cette approche induit des problématiques de sélection et de maintenance des vues matérialisées. En effet, il faut définir des algorithmes qui répercutent les évolutions des données source au niveau des copies de données contenues dans l'entrepôt. Mais il faut aussi définir des algorithmes qui calculent un ensemble optimal de vues à matérialiser de telle sorte que les coûts liés à la maintenance de ces vues ne viennent pas altérer le fonctionnement de l'entrepôt.

Le but de **la modélisation des entrepôts et des magasins** de données est de fournir des abstractions permettant de détacher la manière de représenter les données de leur implantation physique. L'utilisation qui est faite des systèmes décisionnels (analyses décisionnelles au travers de processus OLAP) nécessite des représentations des données différentes de celles qui sont proposées dans les bases de données classiques [Codd 93] [Kimball 96]. Il faut organiser les données décisionnelles en fonction des analyses multidimensionnelles effectuées (analyses des données suivants plusieurs axes). D'autre part, la modélisation des données doit prendre en compte leur aspect évolutif (historisation des données) ; en effet, par nature, l'entrepôt de données conserve l'historique des informations décisionnelles [Inmon 94] [Chaudhuri, Dayal 97]. En outre, ces nouvelles représentations induisent de nouveaux besoins en terme de manipulation et d'interrogation des données ; ceci nécessite d'étendre les langages d'interrogation actuels [RedBrick 98].

L’objet de cette thèse concerne particulièrement la modélisation des entrepôts et des magasins de données.

IV. Conclusion

Les entrepôts de données répondent aux besoins des décideurs et des entreprises de disposer d'outils puissants et adaptés pour exploiter l'énorme masse de données rendues potentiellement accessibles avec l'émergence de l'Internet et de l'Intranet. Un entrepôt de données regroupe dans un format homogène et utile pour l'aide à la décision des données provenant de plusieurs sources de production pouvant être réparties et avoir des formats variés.

L'objet de ce mémoire est de rendre compte de la variété des techniques et des approches qui ont commencé à voir le jour tout en essayant de dégager les principaux problèmes sur lesquels un effort de recherche est encore nécessaire.

I. Introduction

Dans ce chapitre nous allons présenter une méthode proposée par S.Rizzi et M.Golfarelli ; l'objectif de cette méthode est d'ébaucher les phases essentielles dans la conception de l'entrepôt de données qui motive la séquence des étapes, et qui discute les relations entre les différentes étapes et les difficultés dans les entrepôts de données. Cette méthode semi automatique est basée sur un modèle conceptuel de DW appelé le Modèle Dimensionnel de Fait « Dimensional Fact Model ».

II. Le Modèle de Données Utilisé

Dans cette section nous allons présenter en détail le Modèle Dimensionnel de Fait « DFM » ; nous précisons que ce modèle est proposé par les mêmes auteurs de cette méthode. Ce modèle subit l'influence du modèle dimensionnel qui est étendu pour prendre en compte les caractéristiques des entrepôts de données.

II. 1. Le Modèle Dimensionnel de Fait [M.Gol, D.Mai, et S.Riz 98]

La présentation de la réalité établie par l'utilisation du DFM s'appelle le Schéma Dimensionnel et se compose d'un ensemble de schéma de fait (un pour chaque fait) dont les éléments de base sont des faits, des dimensions et des hiérarchies. Le modèle de fait dimensionnel vise a :

- Supporter efficacement le schéma conceptuel ;
- Fournir un environnement expressif où l'utilisateur peut intuitivement formuler des questions ;
- Permettre au concepteur et aux utilisateurs de discuter de manière constructive afin de raffiner les spécifications de condition ;
- Représenter une plateforme solide pour l'étape de conception logique.

Définition 1. Soit g = (V, E) un graphe dirigé, acyclique et faiblement connexe. Nous disons que g est un arbre avec la racine $v_0 \in V$ si le sommet $v_j \in V$ peut être atteint de v_0 par au moins un chemin directe. Nous dénoterons avec le $chemin_{ij}$ (g) $\subseteq$ g un chemin directe (s'il existe) démarrant dans v_i et finissant dans v_j; nous dénoterons avec le sub (v_i) $\subset$ g que l'arbre s'est enraciné dans $v_i \neq v_0$ [M. Gol, S. Riz, D. Mai 98].

Définition 2. Le schéma de fait est un six tuplé.

$$f = (M, A, N, R, O, S)$$

Si

- M est un ensemble de mesures; chaque mesure $m_i \in M$ est définie par une expression numérique ou booléenne qui implique des valeurs acquises du système d'information opérationnel.
- A est un ensemble d'attributs de dimension. Chaque attribut de dimension $a_i \in A$ est caractérisé par un domaine discret des valeurs, dom (a_i).
- N est un ensemble d'attributs de non_dimensionelle.
- R est un ensemble de couples ordonnées, chacun a la forme (a_i, a_j) si $a_i \in A \cup N$ ($a_i \neq a_j$), tel que le graphe qt (f) = ($A \cup N \cup \{ a_i \}$, R) est un arbre avec la racine a_0. a_0 est un attribut factice jouant le rôle du fait sur lequel le schéma est porté. Le couple (a_i, a_j) est une relation (2, 1) entre les attributs a_i et a_j. Nous appelons un domaine de dimension l'ensemble Dim (f) = $\{ a_i \in A \ \exists (a_0, a_1) \in R \}$; chaque élément dans Dim (f) est appelé *Dimension*.
- $O \subset R$ est un ensemble de relations facultatifs.
- S est un ensemble de relations agrégats, chacune se compose d'un triplet (m_j, d_i, Ω) ou $m_j \in M$, $d_i \in$ Dim (f) et Ω est une opération d'agrégats [M. Gol, S. Riz, D. Mai 98].

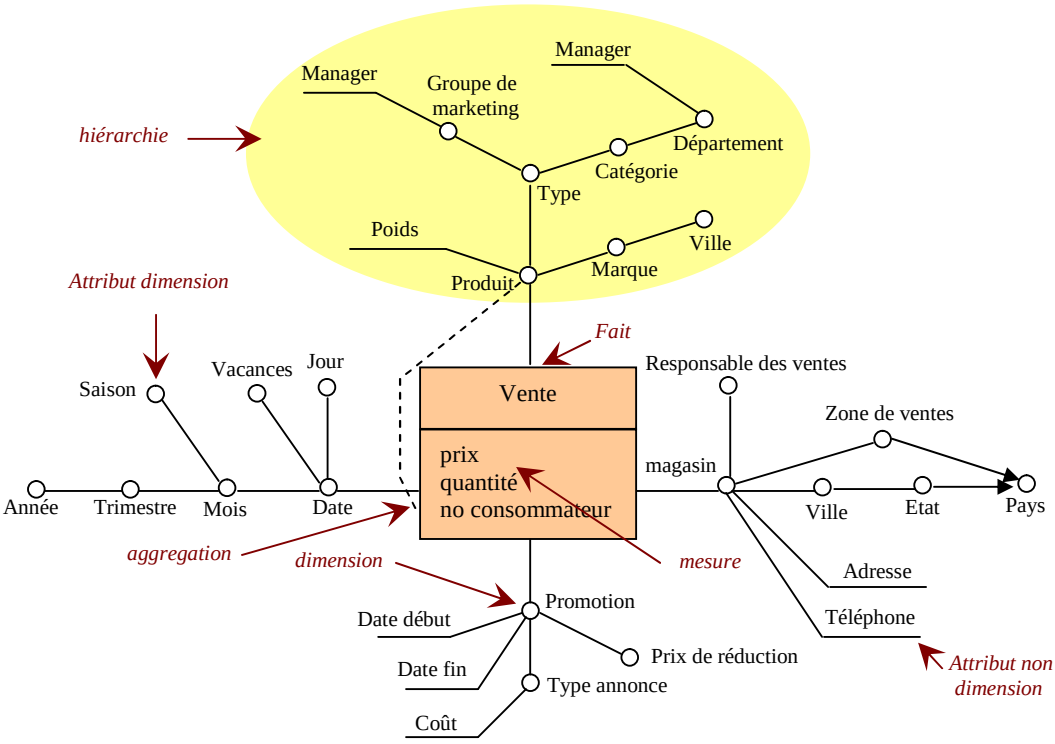


Figure 7. Le Schéma de Fait VENTE.

D'un point de vue graphique, un schéma de Fait est structuré comme un arbre dont la racine est un fait. Un fait est représenté par un rectangle qui rapporte le nom de fait.

Des attributs de dimension sont représentés par des cercles. Chaque attribut de dimension directement attaché au fait est une dimension; les dimensions déterminent la granularité adoptée pour la représentation des faits. Le domaine de dimension du schéma VENTE est (date, produit, magasin).

Les sous arbres enracinés dans les dimensions sont des hiérarchies, et déterminent comment des instances de fait peuvent être agrégés et choisis efficacement pour le processus décisionnel. La dimension dans laquelle une hiérarchie est enracinée définit une granularité d'agrégation plus fine; les attributs dimension dans les sommets le long de chaque chemin de la hiérarchie à partir de la dimension définissent une granularité progressivement plus brute. L'arc qui connecte deux attributs représente une relation (2, 1) entre eux ; ainsi, chaque chemin direct à moins d'une hiérarchie représente nécessairement une relation (2, 1) entre les attributs de début et les attributs de fin.

Le schéma de fait ne peut pas être un arbre, deux trajectoires distinctes peuvent connecter deux attributs de dimension dans une hiérarchie, à condition que chaque trajectoire directe représente une relation (1, 2). Les arcs marqués par un tiret représentent une relation facultative entre les paires d'attributs. Par exemple, le régime d'attribut prend une valeur seulement pour des produits alimentaires.

Une mesure est agrégable sur une dimension si ses valeurs peuvent être agrégées le long de la hiérarchie correspondante par au moins un opérateur, une mesure agrégable est additive si ses valeurs peuvent être agrégées par l'opérateur de somme. Puisque la plupart des mesures sont additives, afin de simplifier la notation graphique dans le DFM, seulement les exceptions sont représentées explicitement. En particulier, si m_j n'est pas additif le long de d_i, m_j et d_i sont reliés par une ligne tirée marquée à tous les opérateurs Ω tels que $(m_j, d_i, \Omega) \in S$ (dans la Figure 7), mesure *no de consommateur* est non agrégable le long de la dimension *produit* .

II. 1. 2. Instance de fait

Soit un schéma de fait f, chaque n-ple des valeurs prises des domaines dans n dimensions définit une cellule élémentaire où une unité d'information peut être représentée pour le DW. Les unités d'information dans le DW sont appelées les instances primaires de fait, chacune est caractérisée exactement par une valeur pour chaque mesure. pf $(\alpha_1,\ldots, \alpha_n)$ est la notation de l'instance primaire de fait correspondant à la combinaison des valeurs $(\alpha_1,\ldots, \alpha_n) \in$ Dom (d_1) x…x Dom (d_n). Dans le schéma de vente, chaque instance primaire décrit les ventes d'un produit pendant une journée dans un magasin.

Définition 3. Soit f un schéma de fait avec n dimensions, le domaine d'agrégation v tel que $(0 \leq v)$ est un ensemble $P= \{a_1,...a_v\}$ si :

1. $\forall i = 1,...v\ (a_i \in A)$;
2. $P \neq Dim\ (f)$;
3. $\forall a_i \in P\ (\nexists a_j \in P,\ a_i \neq a_j \mid a_j \in sub(qt(f), a_i))$ (i.e., aucun chemin direct n'existe entre chaque paire d'attributs dans P) [M. Gol, S. Riz, D. Mai 00].

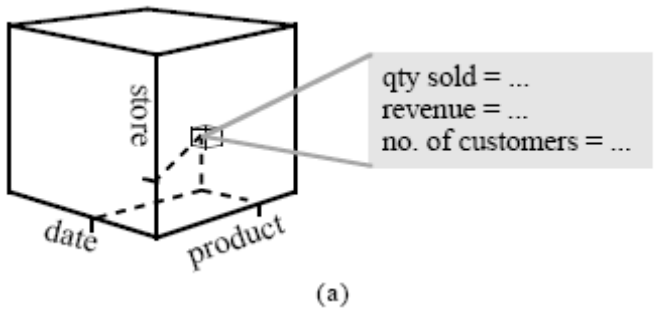
Une dimension $d_i \in Dim(f)$ est dite cachée dans P si aucun attribut de l'hiérarchie sub(qt(f), d_i) n'apparaît dans P. Un domaine d'agrégation P est légal concernant la mesure $m_j \in M$ si

$$\forall d_k \mid \exists (m_j, d_k, \Omega) \in S \qquad d_k \in P$$

Les exemples des domaines d'agrégation dans le schéma de vente sont {*produit*, *quantité*, *mois*, *promotion*}, {*état*, *date*} (produit et promotion sont cachés), {*année*, *saison*} (deux attributs prises de la dimension *date*), {} (toutes les dimensions sont cachées). Le domaine {marque, mois}est illégal concernant le numéro des clients puisque ce dernier ne peut pas être agrégé le long de la hiérarchie de produit. Soit $P=\{ a_1,\ldots a_v \}$ un domaine d'agrégation, et d_{h*} la notation de la dimension dont l'hiérarchie est incluse $a_h \in P$. L'instance de fait secondaire $sf(\beta_1,\ldots \beta_v)$ correspondant à la combinaison des valeurs $(\beta_1,\ldots \beta_v) \in Dom\ (a_1)$ x…x Dom (a_v) agrégées de l'ensemble d'instances de fait primaires.

$$\{ Pf\ (\alpha_1,\ldots,\alpha_n) \mid \forall k \in Dom\ (d_k) \wedge \forall h \in \{ 1,\ldots v \}\ \alpha_{h*}.a_h = \beta_h \}$$

Et elle est caractérisée par une valeur exacte pour chaque mesure pour que P soit légal. Cette valeur est calculée par l'application d'un opérateur d'agrégation aux valeurs que les instances primaires de fait sont agrégées. Dans le schéma vente un exemple d'instance secondaire est celui décrivant les ventes des produits d'une catégorie donnée pendant une journée dans une ville Figure 8a. La Figure 8b montre les instances primaires correspondants au fait. Le numéro de mesure des clients n'est pas rapporté puisqu'il ne peut pas être agrégé le long de la dimension produite.



(a)

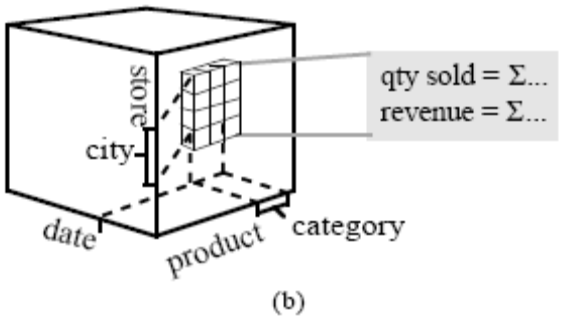


Figure8. L'instance de fait primaire (a) et secondaire (b) du schéma VENTE [M.Gol, D.Mai, et S.Riz 98].

II.1. 3. Le fait vide

Un schéma de fait est vide s'il n'a aucune mesure (M=Ø). Dans ce cas, les instances de fait primaires enregistrent seulement l'occurrence des événements. Considérez par exemple, dans le domaine d'université, le schéma de fait représenté sur la Figure 9. Dans ce cas, chaque instance de fait déclare qu'un étudiant donné a suivi un cours donné pendant une année donnée; aucune mesure n'est utilisée comme moyen de décrire ce fait.

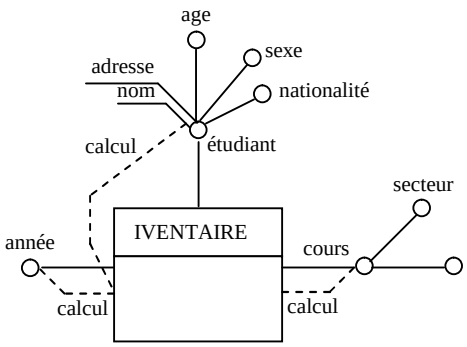


Figure 9. Le schéma de fait INVENTAIRE.

Dans un schéma de fait vide, deux approches au problème d'agrégation peuvent être poursuivies. La première approche emploie les opérateurs ET/OU, l'information diffusée par chaque instance de fait secondaire est liée à l'existence des instances de fait primaires correspondants. Le fait est décrit par une mesure booléenne implicite, qui est vraie si l'événement se produisait et faux autrement : dans ce cas, les deux opérateurs ET et OU sont employés pour l'agrégation, avec respectivement, la sémantique universelle et existentielle. Par exemple:

ATTENDANCE (*course.area*, *student;*

year='1998', *course.area*='Databases', *course.faculty*='Computer Science')

Présente les étudiants qui pendant l'année 98 ont suivi tous les cours de base de données dans la faculté d'informatique (l'opérateur ET), ou les étudiants qui pendant l'année 98 ont suivi au moins un cours de base de données dans la faculté d'informatique (l'opérateur OU).

La deuxième approche emploie l'opérateur COUNT, l'information diffusée par chaque instance de fait secondaire est le nombre d'instances de fait primaires correspondant. D'une manière équivalente, le fait est décrit par une mesure implicite de nombre entier, prend la valeur 1 si l'événement se produisait et 0 autrement, et le schéma de fait est agrégé par l'opérateur de somme. Par exemple:

ATTENDANCE (*course*, *student.sex* ; *year*='1998', *course.faculty*='Computer Science')

Présente, pour chaque cours dans la faculté d'informatique, le nombre d'étudiants de chaque sexe qui ont suivi le cours.

II.1. 4. Concaténation du schéma de fait

Dans un DFM les différents faits sont représentés dans des différents schémas. Cependant, une partie des requêtes que l'utilisateur formule sur le DW peut exiger la comparaison des mesures distinctes prises, bien que reliée, des schémas. Dans cette sous-section nous allons voir que deux schémas de fait reliés peuvent être concaténés dans un nouveau schéma; puisque le même attribut a_i apparaît dans des différents schémas de fait, probablement avec différents domaines, le domaine de a_i dans le schéma f est noté avec $Dom_f (a_i)$.

Définition 4. Deux schémas de fait f' = (M', A', N', R', O', S') et f'' = (M'', A'', N'', R'', O'', S'') sont compatibles s'ils partagent au moins un attribut de dimension :

$A' \cap A'' \neq \emptyset$. L'attribut a_i est considérée commun entre f' et f'' si, dans les deux schémas, il a la même sémantique et si $Dom_{f'} (a_i) \cap Dom_{f''} (a_i) \neq \emptyset$ [M. Gol, S. Riz, D. Mai 00].

Définition 5. Etant donnés un arbre $t = (V \cup \{ a_0 \}, E)$ avec la racine a_0, et un sous-ensemble de sommets $l \subseteq V$, nous définissons la contraction de t sur l comme un arbre :

$cn (t, l) = (l \cup \{ a_0 \}, E)$ ou :

$$E^* = \{ (a_i, a_j) \mid a_i \in l \cup \{ a_0 \} \wedge a_j \in l \wedge \exists path_{ij} (t) \wedge \forall a_k \in l - \{ a_i, a_j \} \; a_k \notin path_{ij} (t) \}$$

Les arcs du cnt (t, l) sont des chemins directs qui, à l'intérieur de t, relient des paires de sommets de l sans inclure d'autres sommets de l. La Figure 9 montre un arbre et sa contraction sur un sous-ensemble des sommets [M. Gol, S. Riz, D. Mai 98].

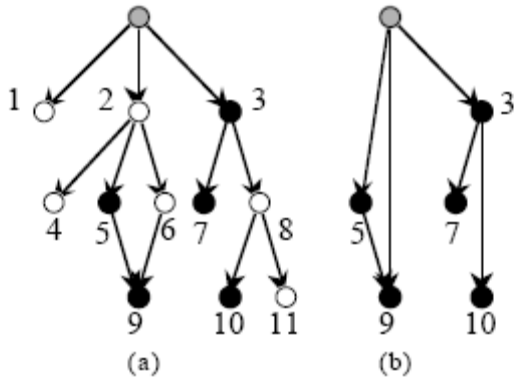


Figure 10. Un arbre (a) et sa contraction sur les sommets gris (b); la racine est en noir [M.Gol, D.Mai, et S.Riz 98].

Définition 6. Soient deux schémas de fait compatibles :

f' = (M', A', N', R', O', S') et f'' = (M'', A'', N'', R'', O'', S''), et donnez l = A' ∩ A''. Les schémas f' et f'' seraient strictement compatibles si cnt (qt (f'), 1) et cnt (qt (f''), 1) sont égaux. Deux schémas compatibles f' et f'' peuvent être concaténés pour créer un schéma résultant f' Θ f'' ; si la compatibilité est stricte, les dépendances inter attributs dans les deux schémas ne sont pas en conflit et f' Θ f '' peut être défini comme suit.

Définition 7. Soient deux schémas strictement compatibles f' et f'', nous définissons le chevauchement (concaténation partielle) de f ' et de f '' comme un schéma f' Θ f'' = (M, A, N, R, O, S) où :

$M = M' \cup M''$

$A = A' \cap A''$

$\forall a_i \in A\ (Dom_{f' \Theta f''}(a_i) = Dom_{f'}(a_i) \cap Dom_{f''}(a_i))$

$N = N' \cap N''$

$R = \{ (a_i, a_j) | (a_i, a_j) \in cnt(qt(f'), A \} = \{ (a_i, a_j) | (a_i, a_j) \in cnt(qt(f''), a) \}$

$O = \{ (a_i, a_j) \in R \mid \exists (a_w, a_z) \in O' | (a_w, a_z) \in path_{ij}(qt(f')) \vee \exists (a_w, a_z) \in path_{ij}(qt(f'')) \}$

$S = \{ (m_j, d_i, \Omega) | d_i \in Dim(f' \Theta f'') \wedge (\exists (m_j, d_i, \Omega) \in S' \wedge d_i \in sub(qt(f'), d_k)) \vee$

$(\exists (m_j, d_i, \Omega) \in S'' \wedge d_i \in sub(qt(f''), d_k)) \}$ [M. Gol, S. Riz, D. Mai 98].

La Figure 11 montre une concaténation entre les deux schémas strictement compatibles *INVENTORY* et *SHIPMENT*, qui partagent la dimension de temps et de produit. Le schéma résultat de la concaténation peut être utilisé, par exemple, pour comparer les quantités embarquées et stockées pour chaque produit.

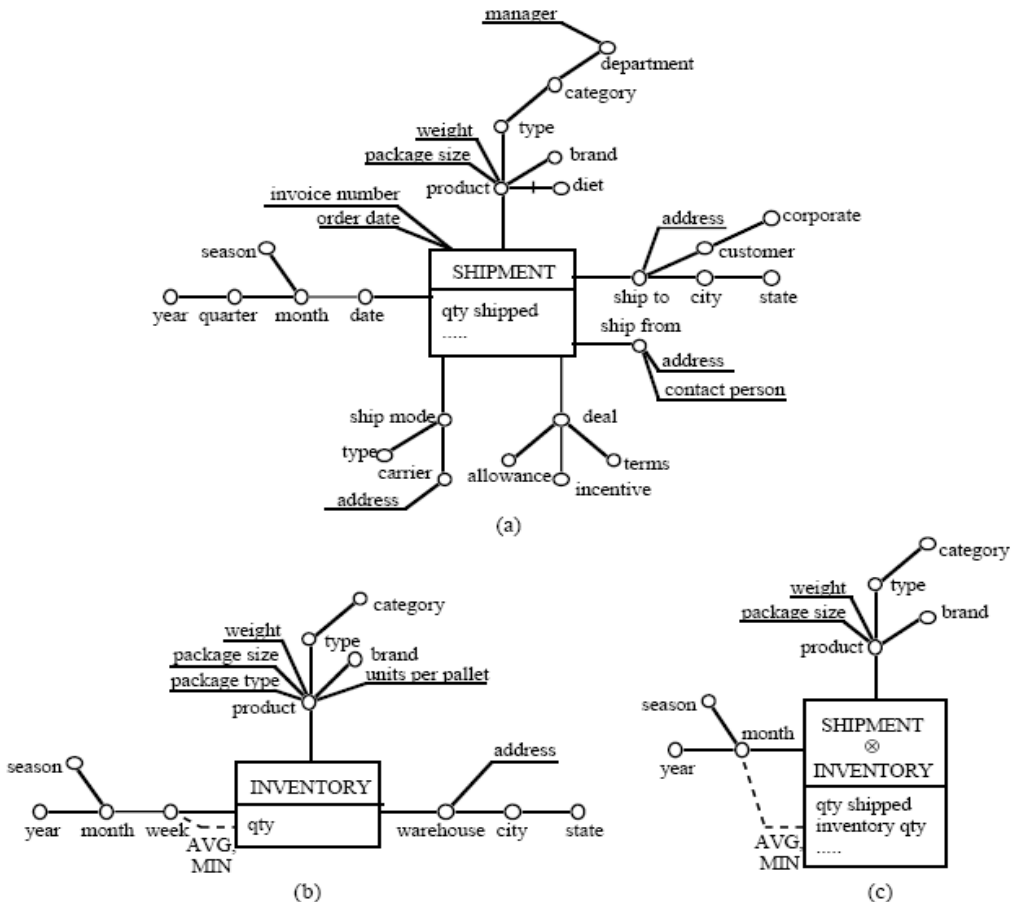


Figure 11. Le Schéma *SHIPMENT* (a), Le Schéma *INVENTORY* (b) et leurs Chevauchement (c) [M.Gol, D.Mai, et S.Riz 98].

III. Présentation de la méthode

La méthode proposée par [M.Gol, S.Riz 99] est caractérisée par les six étapes suivantes :

1. L'analyse de système d'information,
2. La spécification des demandes,
3. La production de schéma conceptuel,
4. Raffinement et validation du schéma dimensionnel,
5. La conception logique,
6. La conception physique.

III. 1. L'analyse de système d'information

Le but de cette étape est de rassembler la documentation concernant le système d'information opérationnel pré existant. Cette étape fait participer les concepteurs de l'entrepôt de données, dont une forte collaboration avec les gens qui gèrent le système d'information ; produit en sortie le schéma (conceptuel ou logique) de la totalité ou d'une partie du système d'information.

Le concepteur en analysant le système opérationnel doit :

- Exploitez l'expérience du gestionnaire de base de données afin de découvrir les données en sortie possible ou les données anormales,
- Sélectionner les données sources opérationnelles en considérant la qualité de données et la stabilité de leurs schémas,
- Déterminez quelles données peuvent être utiles pour l'intégration afin d'obtenir une vue complète du domaine de base de données.

III. 2. La spécification des demandes

Cette étape concerne la collection et le filtrage des demandes des utilisateurs. Elle implique le concepteur et les utilisateurs final de l'entrepôt de données, et produit en sortie les spécifications concernant le choix des faits d'un coté, les indications préliminaires concernant le cahier de charge d'un autre côté.

En particulier, le choix des faits est basé sur la documentation de système d'information produit par l'étape précédente. Le fait est un concept fondamental dans le processus de prise de décision, et correspond typiquement aux événements qui se produisant dynamiquement dans le monde d'entreprise. Si le système d'information opérationnel est doté d'un ou de plusieurs schémas E/R, un fait peut être représenté soit par une entité ou par une relation. En général, les entités ou les relations qui représentent fréquemment les archives de mise à jour sont des bons candidats pour définir des faits.

Le cahier de charge préliminaire est exprimé en langage naturel et il vise à permettre au concepteur d'identifier les dimensions et les mesures durant la conception physique ; pour chaque fait, il devrait indiquer les mesures et les agrégations les plus intéressantes.

III. 3. La production de schéma conceptuel

Malgré son importance le schéma conceptuel est l'un des sujets les moins discutés dans la littérature de DW. Dans [M.Gol, S.Riz 99] une technique (méthodologie) semi-automatique est proposée pour produire le schéma conceptuel commençant respectivement par un schéma E/R et par le modèle logique qui le décrit.

III. 3. 1. Conception de schéma conceptuel à partir du schéma relationnel

La méthodologie proposée pour établir un modèle de dimensionnel de fait à partir de la documentation décrivant la base de données relationnelle opérationnelle comprend les étapes suivantes:

1. Définition des faits.

2. Pour chaque fait :
    a- Construction de l'arbre d'attribut.
    b- Epuration et raffinage de l'arbre d'attribut.
    c- Définition des dimensions.
    d- Définition des mesures.
    e- Définition des hiérarchies.

Cette méthodologie peut être appliquée, avec des différences mineures, démarrant à partir d'un schéma E/R et d'un schéma logique. Dans les sous section suivantes nous allons voir les étapes d'un exemple de VENTE, considérant deux sources alternatives la conception et sa documentation logique. Le schéma E/R simplifié de VENTE est présenté dans Figure 11. Chaque instance de la relation VENTE représente la référence d'un article pour chaque produit dans un billet d'achat. L'attribut UnitPrice (Prix Unitaire) est placé dans VENTE au lieu de PRODUIT puisque le prix de produit peut changer tout le temps. Le schéma logique correspondant est montré ci-dessous (Les clefs primaires sont souligné ; pour chaque clef étrangère). Pour la simplicité, aucun code artificiel n'est présenté pour identifier les schémas relationnels.

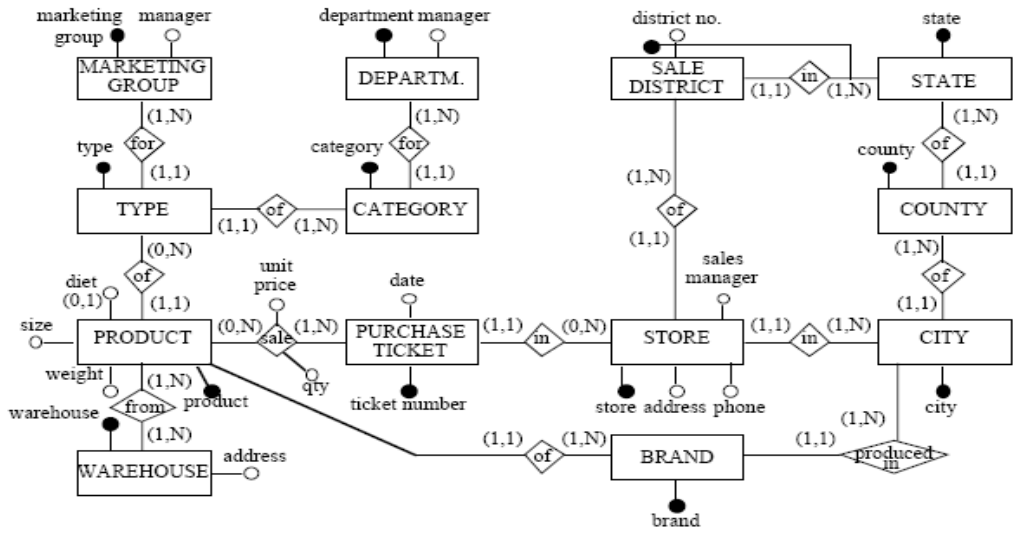


Figure 12. Le schéma E/R simplifier pour un schéma de fait VENTE [M.Gol, et al 98].

III. 3. 1. 1. Définition des faits

Les faits sont des concepts de bases pour le processus décisionnel ; ils correspondent Typiquement aux événements qui se produisant dynamiquement dans le domaine d'entreprise.

Dans le schéma E/R : un fait peut être représenté par une entité F ou par une relation R entre les entités $E_1 \ldots E_n$. Dans le cas précédent, pour la raison de simplicité, il vaut la peine de

transformer R à une entité F en remplaçant chaque branche E_i par une relation binaire R_i entre F et E_i ; si on note respectivement par min(E, R) et max(E, R), le minimum et le maximum cardinalité pour chaque entité E participant dans la relation R, donc :

$$\min(F,R_i) = 1,\ \max(F,R_i) = 1,$$
$$\min(E_i,R_i) = \min(E_i,R),\ \max(E_i,R_i) = \max(E_i,R),\ i=1,...n$$

Les attributs de la relation deviennent des attributs de F; l'identifiant de F c'est la combinaison des identifiants de E_i, i=1…n

Dans le schéma logique : un fait correspond à un schéma EntitéRelation F.

Les entités ou les relations (Schéma EntitéRelation) représentent fréquemment les archives de mis à jour telle que VENTE, sont des bons candidats pour définir des faits. Chaque fait identifié dans un schéma source devient une racine pour des différents schémas de fait. Dans les sous-sections suivantes, la discussion sera concentrée sur un seul fait, celui correspondant à l'entité F. Dans l'exemple de VENTE, le fait de base pour l'analyse commerciale est la vente d'un produit, représenté respectivement dans le schéma E/R et dans le schéma logique, par la relation SALE et par le schéma EntitéRelation SALES. La figure 13 montre comment la relation SALE est transformée en entité.

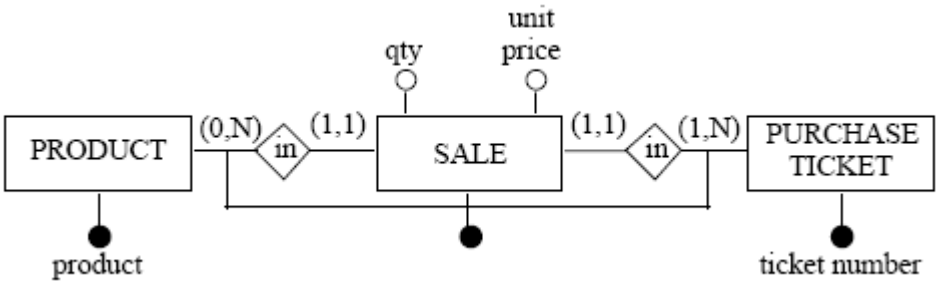


Figure 13. Transformation de la relation VENTE en entité [M.Gol, et al 98].

III. 3. 1. 2. Construction de l'arbre d'attribut

Soit une portion d'un schéma source et d'une entité F appartenant au schéma EntitéRelation ; l'arbre d'attribut s'appelle un sous arbre tel que :

- Chaque sommet correspond à un attribut simple ou appartient à un schéma ;
- La racine correspond à l'identifiant (clef primaire) de F ;
- Pour chaque sommet v, l'attribut correspondant détermine fonctionnellement tous les attributs correspondant aux descendants de v.

Dans le schéma E/R :

Soit Identifiant (E) l'ensemble des attributs qui composent l'identificateur de l'entité E. L'arbre d'attribut de F peut être construit automatiquement par application de la procédure récursive suivante :

```
root=newVertex(identifier(F));
// newVertex(<attributeSet>) renvoie un nouveau sommet composé
// avec la concaténation des noms des attributs dans
// l'ensemble de la procédure translate(F,root);
```

Where

```
translate(E,v):
// E est l'entité courante, v est le sommet courant
{ for each attribute a∈E | a≠identifier(E) do
 addChild(v,newVertex({a})); // adds child a to vertex v
for each entity G connected to E
by a relationship R | max(E,R)=1 do
{ for each attribute b∈R do
addChild(v,newVertex({b}));
next=newVertex(identifier(G));
addChild(v,next);
translate(G,next);
}
}
```

Dans ce qui suit nous allons voir dans un mode pas à pas comment la procédure `translate` génère une branche de l'arbre d'attribut dans l'exemple de VENTE ; l'arbre d'attribut résultant est montré dans la Figure 14.

```
root=newVertex(ticketNumber+product)
translate(E=SALE,v=sale):
   addchild(v,qty); addchild(v,unitPrice);
   for G=PURCHASE TICKET:
       addchild(v,ticketNumber);
       translate(PURCHASE TICKET,ticketNumber);
   for G=PRODUCT:
       addchild(v,product); translate(PRODUCT,product);

translate(E=PURCHASE TICKET,v=ticketNumber):
   addchild(v,date);
   for G=STORE:
       addchild(v,store); translate(STORE,store);

translate(E=STORE,v=store):
  addchild(v,address); addchild(v,phone);
  addchild(v,salesManager);
  for G=SALE DISTRICT:
      addchild(v,districtNo+state);
      translate(SALE DISTRICT,districtNo+state);
```

```
  for G=CITY:
      addchild(v,city); translate(CITY,city);

translate(E=SALE DISTRICT,v=districtNo+state):
  addchild(v,districtNo);
  for G=STATE:
      addchild(v,state); translate(STATE,state);
translate(E=STATE,v=state):
```

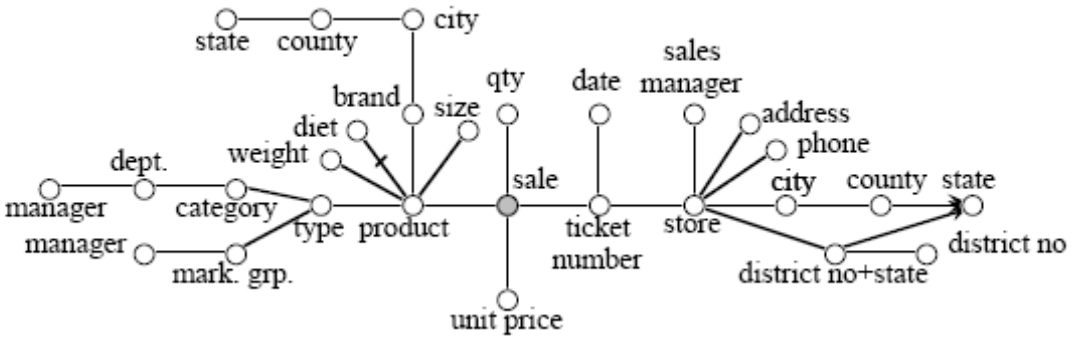


Figure 14. L'arbre d'attribut pour l'exemple de VENTE (racine en gris) [M.Gol, S.Riz 99].

Dans le schéma logique

Soit pk(R) et fk(R, S) deux notation des ensembles des attributs de R formant, respectivement, la clef primaire de R et la clef étrangère notée par S. L'arbre d'attribut pour F peut être construit automatiquement par application de la procédure suivante :

```
root=newVertex(pk(F));
// newVertex(<attributeSet>) renvoie un nouveau sommet composé
// avec la concaténation des noms des attributs dans
// l'ensemble de la procédure translate(F,root);
```

where

```
translate(R,v):
// R est le schéma relationnel courant,
// v est le sommet courant,
{ for each attribute a∈R | (a≠pk(R) ∧(∃/S | a∈fk(R,S)))
addChild(v,newVertex({a})); // adds child a to vertex v
for each attribute set A⊂R | (∃S | A=fk(R,S))
{ next=newVertex(A);
addChild(v,next);
translate(S,next);
}
for each relational scheme T | pk(T)=fk(T,R)
{ for each attribute b∈T
| (b≠pk(R) ∧(∃/S | b∈fk(T,S)))
addChild(v,newVertex({b}));
for each attribute set B⊂T | (∃S≠R | B=fk(T,S))
{ next=newVertex(B);
addChild(v,next);
translate(S,next);
}
```

```
}
}
```

La procédure `translate` construit l'arbre par les dépendances fonctionnelles représentées dans le schéma de base de données. Le premier cycle traite les dépendances entre la clef primaire de R et chaque attribut de R (inclure, si la clef est composée, les attributs simples qui le composent mais exclure ceux appartenant aux clefs étrangères). Le second cycle traite les dépendances entre la clef primaire et chaque clef étrangère qui référence le schéma relationnel S. Le troisième cycle traite la situation :

```
R(kR,...)
T(kT:R,...kS:S)
S(kS,...)
```

Dans chaque relation (1, n) entre R et S est représentée par la troisième forme normale T.

III. 3. 1. 3. Epuration et raffinage de l'arbre d'attribut

Probablement, tous les attributs représentés dans l'arbre d'attribut ne sont pas intéressants pour le DW. Ainsi, l'arbre d'attribut peut être épuré et raffiné afin d'éliminer les niveaux de détail inutiles.

L'épuration est effectuée par élimination (chute) des sous arbres de l'arbre supérieure. Les attributs supprimés ne sont pas inclus dans le schéma de fait, par conséquence il sera impossible de les employer pour agréger les données. Par exemple, dans l'exemple de VENTE, les sous arbre enraciné dans *County* peut être supprimé de la branche *Brand*.

Le raffinage est employé quand, cependant un sommet de l'arbre exprime une information non utile, ses descendants doivent être préservés ; par exemple, on peut classifier des produits directement par catégorie, sans prendre en considération l'information de leurs types. Soit v un sommet a éliminé :

```
graft(v):
{ for each v' | v' is father of v do
      for each v" | v" is child of v do
           addChild(v',v");
  drop v; }
```

Le raffinage est effectué par un déplacement entier du sous arbre enraciné dans v par son parent v' ; si l'arbre d'attribut est noté par t et l'ensemble de ses sommets est noté par I, la procédure `graft(v)` devient cnt (t, I – {v}). Comme résultat, l'attribut v ne sera pas inclus dans le schéma de fait et l'agrégation correspondante au niveau sera perdue ; d'autre part, tous les niveaux des descendants seront maintenus. Dans l'exemple de VENTE, le détail des billets d'achat n'est pas intéressant et le sommet *ticket number* peut être raffiné. En général, le

raffinement d'un enfant (descendant) d'une racine correspond à rendre la granularité du schéma de fait plus brute et, si le nœud raffiné a deux enfants (descendants), ce la mène à augmenter le nombre de dimensions dans le schéma de fait.

III. 3. 1. 4. Définition des dimensions

Les dimensions déterminent comment les instances de fait peuvent être agrégée considérablement pour le processus décisionnel. Les dimensions doivent être choisies dans l'arbre d'attribut parmi les sommets fils de la racine (y compris les attributs qui vont être fils de la racine après le raffinage de l'arbre). Le choix des dimensions est crucial pour la conception de DW puisqu'il détermine la granularité des instances de fait.

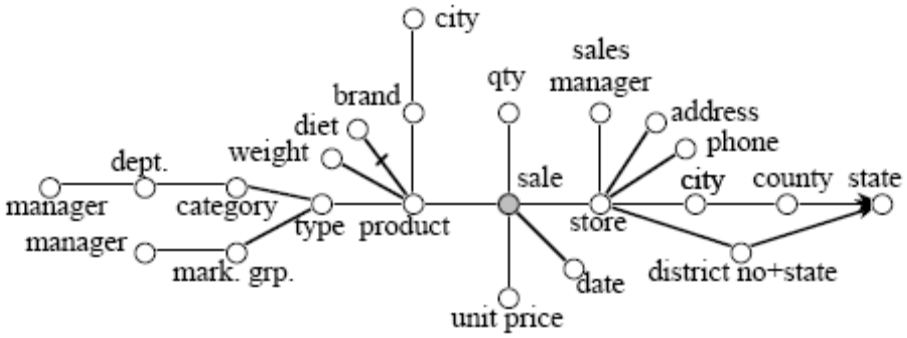


Figure 15. L'arbre d'attribut pour l'exemple de VENTE après épuration et raffinage [M.Gol, S.Riz 99].

Chaque instance primaire de fait " récapitule " toutes les instances de l'entité F correspondant à une combinaison des valeurs de dimension. Si le schéma de dimension inclut tous les attributs qui constituent l'identifiant (clef primaire) de F, chaque instance primaire correspond à une instance (tuple) de F ; souvent, un ou plusieurs attributs qui identifient F sont épurés ou raffinés, ainsi, chaque instance primaire peut correspondre à plusieurs instances (tuples) de F. Dans l'exemple de VENTE, les attributs choisis comme dimensions sont *product*, *store* et *date*. A ce stade, le schéma de fait peut être tracé par l'addition des dimensions choisies au fait de la racine.

III. 3. 1. 5. Définition des mesures

Les mesures sont définies par l'application d'une fonction d'agrégation sur les attributs numérique de l'arbre d'attribut ; cette fonction opère sur toutes les instances (tuples) de F correspondant à chaque instance de fait primaire. La fonction d'agrégation ressemble typiquement aux expressions somme/moyenne/maximum/minimum ou à la fonction *Count* de nombre d'instances (tuples) d'entité. Un fait peut ne pas avoir d'attribut, si la seule information à enregistrer est une occurrence de fait.

Les mesures déterminées sont rapportées dans le schéma de fait. A ce stade, il est utile pour l'étape de conception logique de construire un glossaire qui associe chaque mesure à une expression décrivant comment le schéma source peut être calculé à partir des attributs.

A partir de l'exemple de VENTE et son schéma logique, le glossaire construit et compilé dans SQL est le suivant :

qty sold =
```
SELECT SUM(S.qty)
FROM SALES S,TICKETS T
WHERE S.tickNo = T.tickNo
GROUP BY S.product,T.date,T.store
```

revenue =
```
SELECT SUM(S.qty *S.unitPrice)
FROM SALES S,TICKETS T
WHERE S.tickNo = T.tickNo
GROUP BY S.product,T.date,T.store
```

no. of customers =
```
SELECT COUNT(*)
FROM SALES S,TICKETS T
WHERE S.tickNo = T.tickNo
GROUP BY S.product,T.date,T.store
```

III. 3. 1. 6. Définition des hiérarchies

La dernière étape dans la conception de schéma de fait est la définition des hiérarchies sur les dimensions. Le long de chaque hiérarchie, les attributs doivent être organisées dans l'arbre tel que le lien entre chaque nœud et ces descendants est une relation.

L'arbre d'attribut montre déjà une organisation plausible pour des hiérarchie ; à ce stade, il est encore possible d'épurer et de raffiner l'arbre afin d'éliminer les attributs non pertinents. Il y a également la possibilité d'avoir des nouveaux niveaux d'agrégation en définissant des nouveaux rangs pour les attributs numériques ; comme dans l'exemple de VENTE, la dimension *Temps* est enrichie par l'introduction des attributs *month*, *quarter*, etc.

Pendant cette phase, les attributs qui sont utilisées pour un but informationnelle et non pour l'agrégation, peuvent être définies comme des attributs non-dimensionnelle (par exemple, *adress*, *weight*, etc.). Il est noté que les attributs non-numérique qui sont des fils de la racine mais ne sont pas choisies comme des dimensions doivent nécessairement être soit raffinés (si la granularité des instances primaires de fait est plus grande que celui du fait) ou soit être représentés comme non dimensionnelle (si les deux granularités sont égales).

III. 4. Raffinement et validation du schéma dimensionnel

Cette étape vise à raffiner le cahier de charge préliminaire par la reformulation de ce dernier dans un détail plus profond sur le schéma dimensionnel ; la sous section 4.1 présente un langage simple pour formuler les requête (questions des utilisateurs) selon le DFM. Un autre

aspect significatif est discuté dans la sous section 4.2 concernant le calcul de volume de données prévus.

Cette étape vise également à valider le schéma conceptuel produit dans l'étape précédente.

III. 4.1. Les requêtes

Dans la structure de DFM, une requête typique de DW peut être représentée par l'ensemble d'instances de fait, à n'importe quel niveau d'agrégation, dont les valeurs de mesure doivent être recherchées. Dans cette sous section nous allons voir comment des ensembles d'instances de fait peuvent être exprimées par l'écriture des expressions d'instances de fait ayant en général la forme :

<*fact instance expression*> ::=
<*fact name*> (<*pattern clause*> ; <*selection clause*>)
<*pattern clause*> :: = comma-list of <*pattern elements*>
<*pattern elements*> :: = <*dimension name*> |
<*dimension name*>.<*attribute name*>
<*selection clause*> :: = comma-list of <*predicate*>

La valeur proposée par une mesure dans l'instance de fait décrite par l'expression d'instances de fait est écrite comme le suivant :

<*measure values*> :: = <*fact instance expression*>.<*measure*>

Soit un schéma de fait contient n dimensions $d_1,\ldots d_n$, considéré l'expression d'instances de fait

$$f(d_1,...d_p,a_{p+1},...a_v\ ;\ e_1(b_{i1}),...e_h(b_{ih})) \qquad (1)$$

la première formule d'agrégation p contient une dimension et d'autres v – p contient un attribut dimension. Chaque prédicat booléen e_j contient un attribut b_{ij} appartenant à l'hiérarchie qui a comme racine d_{ij*}.

Si p = v = n ; l'expression (1) représente l'ensemble d'instances primaires de fait

$$\{\ pf(\alpha_1,...\alpha_n) \mid \forall k \in \{1,...n\}\ \alpha_k \in Dom(d_k) \wedge \forall j \in \{1,...h\}\ e_j(\alpha_{ij*}.b_{ij})\ \}$$

Autrement (p < v et /ou au moins une dimension est cachée), soit p un domaine d'agrégation décrit par un domaine de clause. Soit b_{ij} un attribut contenu dans e_j ; e_j est externe si $\exists\ a_{ij*} \in P \mid a_{ij*} \in path_{0ij}\ (qt(f))$, interne autrement. Les attributs externes restreignent l'ensemble d'instances de fait secondaire d'être retourner, où les attributs internes déterminent quelle instance de fait primaire formera chaque instance de fait secondaire. Soit $e_1,\ldots e_r$ et $e_{r+1},\ldots e_h$ être, respectivement, des attributs externes et internes ($0 \leq r \leq h$) ; dans ce cas, l'expression (1) représente l'ensemble d'instances de fait secondaire

$$\{\ sf(\beta_1,...\beta_v) \mid \forall k \in \{1,...v\}\ \beta_k \in Dom(a_k) \wedge \forall j \in \{1,...r\}\ e_j(\beta_{ij*}.b_{ij})\ \}$$

Ou chaque $sf(\beta_1,...\beta_v)$ est agrégat de l'ensemble d'instances de fait primaire

$\{ pf(\alpha_1,...\alpha_n) \mid \forall k \in \{1,...n\}\ \alpha_k \in Dom(d_k) \wedge \forall h \in \{1,...v\}\ \alpha_{h^*}.a_h=\beta_h) \wedge \forall j \in \{r+1,...h\}\ e_j(\alpha_{ij^*}.b_{ij}) \}$

Soit q définie par l'expression d'instances de fait f(P, <sel>) ou $f = f_1 \otimes ... \otimes f_m$. Chaque instance de fait retourné par les m requêtes $q_1 \ldots q_m$, ou $q_i = f_i(P ; <sel>, d_1 \in Dom_f(d_1), \ldots d_n \in Dom_f(d_n)$ et $d_1, \ldots d_n$ sont des dimensions de f.

III. 4. 2. Les volumes de données

Les volumes de données primaires sont calculés pour chaque schéma de fait f par la considération des faits et de cardinalité d'attributs dimension. Soit nk_{ai} un domaine de cardinalité de l'attribut a_i dans f ; le nombre maximum d'instances de fait primaire est

$$cp = \prod_{di \in Dim(f)} nk_{di}$$

La notation np est le nombre réel d'instances de fait primaire, le quel est supposé être connu ; np << cp.

Dans la plupart des DW, le coût d'une requête q est supposé proportionnel au nombre de n-ples dans la vue sur laquelle q est exécuté. Puisque des vues peuvent être matérialisées à n'importe quel niveau d'abstraction, il est nécessaire de calculer le nombre d'instances de fait secondaires correspondant à un modèle d'agrégation P. Le nombre maximum d'instances de fait secondaire correspondant à P est

$$cp = \mathop{X}_{a_i \in P} nk_{ai}$$

Le nombre réel d'instances de fait secondaire, ns(P), peut être calculé en utilisant la formule Yao :

$$ns(P) = cs(P) \times \left(1 - \frac{\left(cp - \frac{cp}{cs(P)} \right)}{\binom{cp}{np}} \right)$$

En utilisant la formule Cardenas ; Quand cp/cs(P) est suffisamment grand, cette formule devient approximativement :

$$ns(P) \approx cs(P) \times \left(1 - \left(1 - \frac{1}{cs(P)} \right)^{np} \right)$$

III. 5. La conception logique

Plusieurs questions doivent être abordées afin d'obtenir une définition correcte du schéma logique de DW. La conception logique reçoit en entrée le schéma dimensionnel, une charge de travail et un ensemble d'information additionnelle (fréquences de mise à jour, espace

disque total disponible, etc.) pour produire un schéma de DW qui devrait réduire au minimum le temps de réponse des demandes en respectant la contrainte d'espace disque. La mise à jours des DWs est faite périodiquement, et pendant ce processus l'entrepôt est indisponible pour répondre aux questions des utilisateurs. Ainsi, le processus de mise à jour n'est pas directement affecté à l'exécution de DW.

Maintenant nous allons voir les étapes à suivre dans la conception logique.

III. 5. 1. Matérialisation des vues

Une technique généralement employée afin de réduire le temps de réponse global est de pré calculer l'information qui peut être utile pour répondre à des questions fréquentes. Des tables de fait rapportant des données consolidées d'autres tables de fait s'appellent souvent les *vues*; chaque vue permet de réduire le coût de l'ensemble des requêtes (questions formulées par les utilisateurs) mais augmente le coût de mise à jours et pose un problème d'occupation d'espace disque.

III. 5. 2. Transformation dans des tables

Pendant cette phase, les tables de fait et de dimension sont créées à partir du schéma dimensionnel et selon le modèle logique adopté. Dans le cas le plus simple, ou le schéma en étoile est adopté, chaque schéma de fait f = (M, A, N, R, O, S) ayant Dim(f) = $\{d_1,...d_n\}$ et
M = $\{m_1,...m_z\}$ est transformer en une seule table de fait

$FT\_f(k_1,...k_n, m_1,...m_z)$

et en n tables dimension

$DT\_d_1(k_1, a_{11},...a_1v_1, a'_{11},...a'_1u_1)$

.....

$DT\_d_n(k_n, a_{n1},...a_nv_n, a'_{n1},...a'_nu_n)$

III. 5. 3. Division verticale des tables de fait

Les schémas de fait incluent habituellement plusieurs mesures qui décrivent le même fait. La division verticale vise à réduire le temps de réponse global de requête par l'optimisation des requêtes qui exigent un sous-ensemble de mesures. Soit la table de fait $FT\_f(k_1,...k_n, m_1,...m_z)$, la division verticale est effectué par la définition d'une division de l'ensemble de mesures M = $\{m_1,...m_z\}$ dans les sous ensembles α ($\alpha \geq 2$) et par dédoublement en conséquence FT_f dans α tables chacune contient la clef complète $k_1,...k_n$ et un des sous-ensembles de mesure.

III. 5. 4. Division horizontale des tables de fait

La division horizontale vise à réduire le temps de réponse des questions (requêtes) prenant en considération la sélectivité de chaque question. La plupart des questions n'accéderont pas à tous les n-ples dans la table de fait, mais seulement un sous ensemble déterminé par la sélection de prédicat contenant un ou plusieurs attributs dimension. Soit la table de fait `FT_f`, la division horizontale est effectuée par la détermination d'un ensemble optimal d'attributs dimension et par redistribution de n-ples dans `FT_f` à un ensemble de tables FT_f_1,...FT_f_β chacun ayant le même schéma relationnel comme `FT_f` et associé à un élément donné du produit Cartésien entre le domaine d'attributs dimension qui contient.

III. 6. La conception physique

La conception physique concerne la sélection d'index optimal. La sélection d'index a un rôle crucial pour déterminer les performances de l'entrepôt de données. La maintenance des indexes pendant la mise à jours n'est pas nécessaire, et l'obtention d'accès à plusieurs structures complexes est possible.

L'étape de sélection d'index vise à déterminer un meilleur sous ensemble d'index pour chaque type d'index, par rapport à la fonction de coût. Un meilleur sous ensemble est celui qui réduit au minimum le coût d'accès des requêtes sous une contrainte d'espace variable d'une application à une autre. Puisque les requêtes exigent habituellement une ou plusieurs jointure pour s'exécuter, la sélection d'index prend en considération différent algorithmes de jointure.

IV. Conclusion

Le long de ce chapitre nous avons vu la proposition des auteurs [M. Gol, S. Riz 99] d'un modèle conceptuel pour la conception des entrepôts de données et une méthodologie semi automatique pour dériver la documentation décrivant le système d'information de l'entreprise. Ce travail est exprimé dans un cadre méthodologique général pour la conception de DW, basé sur le modèle dimensionnel de fait (DFM) et divisé en six étapes principales.

I. Introduction

Dans ce chapitre nous allons voir une méthodologie de conception des entrepôts et magasins de données proposée par Daniel L. Moody et Mark A.R. Kortink ; les deux auteurs proposent de partir d'un modèle existant, le modèle d'entreprise, puis de le transformer en modèle dimensionnel. La première section est une présentation du modèle de données utilisé, alors que la deuxième section est une présentation de la méthode.

II. Le Modèle de Données Utilisé

Les auteurs [Kortink, al 99] proposent de partir d'un modèle existant le modèle de l'entreprise, puis le transformer en modèle dimensionnel. La modélisation d'un entrepôt de données est une étape cruciale car il faut concevoir un modèle qui permettra d'historiser des données et de répondre à des questions que les décideurs se posent. Dans cette méthode le modèle de données utilisé est le modèle dimensionnel. Nous avons déjà passé en revue dans le chapitre état de l'art la définition, les avantages et concepts de bases du modèle dimensionnel.

Dans le modèle dimensionnel chaque donnée est modélisée en n dimensions et l'on va pouvoir extraire les informations par ligne, par colonne ou par tranche.

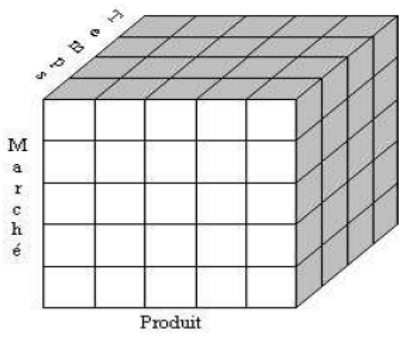


Figure 16. Schéma du cube dimensionnel

Le modèle dimensionnel comporte une table de **faits** et plusieurs tables de dimensions. Les tables de dimension sont de petites tables contenant un minimum de champs, décrivant chacune un axe de l'activité. Ainsi, la table de faits est la table centrale qui va contenir tous les enregistrements qui seront analysés par les utilisateurs.

Il existe plusieurs modèles dimensionnels :

1. Le schéma en étoile (Star)
2. Le schéma en flocon de neige (Snow Flacke)
3. Le schéma en grappe (Cluster)
4. Le schéma en constellation (Constellation)

Deux modèles dimensionnels supplémentaires

Le schéma à plat :

Ce schéma est le plus simple à réaliser sans perte de données. Il est formé par agrégation des entités dans le modèle de données vers une entité minimale. Cela minimise le nombre de tables dans la base de données, mais sans perdre aucune information du modèle original, comme le montre la figure 17.

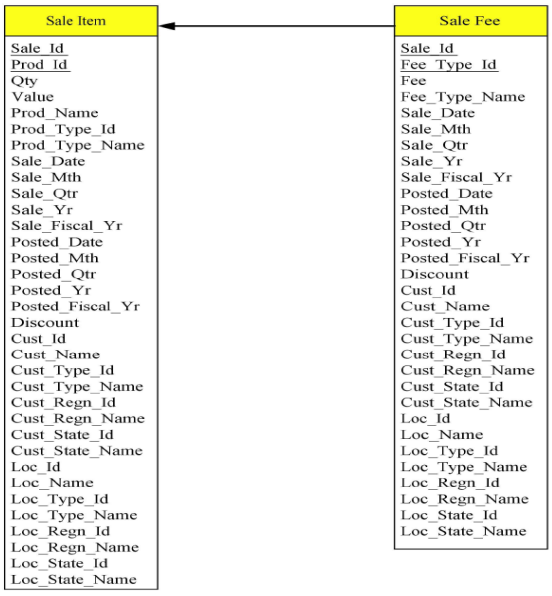


Figure 17. Le schéma à plat.

Le problème de ce type de schéma est qu'il peut engendrer des erreurs lorsqu'il existe une relation entre les entités de transaction. Lorsque l'on agrège les montants numériques d'une entité de transaction vers une autre, cet agrégat est alors répété. Dans l'exemple de la figure16, si une vente (Sale) contient trois articles (Sale Item), le montant des ventes sera répété dans trois enregistrements différents dans la table Sale Item. Ainsi, ces ajouts engendrent deux, voire trois, comptages. Un autre problème de ce schéma est qu'il propose un grand nombre d'attributs dans les tables, ce qui ne facilite pas la lecture et la compréhension. Lorsque le nombre de tables est minimal, la complexité de chaque table s'accroît.

Le schéma en terrasse :

Ce schéma est créé à partir d'une agrégation d'entités en dessous des entités maximales, on s'arrête lorsque l'on rencontre une entité de transaction. Le résultat est une table pour chaque entité de transaction dans le modèle de données. Ce schéma n'est pas très apprécié car il peut poser des problèmes de compréhension aux utilisateurs non avertis, dus à la séparation entre les niveaux de transaction.

Figure 18. Le schéma en terrasse.

Tous ces modèles sont des modèles dimensionnels, que l'on peut ou non implémenter lors de la création d'un entrepôt de données. Cependant, il faut savoir que ces modèles augmentent soit la complexité des tables, soit la redondance des données. Chaque cas est unique, et il faut bien réfléchir au schéma que l'on va adopter. La figure 19 illustre le classement des modèles présentés en fonction de la complexité ou de la redondance des données.

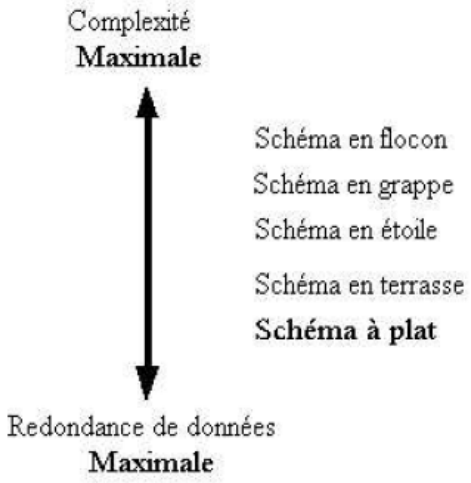


Figure 19. Complexité Vs redondance

Nous allons maintenant voir une méthodologie de conception d'un entrepôt de données.

III. Présentation de la méthode

Une méthode de construction : du modèle relationnel vers le modèle dimensionnel

D'après [Kortink, al 99], on peut arriver à un modèle dimensionnel en partant du modèle d'entreprise s'il existe. En effet, Kimball [Kimball 96] a introduit la notion d'entrepôt de données, mais sans donner de méthodologie de construction, mis à part qu'il faut une base de données dénormalisée et qu'il est prudent d'oublier tout ce qu'on a appris en modélisation de base de données.

Mark A.R. Kortink et Daniel L. Moody [Kortink, al 99] préfèrent rassurer les concepteurs d'entrepôts et les décideurs, en donnant une méthode de conception basée sur un modèle existant. Ils proposent de définir le modèle de l'entrepôt de données à partir du modèle d'entreprise, dont la mise en place est en général très onéreuse.

Dans cette partie du chapitre nous allons voir en détail une méthodologie de construction du modèle dimensionnel à partir du modèle de données de l'entreprise. Cette méthode est définie par les quatre étapes suivantes :

1. Classification des entités.
2. Identification des hiérarchies.
3. La production du modèle dimensionnel.
4. Validation et Raffinement.

III. 1. Classification des entités

La première étape pour créer un modèle dimensionnel à partir du modèle d'entreprise est de classer ses entités. Elles se décomposent en trois catégories :

Entité de transaction :

Elles enregistrent les détails d'événements particuliers comme les salaires, les réservations d'hôtels. Ce sont ces événements, entre autres, que les décideurs veulent analyser. Les caractéristiques des entités de transaction sont :

1. Elles décrivent un événement qui se produit à un instant donné.
2. Elles contiennent les mesures, comme le montant en devise, le poids, les volumes. Ces mesures forment la base des résultats que l'entrepôt permet d'étudier.

Les entités de transaction forment le noyau à partir duquel la table des faits d'un schéma en étoile peut être créée. Toutes les entités de transaction ne sont pas bonnes pour l'aide à la décision, et l'on doit choisir et identifier les plus intéressantes.

Entité Composante :

Elle est directement liée à l'entité de transaction par une relation multiple. Ces entités définissent la finesse des détails ou composants de chaque transaction. Les composants

répondent aux questions *qui*, *quoi*, *quand*, *où*, *combien* et *pourquoi*, d'un événement commercial. Les entités composantes forment la base des tables de dimension dans les schémas en étoile. Chaque table de dimension correspond à une ou plusieurs entités composantes.

Entité de classification :

Ce sont des entités qui sont apparentées à des entités composantes par une chaîne de relations multiples. Ces entités représentent la hiérarchie entre les entités dans le schéma en étoile. La figure suivante montre les hiérarchies.

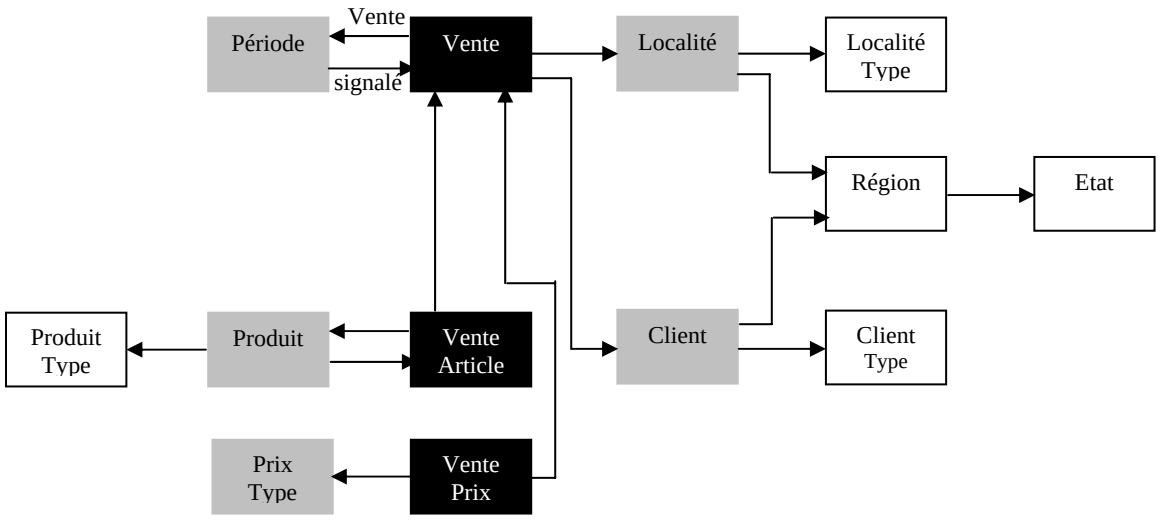


Figure 20. Classification des entités.

Les entités en noir représentent les entités de transaction.

Les entités en gris représentent les entités composantes.

Les entités en blanc représentent les entités de classification.

Ambiguïtés :

Parfois, des entités sont présentes dans plusieurs catégories. Pour résoudre cette ambiguïtés les auteurs [Kortink, al 99] définissent une priorité des hiérarchies comme suit :

1. L'entité de transaction : plus haute.
2. L'entité de classification.
3. L'entité composante : plus basse.

III. 2. Identification des hiérarchies

La hiérarchie est un concept extrêmement important dans le modèle dimensionnel, car le passage du modèle relationnel vers le modèle dimensionnel se fait par la définition des

hiérarchies. Une hiérarchie est un modèle d'entité-association qui est identifié par des séquences d'entités reliées par des relations multiples, toutes alignées dans le même sens. Nous parlons alors d'un schéma normalisé. La figure 21 ci-dessous illustre un exemple d'hiérarchie.

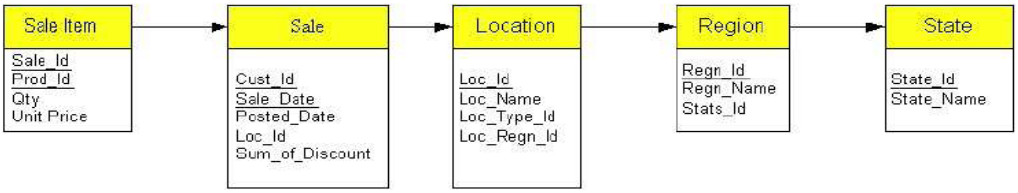


Figure 21. Exemple d'hiérarchie.

On peut donc voir que State est un parent de Region. Region est le fils de State. Sale Item, Sale, Location et Region sont des descendants de State. Sale, Location, Region et State sont les ancêtres de Sale Item. Ce sont des termes que l'on retrouve dans les langages de programmation objet, avec la notion d'héritage.

Hiérarchie maximale :

Une hiérarchie est dite maximale si l'on ne peut l'étendre vers le haut ou vers le bas en ajoutant de nouvelles entités. Une entité est dite minimale si elle est placée au bas d'une hiérarchie maximale. Une entité est dite maximale si elle est située au plus haut de la hiérarchie. Dans l'exemple de la figure 20, il y a deux entités minimales, Sale Item et Sale Fee, alors qu'il y a six entités maximales : Period, Customer Type, State, Location Type, Product Type et Fee Type. Les entités minimales sont facilement identifiables car ce sont des entités qui n'ont pas de relations multiples. A l'inverse, les entités maximales se décrivent par des entités qui n'ont pas de relations monovaluées.

III. 3. Production du Modèle Dimensionnel

Les auteurs [Kortink, al 99] utilisent deux opérateurs pour produire le modèle dimensionnel à partir du modèle Entité Relation :

Opérateur 1 : *Réduction de la hiérarchie*

Des entités de rang supérieur peuvent être concaténées à des entités de rang inférieur en respectant la hiérarchie. Ainsi, l'attribut State_Name peut être mis dans la table Region. Cela introduit une redondance et forme une dépendance transitive, ce qui viole la troisième forme normale de Codd écrite en 1970. On introduit alors le concept de base dénormalisée, terme désigné par Tonkin en 1991.On peut aller plus loin, en concaténant les attributs

Region_Name, State_Id et State_Name à l'entité Location. Il faut alors continuer dans le même sens pour se retrouver avec une seule table, la table Sale_Item qui contient la concaténation de tous les attributs des autres tables.

Opérateur 2 : *L'agrégation ; une fonction des entités de transaction*

Une agrégation est peut être appliquée à une entité de transaction pour former une nouvelle entité de transaction, constituée d'une synthèse de données. Cette fonction n'est possible qu'avec des attributs numériques, afin de faire ressortir des champs calculés, par exemple. La clé de cette nouvelle entité est une combinaison des attributs utilisés par l'agrégation. Attention, l'agrégation perd des informations, et l'on ne peut reconstruire les détails de la table Sale Item depuis la table Product Summary, comme le montre la figure 22 ci-dessous.

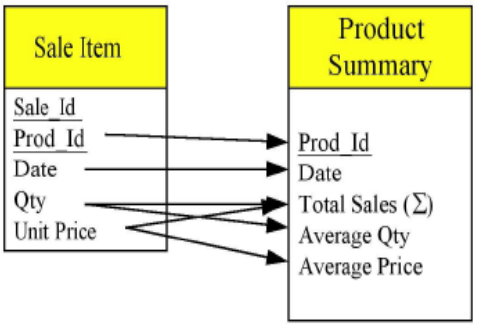


Figure 22. Agrégation d'entité

Les auteurs [Kortink, al 99] proposent une méthodologie pour construire un modèle dimensionnel. Les schémas cibles sont décrits en état de l'art de ce mémoire et sont rappelés en première section de ce chapitre, à savoir : le schéma en étoile, le flocon de neige, la constellation, la galaxie, le schéma en grappe, le schéma à plat et le schéma en terrasse.

Quelque soit le modèle choisi, la procédure de création de la table des faits et la création de ses clés est :

1. Une table des faits est formée pour les entités de transaction les plus pertinentes. La clé de cette table est une combinaison des clés des tables de dimension.
2. Si une relation hiérarchique existe entre deux entités de transaction, l'entité fille hérite de tous les attributs de son père.
3. Les attributs numériques appartenant aux entités de transaction doivent êtres agrégés par les clés.
4. Les tables des faits avec les mêmes clés primaires doivent être fusionnées.

De plus, pour chaque schéma, il faut rajouter certaines spécificités propres à chacun. Par exemple :

1. Dans le cas d'un modèle en étoile : une table de dimension est formée pour chaque entité composante par agrégation de la hiérarchie constituée de ses entités de classification.
2. Pour le modèle en flocon de neige : chaque entité composante devient une table de dimension. A partir de chaque entité composante et ses entités de classification on dérive la hiérarchie d'une dimension.

III. 4. Validation et Raffinement

Dans la pratique, la modélisation dimensionnelle est un processus itératif. La procédure présentée dans l'étape précédente est utile pour produire une conception primaire, mais tous ça doit être raffinés pour produire une conception finale des magasins de données.

III. 4. 1. Combinaison des tables de fait

Les tables de fait doivent être combinées avec les même clefs primaires (i.e. même dimensions). Ceci réduit le nombre de schémas en étoile et facilite la comparaison entre les faits relatifs.

III. 4. 2. Combinaison des tables de dimension

Souvent la création des tables de dimension pour chaque entité composante a comme conséquence un grand nombre de tables de dimension. Pour simplifier la structure des magasins de données, des dimensions relatives devraient être consolidées toutes dans une simple table de dimension.

III. 4. 3. Traitement des relations multiples (n, n)

La plupart des complexités qui surgissent lors de passage du modèle relationnel vers le modèle dimensionnel ; résultent des relations multiples (n, n) ou des entités d'intersection. Les relations multiples causent des problèmes lors de la modélisation dimensionnelle car elles représentent des coupures dans la chaîne hiérarchique, et ne peuvent pas être cachées. Il y a un certain nombre de critères pour traiter des relations multiples:

1. Ignorer l'entité d'intersection,
2. Convertir la relation multiple en une relation un à plusieurs (1, n), par définition de la relation primaire,
3. Inclure les relations multiples dans les magasins de données sachant que les entités peuvent être utiles aux analyseurs mais ne sont pas favorables à l'analyse d'aide à la décision.

III. 4. 4. Transformation des relations SuperTypes

Les relations supertype peuvent être converties en structure hiérarchique par déplacement de supertype et création d'entité de classification pour distinguer entre les supertypes. Ceci peut alors être converti en modèle dimensionnel d'une façon directe.

IV. Conclusion

Dans ce chapitre, nous avons passé en revue une méthode de conception des entrepôts et magasins de données à partir de modèle d'entreprise. Les étapes principales de cette méthode de conception sont :

1. Développer le modèle d'entreprise (s'il n'existe pas).
2. Classifier les entités dans le modèle de données avec un certain nombre de catégories.
3. Identifier les hiérarchies qui existent dans le modèle
4. Production du Modèle Dimensionnel.
5. Validation et raffinement.

I. Introduction

Dans ce chapitre, nous abordons une méthode de modélisation des entrepôts et magasins de données proposée par Frank Ravat, Olivier Teste et Gilles Zurfluh. Ces auteurs exigent que ; la conception d'un système décisionnel doit passer par la séparation de l'entrepôt de données et des magasins de données. Ces différences de fonction et d'objectif se répercutent dans la modélisation de ces deux espaces de stockage. Pour la partie entrepôt les auteurs [Ravat, Teste, et Zurfluh 00] proposent un modèle permettant de décrire l'entrepôt comme un référentiel centralisé de données complexes, temporelles et extraites d'une source d'information. Ce modèle intègre trois concepts : l'objet entrepôt, la classe entrepôt et l'environnement. En suite, les auteurs [Ravat, Teste, et Zurfluh 00] définissent un processus d'élaboration d'entrepôt à partir d'une source globale. Les magasins de données sont dédiés aux analyses décisionnelles de type OLAP. La modélisation multidimensionnelle est utilisée à ce niveau puisqu'elle s'avère adaptée à ce type d'activité [Codd 94] [Kimball 96].

II. Contribution des auteurs [Ravat, Teste, et Zurfluh 00]

Comme il est déjà inscrit dans l'introduction, la conception d'un système décisionnel doit être basée sur la séparation de l'entrepôt et des magasins de données. En effet, l'objectif de ces deux espaces de stockage est différent et les problèmes à résoudre divergent : l'entrepôt regroupe toute l'information décisionnelle tandis que les magasins contiennent une partie de cette information, dédiée à un thème, un métier, une analyse.

L'**entrepôt** est le lieu de stockage centralisé d'un extrait des sources. Il intègre et « *historise* » les données utiles pour la décision. Son organisation doit faciliter la gestion efficace des données et la conservation des évolutions [Test 00b].

Le **magasin** est un extrait de l'entrepôt. Les données extraites sont adaptées à une classe de décideurs ou à un usage particulier. L'organisation des données doit suivre un modèle spécifique qui facilite les traitements décisionnels [Test 00b].

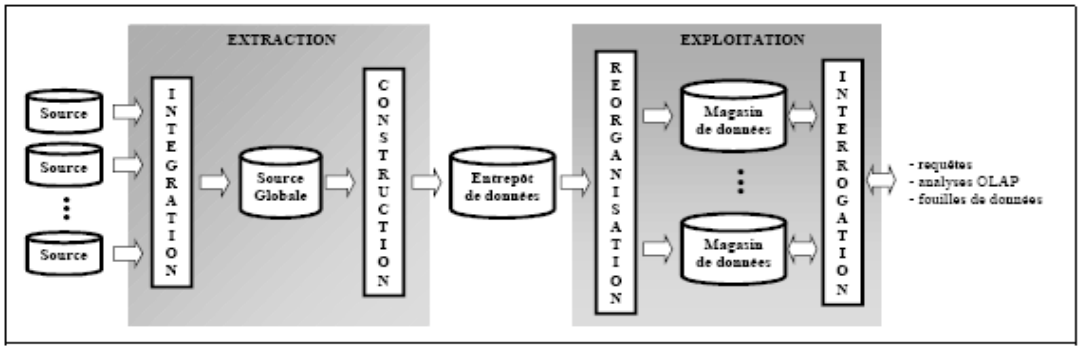


Figure 23. Architecture des systèmes d'aide à la décision.

En s'appuyant sur cette dichotomie, les auteurs [Ravat, Teste, et Zurfluh 00] définissent l'architecture des systèmes décisionnels décrite dans la figure 23. Cette architecture fonctionnelle permet d'identifier les problématiques de recherche [Test 00b].

– L'**intégration** se propose de résoudre les problèmes d'hétérogénéité des différentes sources de données en intégrant celles-ci dans une source globale virtuelle (les données restent stockées dans les sources et sont extraites au moment des mises à jour de l'entrepôt). La source globale est décrite par le modèle de données standard de l'ODMG [Cattel 95] ; le choix du paradigme objet se justifie car il s'avère parfaitement adapté pour l'intégration de sources hétérogènes [Bukhres, Elmagarmid 93] couramment utilisées dans le milieu médical [Pedersen, Jensen 98].

– La **construction** consiste à extraire les données pertinentes pour la prise de décision, puis à les recopier dans l'entrepôt de données, tout en conservant, le cas échéant, les changements d'états des données. Le modèle de l'entrepôt doit supporter des structures complexes [Pedersen, Jensen 98] et supporter l'évolution des données au cours du temps [Mendelzon, Vaisman 00].

– La **réorganisation** permet de restructurer les données dans des magasins ; la réorganisation des données vise à supporter efficacement les processus d'interrogation et d'analyse tels que les applications OLAP et la fouille de données (« *data mining* »). Pour ce faire, les données des magasins sont organisées de manière multidimensionnelle [Kimball 96].

– L'**interrogation** consiste à manipuler les données afin d'analyser les tendances passées pour prendre des décisions. Les données sont représentées sous une forme qui facilite leur compréhension et leur manipulation par les décideurs non informaticiens (tableaux, graphiques,…).

Notre étude se focalise sur le second et le troisième niveau de ce système et aborde le problème de la conception de l'entrepôt, ainsi le problème de conception des magasins de données. Dans ce qui suit nous allons voir deux modèles de données un pour l'entrepôt, et l'autre pour les magasins de données. Puis nous allons étudie : un processus d'extraction mis en jeu dans l'élaboration des structures de données de l'entrepôt ainsi que des comportements de ces données et une démarche de transformation des données dans des magasins dimensionnels.

III. Modèle de données utilisé

III. 1. Modèle de données de l'entrepôt

Dans cette section, nous allons voir un modèle de données pour les entrepôts, basé sur le paradigme objet. Ce modèle proposé par [Ravat, Teste, et Zurfluh 00] subit l'influence du

modèle objet standard de l'ODMG [Cattel 95] qui est étendu pour prendre en compte les caractéristiques des entrepôts de données. Notamment, ce modèle intègre la dimension temporelle d'une manière flexible en permettant l'archivage des données temporelles.

III. 1. 2. Concept d'objet entrepôt

Chaque information extraite (objet, partie ou groupe d'objets source) est représentée dans l'entrepôt par un **objet entrepôt** qui conserve ses évolutions de valeur au cours du temps (tandis que la source de données ne contient que l'état courant [Chaudhuri, Dayal 97], ou bien, ne conserve qu'une partie récente des évolutions, insuffisante pour la prise de décision [Yang, Widom 00]). Dans un entrepôt, l'administrateur peut décider de conserver :

- l'image de l'information extraite c'est-à-dire l'**état courant**, ainsi que
- les états successifs que prend au cours du temps l'information extraite, c'est-à-dire ses **états passés**,
- uniquement un résumé des états passés successifs, c'est-à-dire l'agrégation de certains états passés, appelée **état archivé**. Les états passés ainsi résumés sont supprimés de l'entrepôt afin de limiter l'accroissement du volume des données.

Un **objet entrepôt** est donc défini par le quadruplet (*oid*, S_0, *EP*, *EA*) où *oid* est l'identifiant interne, S_0 est l'état courant, $EP = \{S_{p1}, S_{p2},\ldots, S_{pn}\}$ est un ensemble fini contenant les états passés et $EA = \{S_{a1}, S_{a2},\ldots, S_{am}\}$ est un ensemble fini contenant les états archivés.

Un **état** S_i d'un objet entrepôt est défini par le couple (v_i, h_i) où v_i est la valeur de l'objet pour les instants de h_i et $h_i = < [td^1, tf^1[;\ldots;[td^h, tf^h[>$ est le domaine temporel (ensemble ordonné d'intervalles disjoints deux à deux) définissant les instants durant lesquels la valeur de l'état S_i est courante.

La modélisation des domaines temporels s'effectue au travers d'un modèle temporel, linéaire, discret qui définit le temps par le biais d'unités temporelles ; l'espace continu du temps, représenté par une droite de réels, elle-même décomposée en une suite d'intervalles consécutifs disjoints [Fauv 99]. Chaque partition correspond à une unité temporelle caractérisée par la taille des intervalles décomposant la droite du temps. Ce modèle gère un ensemble d'unités temporelles nommées (*année*, *semestre*, *trimestre*,...) muni d'une relation d'ordre partiel *est-plus-fine* permettant de comparer les unités. Les auteurs [Ravat, Teste, et Zurfluh 2000] définissent plusieurs types temporels de base : l'instant, l'intervalle ainsi que le **domaine temporel**. Ce dernier est un ensemble ordonné d'intervalles disjoints deux à deux et non contigus, noté $h_i = <[td^1, tf^1[; [td^2, tf^2 [;\ldots;[td^h, tf^h[>$ où chaque intervalle est non vide ($\forall k \in [1..h]$, $td^k < tf^k$) et possède une même unité temporelle ($\forall k \in [1..h]$, $\forall j \in [1..h]$, $unit([td^k, tf^k[) = unit([td^j, tf^j[$ où la fonction *unit*(*Int*) retourne l'unité temporelle de *Int*).

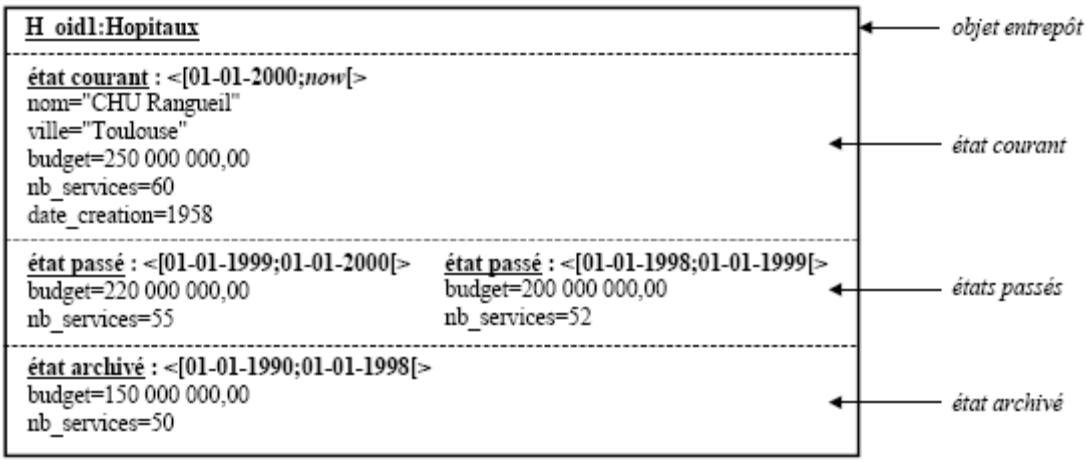


Figure 24. Représentation graphique d'un objet entrepôt [Test 00].

III. 1. 2. Concept de classe entrepôt

III. 1. 2. 1. Définition

Les objets entrepôt qui ont la même structure et le même comportement, sont regroupés dans une même classe ; **classe entrepôt** *c* caractérisé par le septuplet :

- du nom de la classe Nom^c,

- d'un type $Type^c$ définissant la structure $Structure^c$ et le comportement $Comportement^c$ des objets entrepôt de *c* (à chaque classe entrepôt correspond un type),

- d'un ensemble fini de super classes $Super^c$ (c_i est une super classe de *c*, notée $c \leq c_i$ si et seulement si,

$Type^c \supseteq Type^{ci}$ et $Extension^c \subseteq Extension^{ci}$),

- d'une extension $Extension^c = \{o_1, o_2, \ldots, o_x\}$,

- d'une **fonction de construction** $Mapping^c$ qui permet de spécifier le processus d'extraction et de transformation mis en jeu pour créer la structure et le peuplement de la classe *c* à partir de la source globale,

- d'un **filtre temporel** $Tempo^c$ définissant l'ensemble des propriétés temporelles de *c* (une propriété est temporelle lorsque ses évolutions sont conservées par des états passés). Le filtre temporel caractérise la structure des états passés des objets de la classe.

- d'un **filtre d'archives** $Archi^c$ définissant l'ensemble des propriétés archivées de *c* (une propriété est archivée lorsque ses évolutions passées sont résumées dans des états archivés). Le filtre d'archives caractérise la structure des états archivés des objets de la classe.

III. 1. 2. 2. Mécanisme d'archivage

Le **filtre d'archives** $Archci^c = \{(a_1, f_1), (a_2, f_2), \ldots, (a_s, f_s)\}$ caractérise les propriétés archivées de la classe entrepôt. Il est constitué d'un ensemble de couples (a_j, f_j) où *aj* est un attribut et f_j est une fonction d'agrégation. L'ensemble des attributs archivés est un sous-ensemble des

attributs temporels. Chaque attribut archivé est associé à une fonction d'agrégation qui définit la manière dont sont résumées les valeurs temporelles.

Les propriétés archivées sont associées à une fonction d'agrégation qui indique comment sont résumées les évolutions détaillées de la propriété temporelle correspondante. Ce modèle supporte plusieurs catégories de fonctions d'agrégation :

- les fonctions d'agrégation forte (*avg, sum, count, max, min*) résument les états passés sélectionnés pour l'archivage dans un seul état archivé ;
- les fonctions d'agrégation modérée (*avg_t, sum_t, count_t, max_t, min_t*) résument les états passés sélectionnés pour l'archivage avec plusieurs états archivés. Les états passés sélectionnés sont regroupés par grain de temps à une unité temporelle supérieure.

III. 1. 2. 3. Mécanisme d'historisation

Le **filtre temporel** $Tempo^c = \{(p_1, f_1), (p_2, f_2), \ldots, (p_t, f_t)\}$ caractérise les propriétés temporelles d'une classe entrepôt. Il est constitué d'un ensemble de couples (p_j, f_j) où

- p_j est une propriété temporelle et
- fj est soit un attribut, soit une relation, soit une opération retournant un résultat (fonction).

Les évolutions détaillées des propriétés temporelles sont conservées au travers d'états passés. S'il s'agit d'une opération, les évolutions de son résultat sont conservées à chaque point d'extraction (rafraîchissement de la classe).

III. 1. 3. Environnement

Pour supporter efficacement les processus d'analyse décisionnelle, l'entrepôt de données doit être muni d'un mécanisme permettant de définir les parties temporelles, dont les évolutions de valeur seront conservées. En effet, les filtres associés aux classes entrepôt caractérisent comment sont résumés les évolutions de valeur des objets entrepôt, mais il est nécessaire de définir dans l'entrepôt les parties ayant un comportement temporel homogène (période de rafraîchissement, critères d'archivage,…) Pour cela, les auteurs de cette approche proposent le concept d'**environnement** pour définir ces parties temporelles cohérentes. Un **environnement** Env^i est défini par un triplet constitué d'un nom Nom^{Envi}, d'un ensemble fini de classes de l'entrepôt $C^{Envi} = \{c^{Envi}_1, c^{Envi}_2, \ldots, c^{Envi}_{ni}\}$ et d'un ensemble de règles de configuration $Config^{Envi}$ visant à définir le comportement temporel de l'environnement. Un environnement constitue donc une partie temporellement homogène dans l'entrepôt, ayant ses propres configurations locales $Config^{Envi}$. Ce concept d'environnement aide l'administrateur à définir

différentes parties temporelles dans l'entrepôt. Ceci permet de concevoir un entrepôt flexible qui s'adapte aux différentes exigences des décideurs.

III. 1. 4. Schéma de l'entrepôt

Un entrepôt se caractérise par son **schéma** S^{ED} défini par un nom Nom^{ED}, l'ensemble fini des classes de l'entrepôt $C^{ED} = \{c_1, c_2,\ldots, c_n\}$, l'ensemble fini des environnements $Env^{ED} = \{Env_1, Env_2,\ldots, Env_{ne}\}$ et un ensemble de règles de configuration $Config^{ED}$, visant à définir les différents paramètres de configuration globale de l'entrepôt (période de rafraîchissement,…).

Dans la section suivante nous allons voir le modèle de données pour la partie magasins.

III. 2. Modèle de données des magasins

La proposition des auteurs [Ravat, Teste, et Zurfluh 00] consiste à modéliser les magasins de données à un niveau d'abstraction élevé. Le modèle dimensionnel proposé par [Ravat, Teste, et Zurfluh 00] constitue une généralisation des modèles dimensionnels classiques [Agr 95] [Kimball 96]. Dans ce sens, les auteurs [Ravat, Teste, et Zurfluh 00] définissent un modèle conceptuel.

- Les magasins de données sont modélisés au travers d'un schéma dimensionnel qui intègre les concepts de faits et de dimensions ; en effet, la modélisation dimensionnelle représente l'information de manière adaptée aux analyses OLAP.
- Tandis que les modèles existants se basent sur des schémas en étoile comportant un unique fait, la modélisation proposée ici permet de partager chaque dimension entre plusieurs faits formant ainsi un schéma en constellation [Kimball 96]. Le partage des dimensions entre les faits limite les redondances et la complexité.
- Les paramètres des dimensions sont ordonnés suivant une hiérarchie relative à leur granularité. Le modèle proposé est suffisamment flexible pour supporter des hiérarchies multiples pour chaque dimension.
- Un ensemble d'opérations est défini par [Ravat, Teste, et Zurfluh 00]. Il englobe les principales opérations introduites dans les systèmes commerciaux et les travaux de recherche actuels. De plus, les auteurs [Ravat, Teste, et Zurfluh 00] définissent des opérations inhérentes à cette modélisation.

Soit les notations ; [ ] un n-uplet, { } un ensemble et < > une liste. D'autre part, soit **A** un ensemble de noms d'attributs tel que chaque attribut $a \in \mathbf{A}$ est associé à une liste de valeurs $dom(a)$; soit **D** un ensemble de noms de dimensions ; soit **F** un ensemble de noms de faits.

III. 2. 1. Fait

Le sujet analysé est représenté par le concept de fait. Les mesures (attributs) d'un fait sont numériques ; on peut les additionner, les dénombrer ou bien calculer le minimum, le maximum ou la moyenne.

Définition 1. Un **fait** *f* est un couple ($fnom$, $mesures^{fnom}$) où

– $fnom \in$ **F** est l'identifiant ;

– $mesures^{fnom} \subseteq$ **A** est un ensemble d'attributs formant les mesures d'activité du fait ;

$mesures^{fnom} = \{m_1, m_2, \ldots, m_m\}$ [Ravat, Teste, et Zurfluh 00].

III. 2. 2. Dimensions

Les dimensions servent à enregistrer les valeurs pour lesquelles sont analysées les mesures de l'activité. Une dimension est généralement formée de paramètres (attributs) textuels (pour restreindre la portée des requêtes) et discrets (les valeurs possibles sont bien déterminées et constantes) [Kimball 96].

Définition 2. Une **dimension** *d* est un triplet ($dnom$, $parametres^{dnom}$, $hierarchies^{dnom}$) où

– $dnom \in$ **D** est l'identifiant ;

– $parametres^{dnom} \subseteq$ **A** est un ensemble d'attributs formant les paramètres de la dimension ; l'ensemble des paramètres de la dimension contient un paramètre particulier, noté *all* tel que $dom(all) = \{All\}$;

– $hierarchies^{dnom} = < H^{dnom}_1, H^{dnom}_2, \ldots, H^{dnom}_h >$ est une liste de hiérarchies telles que $\forall\ i \in [1,\ldots, h]\ H^{dnom}_i = <p^i_1, p^i_2, \ldots, p^i_{li}>$ avec $\forall\ i \in [1,\ldots, li]$, $p^i_j \in parametres^{dnom}$; on distingue H^{dnom}_1 comme étant la hiérarchie courante [Ravat, Teste, et Zurfluh 00].

Les paramètres d'une dimension sont organisés en hiérarchies de la granularité la plus fine vers la granularité maximale. Les valeurs des paramètres sont associées à une famille de fonctions, notée ρ_{roll}, relatives aux hiérarchies. Une telle fonction $\rho_{roll}^{H(pj \longrightarrow pj)}(v) = v'$ associe une valeur v d'un paramètre p_j à une valeur v' d'un paramètre p'_j situé à une granularité supérieure selon ***H***.

III. 2. 3. Schéma dimensionnel

Un magasin est caractérisé par un schéma dimensionnel composé d'une constellation de faits et de dimensions.

Définition 3. Un **schéma dimensionnel** *S* est un n-uplet ($snom$, FAI^{snom}, DIM^{snom}, $Param^{snom}$) où

- *snom* est l'identifiant ;
- $FAI^{snom} = <f_1, f_2, \ldots, f_u>$ est une liste de faits ; on distingue *f1* le fait courant ;

- $DIM^{snom} = \{d_1, d_2, \ldots, d_v\}$ est l'ensemble des dimensions associées aux faits ;
- $Param^{snom} : FAI^{snom} \rightarrow 2^{DIMsnom}$ est une fonction qui associé un fait $f_i \in FAI^{snom}$ à la liste des dimensions qui lui sont associées de telle sorte que $\forall i \in [1..u]$,

$Param^{snom}(f_i) = <d^i_1, \ldots, d^i_{wi}>$ avec $\forall j \in [1..wi] \quad d^i_j \in DIM^{snom}$; on distingue les dimensions courantes d^i_1 et d^i_2 associées au fait f_i [Ravat, Teste, Zurfluh 00].

Afin de visualiser les informations contenues dans un magasin, les auteurs [Ravat, Teste, et Zurfluh 00] utilisent une présentation sous la forme d'une « *n-table* ». Ce choix est motivé par la simplicité de la représentation en tableau qui permet de visionner l'information de manière intuitive ; il s'agit d'une représentation très répandue à laquelle les utilisateurs sont habitués [Agr 95] [Gyssen, Lakshmanan 97].

Un schéma dimensionnel est donc visualisé sous forme de lignes, de colonnes et de plans ; les mesures du fait courant *f1* sont placées à l'intersection d'une ligne et d'une colonne pour un plan donné. La liste des autres faits est disponible à droite de la « *n-table* ». Les deux dimensions courantes *d1* et *d2* de $Param^{snom}(f1)$ sont affichées en lignes et colonnes définissant le plan visualisé. Les autres dimensions de $Param^{snom}(f1)$ sont disponibles en bas de la « *n-table* ». Pour chaque dimension courante *d1* et *d2*, les paramètres affichés sont ceux de la hiérarchie courante associée à la dimension. Initialement, seul le paramètre de granularité maximale est affiché.

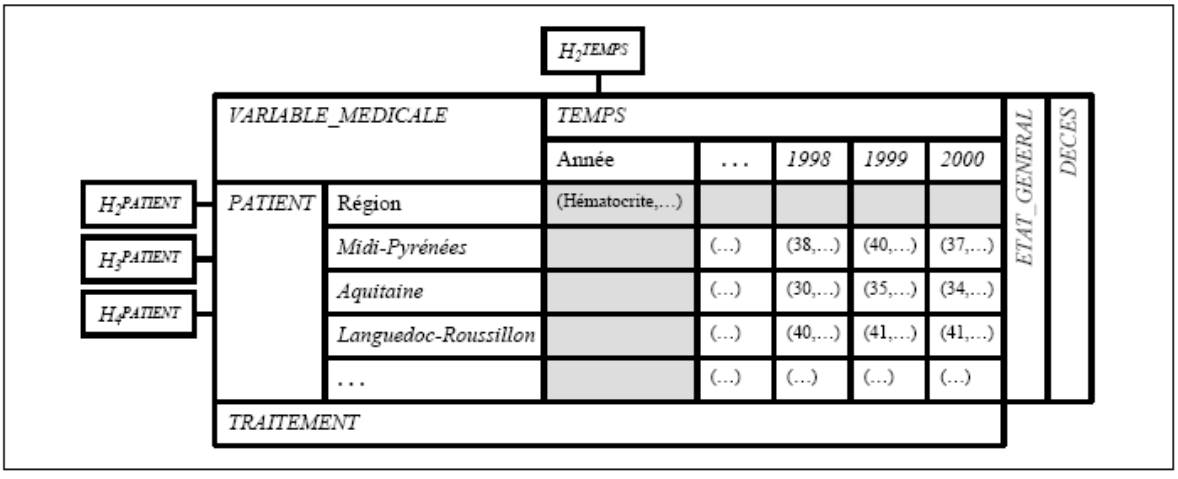


Figure 25. Représentation initiale d'un schéma dimensionnel dans une « *n-table* » [Test 00].

IV. Présentation de l'approche

Dans cette section nous allons voir un processus d'élaboration d'entrepôts de données par extractions pour la partie entrepôt de données. Ainsi qu'une démarche pour la transformation des données de l'entrepôt dans un magasin de données pour la partie magasins de données.

IV. 1. Processus d'élaboration d'entrepôt

Les auteurs [Ravat, Teste, et Zurfluh 00] ont spécifié une solution permettant de définir l'entrepôt de données en deux étapes successives.

- La définition de l'aspect statique des classes entrepôt ; il s'agit de spécifier, à partir de la source globale, les données pertinentes et les structures de ces données pour définir celles des classes entrepôt [Teste 00].
- La définition de l'aspect dynamique des classes entrepôt ; il s'agit de définir le comportement des classes entrepôt en fonction de celui des classes source et des éléments (propriétés et opérations) présents dans l'entrepôt [Ravat, Teste, et Zurfluh 00a].

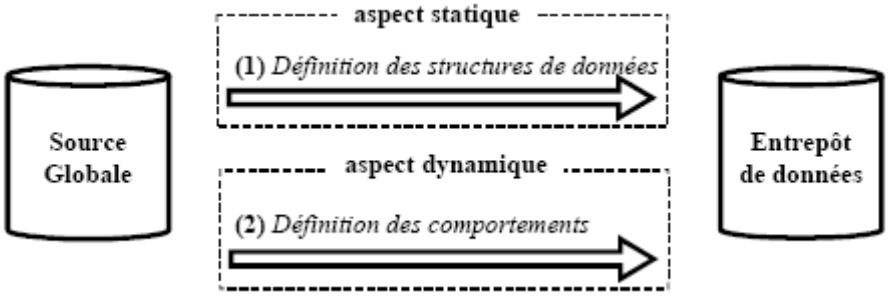


Figure 26. Principe d'extraction pour l'élaboration de l'entrepôt

Le processus de définition des structures et des données des classes entrepôt est réalisé au travers de la fonction de construction (pour une classe entrepôt *c*, $Mapping^c$ est la fonction de construction). Cette fonction de construction est définie par une expression constituée d'une succession d'opérations de base.

La définition du comportement des classes entrepôt s'appuie sur des matrices d'usage. Cette approche consiste à déterminer de manière semi-automatique si les éléments (c'est à dire les attributs, les relations et les opérations) utilisés par une opération sont tous présents dans l'entrepôt.

IV. 1. 1. Définition de l'aspect statique des classes entrepôts

Le mécanisme de définition de l'aspect statique des classes entrepôt consiste à définir la structure des classes entrepôt ainsi que leur extension. Plus précisément, il s'agit pour l'administrateur de

- désigner les objets source qu'il souhaite recopier dans l'entrepôt,
- définir la structure des objets entrepôt créés.

La définition de l'aspect statique d'une classe entrepôt *c* repose sur sa fonction de construction $Mapping^c$. Cette fonction est formée d'une succession d'opérations de base parmi lesquelles les catégories suivantes :

- les **opérations de structuration** (FS) définissent la structure (attributs et relations) des classes entrepôt ;
- les **opérations de qualification** (FQ) déterminent les objets source à partir desquels l'extension de la classe entrepôt est calculée ; ces opérations offrent également des mécanismes pour combiner les objets afin de constituer des objets entrepôt adaptés aux besoins des décideurs ;
- les **opérations ensemblistes** (FE) issues des algèbres objet [Shaw, Zdonik 90] ; elles offrent des mécanismes puissants pour transformer les classes afin de constituer des classes entrepôt adaptées aux besoins des décideurs ;
- les **opérations de hiérarchisation** (FH) organisent la hiérarchie d'héritage dans l'entrepôt en créant des super-classes et des sous-classes dans l'entrepôt.

IV. 1. 1. 1. Principe de l'extraction des données

Une classe entrepôt *c* est définie par (Nom^c, $Type^c$, $Super^c$, $Extension^c$, $Mapping^c$, $Tempo^c$, $Archi^c$) où $Type^c$ est le type de la classe tel que $Type^c$ = ($Structure^c$, $Relation^c$, $Comportement^c$).

Un objet entrepôt *o* est défini par (*oid*, S_0, *Histoire*, *Archive*, *Origine*) où *oid* est son identifiant et

- $S_0 = (DomT_0, V_0)$ est son état courant,
- *Histoire* est l'ensemble de ses états passés,
- *Archive* est l'ensemble de ses états archivés et
- *Origine* est l'ensemble des objets source dont il est issu.

L'extraction permet de recopier la valeur d'un objet source pertinent pour les décideurs dans un objet entrepôt. Une classe source *cs* est définie par (Nom^c, $Type^c$, $Super^c$, $Extension^c$). Afin de pouvoir appliquer une succession d'opérations pour extraire les données source, les objets source sont considérés comme des objets entrepôt particuliers *os* = (*s_oid*, S_0, *Histoire*, *Archive*, *Origine*) où *s_oid* est l'identifiant, S_0 est l'état courant contenant la valeur V_0 de l'objet, *Histoire* = Ø, *Archive* = Ø et *Origine* = {*os*}.

Les extractions sont effectuées à un point de synchronisation (instant pendant lequel les transactions de mise à jour des données sources sont toutes terminées), appelé un **point d'extraction**. A chaque point d'extraction, la valeur courante d'un objet entrepôt *o* est rafraîchie par la valeur courante de son origine source :

- chaque attribut dérivé prend la valeur de l'attribut source qu'il représente,
- chaque attribut calculé prend la valeur retournée par le calcul appliqué sur la source,

- chaque relation dérivée relie les objets entrepôt représentant les objets source liés par la relation source.

IV. 1. 1. 2. Fonctions de structuration

Les fonctions de structuration regroupent la fonction de projection (π), la fonction de masquage (μ) et la fonction d'accroissement (α) ; FS = $\{\pi, \mu, \alpha\}$. Leurs rôle est de définir les propriétés constituant la structure des classes entrepôt.

La fonction de **Projection** $\pi_{\{p1, p2, pp\}}$ (cs) dérive les propriétés de la classe source spécifiées dans l'ensemble $\{p_1, p_2, \ldots, p_p\}$ et l'extension $Extension^c$ de la classe entrepôt *c* est calculée à partir de tous les objets de la classe source *cs*. Inversement, la fonction de **masquage** $\mu_{\{p1, p2, \ldots pq\}}$ (cs) dérive les propriétés de la classe source non spécifiées dans l'ensemble $\{p_1, p_2, \ldots, p_q\}$. La fonction d'**accroissement** $\alpha_{\{p1 : f1, p2 : f2, \ldots, pr : fr\}}$ (cs) permet de créer de nouvelles propriétés $\{p_1, p_2, \ldots, p_r\}$, qui sont soit des attributs calculés, soit des propriétés spécifiques.

IV. 1. 1. 3. Fonctions de qualification

Les fonctions de qualification comprennent la fonction de sélection (σ) et la fonction de jointure ($\triangleright\triangleleft$) ainsi que les fonctions de groupement, « *nest* », (η) et dégroupement « *unnest* » (η^{-1}) ; FQ = $\{\sigma, \triangleright\triangleleft, \eta, \eta^{-1}\}$. Leur rôle est de déterminer l'extension des classes entrepôt en définissant des critères qualifiant les objets source pertinents à recopier dans l'entrepôt, ou bien, en regroupant des objets ou dégroupant des ensembles d'objets.

La fonction de **sélection** $\sigma_p(cs)$ génère une classe entrepôt dont la structure est dérivée de celle de *cs* et dont l'extension est calculée à partir d'une restriction de l'extension source. La fonction de **jointure** $\triangleright\triangleleft_p(cs_1, cs_2)$ génère une classe entrepôt dont la structure est l'union de celles de cs_1 et cs_2 et dont l'extension est calculée en filtrant par le prédicat de jointure *p*, le produit cartésien entre les deux classes source cs_1 et cs_2.

La fonction **nest** $\eta_{\{p1, p2, \ldots, pn\} :: attr}(cs)$ et la fonction **unnest** $\eta^{-1}_{\{p1, p2, \ldots, pm\}}(cs)$ permettent respectivement de grouper et de dégrouper les objets source pour peupler la classe entrepôt.

IV. 1. 1. 4. Fonctions ensemblistes

Les fonction ensemblistes regroupent les fonctions traditionnelles de l'algèbre objet [SHAW 90], c'est-à-dire l'union ($\cup$), l'intersection ($\cap$) et la différence (-) ; FE = $\{\cup, \cap, -\}$. Leur rôle est de combiner plusieurs classes sources afin de définir une nouvelle classe entrepôt.

IV. 1. 1. 5. Fonctions de hiérarchisation

Les précédentes fonctions génèrent des classes entrepôt dont la hiérarchie d'héritage n'est pas organisée ($Super^c = \emptyset$). Par conséquent, les auteurs [Ravat, Teste, Zurfluh 2000] introduisent deux nouvelles fonctions visant à généraliser (Λ) et à spécialiser ($\sum$) les classes entrepôt afin de construire une hiérarchie d'héritage adaptée aux exigences spécifiques de l'entrepôt et de ses utilisateurs ; FH = { Λ , $\sum$ }.

La fonction de **généralisation** $\Lambda_{\{p1, p2, ..., ps\}}(c_1, c_2, \ldots, c_n)$ génère une super classe entrepôt à partir d'une ou plusieurs classes, en regroupant l'ensemble $\{p_1, p_2, \ldots, p_s\}$des propriétés communes, spécifiées par l'administrateur.

La fonction de **spécialisation** $\sum_p(c_1, c_2, \ldots, c_n)$ génère une sous classe entrepôt à partir d'une ou plusieurs classes entrepôt $c_1, c_2, \ldots, c_n$.

IV. 1. 1. 6. Traitement des hiérarchies existantes

L'entrepôt de données et la source de données sont deux systèmes autonomes aux objectifs différents : la source globale contient l'ensemble des informations issues de l'intégration de sources distribuées, autonomes et hétérogènes tandis que l'entrepôt de données centralise les informations utiles pour les décideurs. Ces divergences se répercutent par des différences entre la hiérarchie d'héritage de l'entrepôt et la hiérarchie d'héritage de la source globale.

Cependant, il est fastidieux et même peu satisfaisant pour l'administrateur de reconstruire une hiérarchie existante. Toutes les opérations de structuration, de qualification et ensemblistes sont étendue pour indiquer :

- si la hiérarchie d'héritage des sous-classes à une classe source extraite est conservée ou non (par défaut, elle ne l'est pas),
- si la hiérarchie de composition des classes composantes une classe source extraite est conservée ou non (par défaut, elle ne l'est pas).

IV. 1. 2. Définition de l'aspect dynamique des classes entrepôt

Le paradigme objet adopté par cette approche, encapsule dans une même entité structure et comportement. Cette section présente la dérivation du comportement de ces données.

IV. 1. 2. 1. Extraction des comportements

Pour extraire le comportement dérivé des classes entrepôt, les auteurs [Ravat, Teste, et Zurfluh 00] proposent un processus automatique, répond aux critères suivants :

- Déterminer si les propriétés nécessaires à une méthode sont dérivées dans l'entrepôt,
- Déterminer si les objets manipulés par une méthode sont dérivés dans l'entrepôt,

- Déterminer si les méthodes indispensables à une autre méthode sont dérivées dans l'entrepôt.

IV. 1. 2. 2. Technique des matrices d'usage

Pour dériver les méthodes d'une classe source, il faut déterminer si l'ensemble des éléments (propriétés, objets, méthodes) requis par chaque méthode est présent dans l'entrepôt. Les auteurs [Ravat, Teste, et Zurfluh 00] inspirent une solution de la technique des matrices d'usage utilisées dans la conception des bases de données réparties [Ravat, Zurfluh 95]. Le concept de matrice d'usage initialement proposé dans les bases de données réparties est étendu afin de l'adapter au contexte de la conception des entrepôts objets.

Dans ce contexte, une matrice d'usage est construite de la manière suivante :

- Les opérations éventuellement dérivables forment les lignes et les critères à analyser constituent les colonnes de la matrice. Pour chaque opération, la valeur 1 indique les critères nécessaires à l'opération.
- Une ligne supplémentaire, nommée "*Dérivé*", précise les critères présents (ou dérivés) dans l'entrepôt.
- Une colonne supplémentaire, nommée "*Dérivable*", permet de décider de la "*dérivabilité*" ou non de chaque opération en fonction de la présence dans l'entrepôt des critères requis.

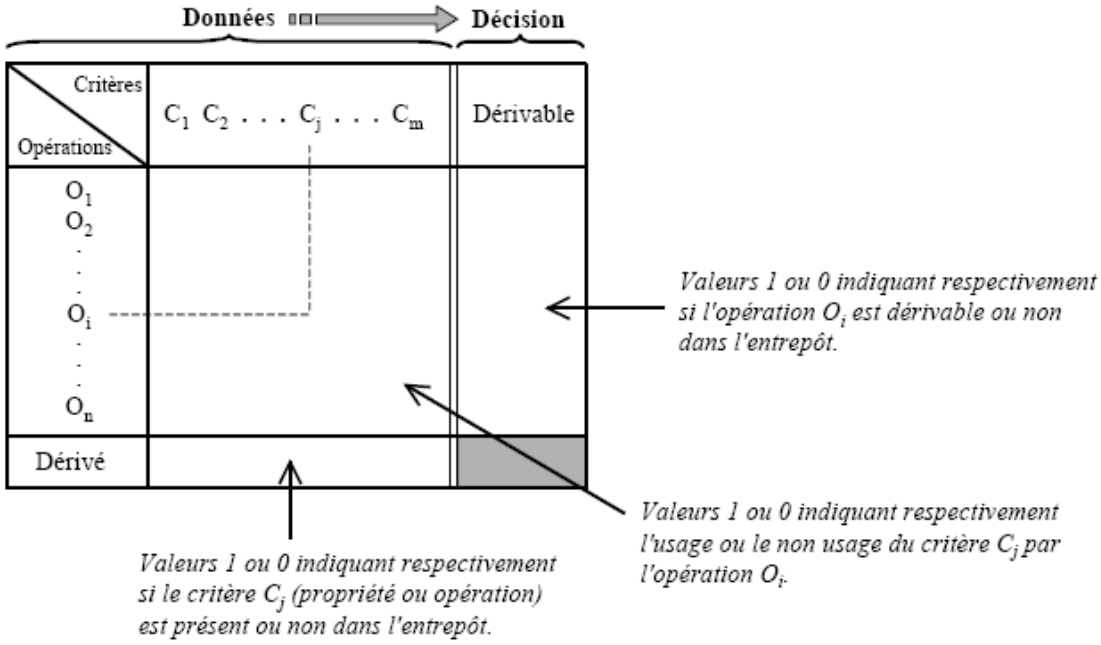


Figure 27. Concept de matrice d'usage dans le contexte des entrepôts objet.

Trois matrices d'usage sont utilisées :

- la **matrice d'usage des propriétés** (MUP) se propose de déterminer si les propriétés nécessaires aux méthodes sont dérivées dans l'entrepôt,

- la **matrice d'usage des opérations** (MUO) sert à déterminer si les opérations utilisées par chaque opération sont disponibles dans l'entrepôt.
- la **matrice d'usage des méthodes** (MUM) sert à déterminer si les méthodes utilisées par chaque méthode sont disponibles dans l'entrepôt.

Chaque fonction *Mapping*^{c} génère une classe entrepôt dont il est nécessaire de définir le comportement dérivé de la source. Par conséquent, le processus d'extraction des comportements intervient à deux niveaux :

- **Localement** à chaque classe entrepôt générée. Une MUP et une MUO sont définies pour chaque classe entrepôt pour déterminer localement si toutes les propriétés nécessaires aux méthodes dérivables sont disponible et si tous les objets manipulés sont dérivés.
- **Globalement** à l'entrepôt. Une MUM globale à tout l'entrepôt est construite pour déterminer si toutes les méthodes nécessaires aux méthodes dérivables sont disponibles.

Une méthode est dite dérivable si tous les éléments (propriétés, objets, autres méthodes) utilisés par la méthode sont disponibles.

IV. 1. 2. 3. Définition du comportement

L'ensemble des opérations dérivables est déterminé, par l'analyse des matrices d'usage locales (MUP) et la matrice d'usage globale (MUO).

Lorsqu'une opération n'est pas dérivée, le processus indique les éléments (propriétés, opérations) nécessaires manquants, afin que l'administrateur puisse ajuster l'entrepôt à ses besoins en dérivant les éléments nécessaires à une opération qu'il souhaite extraire.

L'algorithme qui définie le processus de définition du comportement des classes entrepôt est dans [Test 00]. Les analyses des matrices locales sont réalisées puis l'analyse de la matrice globale est effectuée. A chaque opération analysée localement ou globalement, l'ensemble des éléments manipulés manquants est affiché à l'administrateur. Lorsque les analyses sont terminées, il est possible d'indiquer le comportement dérivé des classes entrepôt.

IV. 2. Une Démarche pour la transformation des données de l'entrepôt dans un magasin

Selon les auteurs dans [Ravat, Teste, et Zurfluh 00], les magasins de données sont élaborés à partir de l'entrepôt. Dans [Ravat, Teste, et Zurfluh 01] les auteurs proposent une démarche à suivre pour construire un magasin comportant quatre étapes :

- détermination des faits représentant les sujets analysés,
- détermination des dimensions représentant les perspectives de l'analyse,

- définition des granularités des données de l'analyse,
- organisation des paramètres des dimensions selon des dépendances de hiérarchie pour supporter les analyses à différents niveaux de détail.

IV. 2. 1. Détermination des faits

La première étape consiste à déterminer le sujet de l'analyse, c'est à dire la liste *FAI* du schéma dimensionnel qui caractérise le magasin. Le sujet est représenté par des faits qui sont issus des classes entrepôt. Autrement dit, l'élaboration des faits est réalisée par dérivations de classes entrepôt particulières dites **classes représentatives**. Une classe représentative est désignée par l'administrateur ; il s'agit d'une classe contenant l'information qui doit être analysée (mesures de l'activité). L'entrepôt de données contient généralement plusieurs classes représentatives à partir desquelles sont créés les faits d'un magasin. Soit (Nom^{ED}, C^{ED}, Env^{ED}, $Config^{ED}$) un entrepôt de données.

Définition 1 : La **détermination d'un fait** F_i est réalisée par l'opération $Fact(nom^{Fi}, CR, Mes)$ où

- *nom* est le nom du fait,
- $CR \in C^{ED}$ est une classe représentative contenue dans l'entrepôt,
- $Mes = \{(A_i, f_i) \mid i \in [1..f]\}$ est un ensemble de couples (A_i, f_i) où A_i est une mesure de l'activité contenue dans le fait et f_i est la fonction qui calcule les valeurs de A_i en l'appliquant sur la classe représentative *CR* [Ravat, Teste, et Zurfluh 01].

Les fonctions de calcul f_i permettent de calculer les valeurs des mesures contenues dans le fait. Ces valeurs sont obtenues à partir des objets d'une classe représentative *CR* en appliquant une fonction de calcul qui peut être :

- un attribut de *CR* (la mesure est calculée à partir de l'attribut lui-même),
- une opération de type fonction de *CR* (la mesure représente le résultat retourné par la fonction).

Ainsi, chaque mesure est construite à partir d'un attribut ou d'une opération de *CR*, ou bien à partir d'une fonction d'agrégation.

Les valeurs contenues dans le fait sont calculées à partir des objets de la classe représentative. Cependant, en fonction de l'objectif décisionnel du magasin, il n'est pas systématiquement utile d'extraire dans un magasin l'ensemble des valeurs contenues dans l'extension de *CR*. Par conséquence, la détermination du fait peut être complétée par une sélection, qui indique le sous-ensemble des objets de la classe représentative qui doit participer à la génération des valeurs du magasin. L'opération de détermination du fait est alors définie par $Fact(nom^{Fi}$,

σ(*CR, pred*), *Mes*) où σ est l'expression d'une sélection portant sur *CR* et *pred* est un prédicat valide exprimant une condition [Test 00].

IV. 2. 2. Détermination des dimensions

La seconde étape de cette démarche consiste à créer les dimensions. Autrement dit, cette étape consiste à définir l'ensemble *DIM* et la fonction *Param* qui associe chaque fait précédemment défini aux dimensions de l'ensemble *DIM*.

Les dimensions sont élaborées à partir de classes entrepôt dites **dépendantes** des classes représentatives des faits. Chaque classe représentative d'un fait possède un ensemble de classes dépendantes à partir desquelles il est possible de construire les dimensions du fait. Cet ensemble de classes dépendantes est construit automatiquement en suivant un principe de dépendance entre classes.

Définition 2 : Le **principe de dépendance** entre classes entrepôt permet de spécifier l'ensemble des classes dépendantes d'une classe représentative d'un fait. Une classe entrepôt $C_1 \in C^{ED}$ est dépendante d'une classe $C_2 \in C^{ED}$ (soit la notation : $C_1 \Rightarrow C_2$) si elle satisfait au moins une des règles suivantes [Ravat, Teste, et Zurfluh 01] :

- $C_1 = C_2$ (la classe représentative est elle-même une classe dépendante), (*R1*)
- C_1 est reliée à C_2 par une relation d'héritage telle que C_1 est sous-classe de C_2, (*R2*)
- C_1 est reliée à C_2 par une relation d'association de cardinalité (*x*, *1*), (*R3*)
- C_1 est reliée à C_2 par une relation de composition
  - C_1 est une classe composante de C_2 avec la cardinalité (*x*, *1*), (*R4'*)
  - C_1 est une classe composée de C_2 avec la cardinalité (*1*, *x*). (*R4''*)

IV. 2. 3. Définition des granularités

La troisième étape consiste à spécifier comment doivent être agrégées les données dans le magasin. L'énoncé précis des dimensions d'un fait détermine la granularité des mesures du fait. Suivant le niveau de granularité, plusieurs valeurs d'une mesure sont obtenues pour chaque combinaison des dimensions du fait.

Définition 3 : La **définition de la granularité** des mesures d'un fait est réalisée par l'opération *Gran*(F_i, *Agreg*) où

- F_i est un fait,
- *Agreg* = $\{(A_i, f_i) \mid i \in [1..f]\}$ est un ensemble de couples (A_i, f_i) où A_i est une mesure de l'activité contenue dans le fait et f_i est une fonction d'agrégation telle que *avg*( ), *sum*( ), *count*( ), *min()*, *max*( ), *first*( ), *last*( ) ou *nth*(*i*) qui permet de regrouper les valeurs [Ravat, Teste, Zurfluh 01].

IV. 2. 4. Hiérarchisation des dimensions

La dernière étape dans l'élaboration d'un magasin de données consiste à organiser la hiérarchie des dimensions. Il s'agit de définir, pour chaque dimension, la liste H^D des hiérarchies des paramètres.

Définition 4 : La **hiérarchisation d'une dimension** est réalisée par l'opération *Hiera*(*D*, *H*) où

- $D = (Nom^D, Attribut^D, H^D)$ est une dimension,
- $H = <A_1, A_2, \ldots, A_h>$ est une hiérarchie telle que $i \in [1..h]$, $A_i \in Attribut^D$ [Ravat, Teste, Zurfluh 01].

V. Conclusion

Dans ce chapitre nous avons présenté une méthode de conception des entrepôts et magasins de données qui repose sur une distinction de deux espaces de stockage ; l'entrepôt et les magasins de données. A partir de cette séparation, existe deux modélisations différentes :

- modélisation de l'entrepôt de données. Basé sur un modèle qui subit l'influence du modèle objet standard de l'ODMG [Cattel 95] qui est étendu pour prendre en compte les caractéristiques des entrepôts de données.
- modélisation des magasins de données. Le modèle conceptuel proposé par les auteurs pour les magasins dimensionnels constitue une généralisation des modèles dimensionnels habituellement proposés (constellation de faits et dimensions munies de hiérarchies multiples).

I. Introduction

Ce chapitre présente une quatrième approche de conception d'entrepôt de données proposée par Michael Böhnlein, et Achim Ulbrich-vom Ende [M.Boh, A.Ulb 00]. C'est une approche basée sur la dérivation des structures de l'entrepôt de données à partir des modèles de processus d'affaires.

II. Le modèle de données utilisé

Cette approche utilise le modèle dimensionnel dans la partie présentation de données ; comme on a déjà passé en revue dans le chapitre état de l'art et dans le troisième chapitre la définition, les avantages et concepts de bases de ce modèle nous présenterons directement le modèle de processus d'affaires.

II.1. Modèle de processus d'affaires

Le processus d'affaires a plusieurs significations dans la littérature. La méthode qui spécifie les systèmes d'affaires et les systèmes d'applications d'affaires fait partie du Modèle d'Objet Sémantique (SOM) ([FeSi 90], [FeSi 91], [FeSi 93a], et [FeSi +94]). Cette méthode est composée de trois étapes principales :

Plans de l'entreprise : Identification de l'univers de discours de la compagnie, son environnement, ses services, ses buts et objectifs, ses facteurs de succès, et ses séquences de valeurs.

Modèle de processus d'affaires : D'un point de vue comportemental, une compagnie est composée d'un ensemble de processus d'affaires. Les processus principaux contribuent directement aux buts de la compagnie, et les sous processus soutiennent les processus principaux par de divers services. Les relations entre les processus d'affaires suivent le modèle client serveur. Un processus client engage un processus serveur pour fournir un certain service.

Spécifications des systèmes d'application d'affaires : Le but d'un système d'application d'affaires est d'automatiser une certaine partie d'un processus d'affaires. Des systèmes d'application sont identifiés et séparés dans l'ensemble de processus d'affaires et sont indiqués par l'utilisation d'une notation orientée objet [Fer.O, Sin.E 94].

II.1.1. Méta modèle pour spécification de processus d'affaires

Dans cette section nous allons voir un modèle de processus d'affaires basé sur un méta modèle qui est représenté sur la Figure 28. Le but d'un processus d'affaires est de produire un

ou plusieurs genres de services et de communiquer les services aux clients de ce processus. La spécification de processus d'affaires nécessite les éléments suivants :

- Spécification des services que le processus d'affaires traite.
- Spécification d'un objet d'affaires et d'un ensemble de transactions qui produisent et transmettrent les services. En ce qui concerne la délimitation de l'univers de discours, les objets internes et externes sont distingués.
- Chaque transaction est associée exactement à deux tâches. Ces tâches peuvent être considérées comme des conducteurs d'une transaction. Les tâches effectuent un protocole qui est nécessaire pour transmettre un paquet de services ou un message d'un processus de serveur à un processus de client. Les tâches d'une transaction sont assignées aux objets d'affaires qui sont reliés par cette transaction. De cette façon, un objet est associé à un ensemble de tâches, chaque tâche conduit une transaction particulière. D'un point de vue objet d'affaires, des tâches peuvent être considérées comme des opérations de l'objet.
- En plus des transactions, deux genres d'évènements sont employés pour commander l'exécution de tâches. Les évènements internes relient des tâches dans un objet. Les évènements externes définissent l'environnement des conditions préalables pour l'exécution des tâches.

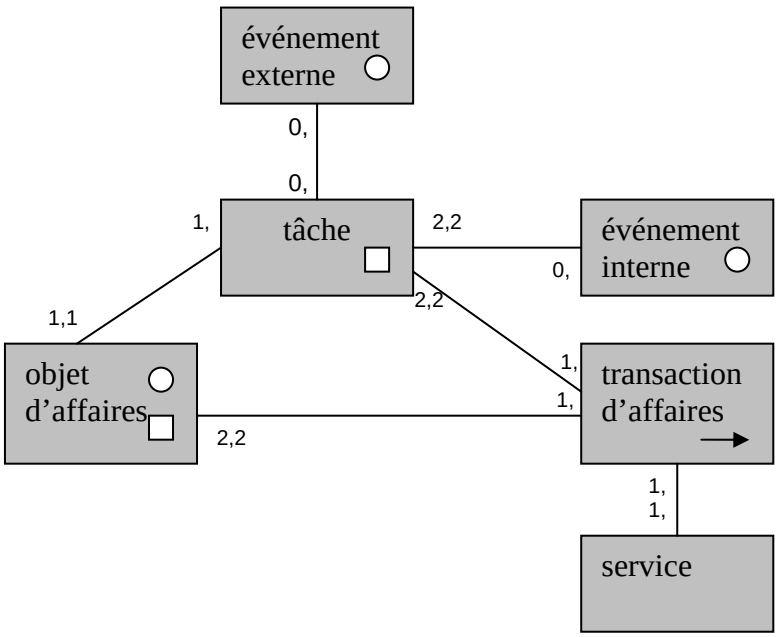


Figure 28. Méta modèle pour spécification de processus d'affaires

Il y a des processus d'affaires qui sont trop complexes pour être présentés sous une même forme. Par conséquent, deux vues différentes sont employées pour la représentation des processus d'affaires. Ces vues peuvent être dérivées du méta modèle.

1. La première vue s'appelle le schéma de transaction et présente la structure statique d'un processus d'affaires. Le schéma de transaction contient des objets et des transactions qui ont un rôle de voies de transmission.
2. La deuxième vue s'appelle le schéma de tâches_évènements et présente le comportement dynamique d'un processus d'affaires. Le schéma de tâches_évènements est composé des tâches, des évènements, et des transactions.

II.1.2. Coordination des objets d'affaires

Un processus d'affaires décrit selon un méta modèle peut être raffiné périodiquement à un niveau plus détaillé. Ce raffinage est fait par la décomposition des objets d'affaires et des transactions. La décomposition des objets et des transactions définie les mécanismes de base pour la coordination entre les objets.

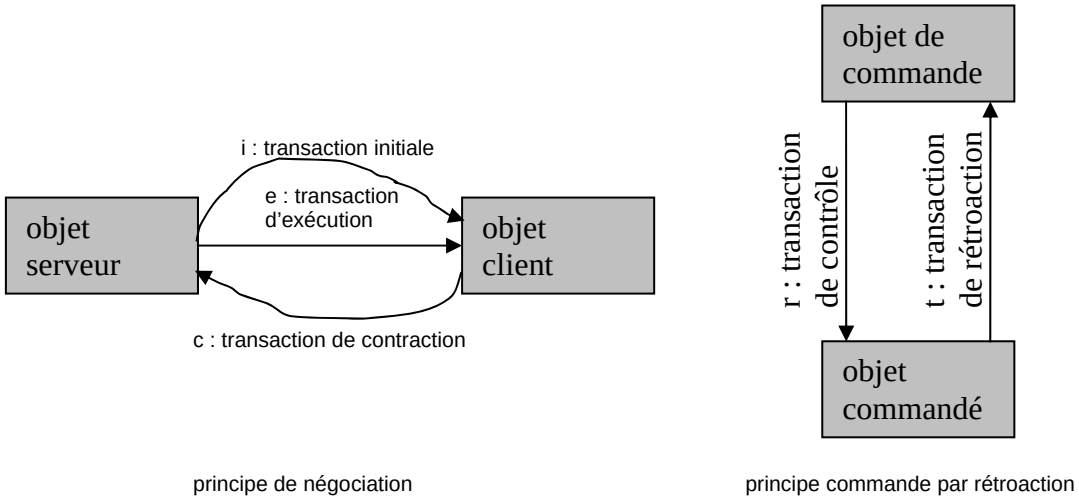


Figure 29. Mécanisme de base pour la coordination entre les objets

- Principe de négociation :

Une transaction entre deux objets est décomposée en une séquence de transactions secondaires : (1) la transaction initiale, (2) une transaction de contraction, (3) une transaction d'exécution. Pendant la transaction initiale, les objets apprennent à se connaître et à échanger l'information aux services livrables. Dans la transaction de contraction les deux objets sont d'accord pour un contrat sur l'échange d'un service. Le but de la transaction d'exécution est d'échanger le service entre les objets.

- Principe de commande rétroactive :

Un objet est décomposé en deux objets secondaires et deux transactions. L'objet secondaire de contrôle prescrit les objectifs où il envoie les messages de contrôle à l'objet secondaire contrôlé par la transaction de contrôle [Fer.O, Sin.E 94].

II.1.3. Distribution de processus d'affaires

Initialement le processus d'affaires est composé de trois élément : (1) un *fournisseur* d'objets internes, (2) un *client* d'objets externes, et (3) un *service* de transaction.

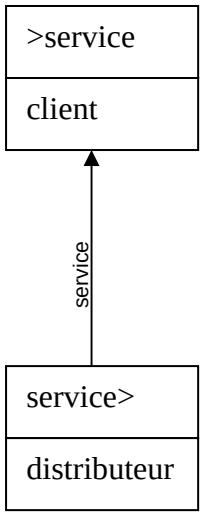


Figure 30. Le schéma d'interaction et le schéma tâches_évènements pour la distribution du processus d'affaires (1er niveau)

Dans le schéma tâches_évènements, les noms initiaux de tâches sont dérivés du nom de transaction. Par exemple, le nom de tâche *service* > (c à d « *service d'envoie* ») et > *service* (c à d « *service d'acquisition* » sont dérivés du *service*.

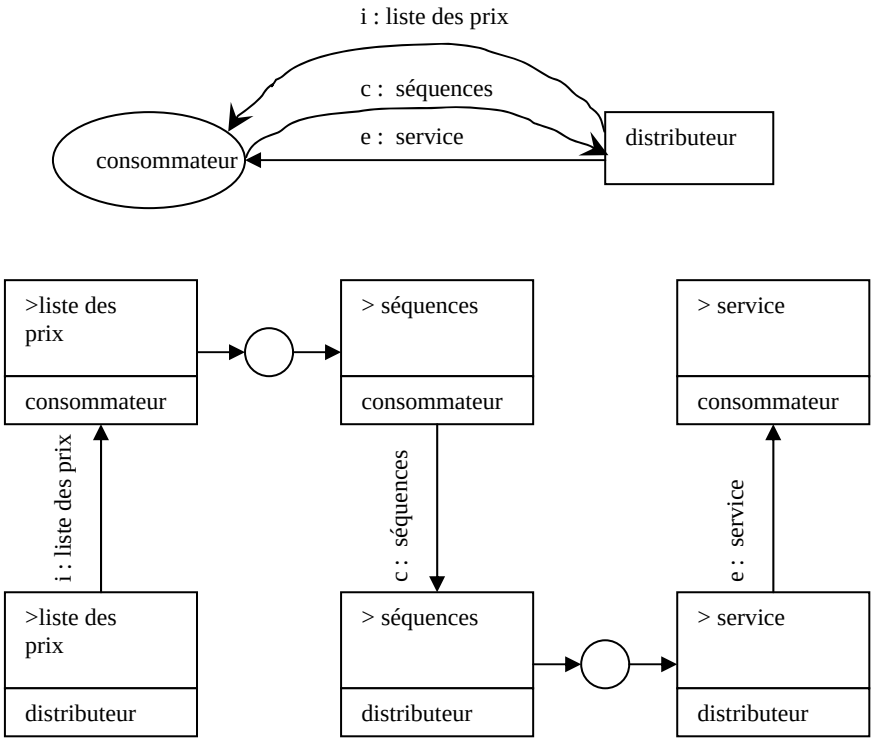


Figure 31. Le schéma d'interaction et le schéma tâches_évènements pour la distribution du processus d'affaires (2ème niveau)

Le deuxième niveau est une amélioration du premier niveau. Ici la transaction *service* est décomposée pour avoir la coordination entre *fournisseur* (objet fournisseur) et *client* (objet client). Comme les deux objets négocient sur le sujet livraison d'un service, selon le principe de négociation la transaction est décomposée aux transactions secondaires *i : liste des prix* (initiale), *c : séquence* (contraction), et *e : service* (transaction d'exécution). Parce que les transactions secondaires s'exécutent séquentiellement, le schéma tâches_évènements et en même temps c'est le schéma d'interaction.

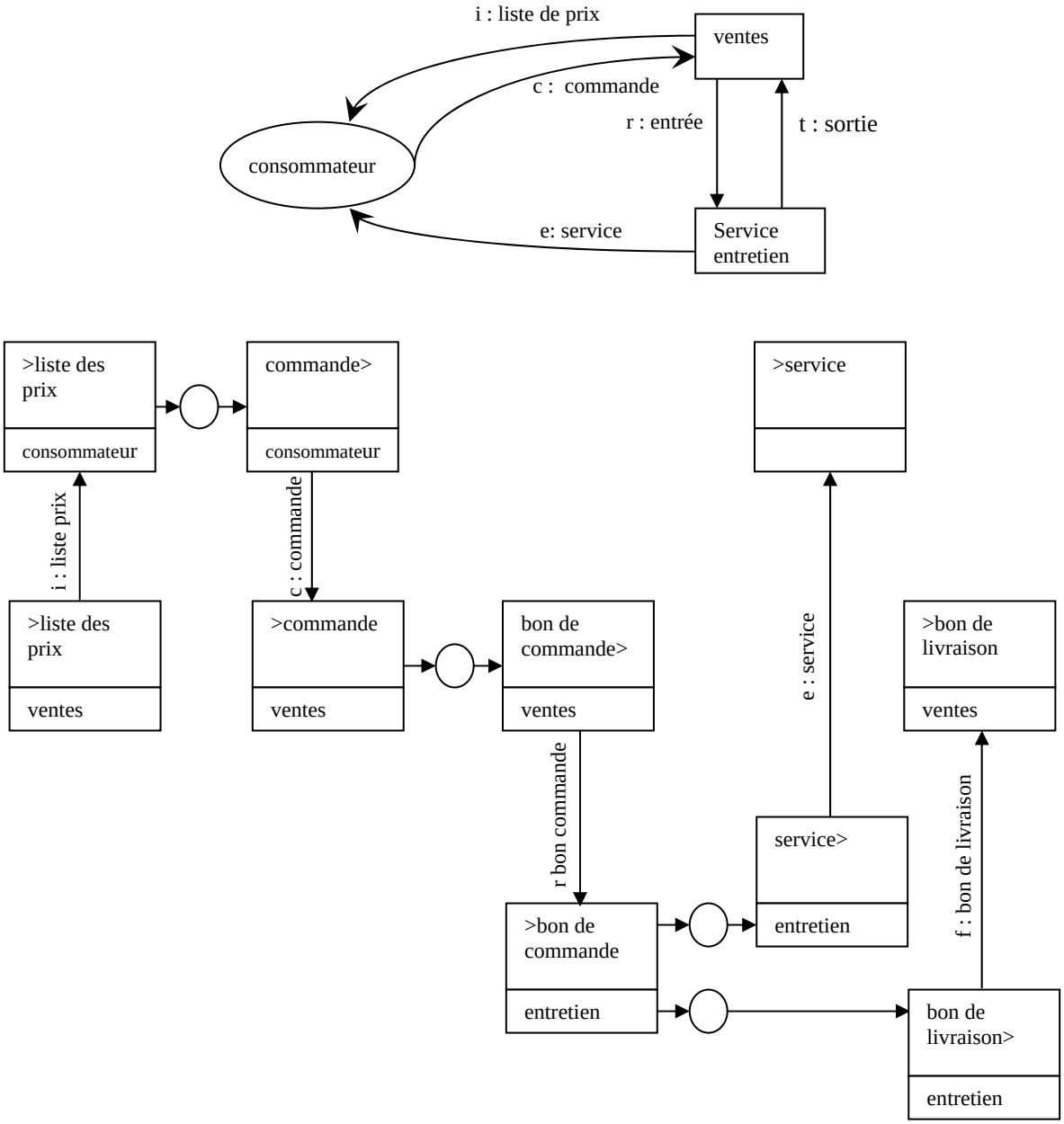


Figure 32. Le schéma d'interaction et le schéma tâches_évènements pour la distribution du processus d'affaires (3ème niveau)

Au troisième niveau, l'objet *fournisseur* est décomposé selon le principe de commandes rétroactives. Le résultat de la décomposition sont les objets secondaires suivantes ; *Ventes* (objet secondaire de contrôle) et *système entretien* (objet secondaire contrôlé). Au même temps les transactions secondaires *liste des prix*, *ordre*, et *service* sont à nouveau attribuées aux objets secondaires.

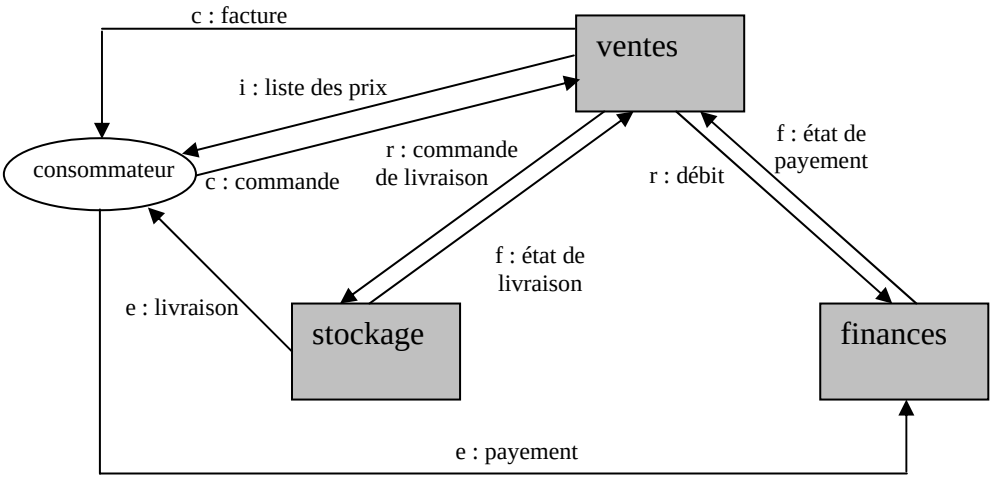


Figure 33. Le schéma d'interaction et le schéma tâches_évènements pour la distribution du processus d'affaires (4ème niveau)

La dernière décomposition est le quatrième niveau. D'abord, la transaction *e : service* est décomposée en deux séquence *e : livraison* et *e : payement cash*. La transaction *payement cash* est encore décomposée selon le principe de négociation en deux séquences *c : facture* et *e : payement*.

III. Présentation de la Méthode

III. 1. Dérivation des structures initiales de l'entrepôt de données

Pour montrer l'application de cette approche les auteurs [M.Boh, A.Ulb 00] présentent un modèle typique du processus d'affaires d'une université allemande. Le modèle est basé sur des résultats du projet OptUni (Optimisation of University Processes). Une université peut être considérée comme fournisseur (provider) de services. L'université est principalement organisée par des processus services Figure 34. Deux processus principaux peuvent être indiqués. Les deux processus *études* et *études supérieures* fournissent un service *études supérieures* aux étudiants et le processus *recherche* fournit des *activités de recherches* aux associés *partenaires de recherches*. Plusieurs processus service comme le service *commerciale*, *personnel*, et *bibliothèque* aident a accomplir les buts des processus principaux en fournissant des services spéciaux.

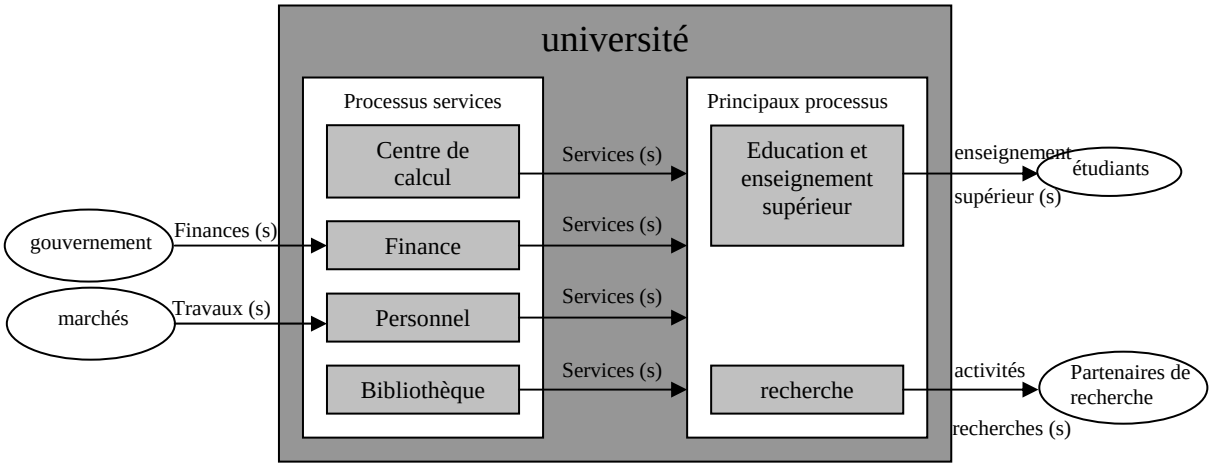


Figure 34. La structure de processus d'affaires d'une université

III.1. 1. Le processus de dérivation

Dans cette section nous allons voir la proposition des auteurs [M.Boh, A.Ulb 00] d'un modèle d'actions orienté par processus d'affaires pour dériver les structures initiales de l'entrepôt de données. La suggestion est basée sur la méthode (SOM). Cette approche est composée de quatre étapes principales Figure 35. Tandis que les trois premières étapes correspondent largement à la méthode (SOM) et sont spécialisées pour adapter les besoins des entrepôts de données, la quatrième étape permet une identification des structures de l'entrepôt de données.

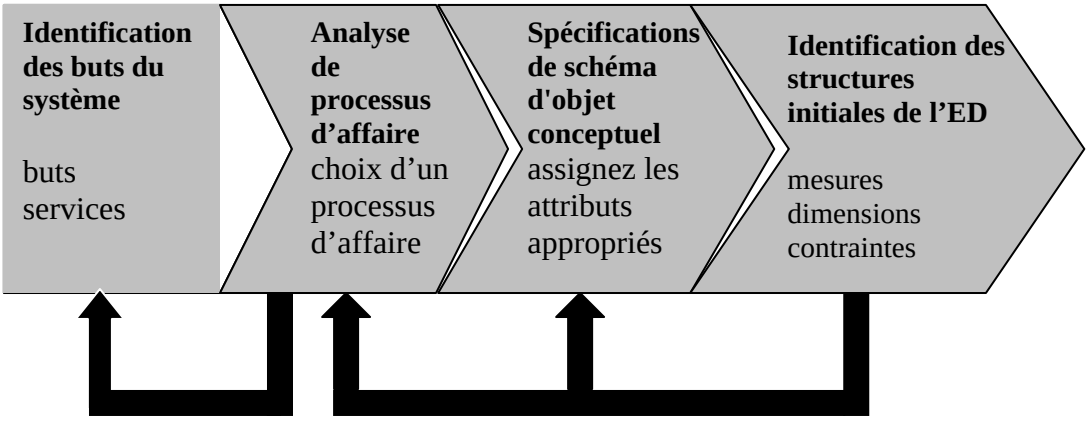


Figure 35. Les principales étapes dans le processus de dérivation

Après l'identification des buts du système (étape 1), le réseau des processus d'affaires est analysé pour obtenir une profonde compréhension du domaine d'application (étape 2). Un processus d'affaires est choisi pour des considérations supplémentaires (étape 3). Le schéma d'objet conceptuel (COS) est dérivé à partir de processus d'affaires qui est déjà choisi pendant la deuxième étape. Un COS est un schéma conceptuel de données enrichi avec des concepts orientées objet. Il décrit les structures de données nécessaires pour compléter le processus

d'affaires choisi. Pour obtenir un schéma (COS) initial beaucoup de transformations des informations possibles sont effectuées sur le modèle de processus d'affaires au niveau du système d'application. Dans le (COS) il est facile d'identifier les structures initiales de l'entrepôt de données (étape 4). Les structures initiales de l'entrepôt de données sont visualisées par des schémas en étoile et flocon de neige.

III.1.1.1. Spécification des buts et des services correspondants au système

En premier lieu les auteurs [M.Boh, A.Ulb 00], spécifient des objectifs et des services correspondant que l'université fournit à ces clients. Deux types d'objectifs sont distingués ; le premier indique les produits et les services à fournir, le seconde détermine dans quelle mesure les buts doivent être poursuivi. Des conditions générales sont prescrites par des lois et ordonnances du ministère et gouvernement. En outre les buts de l'université sont influencés par la gestion de l'université, les étudiants aussi bien que par le publique. Les auteurs [M.Boh, A.Ulb 00] prennent en considération seulement deux conditions pour indiquer les buts de l'université. Ce sont deux lois essentielles pour l'université de **Bavarian** prescrites par le gouvernement. Seulement les buts qui peuvent être mesurés sont fonctionnels et peuvent être facilement décomposé en buts secondaires. Les quatre buts prescrits par les auteurs [M.Boh, A.Ulb 00] sont les suivants :

1. Offre l'enseignement supérieur aux étudiants.
2. Préparation des étudiants par l'application des méthodes et des connaissances scientifique.
3. Encourager la coopération internationale par l'échange de formations entre les universités allemandes et les universités étrangères.
4. Disponibilité des conseils scientifique.

Selon le modèle initial du processus d'affaires proposé par les auteurs [M.Boh, A.Ulb 00] sur la Figure 30. Les services principaux d'une université sont :

- Des études supérieures aux étudiants.
- Activités et coopération de recherches avec des partenaires de recherches.

Par conséquent les auteurs [M.Boh, A.Ulb 00] ont choisi le but « Offre l'enseignement supérieur aux étudiants ». Cet objectif se conforme avec le service principal ; *études supérieures* pour les *étudiants*.

III.1.1.2. Analyse de processus d'affaires

Le processus décrit un procédé pour réaliser le système des buts et des services correspondant à une université. Ainsi il fournit la solution au plan de l'université. On peut distinguer des processus principaux et des processus services [M.Boh, A.Ulb 00]. Tandis que les processus principaux contribuent directement aux buts de l'université et fournissent un service « *recherches et études supérieures* » aux clients « *étudiants* et *partenaires de recherches* ». Le processus principal *recherches et études supérieures* se conforme au but choisi dans la section précédente. La Figure 36 montre une vue plus détaillée du schéma d'interaction de la Figure 32 décrivant le processus *recherches et étude supérieure*. L'analyse du processus d'affaires peut indiquer des services secondaires importants dans le service principale *étude supérieure*. Dans la Figure 36 les transactions qui fournissent des services peuvent être distinguées aux transactions restantes par la lettre S entre parenthèses.

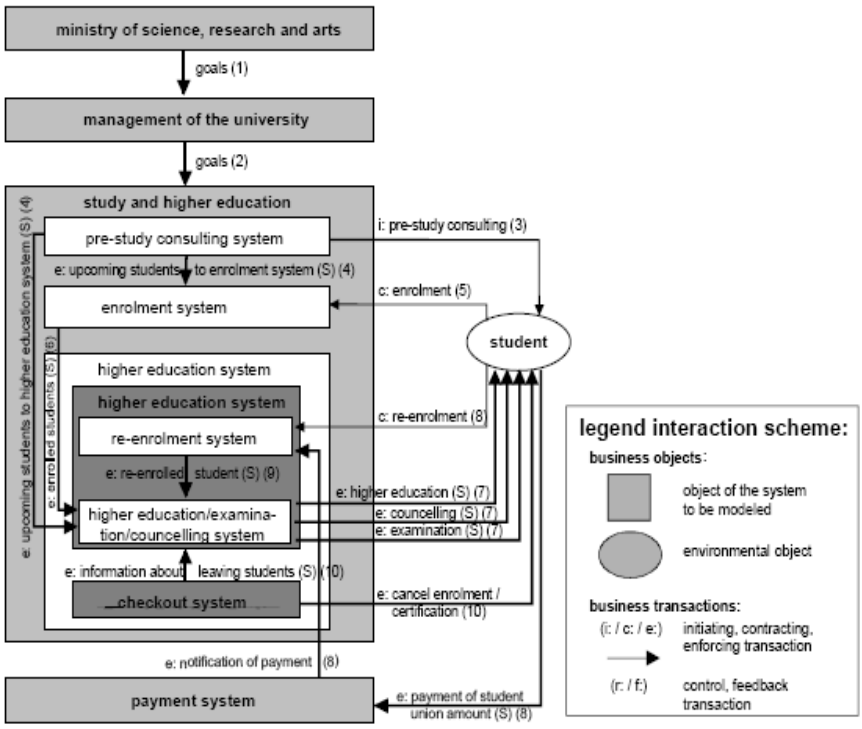


Figure 36. Schéma d'interaction du processus d'affaires *recherches et études supérieures* [M.Boh, A.Ulb 00].

Le ministère de la science, de la recherche et des arts définit des buts des universités de la ville de Bavarian (1). Ces buts sont améliorés et réalisés par la gestion de chaque université (2). Pour fournir des études supérieures aux étudiants l'objet d'affaires *recherches et études supérieures* doit être décomposé. La décomposition indique souvent des services secondaires dans le processus d'affaires. Initialement le *système consultation pré_étude* aide l'*étudiant* de

se rapprocher de l'université (3). Il fournit des services secondaires *informations sur le futur étudiant* aux *études supérieures/examen/conseils scientifiques* et au *système d'inscription* (4). Le *système d'inscriptions* rapporte des *inscriptions* par les *étudiants* (5). En tant que service secondaire *information d'inscription* aux *études supérieures/examen/ conseils scientifiques* (6). Le *système études supérieures* est décomposé en : *études supérieures/examen/ conseils scientifiques*, un *système d'inscription* et un *système de contrôle*. Si on ne prend pas en considération *études supérieures/examen/ conseils scientifiques*. Il fournis les principaux services *études supérieures*, *examen* et *conseil* (7). Chaque semestre les étudiants doivent informer le *système de réinscription* qu'ils veulent continuer leurs études et payer des frais au *système de paiement* (8). Le *système de réinscription* informe les systèmes *études supérieures/examen/ conseils scientifiques* au sujet du *nombre d'étudiants réinscrits* (9). Le *système contrôle* informe *études supérieures/examen/ conseils scientifiques* au sujet de la *certification* et des *annulations* des *inscriptions* (10).

Après cette courte description du processus *études supérieures*, nous allons voir dans la section suivante la dérivation de schéma conceptuel d'objet (COS) du schéma d'interaction tâches_évènements.

III.1.1.3. Dérivation du schéma d'objet conceptuel

L'objectif de cette étape est de transférer autant d'informations possibles à partir de la deuxième étape (modèle de processus d'affaire) vers la troisième étape (spécification de ressource). Le secteur de système d'application d'affaires est représenté par le schéma d'objet conceptuel (COS) et un schéma de tâches. Les structures de données font partie du COS. Par conséquent la transformation de l'information est déjà contenue dans des modèles de processus d'affaires aux structures de données conformément au COS. Un COS peut être considéré comme un schéma de données conceptuel dans le modèle Entité/Relation. Les entités conceptuelles encapsulent les états de tâches d'un objet d'affaires aussi bien que les états de transactions et de services correspondants. Une caractéristique spéciale du COS est la visualisation d'existence de dépendances. Dans un COS en trouve les entités indépendantes sur la partie gauche du schéma, et les entités dépendantes sur la partie droite du même schéma. Ce concept permet une augmentation de dépendance séquentielle entre les entités de gauche et celles de droite. Par exemple l'*inscription* dépend de *consultation pré_étudiant* qui elle-même dépend de *étudiant*. La structure initiale du schéma conceptuel de classes est dérivée d'un niveau plus détaillé du schéma d'intéraction Figure 36. Pour arriver à la Figure 37 il faut appliquer les règles de transformations suivantes :

Les objets d'affaires sont indépendant et peuvent être visualisés en tant qu'entités conceptuelles situées dans la partie gauche du diagramme. Par conséquent les objets d'affaires *études supérieures/examen/ conseils scientifiques*, *système consultation pré_étudiant, étudiant, système d'inscription, système de réinscription, système de contrôle,* et *système de payement* sont schématisées l'un en dessous de l'autre sur la partie gauche du diagramme.

Les transactions dépendantes des objets d'affaires correspondantes sont visualisées en tant qu'entités dépendantes du côté droit des entités conceptuelles représentant ces objets d'affaires.

Le schéma tâche_évenement offre une séquence d'exécution pour les processus de transformation. Des séquences de transactions sont graduellement transformées en séquences d'existence de dépendances conceptuelles entre les entités.

Le COS peut encore être raffiner s'il y a lieu. Parfois il est utile de prendre en considération les cardinalités des relations entre les entités conceptuelles.

Pour bien préparer la quatrième étape les auteurs [M.Boh, A.Ulb 00] assignent des attributs aux entités conceptuelles. Par exemple les attributs comme *age*, *nationalité* (*ville*, *état*, *pays*) sont assignés à *étudiant*, *date*, date en heure de *consultation pré_étudiant*, et *type d'inscription* (nouvelle inscription initiale ou l'inscription initiale), *faculté, sujet* à *inscription*.

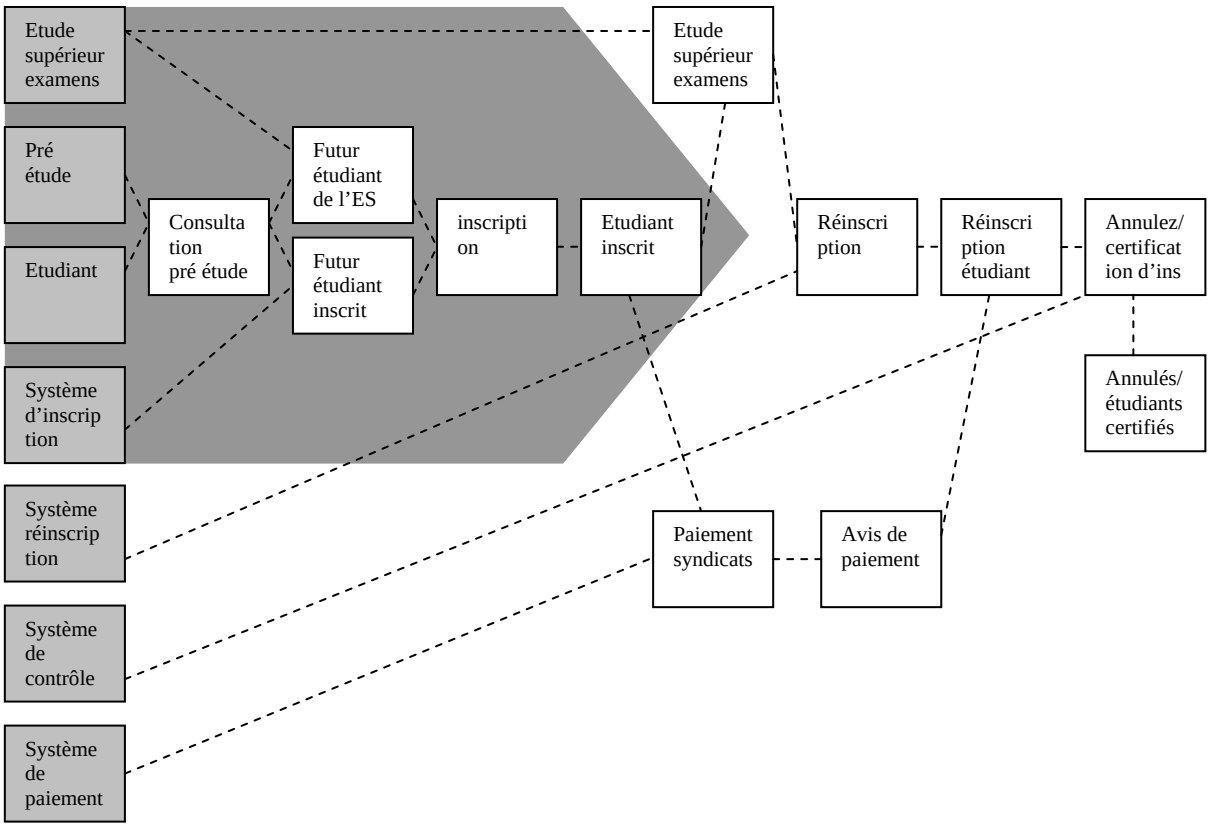


Figure 37. Schéma d'objet conceptuel (COS)

III.1.1.4. Identification des structures initiales de l'entrepôt de données

Maintenant nous allons voir la dérivation des structures initiales de l'entrepôt de données du schéma d'objet conceptuel indiqué dans (l'étape 3). Les structures de l'entrepôt de données sont représentées avec le schéma en étoile du modèle dimensionnel.

III.1.1.4.1. Identification des mesures

L'importance des processus d'affaires peut être évaluée selon leurs contributions aux services et aux buts de l'université. Il est obligatoire de trouver les mesures appropriées pour évaluer ces processus. A partir de la structure de données du COS. Les mesures sont indiquées suivant la chaîne d'argumentation « but-processus d'affaires-services-mesures ». Dans l'exemple précédent le but choisi offre des *études supérieures* aux *étudiants* pour un *examen plus approfondi* (étape 1). Dans (l'étape 2) le processus d'affaires qui correspond aux *étude* et *études supérieures* est analysé en détail. Pendant la décomposition du processus d'affaires beaucoup de nouveaux services secondaires qui correspondent également au but choisi sont apparus. Il est possible de trouver des mesures appropriées pour chaque service secondaires. Les auteurs [M.Boh, A.Ulb 00] se focalisent seulement sur le service secondaire *inscription* des *étudiants*, parce qu'avant l'identification des structures de l'entrepôt de données ont déjà procédés à la dérivation de COS initial de ce dernier processus et ont déjà assigner des attributs appropriées dans (l'étape 3). Maintenant nous allons voir la procedure de recherche sur les mesures et les dimensions correspondant pour faire l'évaluation de service choisi pour le processus d'affaire indiqué dans (l'étape 4).

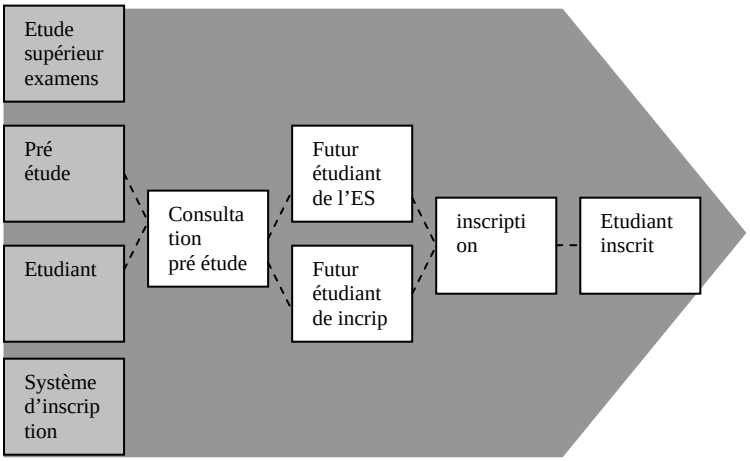


Figure 38 (a). COS et schéma d'étoile correspondant

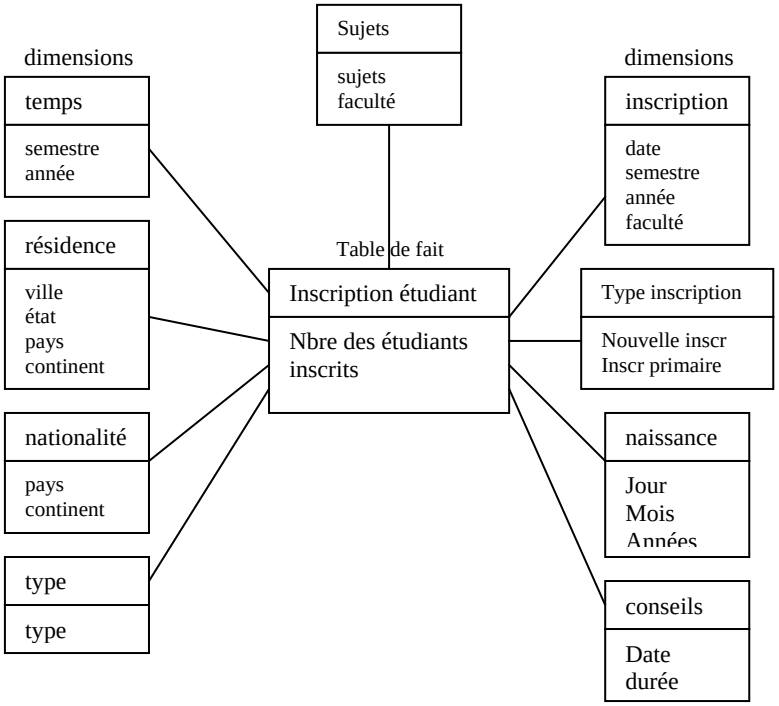


Figure 38 (b). COS et schéma d'étoile correspondant

Comment un service donné peut être mesuré ? L'exécution de service et l'accomplissement de l'ensemble de buts devraient être évalué par une mesure quantitative proportionnée. L'*inscription* peut être mesurée par exemple par les deux mesures : le *nom* et le *taux des étudiants inscrit*. Les mesures de bases et les mesures dérivées sont distinguées les une des autres. Chaque mesure forme un attribut au centre du schéma en étoile comme dans la Figure 37.

III.1.1.4.2. Identification des dimensions et des hiérarchies de dimensions

La visualisation d'exécution des dépendances dans le COS aide à identifier des dimensions et des hiérarchies de dimensions potentielles. Définir des dimensions et hiérarchies de dimensions exigent la créativité et la connaissance considérable du domaine d'application qui ne peut pas être décrit par un algorithme formel. Les dimensions ou hiérarchies de dimensions découvertes sont ajoutées au schéma en étoile. L'entité conceptuelle *inscription* peut contenir des attributs comme *nouvelle inscription, inscription primaire, thème, faculté, date inscription, semestre* et *année d'inscription*. Tous ça mène à la définition des dimensions : *inscription, type d'inscription* et *thème*. L'entité conceptuelle *consultation pré_études* à comme attributs : *date* et *durée de consultation pré_études* ceci mène à la définition d'une dimension du même nom. *Année de naissance* et *résidence* sont des exemples des dimensions concernant l'entité conceptuelle *étudiant*. L'information heuristique peut être utile pour ajouter des dimensions additionnelles.

III.1.1.4.3. Identification des contraintes

Dans le cas des schémas complexes on peut trouver des dimensions partagées entre les mesures. Ces dimensions partagées devraient toujours avoir la même information dans un processus d'affaire. On ne peut pas utiliser deux dimensions de temps différentes dans un processus d'affaires. Il est important de prendre en considération des fonctions d'agrégation le long des hiérarchies de dimensions. Les éléments d'un niveau de hiérarchie plus bas sont récapitulés à un niveau de hiérarchie plus élevé. Cette règle ne peut pas être présentée dans un schéma en étoile ou dans un schéma en flocon standard. Les mesures peuvent être classifiées en : mesures *semi_additives*, et mesures *non_additives* basées sur des fonctions d'agrégation selon leurs dimensions. Dans la Figure 36 ; la mesure *nombre d'étudiant inscrit* est *non_ additives* par rapport à la dimension *temps*. Une relation rétroactive peut être identifiée entre les étapes 2, 3 et 4. Par conséquent il y a seulement une relation transitive entre l'étape 3 ou 4 et l'étape 1. Des changements peuvent être déclanchées par l'université ou par son environnement. Donc il est obligatoire de passer par les mêmes étapes quand des changements sur des buts et sur des services se produisent ; parce que la définition des mesures et des dimensions est obligatoire.

IV. Conclusion

Ce chapitre est une présentation d'une nouvelle approche orientée processus pour le développement des structures d'entrepôt de données. Nous avons vu que les auteurs de cette méthode emploient les modèles de processus d'affaires au lieu des modèles opérationnels de données pour dériver les structures initiales d'entrepôt de données. Les structures de l'entrepôt de données sont représentées avec le schéma en étoile du modèle dimensionnel.

I. Introduction

Dans ce chapitre, nous abordons une méthode de modélisation dimensionnelle des entrepôts et magasins de données proposées par Ralph Kimball ; dans son livre « Concevoir et déployer un data warehouse ». La proposition de l'auteur est construite autour de deux axes principaux :

1. A partir de l'évaluation des besoins, il est possible d'élaborer une série de data marts performants et faciles à appréhender, fondés sur des modèles dimensionnels en étoile.
2. L'élaboration des data marts qui construiront petit à petit l'entrepôt de données global.

II. Le modèle de données utilisé

Ralph Kimball a consacré sa carrière à la conception et à l'exploitation de base de données d'aides à la décision. L'expérience lui a permis d'établir que l'approche de la modélisation dimensionnelle apporte les meilleurs résultats à la fois en matière de facilité d'utilisation et de performances.

II.1 Avantage de la modélisation dimensionnelle

Le modèle dimensionnel possède un grand nombre d'avantages dont le modèle entité/relation est dépourvu. Premièrement, le modèle dimensionnel est une structure prévisible et standardisée. Les générateurs d'états, outils de requête et interfaces utilisateurs peuvent reposer fortement sur le modèle dimensionnel pour faire en sorte que les interfaces utilisateurs soient plus compréhensibles et que le traitement soit optimisé.

La deuxième force du modèle dimensionnel est que la structure prévisible du schéma en étoile résiste aux changements de comportement inattendus de l'utilisateur. Toutes les dimensions sont équivalentes. Toutes peuvent être vues comme des points d'entrée systématiquement identiques dans la table des faits. La conception locale peut être faite indépendamment des schémas de requête.

Le troisième avantage du modèle dimensionnel réside dans le fait qu'il est extensible à loisir pour accueillir des données et des besoins d'analyse non prévus au départ.

La quatrième force du modèle dimensionnel est qu'il existe un certain nombre d'approches standard permettant de gérer des situations de modélisation courantes dans de nombreux secteurs d'activités.

Le dernier avantage du modèle dimensionnel repose sur la cohorte sans cesse grandissante des utilitaires d'administration et de traitement qui gèrent et exploitent les agrégats.

III. Présentation de la méthode

Une fois les besoins collectés et les données audités, le concepteur de l'entrepôt de données est prêt à lancer la conception logique et physique de l'entrepôt de données. La démarche consiste à transformer les données source en structures de data warehouse finalisées. Les conceptions logique et physique représentent la pierre angulaire du data warehouse. Elles permettront de planifier l'extraction des données et les étapes de la transformation [Kimball 00].

Commencer par la méthode matricielle, rassembler les modèles dimensionnels constituent la conception logique du data warehouse. Prendre la décision des modèles dimensionnels à construire par une approche de planification du haut vers le bas appelée *matrice de l'architecture en bus décisionnel.* Cette matrice oblige les constructeurs de l'entrepôt de données à nommer tous les data marts qu'il est possible de construire et toutes les dimensions impliquées dans ces data marts [Kimball 00].

Après avoir identifié tous les data marts et toutes les dimensions possibles, c'est la conception des tables des faits dans ces data marts. Pour mener à bien l'élaboration de chacune de ces tables des faits Ralph Kimball propose une méthode en quatre étapes.

1. Choisir le data mart source unique ou multisources,
2. Déclarer la granularité,
3. Choisir les dimensions,
4. Choisir les faits.

III.1 Construire la matrice

Il est judicieux de considérer un data mart comme un ensemble de faits liés entre eux et qui doivent être utilisés conjointement. Par exemple, dans une banque, un data mart peut être construit a partir d'un ensemble de faits liés aux activités des comptes :

- montant des dépôts ;
- montant des retraits ;
- montant des frais ;
- nombre de transactions ;
- longueur de la file d'attente au guichet.

En règle générale, les éléments les plus utiles d'un data mart sont les faits numériques que nous rencontrons sur le « marché ». Un data mart est une collection pragmatique de faits liés qui n'est pas nécessairement exhaustive ni exclusive. Un data mart est à la fois un *domaine* et une *application*. Un data mart est avant tout une *collection de faits numériques*.

La conception d'un data mart sera facilitée s'il est issu d'un seul processus métier, tel que Transactions des comptes ou Facturation client. L'un des intérêts de l'approche matricielle réside dans la possibilité de combiner des data marts simples et monosources pour former des data marts plus complexes et multisources.

« Au sein d'une grande entreprise, il est judicieux de prévoir 10 à 30 data marts. Vous vous doutez que chacun de ces data marts débouchera ultérieurement sur un groupe de tables des faits liées entre elles et qui seront utilisées conjointement. Si vous avez moins de 10 data marts, vous courez le risque de leur associer des définitions trop générales. En outre, votre data mart ne servira à rien s'il contient tous les faits et toutes les dimensions possibles. En matière de data mart aussi, il faut diviser pour régner… » [Kimball 00].

III.1.1 Etablir la liste des data marts

Commençons à construire notre matrice en établissant une liste de data marts monosources. Un data mart monosource tourne inévitablement autour d'un seul type de données. Prenons l'exemple d'une grosse compagnie de téléphonie, pour construire la matrice de data mart de l'ensemble du projet de data warehouse. Dans ce secteur d'activité, les sources de données dignes d'intérêt sont nombreuses.

Voici un exemple de groupe de data marts monosources :

- facturation client (particuliers et entreprises) ;
- commandes de services et d'installations ;
- journal des incidents ;
- commandes d'annonces publicitaires ;
- service client et demandes de renseignements sur la facturation ;
- promotion marketing et communication client ;
- détail des appels du point de vue de la facturation ;
- détail des appels du point de vue du réseau ;
- inventaire client (matériel, centraux, caractéristiques) ;
- inventaire réseau (relais, lignes, ordinateurs) ;
- inventaire immobilier (pôles, droits de passage, immeubles, boîtiers de rue, installations souterraines) ;
- main-d'œuvre et salaires ;
- traitement informatique et refacturation ;
- achats aux fournisseurs ;
- livraisons des fournisseurs.

III.1.2 Etablir la liste des dimensions

Les lignes de la matrice de l'architecture en bus décisionnel sont les data marts. Les colonnes représentent les dimensions. Il est possible d'établir séparément les listes des data marts et des dimensions, mais il est probablement plus facile de commencer par dresser la liste des data marts, puis de poursuivre en mettant au jour toutes les dimensions possibles pour chacun.

A cette étape de la conception, il n'est pas encore nécessaire de se montrer particulièrement analytique ou restrictif dans la désignation des dimensions. Il suffit de déterminer s'il convient ou non d'inclure telle dimension dans tel data mart, sans tenir compte du fait que nous ayons ou non un exemple de source de données de production liée à la dimension. Par exemple, dans le premier data mart, la facturation client peut induire les dimension suivantes :

- date de facture;
- client (particulier ou entreprise);
- service ;
- catégorie tarifaire (y compris les promotions);
- fournisseurs de services locaux.

Cette conception n'inclut pas le détail des appels téléphoniques qui font partie de la facture. En effet, ces données seront davantage à leur place dans un data mart consacré à la facturation détaillée, dont les dimension pourraient être :

- appelant ;
- appelé ;
- fournisseurs de services longues distance.

Plus nous avançons dans la liste des data marts, plus la liste des dimensions est grande…

III.1.3 Marquer les intersections

Une fois que les lignes et les colonnes de la matrice sont définis, nous marquons systématiquement toutes les intersections qui indiquent l'existence d'une dimension dans un data mart. En reprenant l'exemple de la compagnie de téléphonie, nous obtenons la matrice de la figure 39.

	Date	Client	Service	Catégorie tarifaire	Fournisseur de services locaux	Appelant	Appelé	Fournisseur de service longue distance	Organisation interne	Employé	Lieu	Type d'équipement	Fournisseur	Article fourni	Météo	Etat du compte
Facturation client	✓	✓	✓	✓	✓			✓			✓					✓
Commandes de services	✓	✓	✓		✓	✓		✓	✓	✓	✓	✓			✓	✓
Journal des incidents	✓	✓	✓		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
Annonces publicitaires	✓	✓		✓		✓			✓	✓	✓					✓
Demandes de renseignement client	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓				✓	✓
Promotion et communication	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓
Détail des appels (facture)	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓
Détail des appels (réseau)	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓
Inventaire client	✓	✓	✓	✓	✓			✓	✓		✓	✓	✓	✓		✓
Inventaire réseau	✓		✓						✓	✓	✓	✓	✓	✓		
Inventaire immobilier	✓								✓	✓	✓	✓				
Main d'oeuvre et salaires	✓								✓	✓	✓					
Coûts informatiques	✓	✓	✓		✓			✓	✓	✓	✓	✓	✓	✓		
Achats aux fournisseurs	✓								✓	✓	✓	✓	✓	✓		
Livraisons des fournisseurs	✓								✓	✓	✓	✓	✓	✓		
Opération sur le terrain combiné	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
Gestion des relations client	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Rentabilité client	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Figure 39. Matrice de l'architecture en bus d'entrepôt de données d'une compagnie de téléphonie.

Chaque ligne de la matrice indique la « connecticité » du data mart correspondant. Sur la figure 35, chaque data mart contient en moyenne 10 dimensions.

III.2 Concevoir les tables des faits en quatre étapes

Lorsqu'un data mart et les dimensions associées ont été identifiés et que leur contenu a été communiqué à toutes les parties concernées, il est vraiment intéressant de se lancer dans cette tâche même avant que le premier data mart à implémenter ait été défini.

La conception logique d'un schéma dimensionnel s'effectue en quatre étapes. Ces quatre étapes consistent à faire quatre choix, dans cet ordre :

1. Le data mart.
2. La granularité de la table des faits.
3. Les dimensions.
4. Les faits.

Etape 1. Choisir le data mart : source unique ou multisources

Dans les cas de figure les plus simples, choisir le data mart revient à choisir la source de données opérationnelle. Un data mart contiendra des bons de commande, des bons de livraisons, des ventes au détail, des règlements ou des communications avec le client. Cependant, dans les situations complexes, il est possible de définir un data mart incluant plusieurs sources opérationnelles.

Donc, consultons les intitulés des lignes de notre matrice en vue de choisir un data mart. La première table des faits de la conception doit être issue d'un data mart monosource.

Etape 2. Déclarer la granularité

Généralement, la granularité de la table des faits choisie est la plus fine possible. Une granularité très fine, telle que la transaction unitaire, l'instantané quotidien unitaire ou la ligne (de commande, de facture, etc.) unitaire, offre de nombreux avantages. Plus le niveau de détail est fin, plus la conception est robuste.

Déclarer la granularité consiste à décrire le contenu de l'enregistrement de la table des faits. Si, par exemple, cet enregistrement représente le total des ventes quotidiennes d'un point de vente, nous tenons notre *granularité*. Si l'enregistrement de la table de faits est une ligne sur un bon de commande, la granularité est la ligne de commande. Il en est de même pour un enregistrement de table des faits concernant une transaction effectuée auprès d'un guichet automatique de banque. Les décisions à prendre dans les étapes 3 et 4 reposent sur une perception claire de la granularité.

Etape 3. Choisir les dimensions

Une fois que la granularité de la table des faits est bien établie, le choix des dimensions est assez simple. Bien souvent, la granularité elle-même détermine une série de dimensions principale ou minimale. Par exemple, la série de dimensions minimale d'une ligne de bon de commande doit inclure la date de la commande, le client, le produit et une dimension dégénérée dédiée au numéro de commande. Dans ce cadre, le concepteur peut ajouter sans problème un grand nombre de dimensions supplémentaires. Dans presque tous les cas, celles-ci prendront une valeur unique dans le contexte des dimensions principales. Par conséquent, des dimensions supplémentaires telles que la date de livraison, les conditions du contrat, l'état d'avancement de la commande et le mode de livraison peuvent être ajoutées à la discrétion du concepteur si la source de données est disponible. Dans la plupart de ces cas, l'ajout d'une dimension est aisé, dans la mesure ou il ne modifie pas la granularité de la table des faits.

Donc, si le choix de la granularité est bon, le choix des dimensions de la table des faits est aisé. Bien souvent, la granularité est exprimé en termes de dimensions principales. *Le niveau des stocks quotidien des articles stock dans un centre de distribution* évoque inévitablement la dimension temps, la dimension article stock. Par ailleurs, une rapide vérification permettra de déterminer la comptabilité d'autres dimensions avec la granularité choisie.

Reprenons l'exemple de la facturation client de la compagnie de téléphonie et optons pour une granularité représentée par la ligne de la facture client mensuelle. Cette granularité spécifie manifestement une dimension temps, une dimension client, une dimension service (la ligne de facture) et, éventuellement, une dimension tarif ou promotion.

L'équipe des concepteurs doit alors faire preuve de créativité. Toutes les dimensions du « portefeuille » principal des dimensions possibles doivent être analysées afin de déterminer si elles sont compatibles avec la granularité. Les dimensions qui prennent une valeur unique à ce niveau de détail sont des candidates viables.

Etape 4. Choisir les faits

La dernière étape consiste à ajouter le plus de faits possibles dans le contexte de la granularité déclarée. La granularité de la table des faits permet également le choix des faits et met en évidence la portée qu'ils doivent avoir. Les tables des faits constituées d'instantanés sont celles qui peuvent intégrer le nombre de faits le plus élevé, parce que tout récapitulatif d'une activité périodique trouve sa place dans la table des faits. Par ailleurs, les tables des faits constituées d'instantanés peuvent intégrer des faits supplémentaires en cas d'identification de nouveaux récapitulatifs intéressants. Les tables des faits constituées de lignes peuvent également contenir plusieurs faits dans la mesure ou, par exemple, une ligne peut être décomposée en quantité, prix de gros, ajustement, remise, montant net et TVA. Les faits qui ne s'ajustent pas exactement à la granularité de l'enregistrement de la table des faits sèmeront le désordre au moment ou les outils de requêtes et autres générateurs d'états tenteront de les combiner selon plusieurs dimensions. Il est tout à fait justifié de créer des enregistrements agrégés ou récapitulatifs, notamment pour des raisons de performances, mais ces faits seront impérativement stockés dans *des enregistrements différents de tables des faits différentes*.

Dans l'exemple de la facturation client des appels téléphoniques, si la granularité est la ligne de facture, nous risquons de nous trouver en présence d'une dizaine, voire plus, de lignes de facture représentant des services, tels que les frais fixes, les frais périodiques, les frais d'installation et les taxes.

III.3 Gérer le projet de modélisation dimensionnelle

Gérer un projet de modélisation dimensionnelle consiste essentiellement à faire circuler le résultat de la conception entre les personnes. Une bonne vision du projet est indispensable à une communication efficace. Quatre outils graphiques aident pour la gestion.

- La matrice de l'architecture en bus décisionnel;
- Le diagramme d'une table des faits;
- Le détail d'une table des faits;
- Le détail d'une table dimensionnelle.

Ces diagrammes doivent combiner les informations descriptives afin de fournir un document de conception complet, qui doit inclure une brève introduction aux concepts de la modélisation dimensionnelle et la terminologie associée.

III.3.1 Matrice de l'architecture en bus décisionnel

La matrice peut être utilisée comme une introduction générale à la conception ; elle procure alors à chaque interlocuteur une vue de ce que sera l'entrepôt de données une fois terminé.

III.3.2 Diagramme de la table des faits

Après avoir élaboré la matrice, il faut préparer un diagramme logique de chaque table des faits. La figure 36 représente le diagramme de la table des faits correspondant à l'exemple des lignes de facturation client des appels téléphoniques.

Le diagramme de la table des faits ne se contente pas d'illustrer les spécificités de la table ; il la situe aussi dans le contexte du data mart. Il nome la table des faits, énonce clairement un aperçu de toutes les dimensions qui ont été identifiées pour l'activité concerné. Cette représentation, qui indique le nom et la description de chaque dimension. Donne une vision du modèle globale. Ces descriptions figurent sur le diagramme afin de renforcer l'utilité des modèles.

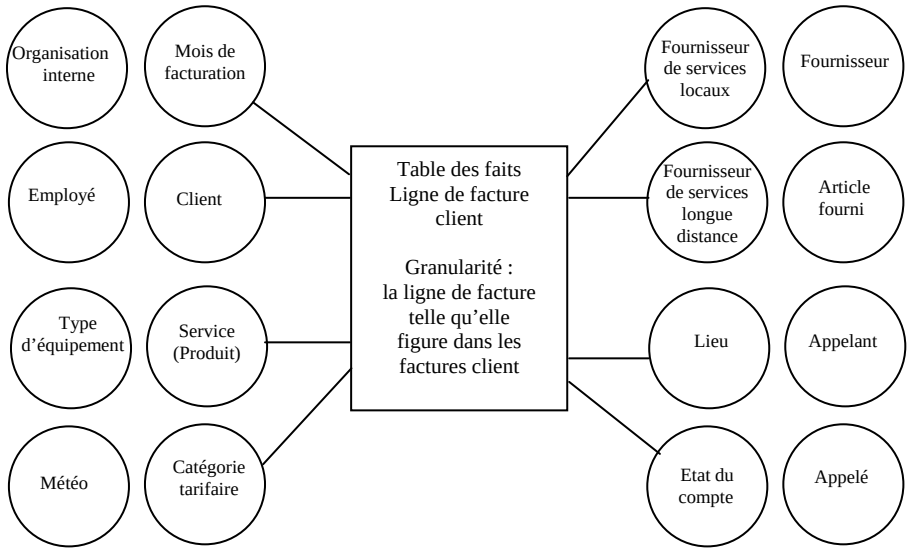


Figure 40. Diagramme de la table des faits de la facturation client des appels téléphoniques.

III.3.3 Détails de la table des faits

Le détail de la table des faits représente la liste complète des faits de la table (voir la figure 41). On y trouve les faits présents physiquement dans la table, les faits dérivés présentés au travers des vues des SGBD et des faits calculés à partir de ceux des deux premiers groupes. Chaque fait doit posséder ses propres règles d'agrégation afin que la personne qui les consulte soit avertie que tel ou tel fait est semi additif ou non additif. Par exemple, les soldes des comptes bancaires sont toujours semi additifs, car ils doivent faire l'objet de moyennes périodiques ; ils peuvent toutefois être additifs dans certaines dimensions (la dimension client, par exemple). D'autres faits, tels que la température, n'est jamais additive, quelles que soient les dimensions, mais des calculs de moyennes peuvent leur être appliqués.

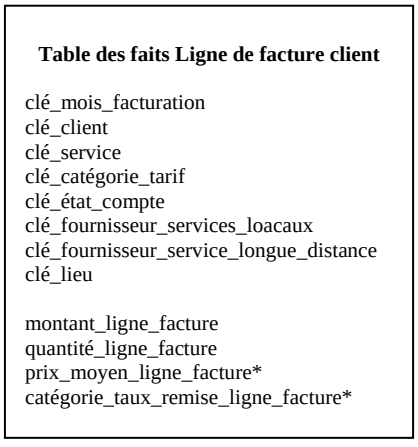


Figure 41. Diagramme de détail de la table des faits présentant les clés dimensionnelles

III.3.4 Détail de la table dimensionnelle

Le deuxième digramme détaillé concerne la table dimensionnelle, comme le montre la figure 42, qui présente les attributs d'une dimension. Chaque dimension a son propre digramme, qui en explicite la granularité. Le diagramme de détail affiche les cardinalités approximatives de chaque attribut dimensionnel et donne à l'utilisateur une vue d'ensemble des nombreuses hiérarchies et relations entre les attributs. Ce diagramme permet également d'inclure des attributs demandés par les utilisateurs mais non encore disponibles ou non prévus dans le projet initial. Les informations nécessaires à ce diagramme comprennent le nom et la description de chaque attribut ainsi que des exemples de valeurs. Il est judicieux de prendre les éventuels noms et descriptions existants à condition qu'ils recouvrent exactement les mêmes notions et qu'il soient rédigés en langage « métier ».

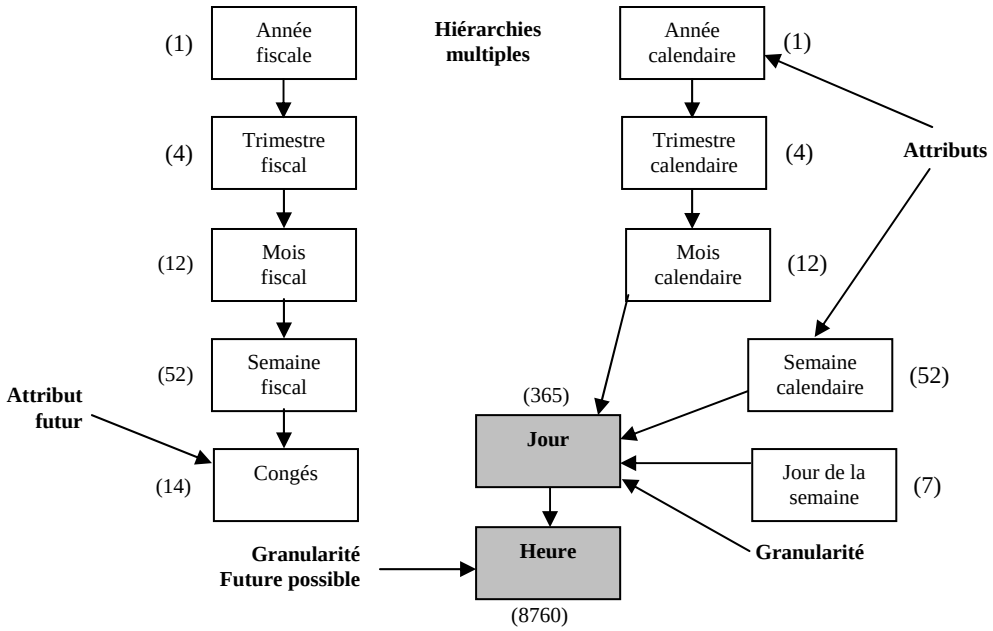


Figure 42. Diagramme de détail de la table dimensionnelle

III.4 Les tâches de l'équipe de modélisation dimensionnelle

Quel que soit le contexte de la modélisation des données, le développement du modèle dimensionnel est un processus itératif. Faire des allers-retours entre les besoins des utilisateurs et les détails des fichiers source sélectionnés.

Les concepteurs de data warehouse les plus expérimentés sont capables de développer seuls une première modélisation dimensionnelle, qu'ils soumettent ensuite à leur équipe pour révision. Cette démarche se pratique, mais elle ne permet pas à l'ensemble de l'équipe d'observer le processus de près et d'apprendre à développer un modèle. Il est important de

garder à l'esprit le double objectif d'un projet de data warehouse : faire en sorte que le travail soit fait et apprendre à l'équipe à devenir autonome.

III.4.1 Créer le modèle initial

La modélisation initiale doit être effectuée par un groupe de modélisation composée de modélisateurs de données, d'administrateurs de bases de données et d'experts des systèmes source, ainsi que de quelques analystes fonctionnels.

Pour développer le modèle initial, la méthode la plus efficace est d'avoir recours aux bons vieux diagrammes. La procédure de développement suivante a prouvé son efficacité [Kimball 00] :

1. Identifier les data marts.
2. Identifier les dimensions possibles souvent en réponse aux propositions de data marts.
3. Remplir la matrice par ces propositions.
4. Identifier les dimensions et la granularité de chaque data mart.
5. Identifier les faits spécifiques de base de chacun des data marts.
6. Affiner et détailler les dimensions.

III.4.2 Identifier les faits de base et les faits dérivés

Les champs des tables des faits *ne se résument* pas aux colonnes issues des données source. Il existe bien d'autres informations à analyser : période de ventes, évolution des pourcentages de réclamations d'une année sur l'autre, rentabilité brute, etc.

Il existe deux types de faits dérivés. Les uns sont additifs et peuvent être calculés à partir des autres faits du même enregistrement de table des faits. Le second type de faits dérivés repose sur des calculs non additifs, tels que les ratios et les faits cumulatifs exprimés à des niveaux de détail différents de celui des faits de base. Un fait cumulatif peut par exemple être un fait de type cumul périodique. Ces faits ne peuvent pas être présentés au seul moyen d'une vue ; ils doivent être calculés au moment de la requête par l'outil de requête ou par le générateur d'états.

III.5 Identifier les sources de chaque table des faits et de chaque table dimensionnelle

L'analyse des données est une tâche préalable qu'il est indispensable de mener à bien pour développer un modèle dimensionnel. Deux niveaux d'analyse sont nécessaires. Premièrement, l'identification des données requises par l'activité. Cette opération aidera à sélectionner les sources de données. Deuxièmement, la compréhension d'une manière approfondie de chacune des sources de données impliquées dans l'itération du data warehouse en cours. Le processus

d'identification des besoins distingue les sources de données formelles et les sources informelles.

III.5.1 Comprendre les sources de données candidates

Les sources de données candidates sont celles qui sont énumérées dans le recueil récapitulatif des besoins des utilisateurs. En général, l'utilisateur déclare avoir besoin d'informations sur les ventes, sur les stocks, et d'informations financières. Il est rare qu'il demande expressément les données concernant les références principales des clients ou l'intégralité des produits. Ce type de source est implicite quand l'utilisateur demande par exemple à connaître les performances par client, par région et par activité.

III.5.2 Origine des sources de données

Il est extrêmement complexe d'établir les responsabilités en matière de qualité et d'intégrité des données du data warehouse. En effet, la plupart des systèmes opérationnels capturent des données opérationnelles essentielles.

III.5.3 Fournisseurs de données

Les responsables des systèmes opérationnels doivent considérer leurs responsabilités de fournir des informations à l'entrepôt de données comme une composante régulière de leurs activités opérationnelles.

III.5.4 Critères de sélection d'une source de données

Il est possible d'identifier les sources de données une fois que les dimensions conformes ont été elles-mêmes identifiées. Ralph Kimball propose quelques critères à prendre en compte :

- Accessibilité des données,
- Longévité de l'alimentation,
- Précision des données,
- Planification du projet.

IV. Conclusion

Dans ce chapitre, nous avons passé en revue la méthodologie utilisée pour la conception d'un entrepôt de données. Nous avons commencé de présenter la matrice de l'architecture en bus, qui permet de mettre sur le papier les data marts et les dimensions. Ensuite, on a présenter la méthode de conception de data marts en quatre étape. Nous avons présenté plusieurs techniques d'élaboration de diagrammes à exploiter en cours du processus de modélisation.

I. Introduction

Pendant les deux dernières décennies les systèmes d'entrepôt de données sont devenus un composant essentiel des systèmes interactifs d'aide à la décision modernes dans de grands organismes. Les systèmes d'entrepôt de données offrent l'accès efficace aux données intégrées et historiques des sources hétérogènes aux directeurs d'entreprise dans leur planification et prise de décision. La construction d'un entrepôt de données par rapport à la technologie de la programmation est encore une jeune discipline et n'offre pas encore des stratégies et des techniques bien établies pour le procédé de développement. Dans cette étude nous allons faire l'évaluation de chacune des cinq méthodologies de développement présentées précédemment selon de divers critères.

II. Les différents critères de comparaisons

Nous avons présenté un ensemble de méthodologies de conception d'entrepôt de données. Ce chapitre propose une étude comparative de ces méthodologies de conception.

Pour cela, nous définissons un ensemble de critères de comparaisons qui prennent en compte les spécificités inhérentes à chaque méthode de conception d'entrepôt de données. Ces critères ont été inspirés des caractéristiques des différentes méthodologies étudiées.

Les critères de comparaison définis sont :

1. ***Le modèle en entrée*** : C'est le modèle existant au niveau des sources de l'entrepôt (modèle d'entreprise, modèle Entité/Relation, modèle de processus d'affaire, la source globale, bases de données dénormalisées).
2. ***Le modèle en sortie*** : C'est le modèle de construction de l'entrepôt de données (le modèle Orienté Objet, modèle multidimensionnel…etc.).
3. ***L'interrogation des données entrepôt*** : ce critère concerne les techniques préconisées par chaque approche pour interroger les données.
4. ***Extraction de comportement*** : concerne la possibilité offerte par une démarche d'extraire le comportement des sources en entrée, en plus de l'extraction classique de données.
5. ***L'architecture du système décisionnel*** : C'est la décomposition d'un tel système, dans le but de distinguer les différentes problématiques de recherche.
6. ***Orientation de la méthodologie*** : Les méthodologies de conception de l'entrepôt de données qui existe dans la littérature peuvent faire partie de trois groupes de base : Orientée Données, Orientée But, Orientée Utilisateur.

II.1. Le modèle en entrée

II.1.1. Méthodologie [M. Gol, S. Riz 99]

La première étape de cette méthodologie de conception des entrepôts de données consiste à rassembler la documentation concernant le système d'information opérationnel pré existant. La conception de ce système d'information opérationnel est faite par le modèle Entité/Relation. Donc le modèle en entrée pour cette méthodologie est le modèle Entité/Relation.

II.1.2. Méthodologie [Kortink, al 99]

D'après [Kortink, al 99], on peut arriver à un modèle dimensionnel en partant du modèle d'entreprise. Les auteurs proposent de définir le modèle de l'entrepôt de données à partir du modèle d'entreprise.

II.1.3. Méthodologie [Ravat, Teste et Zurfluh 00]

Dans cette méthodologie, la construction de l'entrepôt de données à partir d'extraction de données issues de la source globale. La source globale, à partir de laquelle l'entrepôt de données est construit, est obtenue par intégration des systèmes qui supportent l'activité de gestion des transactions courantes de l'entreprise. La source globale peut être représentée au travers d'un modèle classique dédié à la gestion des données transactionnelles (modèle relationnel, modèle E/A).

II.1.4. Méthodologie [M.Boh, A.Ulb 00]

Ici, les structures de l'entrepôt de données sont dérivées à partir des modèles de processus d'affaire. Donc le modèle en entrée pour cette approche de conception des entrepôts de données est le modèle de processus d'affaire.

II.1.4 Méthodologie [Kimball 00]

Cette méthodologie consiste à transformer les données sources (bases de données dénormalisées) en structures d'entrepôt de données finalisées.

II.2. Le modèle en sortie

II.2.1. Méthodologie [M. Gol, S. Riz 99]

Cette méthode semi automatique est basée sur un modèle conceptuel de DW qui s'appelle le Modèle Dimensionnel de Fait « Dimensional Fact Model ». Ce modèle est proposé par les mêmes auteurs de cette méthodologie. Il subit l'influence du modèle dimensionnel qui est

étendu pour prendre en compte les caractéristiques des entrepôts de données. La présentation de la réalité établie par l'utilisation du DFM s'appelle le Schéma Dimensionnel et se compose d'un ensemble de schéma de fait (un pour chaque fait) dont les éléments de base sont des faits, des dimensions et des hiérarchies. Le modèle de fait dimensionnel vise a :

- Supporter efficacement le schéma conceptuel ;
- Fournir un environnement expressif où l'utilisateur peut intuitivement formuler des questions ;
- Permettre au concepteur et aux utilisateurs de discuter de manière constructive afin de raffiner les spécifications de condition ;
- Représente une plateforme solide pour l'étape de conception logique.

II.2.2. Méthodologie [Kortink, al 99]

L'entrepôt de données dans cette méthodologie est modélisé de manière dimensionnelle. Ce modèle permet de représenter les données d'une manière visuelle. La modélisation dimensionnelle est une méthode de conception logique souvent associée aux entrepôts de données. Elle diffère de la modélisation entité relation traditionnelle. Elle vise à présenter les données sous une forme standardisée intuitive et qui permet des accès performants. Le modèle dimensionnel comporte une table de **faits** et plusieurs tables de dimensions. Les tables de dimension sont de petites tables contenant un minimum de champs, décrivant chacune un axe de l'activité. Ainsi, la table de faits est la table centrale qui va contenir tous les enregistrements qui seront analysés par les utilisateurs. Il existe plusieurs modèles dimensionnels :

1. Le schéma en étoile (Star)
2. Le schéma en flocon de neige (Snow Flacke)
3. Le schéma en grappe (Cluster)
4. Le schéma en constellation (Constellation)

Les avantages du modèle dimensionnel sont les suivants :

- Conçu pour un requêteur : performances;
- Peut être modifié sans peine (faits nouveaux, dimensions nouvelles, attributs dimensionnels nouveaux, granularité variable);
- Doit être capable d'intégrer de nouvelles sources.

II.2.3. Méthodologie **[Ravat, Teste et Zurfluh 00]**

Cette approche est basée sur une dichotomie entre deux espaces de stockage au sein du système décisionnel :

L'**entrepôt** est le lieu de stockage centralisé d'un extrait des sources. Il intègre et « *historise* » les données utiles pour la décision. Son organisation doit faciliter la gestion efficace des données et la conservation des évolutions.

Le **magasin** est un extrait de l'entrepôt. Les données extraites sont adaptées à une classe de décideurs ou à un usage particulier. L'organisation des données doit suivre un modèle spécifique qui facilite les traitements décisionnels [Test 00].

Alors cette approche n'organise pas l'entrepôt de données de manière dimensionnelle. Cette activité est réservée aux magasins de données qui améliorent les performances d'interrogation sans se soucier des redondances d'information ; chaque magasin stocke une partie de l'information disponible dans l'entrepôt afin de répondre à un objectif décisionnel précis ou à un groupe d'utilisateurs ayant les mêmes besoins. Les auteurs [F.Ravat, O.Teste, G.Zurfluh] définissent un modèle de données pour les entrepôts, basé sur le paradigme objet. Ce qui apporte des solutions nouvelles répondant aux exigences des entrepôts (conservation des évolutions de manière adaptée aux besoins des applications décisionnelles). Ces contraintes n'existent pas dans les bases de données classiques et les bases de données temporelles. En particulier, les auteurs [F.Ravat, O.Teste, G.Zurfluh] définissent trois concepts.

Le concept d'**objet entrepôt** étend le concept d'objet par intégration de l'aspect évolutif des données. Il se compose d'un état courant, de plusieurs états passés représentant les évolutions détaillées et d'états archivés représentant certaines évolutions de manière résumées. L'intérêt de cette proposition réside dans le fait que les évolutions des données sont conservées avec un niveau de détail pertinent. Les évolutions sont modélisées à partir d'un modèle temporel linéaire, multigranulaire et discret.

Le concept de **classe entrepôt** étend le concept standard de classe pour prendre en compte les caractéristiques d'historisation et d'archivage des objets entrepôt au travers de filtres temporels et de filtres d'archives. Le filtre temporel caractérise les propriétés temporelles d'une classe entrepôt dont les évolutions sont conservées de manières détaillées. Le filtre d'archives caractérise les propriétés archivées dont les évolutions sont résumées. Deux d'archivages sont possibles, soit un archive fort qui résume fortement les valeurs passées en perdant la trace de l'évolution, soit un archivage modéré qui résume les évolutions à un niveau de détail plus élevé.

Le concept d'**environnement** permet de définir simplement, dans le schéma de l'entrepôt, des parties temporelles homogènes, cohérentes, de taille adaptée aux besoins décisionnels et

configurables par l'administrateur. Ce concept permet d'unifier les niveaux d'historisation en offrant trois niveaux de granularité : attribut, classe ou ensemble. En outre, il octroie une grande adaptabilité à l'entrepôt en permettant à l'administrateur de définir des environnements ayant un comportement temporel spécifique afin de conserver uniquement les évolutions pertinentes.

Ce modèle de données doit prendre en compte cinq caractéristiques :

- Gestion efficace des données décisionnelles ;
- Gestion de données complexes ;
- Gestion de l'origine de données et extraction ;
- Gestion de l'historique des données ;
- Mécanisme d'archivage.

II.2.4. Méthodologie [M.Boh, A.Ulb 00]

Les données de l'entrepôt sont présentées par le schéma en étoile du modèle dimensionnel. Cette approche est basée sur la dérivation des structures de l'entrepôt de données à partir des modèles de processus d'affaires. Le processus d'affaires à plusieurs significations dans la littérature. La méthode qui spécifie les systèmes d'affaires et les systèmes d'applications d'affaires fait partie de Modèle d'Objet Sémantique (SOM).

II.2.5. Méthodologie [Kimball 00]

Ralph Kimball est le fondateur de la modélisation dimensionnelle. Pour lui la modélisation dimensionnelle est la seule technique viable permettant de fournir des données aux utilisateurs finaux dans le cadre d'un entrepôt de données.

Le tableau 1 présente une étude comparative des modèles de données proposées dans les méthodologies de conceptions des entrepôts de données déjà étudiées.

En plus du modèle utilisé, nous considérons les caractéristiques suivantes :

1. Gestion efficace des données décisionnelles,
2. Gestion de données complexes,
3. Gestion de l'origine de données et extraction,
4. Gestion de l'historique des données,
5. Mécanisme d'archivage,
6. Supporter efficacement le schéma conceptuel,
7. Fournir un environnement expressif où l'utilisateur peut intuitivement formuler des questions,

8. Représente une plateforme solide pour l'étape de conception logique,
9. Conçu pour un requêteur : performances,
10. Peut être modifié sans peine (faits nouveaux, dimensions nouvelles, attributs dimensionnels nouveaux, granularité variable),

	[M. Gol, S. Riz, 99]	**[Kortink, al, 99]**	**[Rav, Test, Zur 00]**	**[M.Boh, A.Ulb 00]**	**[Kimball 00]**
Gestion efficace des données décisionnelles			X		X
Conçu pour un requêteur : processus OLAP	X	X		X	X
Peut être modifié sans peine (nouveau fait, nouvelles dimensions, granularité variable)		X		X	X
Gestion de données complexes			X		
Fournir un environnement expressif	X	X	X	X	X
Supporter efficacement le schéma conceptuel	X	X	X	X	X
Mécanisme d'archivage			X		
Plateforme solide pour la conception logique	X	X	X	X	X
Gestion de l'origine de données et extraction			X		
Gestion de l'historique des données			X		

Tableau 1 : Comparatif des modèles de données utilisées.

II.3. L'interrogation de l'entrepôt de données

Les outils OLAP (On Line Analytical Process) reposent sur une base de données multidimensionnelle, destinée à exploiter rapidement les dimensions d'une population de données.

La plupart des solutions OLAP reposent sur un même principe : restructurer et stocker dans un format multidimensionnel les données issues de fichiers plats ou de bases relationnelles. Ce format multidimensionnel, connu également sous le nom d'hypercube, organise les données le long de dimensions. Ainsi, les utilisateurs analysent les données suivant les axes propres à leur métier.

Ce type d'analyse multidimensionnelle nécessite à la fois l'accès à un grand volume de données et des moyens adaptés pour les analyser selon différents points de vue. Ceci inclut la

capacité à discerner des relations nouvelles ou non prévues entre les variables, la capacité à identifier les paramètres nécessaires à manier un volume important de données pour créer un nombre illimité de dimensions et pour spécifier des expressions et conditions inter dimensions. Ces dimensions représentent les chemins de consolidation.

L'interrogation des données entrepôt se fait directement pour les approches basées sur le modèle multidimensionnel car elles supportent le processus OLAP.

Alors que l'approche basée sur le paradigme objet ; dans la partie magasin des données l'interrogation se fait à l'aide du processus OLAP (dite indirecte), dans la partie entrepôt de données les auteurs proposent un algèbre pour la manipulation des données qui s'inspire des principales algèbres temporel objet (interrogation directe). Cet algèbre est destiné aux utilisateurs experts qui effectuent des interrogations directement sur l'entrepôt de données. En effet, il est parfois utile d'obtenir rapidement des informations ou bien d'effectuer des analyses ponctuelles directement sur l'ensemble des données décisionnelles stockées dans l'entrepôt. C'est un algèbre (c'est à dire un ensemble d'opérateurs servant de base à l'implantation de langages textuels ou graphiques) qui prend en compte les spécificités inhérentes au modèle de données proposé dans cette méthodologie pour les entrepôts. En particulier, il est nécessaire de pouvoir manipuler les données :

1. Organisées sous forme d'objets entrepôt constitués de plusieurs états (courants, passés et archivés),
2. Organisées sous forme d'états indépendamment les uns des autres,
3. Organisées sous la forme d'un ensemble d'états chronologiquement ordonnés.

Le tableau 2 présente une étude comparative des modèles proposés dans chaque méthodologie de conception des entrepôts de données. Ces modèles manipulant les structures de données suivantes : tableaux à n dimensions (T), relation (R) et objet (O).

En plus du modèle utilisé, nous considérons les caractéristiques suivantes :

1. La correspondance avec le modèle relationnel permettant d'exprimer plus simplement certaines opérations comme les opérations classiques des bases de données,
2. La représentation explicite des hiérarchies (de niveaux de détail) sur les dimensions,
3. La possibilité de manipuler à la fois le contenu et la structure des données,
4. La prise en compte du temps de manière spécifique,
5. Le type de langage utilisé pouvant être de type algèbre (*Al*), calcul (*Ca*), règles (*Re*),
6. Les opérations associées au modèle de données.

	[M. Gol, S. Riz, 99]	[Kortink, al, 99]	[Rav, Test, Zur 00]	[M.Boh, A.Ulb 00]	[Kimball 00]
Modèle de données	T, R	T	O	T	T
Correspondance relationnelle	X	X		X	X
Hiérarchies des dimensions	X		X		
Contenu et structure	X	X		X	X
Gestion du temps	X	X	X	X	X
Type de langage	Re	Re	Al	Ca	Re

Tableau 2 : Comparatif des modèles multidimensionnels proposés dans chaque méthode.

II.5. Extraction de comportement

Le paradigme objet adopté par l'approche dans [F.Ravat, O.Teste, G.Zurfluh], encapsule dans une même entité structure et comportement ; (un processus automatique pour l'extraction de comportement est proposé par les auteurs [F.Ravat, O.Teste, G.Zurfluh]). Alors que aucune proposition ne traite ce problème dans le domaine des entrepôts ; la raison principale étant que les entrepôts de données sont habituellement développés dans un contexte multidimensionnel ou seules les structures sont dérivées par le biais de la technique des vues matérialisées.

II.6. L'architecture du système décisionnel

La mise en place d'un système décisionnel est une tâche complexe qui recouvre de nombreuses difficultés. Dans le but de définir des problématiques parfaitement identifiées et indépendantes les unes des autres, l'architecture fonctionnelle d'un système d'aide à la décision pour l'approche dans [F.Ravat, O.Teste, G.Zurfluh] ; est décomposée en trois niveaux (intégration, construction, structuration). Alors que l'architecture fonctionnelle d'un système d'aide à la décision pour les quatre autres approches est la même parce que ; elles adoptent le modèle dimensionnel. Cette architecture est décomposée en quatre niveaux (conception, alimentation en données, consultation, présentation des résultats).

***II.6.1. Méthodologie* [Ravat, Teste et Zurfluh 00]**

L'architecture fonctionnelle d'un système d'aide à la décision est décomposée en trois niveaux :

L'**intégration** se propose de résoudre les problèmes d'hétérogénéité (systèmes, modèles, formats et sémantiques des données,…) des différentes sources de données par intégration de celles-ci dans une source globale. Cette source globale est virtuelle, c'est à dire que les données utilisées pour la décision restent stockées dans les sources de données et sont extraites uniquement au moment des mises à jour de l'entrepôt. La source globale est décrite au moyen du modèle de données orientées objet standard de l'ODMG [Cattel 1995]. L'intégration s'appuie sur des techniques de bases de données fédérées et réparties [Ravat, Zurfluh 1995] [Ravat 1996].

La **construction** consiste à extraire les données pertinentes pour la prise de décision, puis à les recopier dans l'entrepôt de données. L'entrepôt de données constitue une collection centralisée, de données matérialisées et historiques (conservation des évolutions), disponibles pour les applications de l'entrepôt. Le modèle de l'entrepôt décrivant les données doit supporter des structures complexes et supporter l'évolution des données au cours du temps [Inmon 94].

La **structuration** réorganise l'information décisionnelle dans des magasins de données afin de supporter efficacement les processus d'interrogation et d'analyse, tels que les applications OLAP ("*On-Line Analytical Processing*") et la fouille de données ("*Data Mining*").

II.6.2. Les quatre autres méthodologies

Les bases de données opérationnelles de l'entreprise sont la source d'alimentation de l'entrepôt. Après plusieurs vérifications, les données sont introduites dans l'entrepôt, ce qui, par le modèle dimensionnel sous-jacent, permet d'avoir des données dans n dimensions. Les quatre grandes problématiques auxquelles sont confrontés les entrepôts de données sont :

La **conception** de l'entrepôt est une tâche critique et difficile à mettre en œuvre. Elle doit répondre de façon spécifique aux besoins de l'entreprise et apporter les bonnes réponses. La modélisation dimensionnelle permet de voir un entrepôt sous la forme d'un **cube** de données dont les arêtes portent une étiquette. Chaque donnée est donc modélisée en n dimensions et l'on va pouvoir extraire les informations par ligne, par colonne ou par tranche.

En ce qui concerne **l'alimentation**, personne n'alimente l'entrepôt au sens de la saisie de données car ce dernier est alimenté par les données extraites des bases en-dessous. Un moniteur (figure 43) est chargé de garder la cohérence des données et de répercuter vers l'entrepôt toutes les mises à jour des bases opérationnelles. Chaque moniteur envoie ses données vers un intégrateur qui va alimenter l'entrepôt de données. Il vérifie et valide les données, les met en forme afin de garder une cohérence entre les données issues de bases

différentes, les analyse et les décrit, afin que les utilisateurs puissent interroger l'entrepôt avec les mots de leur métier.

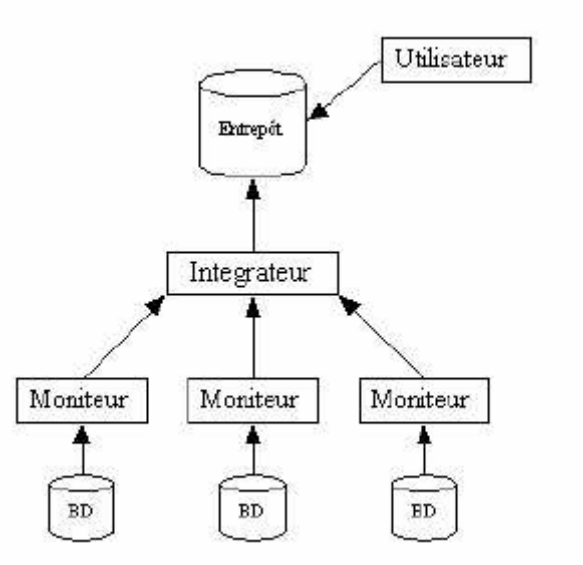


Figure 43. Architecture d'intégration des données vers l'entrepôt

L'interrogation de l'entrepôt pose aussi un problème. En effet, il s'agit de bien cibler le type de requêtes à implémenter, car sur des millions, voire des milliards d'enregistrements, une requête peut prendre plusieurs heures avant d'afficher un résultat. Ainsi, il faudra faire la différence entre l'interrogation prédéfinie et l'interrogation ponctuelle. Une interrogation prédéfinie doit se dérouler en un minimum de temps et est faite périodiquement. A l'inverse, la requête ponctuelle peut être réalisée afin d'approfondir la tendance de certaines mesures lors de l'étude d'une activité. En général, lors de la conception physique d'un entrepôt, on ne prend en compte que les requêtes prédéfinies. En conséquence, le temps de réponse des requêtes ponctuelles est en général bien supérieur à celui des requêtes prédéfinies.

La **présentation** est vue comme l'élément essentiel de l'entrepôt par les personnes qui vont l'interroger. En effet, les utilisateurs ne sont pas des informaticiens et ne connaissent pas le langage SQL ; ils doivent néanmoins poser des questions à l'entrepôt et il n'est plus question d'avoir des pages de résultats comme lors de l'interrogation des bases de données. Un état schématique, avec des mots précis, le tout sous forme graphique, est attendu par les usagers. Il existe plusieurs façons de faire : soit l'entrepôt possède son propre outil de présentation (reporting), soit on pourra faire appel à des logiciels annexes qui permettront ce travail.

Le tableau 3 présente une étude comparative de l'architecture fonctionnelle d'un système d'aide à la décision proposée dans chaque méthodologie de conception des entrepôts de données.

	[M. Gol, S. Riz, 99]	[Kortink, al, 99]	[Rav, Test, Zur 00]	[M.Boh, A.Ulb 00]	[Kimball 00]
Intégration			**Résolution des problèmes d'hétérogénéité**		
Construction (conception)	**Modélisation cube**	**Modélisation cube**	**Extraction des données**	**Modélisation cube**	**Modélisation cube**
Structuration			**Réorganise l'information**		
Alimentation	**Extraction des données**	**Extraction des données**		**Extraction des données**	**Extraction des données**
Interrogation	**Processus OLAP**	**Processus OLAP**	**OLAP partie magasins**	**Processus OLAP**	**Processus OLAP**
Outil de présentation	**Reporting**	**Reporting**	**Logiciels annexes**	**Reporting**	**Reporting**

Tableau 3 : Comparatif de l'architecture fonctionnelle d'un système d'aide à la décision proposée dans chaque méthodologie.

II.7. Orientation de la méthodologie

II.7.1. Méthodologies Orientée Données

« Inmon » un fondateur des entrepôts de données déclare que les conditions sont la dernière chose à considérer dans le cycle de vie de développement de support de décision ils sont compris après l'entrepôt de données.

- Golfarelli, Maio et Rizzi proposent une méthode semi-automatique pour établir un modèle dimensionnel d'entrepôt de données. La stratégie de développement d'entrepôt de données est basée sur l'analyse du modèle de corporation de données et des transactions appropriées.
- Franck Ravat et Olivier Teste proposent un modèle de données pour les entrepôts de données, basé sur le paradigme objet. L'approche ignore les besoins des utilisateurs d'entrepôt de données a priori. Les buts de compagnie et les exigences d'utilisateurs ne sont pas reflétés du tout. Les besoins d'utilisateur sont intégrés dans le deuxième cycle.

- Mark A.R Korkink et Daniel L. Moody préfèrent rassurer les concepteurs d'entrepôt et les décideurs, en donnant une méthode de conception basée sur un modèle existant. Ils proposent de définir le modèle de l'entrepôt de données à partir du modèle d'entreprise.

II.7.2. Méthodologies Orientée Buts

- Böhnlein et Ulbrich-vom Ende proposent une approche qui est basée sur le SOM (Modèle d'Objet Sémantique) afin de dériver la structure initiale d'entrepôt de données. La première étape du processus détermine les buts de l'entreprise. À partir des buts de compagnie, des buts processus-spécifiques d'affaires sont choisis. Chaque but est mesuré par au moins un indicateur. Les indicateurs visent à réaliser des buts à long terme.

II.7.3. Méthodologie Orientée Utilisateurs

- Ralph Kimball propose une méthode à cinq étapes pour la conception des entrepôts de données.

Le tableau 4 présente une étude comparative de l'orientation de chaque méthode de conception des entrepôts de données présentées toute au long de ce mémoire.

	[M. Gol, S. Riz, 99]	**[Kortink, al, 99]**	**[Rav, Test, Zur 00]**	**[M.Boh, A.U0lb 00]**	**[Kimball 00]**
Orientée But				X	
Orientée Données	X	X	X		
Orientée utilisateurs					X

Tableau 4 : Comparatif de l'orientation de chaque méthodologie.

Le tableau 5 présente Les spécifications correspondant pour chaque approche.

	[M. Gol, S. Riz, 99]	**[Kortink, al, 99]**	**[Rav, Test, Zur 00]**	**[M.Boh, A.Ulb 00]**	**[Kimball 00]**
Modèle en Entrée	**Modèle relationnel**	**Modèle d'entreprise**	**Source hétérogène**	**Modèle de processus d'affaires**	**base de données dénormalisée**
Modèle en Sortie	**dimensionnel**	**Orientée Objet**	**dimensionnel**	**dimensionnel**	**dimensionnel**
Processus OLAP	**Supporter le processus OLAP**	**Supporter le processus OLAP**	**Au niveau des magasins**	**Supporter le processus OLAP**	**Supporter le processus OLAP**
Extraction de comportement	**Non**	**Non**	**Oui**	**Non**	**Non**

Tableau 5 : Les spécifications correspondant à chaque approche.

III. Comparaison et Conclusion

L'industrie, ayant emboîté le pas à la recherche, des résultats sont apparus assez rapidement, et plusieurs produits opérationnels, se disputent un marché des plus juteux. En parallèle, plusieurs travaux sont consacrés, à la modélisation, et implémentation des entrepôts et magasins de données.

Le but de ce chapitre était de rendre compte de la variété des techniques et des approches qui ont commencé à voir le jour tout en essayant de dégager les principaux problèmes sur lesquels un effort de recherche est encore nécessaire. Pour cela, nous avons défini un ensemble de critères de comparaisons qui prennent en compte les spécificités inhérentes à chaque méthode de conception d'entrepôt de données.

I. Introduction

Les travaux présentés le long de ce mémoire relatifs aux entrepôts de données abordent globalement deux problématiques :

- La première traite essentiellement de l'organisation des données. Cette organisation dite multidimensionnelle vise à supporter efficacement les analyses OLAP en offrant une vision des données adaptées et les temps de réponse sont accélérés en calculant de nombreux pré-agrégats.
- La deuxième, propose que la conception d'un système décisionnel doit être basée sur la séparation de l'entrepôt et des magasins de données. En effet, l'objectif de ces deux espaces de stockage est différent et les problèmes à résoudre divergent : l'entrepôt regroupe toute l'information, dédiée à un thème, un métier, une analyse.

L'objet de ce chapitre est de définir une première mouture d'une démarche, d'élaboration d'un système décisionnel basé sur la dualité entre les deux méthodologies suivantes :

1. définition des étapes essentielles pour la conception d'un entrepôt de données [M.Gol, S.Riz 99].
2. Une méthode de construction : du modèle relationnel vers le modèle dimensionnel [Kortink, al 99].

II. Pourquoi ce choix

L'approche proposée par Michael Böhnlein, et Achim Ulbrich-vom Ende [M.Boh, A.Ulb 00] est basée sur la dérivation des structures de l'entrepôt de données à partir des modèles de processus d'affaires. Les auteurs proposent un modèle d'actions orienté par processus d'affaires pour dériver les structures initiales de l'entrepôt de données. Le concept de processus d'affaires figure rarement dans la littérature des entrepôts de données. L'approche proposée consiste simplement à supprimer l'information devenue obsolète. Cette approche est donc limitée car elle ne propose pas de mécanismes plus souples comme l'archivage de certaines données.

L'approche proposée par Frank Ravat, Olivier Teste et Gilles Zurfluh [Ravat, Teste, et Zurfluh 00] traite de la modélisation orientée objet pour la conception d'un entrepôt de données. Un des aspects majeurs de cette modélisation est l'extension du concept de classe par celui de classe entrepôt, défini au travers d'un filtre temporel et d'un filtre d'archives ainsi que d'une fonction de construction. Les filtres gèrent l'évolution des données sous une forme pertinente (détaillée ou archivée) pour l'aide à la décision. La fonction de construction définit la structure et les données des classes entrepôt à partir d'un processus d'extraction appliqué sur une source globale intégrée. L'intérêt de processus d'extraction est qu'il combine les structures et le comportement des données. L'extraction

du comportement des classes s'effectue au travers du concept de matrice d'usage. Ainsi une proposition consiste à modéliser les magasins de données à un niveau d'abstraction élevé. Cette approche est limitée car elle traite de la modélisation orientée objet pour la conception d'un entrepôt de données. Ce modèle subit l'influence du modèle objet standard de l'ODMG qui n'a pas réussi dans le domaine de développement des SGBD.

Kimball [Kimball 96] a introduit la notion d'entrepôt de données, mais sans donner de méthodologie formelle de construction, mis à part qu'il faut une base de données dénormalisée.

Les deux autres approches, utilisent la modélisation multidimensionnelle qui est adaptée à l'interrogation et l'analyse des magasins de données. En effet, l'approche multidimensionnelle représente les données conformément aux traitements effectués par les décideurs, suivant les différents axes d'analyses possibles. Le modèle multidimensionnel se compose de **faits** contenant les mesures à analyser et de **dimensions** contenant les paramètres de l'analyse. Dans chaque dimension, les paramètres sont organisés hiérarchiquement en niveaux de détail.

III. Proposition

Dans cette partie, nous prenons en considération la proposition des auteurs [Kortink et al, 1999] qui est : partir d'un modèle existant le modèle de l'entreprise, puis le transformer en modèle dimensionnel ; nous définissons un modèle d'élaboration des systèmes décisionnel. Notre modèle subit l'influence des deux approches proposées respectivement par S.Rizzi et M.Golfarelli [M.Gol, S.Riz 99] et [Kortink et al 99].

Notamment, notre modèle adopte les principales étapes proposées par S.Rizzi et M.Golfarelli [M.Gol, S.Riz 99], intégrer l'étape de *classification des entités* qui est proposée dans [Kortink et al 99], et nous proposons une étape de *définition des objectifs* en utilisant le concept de *cas d'utilisation d'UML*. Ainsi les deux autres étapes pour la *production du schéma conceptuel orienté données / orienté but* et une étape de *confrontation des deux schémas* pour produire le schéma dimensionnel. Nous allons spécifier une solution permettant de définir l'entrepôt de données en sept étapes successives dont les deux premières peuvent être réalisées en parallèle:

1. Conception orienté données :
    a. Analyse de système d'information ;
    b. Classification des entités
    c. Production du schéma conceptuel orienté données,
2. Conception orienté but
    a. Spécification des demandes.
    b. Modélisation fonctionnelle (Cas d'Utilisation d'UML)
    c. Production du schéma conceptuel orienté but,
3. Confrontation des deux schémas,
4. La production du schéma conceptuel (orienté but /donnée),
5. Raffinement et validation du schéma dimensionnel,
6. La conception logique,
7. La conception physique.

Pour apporter plus de clarté dans la présentation de ce modèle ; nous proposons un diagramme qui facilite la présentation des étapes de notre proposition.

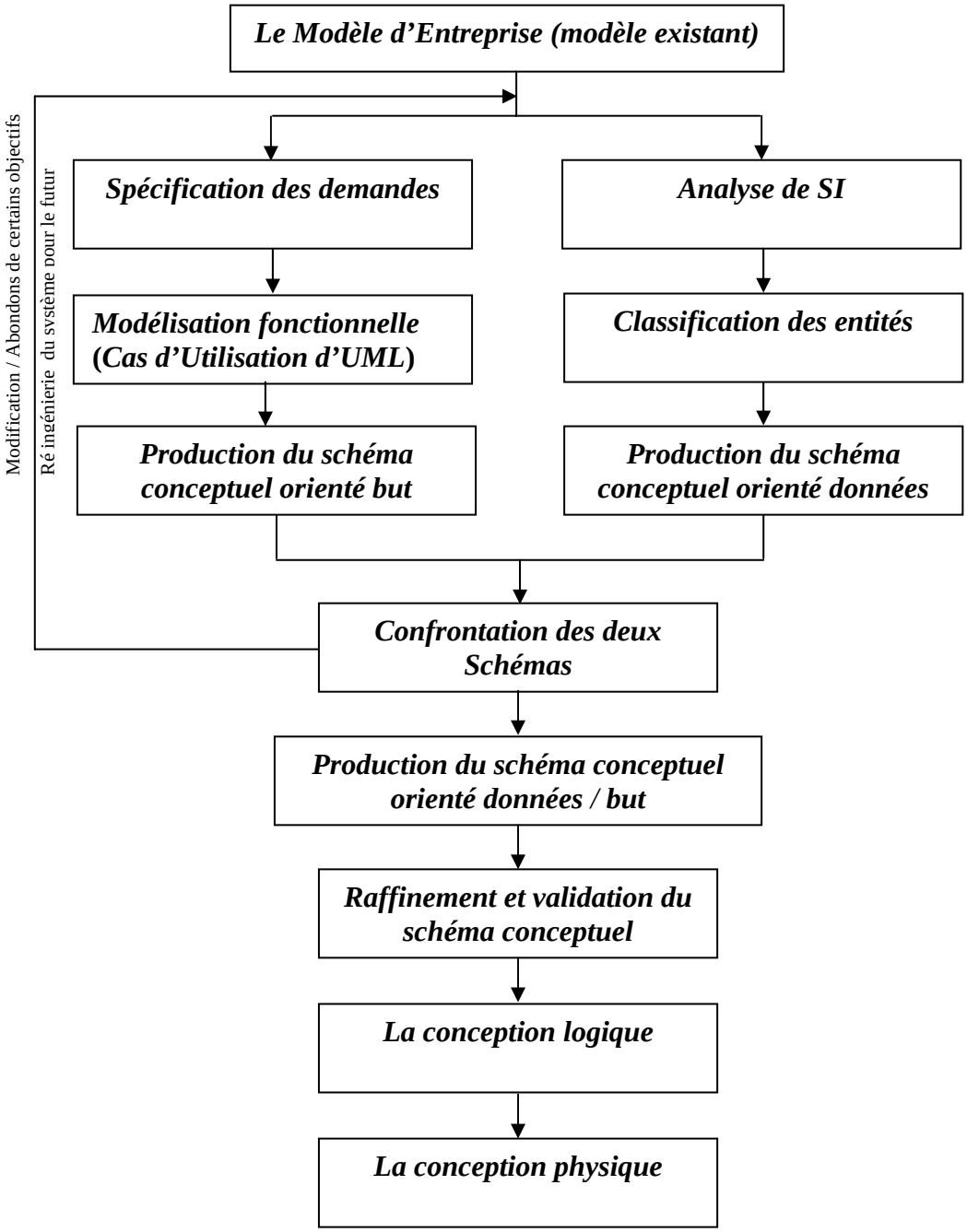


Figure 44. *Digramme de spécification des étapes principales*

III. 1. Conception orientée données

III. 1.1. Analyse de SI

Le but de cette étape est de rassembler la documentation concernant le système d'information opérationnel pré existant. Le concepteur en analysant le système opérationnel doit :

- Exploitez l'expérience du gestionnaire de base de données afin de découvrir les données en sortie possible ou les données anormales,

- Sélectionner les données sources opérationnelles en considérant la qualité de données et la stabilité de leurs schémas,
- Déterminez quelles données peuvent être utiles pour l'intégration afin d'obtenir une vue complète du domaine de base de données.

III. 1.2. Classification des entités

La première étape pour créer un modèle dimensionnel à partir du modèle d'entreprise est de classer ses entités. Elles se décomposent en trois catégories :

1. Entité de transaction : Elles enregistrent les détails d'événements particuliers comme les salaires, les réservations d'hôtels.
2. Entité Composante : Elle est directement liée à l'entité de transaction par une relation multiple.
3. Entité de classification : Ce sont des entités qui sont apparentées à des entités composantes par une chaîne de relations multiples.

III.1.3. Production du schéma conceptuel orienté données

Pour produire le schéma conceptuel orienté données, Nous adoptons la proposition dans [Kortink, al 99] ; qui utilise deux opérateurs pour produire le schéma conceptuel à partir du modèle entité / relation. Ces deux opérateurs sont : Réduction de la hiérarchie, et l'agrégation.
Quelque soit le modèle choisi, la procédure de création de la table des faits et la création de ses clés est :

1. Une table des faits est formée pour les entités de transaction les plus pertinentes.
   La clé de cette table est une combinaison des clés des tables de dimension.
2. Si une relation hiérarchique existe entre deux entités de transaction, l'entité fille hérite de tous les attributs de son père.
3. Les attributs numériques appartenant aux entités de transaction doivent êtres agrégés par les clés.

Les tables des faits avec les mêmes clés primaires doivent être fusionnées.

III. 2. Conception orientée buts :

III. 2.1. Spécification des demandes

Cette étape concerne la collection et le filtrage des demandes des utilisateurs. Elle implique le concepteur et les utilisateurs final de l'entrepôt de données, et produit en sortie les spécifications concernant le choix des faits d'un coté, les indications préliminaires concernant le cahier de charge d'un autre côté.

En particulier, le choix des faits est basé sur la documentation de système d'information produit par l'étape précédente. Le fait est un concept fondamental dans le processus de prise de décision, et correspond typiquement aux événements qui se produisant dynamiquement dans le monde d'entreprise. Si le système d'information opérationnel est doté d'un ou de plusieurs schémas E/R, un fait peut être représenté soit par une entité ou par une relation. En général, les entités ou les relations qui représentent fréquemment les archives de mise à jour sont des bons candidats pour définir des faits.

Le cahier de charge préliminaire est exprimé en langage naturel et il vise à permettre au concepteur d'identifier les dimensions et les mesures durant la conception physique ; pour chaque fait, il devrait indiquer les mesures et les agrégations les plus intéressantes.

III. 2.2. Modélisation fonctionnelle

Après avoir identifié les acteurs qui interagissent avec le système, nous y développons un premier modèle UML, pour pouvoir établir précisément les frontières du système. Ensuite, nous apprenons à identifier et décrire les cas d'utilisation, qui nous permettent de préciser le point de vue fonctionnel, en détaillant les différentes façons dont les acteurs peuvent utiliser le système.

III. 2.2.1. Identification des acteurs

Un acteurs représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié.

Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.

III. 2.2.2. Identification des cas d'utilisation

Un cas d'utilisation (« use case ») représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier.

Chaque cas d'utilisation spécifie un comportement attendu du système considéré comme un tout, sans imposer le mode de réalisation de ce comportement. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera.

L'ensemble des cas d'utilisation doit décrire exhaustivement les exigences fonctionnelles du système. Chaque cas d'utilisation correspond donc à une fonction métier du système, selon le point de vue d'un de ses acteurs.

III. 2.3. Production du schéma conceptuel orienté but

Pour produire le schéma conceptuel orienté but. Nous proposons d'exploiter la modélisation fonctionnelle, pour identifier en premier lieu les mesures susceptibles de répondre aux besoins des différents acteurs. En deuxième lieu, nous proposons d'associer ces mesures aux différents axes d'analyse, toujours en fonction des besoins. Ces différents axes correspondent aux dimensions du modèle multidimensionnel. Nous aboutissons ainsi à un modèle en étoile qui correspond aux objectifs des décideurs, mais qui n'est pas forcément cohérent avec les données disponibles.

III. 3. Confrontation des deux schémas

Deux cas pour la confrontation :

1. Si les données répondent aux besoins alors
    - Produire le schéma orienté données / but
2. Si les données ne répondent pas aux besoins alors plusieurs alternatifs doivent être envisagées :
    - Modification de certains objectifs : pour les adapter aux données existantes
    - Abondons certains objectifs : qui ne peuvent être atteint actuellement
    - Réingénierie du SI pour le futur : il est intéressant de préconiser une réingénierie du système opérationnel existant, pour qu'il puisse répondre aux objectifs décisionnels dans le futur.

III. 4. Production du schéma conceptuel (orienté but/données)

Après confrontation des deux schémas, le nouveau schéma conceptuel correspondra au schéma orienté but (éventuellement modifié), et enrichi des détails du schéma orienté données (attributs des dimensions).

III. 5. Raffinement et validation du schéma dimensionnel

Cette étape vise également à valider le schéma conceptuel produit dans l'étape précédente, pour cette raison nous adoptons la proposition dans [Kortink, al 99] de quatre sous étapes suivantes :

1. Combinaison des tables de fait,
2. Combinaison des tables de dimension,
3. Traitement des relations multiples (n, n),
4. Transformation des relations SuperTypes.

III. 6. La conception logique

Plusieurs questions doivent être abordées afin d'obtenir une définition correcte du schéma logique d'entrepôt de données. La conception logique reçoit en entrée le schéma dimensionnel, une charge de travail et un ensemble d'information additionnelle (fréquences de mise à jour, espace disque total disponible, etc.) pour produire un schéma d'entrepôt de données qui devrait réduire au minimum le temps de réponse des demandes en respectant la contrainte d'espace disque. La mise à jours des entrepôts de données est faite périodiquement, et pendant ce processus l'entrepôt est indisponible pour répondre aux questions des utilisateurs. Ainsi, le processus de mise à jour n'est pas directement affecté à l'exécution de l'entrepôt de données. Quatre étapes à suivre pour l'étape conception logique.

1. Matérialisation des vues,
2. Transformation dans des tables,
3. Division verticale des tables de fait,
4. Division horizontale des tables de fait.

III. 7. La conception physique

La conception physique concerne la sélection d'index optimal. La sélection d'index a un rôle crucial pour déterminer les performances de l'entrepôt de données. La maintenance des indexes pendant la mise à jours n'est pas nécessaire, et l'obtention d'accès à plusieurs structures complexes est possible.

L'étape de sélection d'index vise à déterminer un meilleur sous ensemble d'index pour chaque type d'index, par rapport à la fonction de coût. Un meilleur sous ensemble est celui qui réduit au minimum le coût d'accès des requêtes sous une contrainte d'espace variable d'une application à une autre. Puisque les requêtes exigent habituellement une ou plusieurs jointure pour s'exécuter, la sélection d'index prend en considération différent algorithmes de jointure.

IV. Classification de la proposition

Dans cette section nous classifions cette proposition par rapport aux critères choisis.

- Le modèle en entrée : c'est le modèle d'entreprise (modèle existant),
- Le modèle en sortie : c'est le modèle multidimensionnel,
- L'interrogation de l'entrepôt de données : l'interrogation des données entrepôt se fait directement car elle supporte le processus OLAP,
- Extraction de comportement : L'entrepôt de données est développé dans un contexte multidimensionnel ou seule les structures sont dérivées par le biais de la technique des

vues matérialisées or l'extraction de comportement est bien adapté à l'approche [Ravat, Teste, et Zurfluh 00],

- L'architecture du système décisionnel : est décomposée en quatre niveaux (conception, alimentation en données, consultation, présentation des résultats),
- Orientation de la démarche : c'est une démarche à orientation hybride but / données

V. Conclusion

Dans ce chapitre, on a proposé une démarche pour la définition d'un processus de construction de l'entrepôt de données à partir du modèle d'entreprise. L'intérêt de notre proposition réside dans le fait qu'elle aborde tant les étapes principales pour la conception d'un entrepôt de données proposées par S.Rizzi et M.Golfarelli que l'intégration des deux étapes proposées par Daniel L. Moody et Mark A.R. Kortink. Ainsi l'intégration du concept de *Cas d'utilisation d'UML*, pour produire un schéma multidimensionnel orienté but / orienté données.

Conclusion

Notre travail traite de la modélisation conceptuelle des données dans les systèmes d'aide à la prise de décision. Ces systèmes contiennent l'information utile aux décideurs et conservent l'historique de cette information pour supporter efficacement les analyses et les processus de prise de décision.

Les entrepôts de données répondent aux besoins des décideurs et des entreprises de disposer d'outils puissants et adaptés pour exploiter l'énorme masse de données rendues potentiellement accessibles avec l'émergence de l'Internet et de l'Intranet. Un entrepôt de données regroupe dans un format homogène et utile pour l'aide à la décision des données provenant de plusieurs sources de production pouvant être réparties et avoir des formats variés.

L'objet de ce mémoire était de rendre compte de la variété des techniques et des approches, de conception des entrepôts de données, qui ont commencé à voir le jour tout en essayant de dégager les principaux problèmes sur lesquels un effort de recherche est encore nécessaire.

Après avoir étudié les différentes approches recensées dans la littérature des systèmes décisionnels, les critères de comparaison établis nous ont permis de mettre en évidence les points forts et les points faibles de chaque approche. Parmi ces critères nous avons accordé un intérêt particulier, à l'orientation de la démarche de conception. Ainsi, si les démarches orientés but ou utilisateurs, semblent plus à même de répondre aux exigences des décideurs, les approches orientés données, offrent une meilleure exploitation des données, et parfois même la possibilité d'automatiser partiellement le processus de conception.

Ceci nous a conduit à établir la première mouture d'une démarche à orientation hybride (but / données), basée sur la confrontation de deux schémas conceptuels, afin de vérifier la disponibilité des données nécessaires aux besoins décisionnels, et éventuellement revoir à la baisse les objectifs ou proposer une reconception des systèmes opérationnels pour tenir compte de ces données.

Perspectives

Les perspectives que nous envisageons de conduire sont les suivantes :

- **Proposition d'une méthode pour la conception des systèmes décisionnels**. A l'heure actuelle, aucune méthode complète n'est proposée pour aider les entreprises à construire un système décisionnel. La proposition d'une solution pour concevoir un système décisionnel en se basant sur une méthode complète (modèles, formalismes, démarche et outils).
Cette méthode se basera sur la démarche proposée dans ce mémoire, et aura pour objectif d'offrir un cadre précis et complet pour concevoir des systèmes décisionnels comportant des données complexes et évolutives.

- **Etude de la méta-modélisation des systèmes décisionnels**. Nous voulons avoir une étude globale concernant la méta-modélisation des systèmes décisionnels qui couvre l'ensemble des composantes de l'architecture : l'intégration des sources, la construction de l'entrepôt, la réorganisation dans les magasins et la manipulation multidimensionnelle des données. Cette méta-modélisation doit permettre l'intégration des contraintes de la source globale, de l'entrepôt et des magasins ; ces contraintes portent aussi bien sur les modèles que sur la démarche de conception du système décisionnel.

Références bibliographiques

1. [Agr 1995] AGRAWAL R., GUPTA A., SARAWAGI A., « *Modeling Multidimensional Databases* », IBM Research Report, September 1995, (proceedings of ICDE'97).

2. [Bukhres, Elmagarmid 1993] Bukhres O.A., Elmagarmid A.K., "*Object-Oriented Multidatabase Systems - A solution for Advanced Applications*", Prentice Hall, ISBN 0-13-103813-2, 1993.

3. [Cattel 1995] Cattel R.G.G, "*ODMG-93 Le Standard des bases de données objet*", Thomson publishing, ISBN 2-84180-006-7, 1995.

4. [Chaudhuri, Dayal 1997] Chaudhuri S., Dayal U., "*An Overview of Data Warehousing and OLAP Technology*", ACM SIGMOD Record, 26(1), 1997.

5. [Codd 1993] Codd E.F., "*Providing OLAP (on-line analytical processing) to user-analysts: an IT mandate*", Technical Report, E.F. Codd and Associates, 1993.

6. [Fauvet, Dumas 1999]Fauvet M-C., Dumas M., Scholl P-C., A representation-independent temporal extension of ODMG's Object Query Language, *BDA'99*, Bordeaux (France), Octobre 1999.

7. [Fer.O, Sin.E 1994] Ferstl, O.K.; Sinz, E.J.: *From Business Process Modeling to the Specification of Distributed Business Application Systems - An Object-Oriented Approach***,** Bamberger Beitrag zur Wirtschaftsinformatik, No. 20, Bamberg, June 1994.

8. [FeSi90] Ferstl O.K., Sinz E.J.: *Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell* (SOM). In: Wirtschaftsinformatik Band 32, Heft 6 (1990), 566 – 581.

9. [FeSi91] Ferstl O.K., Sinz E.J.: *Ein Vorgehensmodell zur Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell* (SOM). In: Wirtschaftsinformatik Band 33, Heft 6 (1991), 477 – 491.

10. [FeSi93a] Ferstl, O.K.; Sinz, E.J.: *Der Modellierungsansatz des Semantischen Objektmodells* (SOM); Bamberger Beiträge zur Wirtschaftsinformatik, Nr. 18, Bamberg, 1993.

11. [FeSi+94] Ferstl O.K., Sinz E.J., Amberg M., Hagemann U., Malischewski C.: Tool-Based *Business Process Modeling Using the SOM Approach. Proc*. IFIP World Computer Congress, Hamburg 1994.

12. [Golfarelli, et al 1998] Golfarelli M., Maio D., Rizzi S., "*Conceptual Design of Data Warehouses from E/R Schemes*", In Proceedings of the 31st Hawaii International Conference on System Sciences, Kona (Hawaii, USA), 1998.

13. [Golfarelli, Rizzi 1999] Golfarelli M., Rizzi S., "*Designing the data warehouse: key steps and crucial issues*", Journal of Computer Science and Information Management, vol. 2, n. 3, 1999.

14. [Golfarelli, D. Maio, et S.Rizzi 2000] *Applying vertical fragmentation techniques in logical design of multidimensional databases*. In Proc. 2nd Int. Conf. on Data Warehousing and Knowledge Discovery, pages 11–23, 2000.

15. [Golfarelli, D. Maio, and S. Rizzi 1998] *The Dimensional Fact Model: a Conceptual Model for Data Warehouses*. International Journal of Cooperative Information Systems, 7(2&3) (1998) 215 247.

16. [Gupta, Mumick 1995] Gupta A., Mumick I.S., "*Maintenance of Materialized Views: Problems, Techniques, and Applications*", IEEE Data Engineering Bulletin, 1995.

17. [Gyssen, Lakshmanan 1997] Gyssen M., Lakshmanan L.V.S., "*A Foundation for Multi- Dimensional Databases*", In Proceedings of 23rd International Conference on Very Large Data Bases - VLDB'97, Athens (Greece), August 25-29 1997.

18. [Inmon 1994] Inmon W.H., "*Building the Data Warehouse*", John Wiley&Sons, ISBN 0471- 14161-5, 1994.

19. [Kimball 2000] Ralph.Kimball, « *Concevoir et déployer un data warehouse*».

20. [Kimball 1996] Kimball R., "*The data warehouse toolkit*", John Wiley and Sons, 1996.

21. [Kortink et al, 1999] Mark A.R. Kortink, Daniel L. Moody: From entities to Stars, Snowflakes, Clusters, Constellations and Galaxies: *A Methodology for Data Warehouse Design, Proceedings 18th international conference on Conceptual Modeling* 15-18 Novembre 1999 P58-82.

22. [M.Boh, A.Ulb 1999] Boehnlein, M.; Ulbrich-vom Ende, A.: *Deriving Initial Data Warehouse Structures from the Conceptual Data Models of the Unterlying Operational Information Systems, in: Proceedings of the ACM Second International Workshop on Data Warehousing and OLAP* (DOLAP'1999, Kansas City, 6. November), 1999, pp. 15-21.

23. [M.Boh, A.Ulb 2000] Michael Böhnlein, Achim Ulbrich-vom Ende, « *Developing Data Warehouse Structures from Business Process Models* », University of Bamberg, Feldkirchenstr.21,D96045Bamberg,Germany{michael.boehnlein,achim.ulbrich}@sowi.uni-bamberg.de.

24. [Mendelzon, Vaisman 2000] Mendelzon A.O., Vaisman A.A., "*Temporal Queries in OLAP*", Proceedings of 26th International Conference on Very Large Data Bases - VLDB 2000, Cairo (Egypt), September 10-14, 2000.

25. [Pedersen, Jensen 1998] Pedersen T.B., Jensen C.S., "*Research Issues in Clinical Data Warehousing*", SSDBM'98, July 1998, Capri (Italy).

26. [Ravat, Teste, Zurfluh 2000a] Franck Ravat, Olivier Teste, Gilles Zurfluh, "*Modélisation et extraction de données pour un entrepôt objet*", Actes des 16ième Journées Bases de Données Avancées - BDA'2000, 24-27 Octobre 2000, Blois (Loir et Cher, France).

27. [Ravat, Teste, Zurfluh 2000b] Franck Ravat, Olivier Teste, Gilles Zurfluh, " *A Temporal Object-Oriented Data Warehouse Model: Comprehensive Definition*", Rapport Interne IRIT/00-14-R, Juin 2000, Toulouse (France).

28. [Ravat, Teste, Zurfluh 2001] Franck Ravat, Olivier Teste, Gilles Zurfluh, "*Modélisation multidimensionnelle des systèmes décisionnels*", Accepté et à paraître dans les Actes des 1ères Journées Francophones d'Extraction et de Gestion des Connaissances - EGC 2001, 18-19 Janvier 2001, Nantes (Loire-Atlantique, France).

29. [Ravat, Teste 2000a] Franck Ravat, Olivier Teste, "*Object-Oriented Decision Support System*", Proceedings of the 2nd International Conference on Enterprise Information Systems - ICEIS'00, July 4-7 2000, Stafford (UK).

30. [Ravat, Teste 2000b] Franck Ravat, Olivier Teste, "*An Object Data Warehousing Approach: a Web Site Repository*", Proceedings of Challenges of the Enlarged 4th East-European Conference on Advances in Databases and Information Systems - ADBIS-DASFAA, September 5-8 2000, Prague (Czech Republic).

31. [Ravat, Teste, Zurfluh 1999] Franck Ravat, Olivier Teste, Gilles Zurfluh, "*Towards Data Warehouse Design*", Proceedings of the 8th International Conference on Information and Knowledge Management - CIKM'99, ACM Press - *ed*. Susan Gauch - p359-366, November 2-6 1999, Kansas City (Missouri, USA).

32. [Ravat, Zurfluh 1995] Ravat F., Zurfluh G. "*Répartition d'un schéma conceptuel de bases de données orientées objet*", BDA'95, pp. 145-163, Nancy (France), 1995.

33. [RedBrick 1998] Red Brick Systems, INC., "*Decision-Makers, Business Data and RISQL*", http://www.redbrick.com, 1998.

34. [Shaw, Zdonik 1990] Shaw G.M., Zdonik, S.B., "*A Query Algebra for Object-Oriented Databases*", Proceedings of the Sixth International Conference on Data Engineering - ICDE'90, IEEE Computer Society, ISBN 0-8186-2025-0, Los Angeles (California, USA), February 5-9 1990.

35. [Teste 2000] Olivier Teste, "*Elaboration d'entrepôts de données complexes*", Actes du XVIIIème Congrès INFormatique des ORganisations et Systèmes d'Information et de Décision - INFORSID'00, *ed*. INFORSID - ISBN 2-906855-16-2, p229-245, 16-19 mai 2000, Lyon (Rhône, France).

36. [Teste 2000b] Olivier Teste, "*Modélisation Conceptuelle des Entrepôts de Données*", Rapport Interne IRIT/99-01-R, Janvier 1999, Toulouse (France).

37. [Widom 1995] Widom J., "*Research problems in data warehousing*", Proceedings of the 4th International Conference on Information and Knowledge Management - ACM CIKM'95, November 29-December 2 1995, Baltimore (Maryland, USA).

38. [Yang, Widom 2000] Yang J., Widom J., "*Temporal View Self-Maintenance in a Warehousing Environment*", In Proceedings of the 7th International Conference on Extending Database Technology - EDBT 2000, Konstanz (Germany), March 2000.

Résumé

L'objet de ce mémoire concerne plus particulièrement la modélisation des entrepôts et des magasins de données. Issus originellement de l'industrie, les entrepôts et magasins de données sont devenus aujourd'hui un thème de recherche à part entière. Le but de **la modélisation des entrepôts et des magasins** de données est de fournir des abstractions permettant de détacher la manière de représenter les données de leur implantation physique. Le but de notre travail est d'étudier les concepts de base des entrepôts de données, ainsi que les modèles associés ; rendre compte de la variété des techniques et des approches de conception des entrepôts de données, qui ont commencé à voir le jour tout en essayant de dégager les principaux problèmes sur lesquels un effort de recherche est encore nécessaire ; définir des critères de comparaison, entre ces approches, et déceler leurs avantages et leurs lacunes. Ainsi la proposition d'une démarche de conception pour les entrepôts de données. Notre démarche est basée sur les méthodologies proposées dans la littérature.

Abstract

The object of this memory is related to the modeling of the datawarehouse and the datamart.Resulting originally from industry, the warehouse and stores datamart became today a research topic to whole share. The goal of the modeling of the warehouse and the datamart is to provide abstractions making it possible to detach the manner of representing the data of their physical establishment. The goal of our work is to study the basic concepts of the datawarehouses, as well as the associated models, that is to account for the variety of the techniques and the approaches of designing the datawarehouses, trying to release the problems on which an effort of research is still necessary; to define comparison criterions between these approaches, and to detect their advantages and their Disadvantages. At last we propose a methodology for designing a datawarehouse.

الهدف من هده الأطروحة يخص كيفية تجسيد بنك المعلوما تية بطريقته الجديدة. هدا العمل مخصص لدراسة مختلف الطرق الجديدة لتجسيد بنك المعلوما تيةِ النظر في خصا ئص كل طريقة أهدا فها, نقاط الضعف و نقاط القوة. إجراء مقارنة معمقة و في الأخير طرح فكرة طريقة جديدة استنادا إلى الطرق المذكورة سابقا.