N° d'ordre : 03/2012 – M/INF

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene
Faculté d'électronique et d'informatique

Département d'Informatique

MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTER
en INFORMATIQUE

Spécialité : Systèmes Intelligents et Ingénierie des Logiciels

Par : BOUKHEDOUMA Hocine

Sujet

Approche de Conception d'un Entrepôt de Données Multimédia à base Ontologique

Soutenu publiquement le 16 septembre 2012 devant le jury composé de :

M. BELKHIR A.	Professeur à l'USTHB	Président
Mme. ALIMAZIGHI Z.	Professeur à l'USTHB	Directrice de mémoire
M. BOUKRA A.	Maitre de conférences / A à l'USTHB	Examinateur
M. BOUKHALFA K.	Maitre de conférences / A à l'USTHB	Examinateur

Remerciements

Je tiens à adresser mes sincères remerciements à madame ALIMAZIGHI ZAIA, professeur à la FEI- USTHB pour m'avoir proposé le sujet et orienté dans la réalisation de ce travail. Je la remercie aussi pour sa gentillesse, sa patience et ses conseils.

Je tiens à remercier monsieur BELKHIR Abdelkader, professeur au département informatique de l'USTHB de m'avoir fait l'honneur de présider le jury de ce mémoire.

Je tiens aussi à remercier messieurs BOUKRA Abdelmadjid, BOUKHALFA Kamel et BOUYAKOUB fayçal, maîtres de conférences au département informatique de l'USTHB d'avoir accepté d'examiner et juger mon travail.

Je ne saurais terminer sans remercier vivement ma sœur SOUAD pour son aide précieuse, sa gentillesse, sa patience, ses conseils et sa disponibilité.

Je remercie aussi tous les membres de ma famille pour leur patience et leur soutien moral.

Je remercie Leila pour son soutien moral.

Enfin, Je remercie tous ceux qui ont participé, de près ou de loin, à l'aboutissement de ce travail.

Résumé

Les *entrepôts de données (ED)* ou encore *Data warehouse (DW)* ont été conçus dans le but d'aider les décideurs dans leur prise de décisions, à partir d'analyses effectuées sur un grand volume de données historisées.

La mise en œuvre d'un entrepôt de données passe par trois phases principales à savoir : la planification, la conception et la maintenance. La phase de *conception* est une phase très importante dans les projets d'entreposage de données. En effet, différentes études ont montré que le manque ou l'absence d'une phase de modélisation conceptuelle est une cause possible d'échec des projets d'entreposage.

Plusieurs approches de conception ont été proposées. Ces dernières peuvent être classifiées en trois catégories : les approches *directes*, les approches *traditionnelles* et les approches *ontologiques* assurant une conception à base d'ontologies.

Par ailleurs, dans certains domaines (telles que la médecine, la biologie, la géographie), les données se présentent sous diverses formes (texte, image, son, vidéo) et sont qualifiés de données multimédias. Ces données ont la particularité d'être volumineuses et complexes nécessitant ainsi des traitements et des outils particuliers pour leur stockage et leur exploitation. Les entrepôts qui traitent ce type de données sont appelées *entrepôt de données multimédia (EDM).*

Notre problématique consiste à proposer une démarche de conception d'entrepôts de données multimédia sur la base d'approches existantes. Pour cela, nous avons adopté une approche de conception d'un EDM prenant en compte les sources de données et les besoins des utilisateurs en même temps. Nous optons pour une approche à base *ontologique* permettant ainsi de régler les conflits sémantiques dans la phase d'intégration des données. De plus, l'apport de notre approche par rapport aux approches existantes réside dans la prise en compte des données *multimédia*.

Mots clés

Données multimédia- Entrepôt de données multimédia – Ontologie – Approche de conception – Intégration.

Table des matières

Page

Introduction générale

1. Contexte

De nos jours, la croissance de la masse de données circulant au niveau des entreprises ainsi que leur ancienneté pose des difficultés à accéder à l'ensemble de ces données et à regrouper les différentes bases de données disséminées dans les différents services, afin de les analyser et de prendre des décisions rapidement. Cette analyse et prise de décision apparaissent dans plusieurs domaines tels que l'urbanisme, la médecine, la géographie, les télécommunications, etc.

Pour réaliser cette analyse aisément et rapidement, il est souhaitable, voire nécessaire, de rassembler les informations de l'entreprise dans une seule base spécialement conçue pour l'analyse et la prise de décision, cette base est appelée *entrepôt de données (ED)* ou encore *Data warehouse (DW)*. Cependant, pour être utile, un ED doit être capable de répondre aux attentes des leurs utilisateurs qui sont les décideurs de l'entreprise. Il est donc nécessaire d'accorder une grande importance au développement de l'entrepôt tout au long de son cycle de vie.

La mise en œuvre d'un entrepôt de données passe par trois phases principales à savoir : la planification, la conception et la maintenance. La planification consiste à préparer le terrain pour le développement de l'entrepôt suivant un ensemble de tâches. La conception consiste à développer le schéma de l'entrepôt. La maintenance de l'entrepôt implique l'optimisation de ses performances périodiquement.

La phase de conception est une phase très importante dans les projets d'entreposage de données. En effet, différentes études ont montré que le manque ou l'absence d'une phase de modélisation conceptuelle est une cause possible d'échec des projets d'entreposage [Golfarelli et al 98]. Pour remédier à ce problème, il est donc nécessaire d'utiliser une vraie phase de conception afin de réduire le risque de l'échec.

Plusieurs approches de conception ont été proposées. Ces dernières peuvent être classifiées en trois catégories [Khouri09]: les approches *directes* basées sur la génération directe du schéma logique, les approches *traditionnelles* permettant la mise en place d'une vraie phase de conception et une nouvelle catégorie englobant les approches *ontologiques* assurant une conception à base d'ontologies. Les ontologies étant le regroupement de définitions et de description de concepts sont d'un apport considérable pour le règlement des conflits sémantiques dans les phases d'intégration de données provenant de sources hétérogènes.

2. Problématique et Contribution

Aujourd'hui, les données se présentent sous diverses formes (texte, image, son, vidéo) et sont qualifiés de données *multimédias*. Ces données ont la particularité d'être volumineuses et complexes nécessitant ainsi des traitements et des outils particuliers pour la leur stockage et leur traitement.

D'après l'étude que nous avons faite sur les données multimédias, nous avons remarqué qu'il existe deux manières de les traiter. La première consiste à faire une étude approfondie en traitant le *contenu* des données en question. La deuxième consiste à utiliser des informations sur les données appelées *métadonnées (telles que le nom de fichier, la taille, les mots clés, etc.)* et à effectuer les traitements nécessaires en utilisant ces informations.

Une problématique essentielle qui revient souvent concerne l'intégration des données due aux différents conflits syntaxiques et sémantiques. Les approches ontologiques sont apparues pour remédier à ce problème. A notre connaissance, peu de travaux ont été réalisés dans le contexte de la conception d'ED à base ontologique, nous nous sommes basés sur deux d'entre eux: le premier [Romero et al 07] présente une méthode de conception dérivant le schéma conceptuel de l'entrepôt uniquement à partir des sources de données. Le second travail [Khouri 09] prend en compte les sources de données et les besoins des utilisateurs. En effet, l'implication des besoins des utilisateurs dans la conception d'un ED est très importante ; elle a été prouvée par diverses études faite dans ce contexte [Giorgini et al 05], [Gam et al 06].

A cause de la nature complexe des données multimédia, les conflits syntaxiques et sémantiques liés à leur intégration se fait ressentir plus que pour des données simples (texte). Ainsi, notre contribution s'insère dans le cadre de la conception des ED, manipulant particulièrement des données *multimédia*. Elle consiste à proposer une démarche de conception d'un ED prenant en compte les sources de données et les besoins des utilisateurs en même temps. L'apport de notre approche par rapport aux approches existantes réside dans la prise en compte des données multimédia extrêmement hétérogènes telles que *le son, l'image et la vidéo*.

Ainsi, pour toucher à tous les volets rentrant dans la réalisation de notre travail, ce dernier comportera deux parties : la première représentant un état de l'art sur les différents concepts, méthodes, outils et technologies utilisés et la seconde partie décrivant la démarche proposée pour la conception d'un entrepôt de données multimédia (EDM) à base ontologique.

3. Organisation du mémoire

Le présent travail est constitué de six chapitres :

Le premier chapitre présente une introduction aux entrepôts de données en mettant l'accent sur les principales notions ainsi que les concepts qui y sont liés.

Le deuxième chapitre présente un état de l'art sur les différentes méthodes de conception des projets d'entreposage qui sont classifiées en trois catégories. Pour chaque approche, nous décrivons son principe ainsi que les principales méthodes de conception proposées.

Le troisième chapitre s'intéresse à la modélisation des données multimédias. Pour cela, nous présentons les principales spécificités des données multimédias et les principaux standards permettant de les prendre en charge. Nous abordons aussi le côté technique qui nous permet de voir comment ces données sont traitées par un SGBD.

Le quatrième chapitre porte sur les entrepôts de données multimédias, une catégorie d'entrepôts qui apparait dans plusieurs domaines tels que la médecine, la biologie et la géographie. Un état de l'art synthétisant les travaux liés à ce type d'entrepôts est présenté dans ce chapitre.

Le cinquième chapitre présente les principaux concepts liés aux ontologies ainsi qu'une synthèse des travaux réalisés dans ce domaine concernant notamment, les approches de construction d'ontologies. Nous mettons l'accent aussi sur un type particulier d'ontologies dédiées à des domaines traitant des données multimédias.

Le sixième chapitre décrit les différentes étapes de la démarche que nous proposons pour la conception d'un EDM à base ontologique. Cette démarche est illustrée par un exemple lié au domaine médical, en particulier la détection de pathologies du cerveau humain à partir d'images radiologiques.

Deux annexes sont jointes à ce document : l'annexe A donne quelques détails sur les standards de description de données multimédia et l'annexe B décrit quelques langages de description et quelques outils d'édition d'ontologies.

Chapitre 1

Les Entrepôts de données

1. Introduction

Actuellement, avec l'apparition d'Internet et de nouvelles technologies, la quantité de données stockées au niveau des entreprises s'accroit considérablement. La principale caractéristique de ces données est qu'elles soient souvent hétérogènes et réparties dans plusieurs services de l'entreprise. De ce fait, l'accès à ces données devient de plus en plus difficile, ce qui augmente la complexité de certaines tâches telles que l'analyse et la prise de décision.

Cette analyse et prise de décision apparaissent dans plusieurs domaines. Dans la grande distribution, par exemple, on aimerait déterminer les produits à succès, les modes, les habitudes d'achat des consommateurs globalement ou par secteur géographique. Dans le domaine des services tels que les banques, les entreprises de télécommunication, les assurances et mutualités, on aimerait pouvoir classifier les clients, détecter des fraudes ou les clients qui risquent d'être infidèles, etc.

Afin de faciliter la tâche d'analyse des données, ces dernières doivent être regroupées dans une seule base dédiée pour l'analyse et la prise de décision. Ainsi, les entrepôts de données se sont imposés comme une solution rentable pour faire face aux besoins des entreprises en termes de capitalisation de connaissances et d'aide à la décision.

Dans ce chapitre nous commençons par voir ce qu'est l'entrepôt de données, puis nous examinons l'architecture et les principaux composants d'un entrepôt de données. Nous introduisons aussi les mini-entrepôts (*data marts*) et les notions associées au développement et à la gestion des mini-entrepôts. Par la suite, nous mettons l'accent sur la modélisation multidimensionnelle ainsi que la construction et la maintenance d'un ED. Nous exposons aussi quelques outils associés à l'ED. Nous enchainons par citer les bénéfices et les problèmes potentiels associés à l'entreposage de données.

2. Définition d'un ED

Un entrepôt de données peut être défini comme étant un lieu de stockage des données nécessaires à la prise de décision ; il est alimenté et mis à jour via des extractions de données portant sur les bases de production qui sont considérées dans la chaîne décisionnelle comme les "sources de données".

La définition la plus adaptée est celle de *Bill Inmon* [Verhaegen06], qui le définit comme suit :
«Un entrepôt est une collection de données orientées sujet, intégrées, non volatiles et historisées, organisées pour le support d'un processus d'aide à la décision».
D'après cette définition, les données d'un ED sont [Verhaegen06]:

- *Orientées sujet :* Les données sont orientées métier et donc triées par thèmes. L'intégration dans une structure unique est indispensable pour éviter aux données concernées par plusieurs sujets d'être dupliquées ;

- *Intégrées :* Les données proviennent de plusieurs sources hétérogènes. Avant d'être intégrées dans l'entrepôt, les données doivent êtres mises en forme et unifiées afin d'avoir un état cohérent. Cela nécessite un gros travail de normalisation, de gestion des référentiels et de cohérence ;

- *Non volatiles :* Les données sont stables et non modifiables. Un entrepôt de données doit garantir qu'une requête lancée à différentes dates sur les mêmes données donne toujours les mêmes résultats. De plus, les données d'un entrepôt sont mises à jour périodiquement, ce ne sont donc pas des informations en temps réel ;

- *Historisées :* Les données sont historisées et donc datées : l'historisation est nécessaire pour suivre dans le temps l'évolution des différentes valeurs des indicateurs à analyser. Ainsi, un référentiel temps doit être associé aux données afin de permettre l'identification de valeurs précises dans la durée.

3. Architecture et composants

L'architecture type d'un entrepôt de données est donnée dans la figure 1.1. Les principaux composants constituant un entrepôt de données sont détaillés juste après la figure.

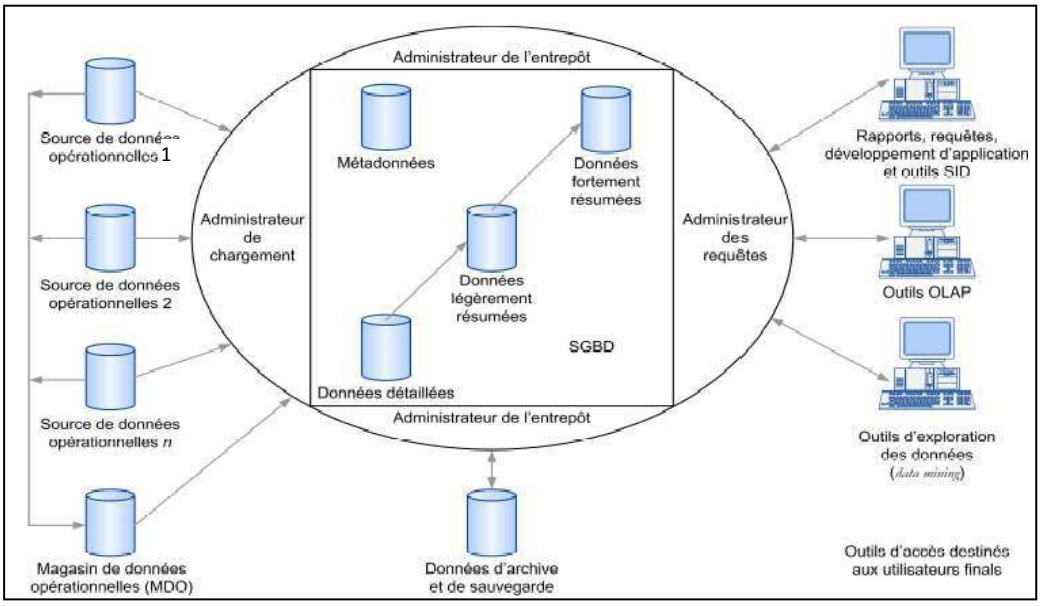


Figure 1.1. *Architecture type d'un entrepôt de données [Connoly et al 05]*

3.1 Données opérationnelles

La source des données de l'entrepôt de données est fournie par :

- Des données opérationnelles d'un ordinateur central, contenues dans des bases de données volumineuses. Des estimations ont montré que la majorité des données opérationnelles d'entreprise est encore contenue dans de tels systèmes [Connoly et al 05].
- Des données départementales contenues dans des systèmes de fichiers propriétaires, tels que VSAM, RMS, et des SGBDR tels que : *Informix* et *Oracle*.
- Des données privées détenues dans des stations de travail et des serveurs privés.
- Des systèmes externes, tels que l'Internet, des bases de données commerciales disponibles ou des bases de données détenues par les fournisseurs ou les clients de l'organisation.

3.2 Magasin de données opérationnelles (ODS, *Operational Data Store*)

Un magasin de données opérationnelles (MDO) est un annuaire de données opérationnelles actualisées et intégrées qui sert aux analyses. Il est souvent structuré et fourni avec des données de la même manière que l'entrepôt de données, mais agit en fait comme une zone temporaire de préparation des données à déplacer vers l'entrepôt.

Le MDO est souvent créé après la constatation que les systèmes opérationnels classiques sont incapables d'atteindre les exigences en matière de rapports. Le MDO apporte aux utilisateurs l'aisance d'utilisation d'une base de données relationnelle tout en demeurant à distance des fonctions de support à la décision de l'entrepôt de données.

L'édification d'un MDO s'avère une étape utile avant la mise en place d'un véritable entrepôt de données, parce que le MDO est en mesure de fournir les données déjà extraites des systèmes sources et nettoyées. Par conséquent le travail restant à effectuer, qui consiste à intégrer et restructurer les données pour l'entrepôt de données, se simplifie.

3.3 Administrateur de chargement

L'administrateur de chargement (*load manager*), dénommé aussi composant *frontal* ou de *frontend*, effectue toutes les opérations associées à l'extraction et au chargement des données dans l'entrepôt. Les données peuvent être extraites directement des sources de données ou, plus fréquemment, du magasin opérationnel.

Les opérations effectuées par l'administrateur de chargement sont éventuellement des transformations simples des données pour les préparer à entrer dans l'entrepôt. La taille et la complexité de ce composant varient selon les entrepôts de données et sa construction s'effectue à l'aide d'une combinaison d'outils d'exportation et de chargement de données spécifiques à des vendeurs et des programmes réalisés sur mesure.

3.4 Administrateur de l'entrepôt

L'administrateur de l'entrepôt (*warehouse manager*), remplit toutes les tâches associées à la gestion des données dans l'entrepôt. Ce composant est édifié à l'aide d'outils d'administration des données des vendeurs et de programmes sur mesure.

Les tâches remplies par l'administrateur de l'entrepôt sont, entre autres :

- L'analyse des données pour en garantir la cohérence;
- La transformation et la fusion de données sources d'une zone de stockage temporaire vers des tables de l'entrepôt;
- La création d'index et de vues sur les tables de base;
- La génération de la dénormalisation si nécessaire;
- La génération d'agrégats si nécessaire;
- La sauvegarde et l'archivage des données.

3.5 Administrateur des requêtes

L'administrateur des requêtes (*query manager)*, aussi nommé le composant principal ou de *backend*) se charge des opérations associées à la gestion des requêtes des utilisateurs. Ce composant est normalement construit à l'aide d'outils d'accès aux données par les utilisateurs, vendus par les éditeurs de logiciels, d'outils de surveillance d'entrepôt de données, d'utilitaires de base de données et de programmes réalisés sur mesure. La complexité de l'administrateur des requêtes est déterminée par les facilités que procurent les outils d'accès aux données par les utilisateurs finaux et la base de données. Les opérations prises en charge par ce composant sont notamment :

- la déviation des requêtes vers les tables adéquates ;
- la planification de l'exécution des requêtes.

3.6 Données détaillées

Cette zone de l'entrepôt stocke toutes les données détaillées dans le schéma de base de données. Dans la plupart des cas, les données détaillées ne sont pas stockées en ligne mais sont mises à disposition par l'agrégation des données jusqu'au niveau de détail suivant.

Cependant, d'une manière régulière, les données détaillées sont ajoutées à l'entrepôt en supplément des agrégats de données.

3.7 Données légèrement ou fortement résumées

Cette zone de l'entrepôt stocke toutes les données légèrement ou fortement résumées au préalable (agrégées) générées par l'administrateur de l'entrepôt. Cette partie de l'entrepôt est transitoire dans la mesure où elle est susceptible de changer de manière permanente afin de répondre aux profils de requêtes, eux aussi changeants.

La raison d'être des informations de synthèse est d'augmenter les performances des requêtes.

3.8 Données archivées et sauvegardées

Cette partie de l'entrepôt emmagasine les données détaillées et résumées pour les besoins d'archivage et de sauvegarde. Même si les données résumées sont générées à partir des données détaillées, il est nécessaire de sauvegarder les données de synthèse en ligne si ces données sont maintenues au-delà de la période de détention des données détaillées, auquel cas, il ne serait plus possible de les générer à nouveau en cas de problème

L'archivage des vieilles données joue un rôle important dans le maintien de l'efficacité et des performances de l'entrepôt, en déplaçant les données anciennes et d'un intérêt limité vers des dispositifs d'archivage tels que des bandes magnétiques ou des disques optiques.

Il faut noter que les entrepôts de données ne contiennent pas systématiquement tous ces types de données, c'est aux concepteurs de les générer si nécessaire.

3.9 Métadonnées

Cette partie de l'entrepôt conserve toutes les définitions de métadonnées (les données à propos des données) employées par tous les processus de l'entrepôt. Les métadonnées servent à une grande variété de desseins, dont :

- Les processus d'extraction et de chargement : Les métadonnées sont utilisées pour faire correspondre des sources de données à une vue commune des données au sein de l'entrepôt;

- Le processus d'administration de l'entrepôt : Les métadonnées servent à automatiser la production des tables de synthèse;

- En tant qu'acteur dans le processus d'administration des requêtes : Les métadonnées sont utilisées pour diriger une requête vers la source de données la plus appropriée.

3.10 Outils d'accès des utilisateurs finaux

La principale raison d'être de l'entreposage de données est de livrer des informations aux utilisateurs commerciaux et administratifs pour les aider à prendre des décisions stratégiques. Ces utilisateurs interagissent avec l'entrepôt, grâce à des outils d'accès aux données spécifiques pour les utilisateurs finaux. L'entrepôt de données doit par conséquent apporter son soutien aux analyses *ad hoc* (ou de circonstance) mais aussi de routine. Les hautes

performances sont obtenues par une planification prédéfinie des exigences de jointure, de synthèse et d'états (rapports imprimés) par les utilisateurs finaux.

Même si les définitions des outils d'accès des utilisateurs finaux se chevauchent souvent, ils sont répartis en cinq catégories distinctes [Connoly et al 05] :

- Les outils de rapport et de requête;
- Les outils de développement d'application;
- Les systèmes d'information pour dirigeants (SID);
- Les outils de traitement analytique en ligne (OLAP);
- Les outils d'exploration des données (*data mining*).

4. Mini-entrepôt de données

L'émergence explosive des entrepôts de données s'est accompagnée de l'apparition du concept apparenté de mini-entrepôts de données (*data marts*). Nous décrivons dans ce qui suit ce que sont les mini-entrepôts, l'intérêt de les mettre en place et les soucis engendrés par le développement et l'usage des mini-entrepôts.

4.1 Définition d'un mini-entrepôt de données

Un mini-entrepôt de données détient un sous-ensemble des données d'un entrepôt de données, normalement sous la forme de données résumées, concernant un département ou un service particulier ou encore d'une fonction professionnelle précise. Le mini-entrepôt est soit autonome, soit relié à l'entrepôt de données central de l'entreprise.

Les caractéristiques déterminantes des mini-entrepôts par rapport aux entrepôts de données sont, entre autres [Connoly et al 05]:

- Un mini-entrepôt se concentre sur les seules exigences des utilisateurs associés à un département ou une fonction professionnelle;

- Les mini-entrepôts ne contiennent en principe pas de données opérationnelles détaillées, contrairement aux entrepôts de données;

- Comme les mini-entrepôts contiennent moins de données, comparés aux entrepôts de données, les mini-entrepôts sont plus aisément compris et parcourus.

Les soucis qui surgissent dans le développement et l'administration des mini-entrepôts sont détaillés dans [Connoly et al 05] ; on les résume dans les points suivants :

- Fonctionnalités du mini-entrepôt ;
- Taille du mini-entrepôt ;
- Performances de chargement du mini-entrepôt ;
- Accès par les utilisateurs aux données placées dans plusieurs mini-entrepôts ;
- Accès à un mini-entrepôt par l'Internet ou un intranet ;
- Administration du mini-entrepôt ;
- L'installation du mini-entrepôt.

4.2 Raisons de créer un mini-entrepôt de données

Parfois, les utilisateurs ne désirent pas accéder à toutes les données contenues dans l'entrepôt, mais plutôt à une partie spécifique liée à un département ou à un service donné. C'est la fonction principale qu'offre un mini-entrepôt de données (*datamart*).

De nombreuses raisons incitent à mettre en place un mini-entrepôt, notamment [Connoly et al 05] :

- Donner aux utilisateurs un accès aux données dont ils ont besoin pour leurs analyses les plus fréquentes.

- Fournir les données sous une forme compatible avec la perception collective des données par un groupe d'utilisateurs d'un département ou d'une fonction professionnelle.

- Améliorer les délais de réponse, grâce à la réduction du volume des données auxquelles les utilisateurs finaux accèdent.

- Les mini-entrepôts utilisent normalement un nombre de données moindre, de sorte que des tâches telles que le nettoyage, le chargement, la transformation et l'intégration des données sont nettement plus faciles et, donc, l'implémentation et le réglage d'un mini-entrepôt sont beaucoup plus simples que dans le cas de l'édification d'un entrepôt de données d'entreprise.

- Les coûts de mise en place des mini-entrepôts sont généralement moindres que ceux engendrés par l'installation d'un entrepôt de données.
- Les utilisateurs potentiels d'un mini-entrepôt sont mieux définis; il est plus facile de cibler leurs besoins et, par conséquent de demander leur coopération dans un projet de mini-entrepôt, que dans le cas d'un projet d'entrepôt de données à l'échelle de toute une entreprise.

4.3 Architecture d'un mini-entrepôt

La figure 1.2 illustre l'architecture type d'un entrepôt de données associé à un mini-entrepôt de données.

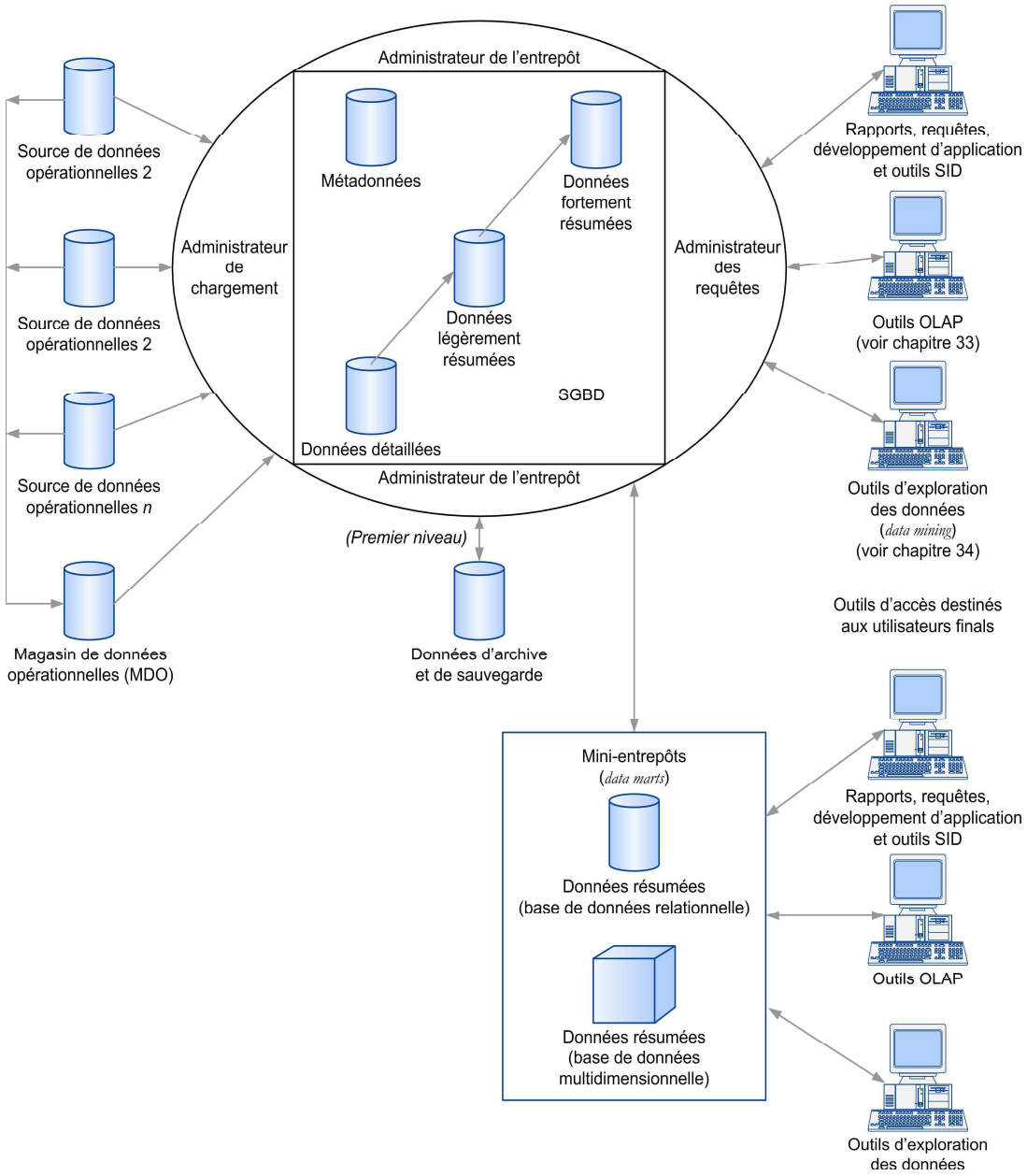


Figure 1.2. *Architecture type de l'entrepôt de données (data warehouse) associé au mini-entrepôt de données (data mart).*

Il existe plusieurs approches possibles pour la mise en place de mini-entrepôts. L'une d'elles consiste à construire plusieurs mini-entrepôts avec, en tête, l'intention à terme de les intégrer dans un entrepôt; une autre approche suggère de construire l'infrastructure d'un entrepôt de données d'entreprise et, en même temps, de mettre en place un ou plusieurs mini-entrepôts pour satisfaire les besoins professionnels immédiats [Connoly et al 05].

Les architectures des mini-entrepôts se présentent sous la forme d'applications à deux ou trois niveaux. L'entrepôt de données, facultatif, constitue le premier niveau (dans le cas où l'entrepôt de données fournit les données au mini-entrepôt), le mini-entrepôt est le deuxième niveau et la station de travail de l'utilisateur final forme le troisième niveau, comme le montre la figure 1.2. Les données se répartissent parmi ces niveaux.

La création d'un mini-entrepôt apporte plusieurs avantages, parmi lesquels on peut citer : l'amélioration des délais de réponse, la réduction des coûts, la facilité du nettoyage et de l'intégration, etc.

5. La modélisation multidimensionnelle

Comme nous l'avons déjà mentionné, un entrepôt de données est orienté sujet et doit permettre d'analyser des données suivant plusieurs dimensions [Connoly et al 05]. Les bases de données multidimensionnelles, à la différence des bases de données relationnelles classiques permettent d'effectuer des traitements sur des données en prenant en compte plus de deux axes : les données ne sont pas modélisées sous forme tabulaire mais sous forme d'hyper-cubes (ou 'cubes de données' ou encore 'data cubes'), comme le montre la figure 1.3.

Ces structures sont essentiellement utilisées par des analystes qui cherchent à trouver des tendances dans de grandes quantités de données. Par exemple, rechercher les facteurs de risques (âge, sexe, traitement, etc.) de maladies pour une étude médicale, caractériser des données (type d'objet, siècle, etc.) dans le cadre de fouilles archéologiques ou déterminer des corrélations entre produits-magasin-vendeur pour le directeur de ressources humaines de chaînes de distribution.

La conception des entrepôts de données est donc basée sur la modélisation multidimensionnelle [Amarouche 06] comprenant la notion de table de faits, ainsi qu'un ensemble de tables dimensionnelles.

Dans ce qui suit, des détails concernant la table des faits et les tables dimensionnelles seront donnés suivis des différentes configurations obtenues par la disposition de la table des faits et de dimensions [Verhaegen 06].

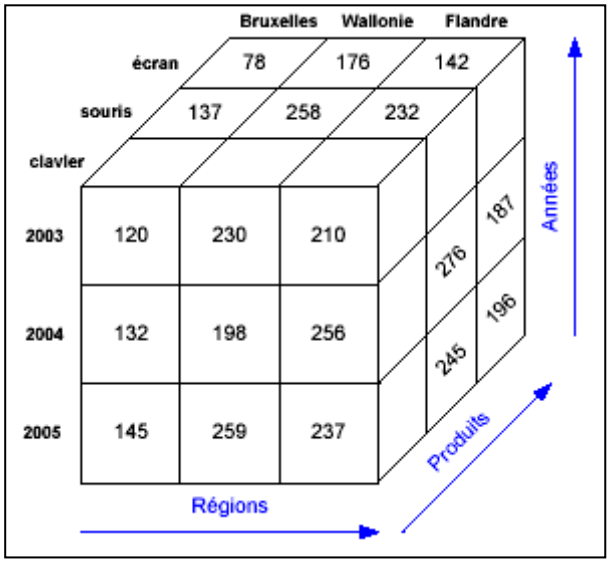


Figure 1.3. *Cube de données*

5.1 Définitions

Dans cette partie, nous donnons quelques définitions liées aux notions de la modélisation multidimensionnelle. Des exemples d'illustrations seront présentés si nécessaire.

Définition 1 (Sujet): Un sujet est défini par un ensemble de mesures et un ensemble de dimensions [Verhaegen 06]. Par exemple, les mesures d'une vente peuvent être son numéro, son prix et sa quantité : ce sont les valeurs numériques que l'on veut comparer. Les dimensions seraient la date, le type de produit et la région : ce sont les points de vue depuis lesquels les mesures peuvent être observées.

Définition 2 (Dimension) : Une dimension est une liste d'éléments organisés de façon hiérarchique [Verhaegen 06]. La granularité d'une dimension est son nombre de niveaux hiérarchiques. Par exemple, pour le temps, nous pourrions avoir la hiérarchie suivante : année, semestre, trimestre, mois, semaine, jour, soit six niveaux.

Les axes de dimensions doivent fournir des règles de calcul d'agrégats pour chaque mesure. Par exemple, le nombre de ventes du premier trimestre est la somme du nombre de ventes de janvier, février et mars. Les dimensions sont stockées dans des tables de dimensions qui contiennent les niveaux hiérarchiques des dimensions ainsi que les formules à appliquer sur les données numériques (les faits) pour passer d'un niveau à un autre.

Définition 3 (fait) : Un fait représente la valeur d'une mesure, mesurée ou calculée, selon un membre de chacune des dimensions [Verhaegen 06]. Par exemple, « 10000 » est un fait qui exprime la valeur de la mesure « coût des travaux » pour le membre « 2002 » du niveau « année » de la dimension « temps » et le membre « Bruxelles » du niveau « ville » de la dimension « géographie ». Les mesures sont stockées dans des tables de faits qui contiennent les valeurs des mesures et les clés vers les tables de dimensions.

5.2 Les implémentations d'un cube de données

Il existe deux manières principales pour construire un système basé sur un modèle multidimensionnel, selon la manière dont le cube est stocké : l'approche MOLAP et l'approche ROLAP [Khouri 09].

5.2.1 L'approche MOLAP

Les systèmes MOLAP (*Multidimensional* On-Line Analytical Processing) implémente le cube sous forme d'un tableau multidimensionnel (SGBD multidimensionnel). Chaque dimension du tableau représente une dimension du cube. Les données de chaque cellule sont stockées. Par exemple, le cube multidimensionnel de la figure 1.4 peut être représenté par le tableau multidimensionnel (tableau 1.1) :

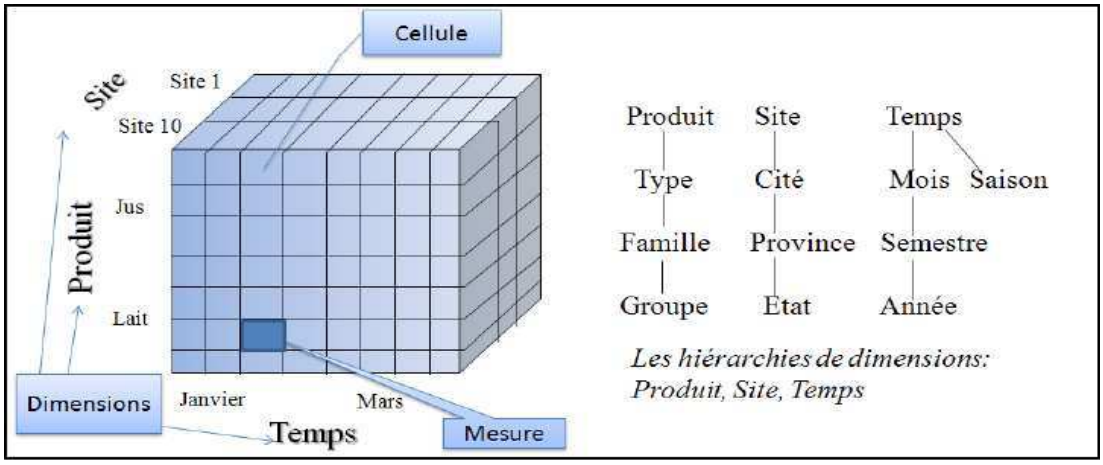


Figure 1.4. *Le cube multidimensionnel Vente*

Produit	Temps	Trimestre 1			Trimestre 2			Trimestre 3			Trimestre 4			Total		
	Site	N	S	Total	N	S	Total	N	S	Total	N	S	Total	N	S	Total
Produits laitiers		12	34	46	22	36	58	24	37	61	33	55	88	91	162	253
Viennoiseries		29	66	95	44	50	94	56	55	111	44	39	83	173	210	383
Viandes		55	34	89	69	27	96	31	26	57	68	70	138	223	157	380
Total		96	134	230	135	113	248	111	118	229	145	114	309	487	529	1016

Tableau1.1. *Représentation du cube multidimensionnel selon MOLAP*

Le principal avantage d'un système MOLAP est sa performance en temps d'accès (l'accès aux données est direct). Outre le fait que ce système ne présente aucun standard, ses principaux inconvénients sont [Bellatreche00]:

- Le besoin de redéfinir des opérations pour manipuler les structures multidimensionnelles.
- Difficulté de la mise à jour et de la gestion du modèle.
- Consommation de l'espace lorsque les données sont éparses, ce qui nécessite l'utilisation des techniques de compression.

5.2.2 L'approche ROLAP

Les systèmes ROLAP (Relational On-Line Analytical Processing) stockent les données du cube en utilisant un SGBD relationnel. Chaque dimension du cube est représentée sous forme d'une table appelée *table de dimension*. Chaque fait est représenté par une *table de fait*. Les mesures sont stockées dans les tables de faits qui contiennent les valeurs des mesures et les clés vers les tables de dimensions. Par exemple, le cube multidimensionnel de la figure 1.4 peut être représenté par les tables de la figure1.5 :

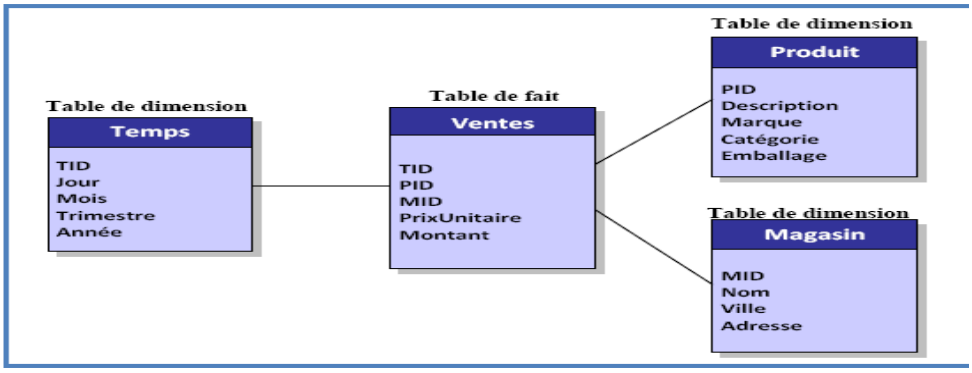


Figure 1.5. *Représentation du cube multidimensionnel selon ROLAP*

Le principal inconvénient de ces systèmes est qu'ils peuvent présenter un temps de réponse aux requêtes élevé. Leurs principaux avantages sont :

- Exploitation des capacités d'un standard bien établi et maîtrisé : le relationnel, ce qui facilite leur intégration dans les SGBD relationnels existants.
- Stockage de grandes quantités de données.

La disposition de la table de faits et de dimensions donne naissance à trois configurations de modèles conceptuels pour modéliser les systèmes ROLAP :

- Le schéma en *étoile*
- Le schéma en *flocon de neige*
- Le schéma en *constellation*

a- Le modèle en étoile (*star*) : C'est la structure de données la plus utilisée et la plus appropriée aux requêtes et analyses des utilisateurs d'entrepôts de données. Elle est simple à créer, stable et intuitivement compréhensible par les utilisateurs finaux. Ce modèle consiste en une grande table de faits et un cercle d'autres tables qui contiennent les éléments descriptifs du fait : les dimensions [Verhaegen06], [Peguiron05].

La figure 1.6 illustre ce modèle. On remarque que ce dernier ressemble à une étoile, d'où sa dénomination 'En étoile'.

L'avantage de ce modèle est la facilité de lecture et la faible complexité des requêtes. En effet généralement, peu de jointures sont nécessaires car le nombre de tables reste faible. Son inconvénient concerne la redondance dans les tables de dimensions, ce qui entraine un stockage lourd et une alimentation complexe de la base de données.

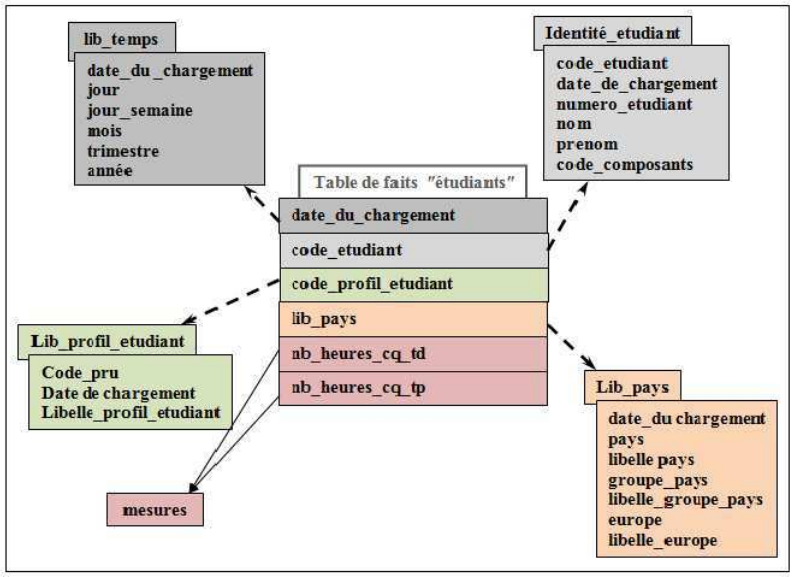


Figure1.6. *Modèle en étoile*

b- Le modèle en flocon de neige (*snowflake*) : C'est une variante du schéma en étoile. Dans la théorie, la différence réside dans la simple normalisation des tables de dimensions en 3FN. Il est donc tout simplement question de mettre les attributs de chaque niveau hiérarchique dans une table de dimension distincte pour éviter la redondance [Verhaegen06].
L'avantage de ce modèle est qu'il permet de gagner de l'espace disque et facilite l'alimentation. Son inconvénient apparait dans l'implication de nombreuses jointures dans les requêtes et donc une difficulté d'écriture de ces dernières. Un exemple est présenté dans la figure1.7.

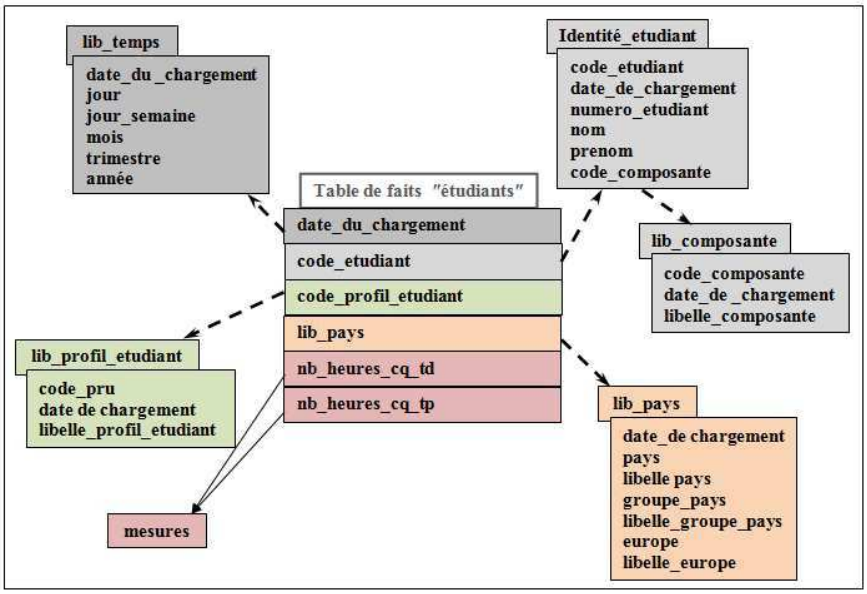


Figure 1.7. *Modèle en flocon de neige*

c- La constellation de faits : La constellation de faits permet de représenter plusieurs tables de faits partageant quelques tables de dimensions [Peguiron05].

La figure 1.8 montre ce type de modèle où les tables de faits *'étudiants'* et *'enseignants'* peuvent être mises en relation par la dimension *temps* et *géographie*.

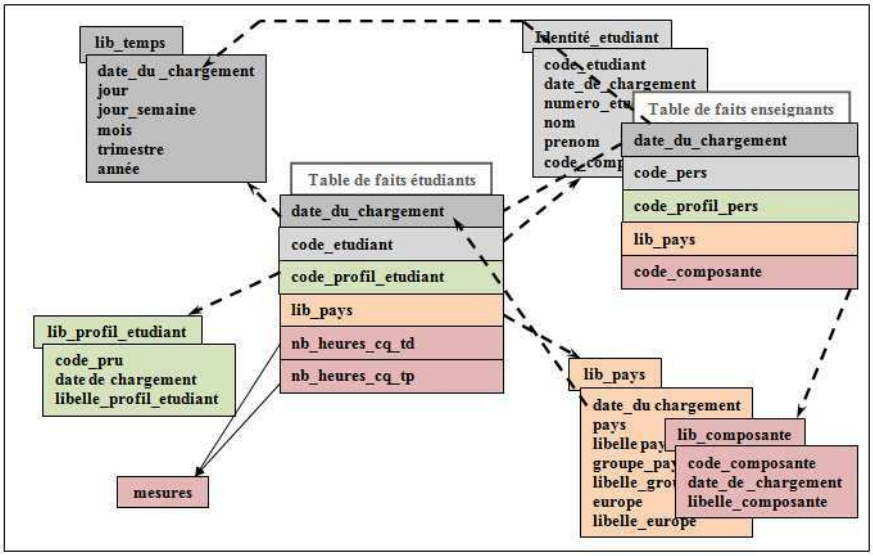


Figure 1.8. *Constellation de faits*

5.2.3 L'approche HOLAP

Les systèmes HOLAP (**Hybrid** On-Line Analytical Processing) sont des systèmes où les données fréquemment utilisées (généralement les données agrégées) sont maintenues par un SGBD multidimensionnel, et les données non fréquemment utilisées dans un SGBD relationnel. Ceci afin de bénéficier des avantages des deux systèmes cités précédemment. La séparation des données doit être transparente à l'utilisateur final [Khouri09].

Une comparaison entre les deux approches ROLAP et MOLAP est donnée dans le tableau suivant :

Stockage	Avantages	Inconvénients
ROLAP	Technologie familière	Lenteur
	Scalable (capacité d'expansion élevée)	
	Ouvert	
MOLAP	Modèle multidimensionnel	Technologie non prouvée
	Traitement de requête spécialisé	Non scalable
	Techniques d'indexation spécialisées	

Tableau 1.2. *ROLAP Vs MOLAP* [Bellatreche00]

Quand il s'agit de choisir une modélisation plutôt qu'une autre, de nombreux paramètres sont à prendre en compte : la nature des requêtes, les besoins d'analyse, les besoins de flexibilité, l'évolution des dimensions dans le temps, etc. Ce n'est donc jamais une question

simple et il convient de l'étudier avec le plus grand soin. En effet, une mauvaise modélisation peut conduire à l'inutilité d'un entrepôt avec les pertes de temps et d'argent que cela implique.

5.3 Opérations sur les cubes de données

Les opérations *slice* et *dice* permettent respectivement d'extraire une tranche du cube et un sous-cube selon des prédicats sur les dimensions. Ces opérations sont illustrées respectivement aux figures 1.9 et 1.10.

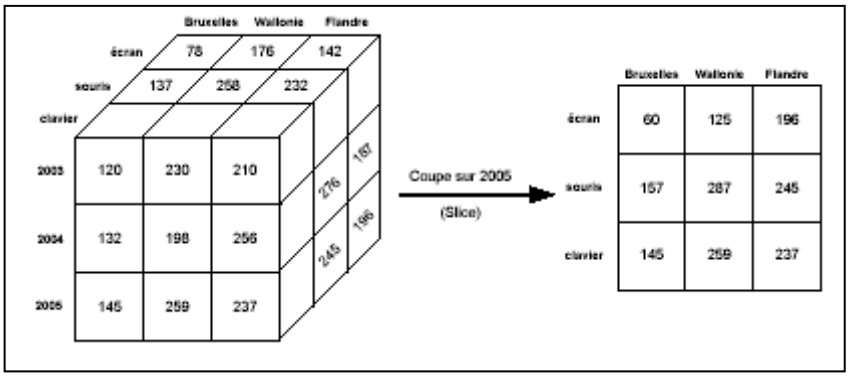


Figure 1.9. *Opération « slice », extraction d'une tranche d'un cube*

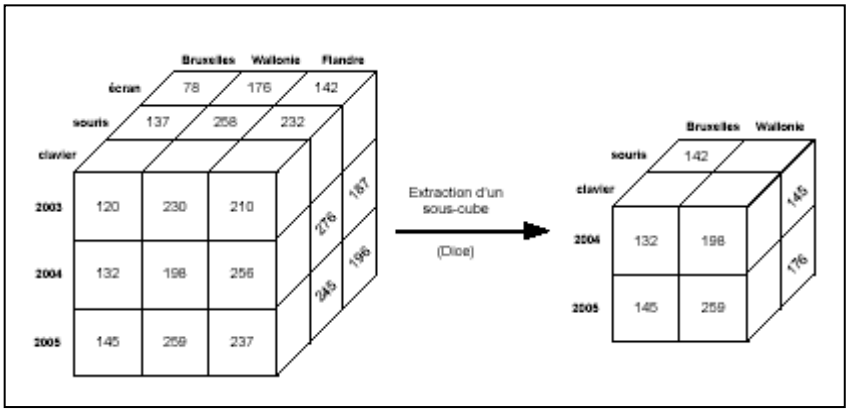


Figure 1.10. *Opération « dice », extraction d'un sous-cube*

Deux autres opérations importantes sont possibles sur les bases de données multidimensionnelles : *roll-up* et *drill-down* [Verhaegen, 06]. Ces deux opérations permettent de naviguer dans les données en suivant les hiérarchies de dimensions. La première, roll-up, permet d'agréger les données suivant une dimension. La deuxième, drill-down, permet de faire le contraire, c'est à dire de détailler les données. Par exemple, un roll-up sur la dimension temporelle nous permet de passer d'une vue par trimestre à une vue par année. Un drill-down permet de passer d'une vue par année à une vue par trimestre.

6. Maintenance d'un entrepôt de données

Après être conçu, un entrepôt de données doit être maintenu lors de l'application de modifications au niveau des sources de données.

Les données de l'entrepôt sont stockées sous forme de vues matérialisées sur les différentes sources de données. Une vue est une requête nommée. Une VM est une table contenant le résultat d'une requête [Bellatreche 00]. Les vues améliorent l'exécution des requêtes en précalculant les opérations les plus coûteuses comme la jointure et l'agrégation, et en stockant leurs résultats dans la base. En conséquence, certaines requêtes nécessitent seulement l'accès aux VM et sont ainsi exécutées plus rapidement. Les vues dans le contexte OLTP ont été largement utilisées pour répondre à plusieurs rôles : la sécurité, la confidentialité, l'intégrité référentielle, etc.

Une vue matérialisée (VM) est définie sur une ou plusieurs tables à partir d'une ou de plusieurs bases. Ces dernières peuvent être homogènes ou hétérogènes. La structure d'une vue peut être ainsi composée d'attributs de types complexes.
Les vues matérialisées permettent d'interroger plusieurs données à la fois, et surtout d'éviter les jointures et/ou des requêtes fréquentes et complexes. Pour mieux comprendre, l'exemple suivant illustre la notion des vues matérialisées.

Exemple 1 : Soient les deux relations *Employé* et *Emplyé-Projet* telles que :

Employé (EmpId, nom, DId, DoB, Picture);
Employé-projet (EmpId, PId, Finished);

La requête SQL suivante est utilisée pour créer la table d'une vue matérialisée VM1 comme montré dan la figure 1.11 :

SELECT EmpId, nom, Picture, PId, Finished,
FROM Employé, Employé-Projet
WHERE Employé.EmpId = Employé-Projet.EmpId

EmpId	Name	DI1	Dob	Picture
1	Jhon	21	3/26/53	…
2	Mary	22	8/25/68	…
3	Joe	20	4/15/73	…

EmpId	PId	Finished
1	456	5/1/95
1	30	10/1/96
2	100	2/21/94
2	30	9/31/96
3	150	7/31/96

Materialized View MVI (virtual view)

EmpId	Name	Did	Dob	Pict	PId	Finished
1	John	21	3/26/53	…	456	5/11/95
1	John	21	3/26/53	…	30	10/1/96
2	Mary	22	8/25/68	…	100	2/21/94
2	Mary	22	8/25/68	…	30	9/31/96
3	Joe	20	4/15/73	…	150	7/31/96

Figure 1.11 Création d'une VM VM1 à partir de tables Employé et Employé-Projet

La maintenance d'un entrepôt de données revient à la maintenance des vues matérialisées. Cette dernière consiste à répercuter les changements apparus au niveau des sources de données. Afin de s'adapter à l'évolution des processus d'analyse, l'évolution d'un entrepôt requiert d'une part, la maintenance de structure et d'autre part, la maintenance de données [Lambollez et al 95]. La maintenance structurelle a pour objectif de maintenir le schéma des

vues de l'ED suite aux changements des structures des sources de données. La maintenance des données consiste à alimenter l'entrepôt suite aux mises à jour des sources de données.

La question qui se pose concernant la maintenance des entrepôts est quand et comment répercuter les mises à jour des sources de données sur l'entrepôt ?

- ***Quand répercuter les MAJ des sources ?***

C'est à l'administrateur de définir à quel moment répercuter les MAJ, en effet, il peut choisir de le faire soit à chaque modification, soit d'une manière périodique.

- ***Comment répercuter les MAJ des sources ?***

On peut soit tout recompiler périodiquement, soit maintenir les vues d'une façon incrémentale en commençant d'abord par détecter les modifications (transactions, règles actives, etc.) et par les envoyer, ensuite, à un intégrateur qui se charge de déterminer les vues concernées, calculer les modifications et les répercuter.

7. Outils et technologies de l'entreposage de données

Plusieurs outils associés aux EDs ont été mis en œuvre menant à une bonne construction et une meilleure utilisation d'un entrepôt de données. Il existe des outils d'extraction, des outils de chargement, des outils d'administration, des outils d'accès aux données par les utilisateurs finaux, etc.

7.1 Outils d'extraction, de purification et de transformation

La sélection des outils corrects d'extraction, de nettoyage et de transformation est une étape cruciale au cours de la construction d'un entrepôt de données. Un nombre croissant de vendeurs de logiciels spécialisés concentrent leurs efforts sur le respect complet des exigences en termes d'implantation d'entrepôt de données au lieu de se contenter de simplement déplacer les données entre les plates-formes matérielles. Les tâches de capture des données d'un système source, le nettoyage et la transformation de ces données, puis le chargement des résultats dans un système cible s'effectuent, soit par des produits distincts, soit par une solution intégrée unique. Les solutions intégrées entrent dans une des catégories suivantes [Connoly et al 05]:

- Les générateurs de code;
- Les outils de duplication des données d'une base de données;
- Les moteurs de transformation dynamiques.

7.2 SGBD de l'entrepôt

Il y a peu de notions d'intégration associées à la base de données de l'entrepôt de données. Du fait de la maturité des produits de ce genre, il est prévisible que la plupart des bases de données relationnelles s'intégreront à d'autres types de logiciels. Par contre, des soucis naissent concernant la taille potentielle de la base de données de l'entrepôt de données. Le parallélisme avec la base de données fait surgir une inquiétude légitime, ainsi que d'autres

notions, telles que les performances, l'évolutivité, la disponibilité et l'aspect gérable, qui doivent être prises en compte lors du choix d'un SGBD.

Les exigences envers un SGBD d'entrepôt de données sont résumées dans les points suivants [Connoly et al 05]:

- Performances de chargement ;
- Gestion du chargement ;
- Gestion de la qualité des données ;
- Performances des requêtes ;
- Extension au-delà du téraoctet ;
- Évolutivité vers une masse d'utilisateur ;
- Mise en réseau de l'entrepôt de données ;
- Administration de l'entrepôt de données ;
- Intégration de l'analyse dimensionnelle ;
- Fonctionnalités de requête évoluées.

7.3 Outils d'administration et de gestion

Un entrepôt de données a besoin d'outils pour prendre en charge l'administration et la gestion d'un environnement si complexe. Ces outils sont relativement rares, surtout ceux qui intègrent bien les différents types de métadonnées et les opérations quotidiennes de l'entrepôt de données. Les outils d'administration et de gestion de l'entrepôt de données doivent être en mesure d'accomplir les tâches suivantes [Connoly et al 05]:

- La surveillance des données provenant de sources multiples;
- Les vérifications de la qualité des données et de leur intégrité;
- La gestion et la mise à jour des métadonnées;
- La surveillance des performances de la base de données pour garantir des délais efficaces de réponse aux requêtes et l'efficience de l'utilisation des ressources;
- L'audit de l'usage de l'entrepôt de données pour répercuter ensuite sur les utilisateurs les informations de frais d'utilisation;
- La duplication, la délimitation de sous-ensembles et la distribution des données;
- L'entretien de l'efficacité de gestion du stockage des données;
- L' « expurgation » des données;
- L'archivage et la sauvegarde des données;
- La prise en charge de la restauration, après une panne ou une défaillance;
- La gestion de la sécurité.

8. Evolution d'un entrepôt de données

Les entrepôts sont amenés à évoluer souvent et considérablement. En effet, La taille d'un entrepôt croît rapidement (de 20giga à 100giga en 2 ans). Cette évolution est due aux raisons suivantes [web 6]:

- L'introduction de nouvelles données à cause de l'extension géographique, le changement de fréquence des historiques, le changement du niveau de détail, etc.

- L'ajout de nouveaux éléments de données au modèle : par exemple l'ajout d'un attribut pour deux millions de n-uplets représente une augmentation considérable.
- La création de nouveaux index et de résumés.
- L'ajout de nouveaux outils tels que les générateurs de requêtes, les outils OLAP, etc.
- L'apparition de nouveaux utilisateurs.
- La complexité des requêtes.

Face à une telle évolution, le problème qui se pose est comment garantir l'extensibilité, la disponibilité et la maintenabilité ? Trois aspects majeurs sont concernés [web 6]:

(1) La **base de données** doit être extensible, disponible, facilement gérable (optimisation, indexation, gestion du disque automatique).

(2) Le **middleware** (gestionnaire de transactions, gestionnaire d'accès,...) doit être performant et cohérent.

(3) **L'intégration des outils** doit se faire avec un souci de compatibilité, les outils doivent être conformes au plus grand nombre de standards.

9. Avantages apportés par les entrepôts de données

La mise en place réussie d'un entrepôt de données apporte des bénéfices majeurs à l'organisation qui l'a menée, notamment [Connoly et al 05]:

- **Un taux de rendement du capital investi (TRCI) potentiellement élevé :** Une organisation doit faire appel à une vaste quantité de ressources pour garantir le succès de l'implantation d'un entrepôt de données et la variation des coûts est énorme : de quelques 75 000 dollars américains à plus de 15 millions de dollars, selon les solutions techniques envisagées. Cependant, une étude menée en 1996 par l'IDC (International Data Corporation) a établi que le taux de rendement du capital investi moyen sur trois ans (TRCI) dans l'entreposage de données a atteint 401 %, avec plus de 90 % des compagnies sondées constatant un TRCI de plus de 40 %, la moitié des entreprises atteignant un TRCI supérieur à 160 % TRCI et un quart des sociétés arborant un TRCI de plus de 600 % (IDC, 1996).

- **Un avantage concurrentiel :** Le fort taux de rendement du capital investi pour les compagnies qui ont réussi à implanter un entrepôt de données montre à l'évidence l'avantage concurrentiel énorme qui accompagne cette technologie. L'avantage concurrentiel se manifeste surtout par la mise à la disposition des décideurs d'informations autrefois indisponibles, inconnues et non collectées par exemple sur les consommateurs, les tendances et les demandes.

- **Une productivité accrue de la part des décideurs d'entreprise :** L'entreposage de données améliore la productivité des décideurs d'entreprise, en créant une base intégrée de données cohérentes, orientées sujet et historiques. L'entrepôt de données intègre des données de nombreux systèmes, à priori incompatibles, en une forme qui offre une vue

cohérente de l'organisation. En transformant les données en informations exploitables, l'entrepôt de données donne aux décideurs d'entreprises la possibilité de mener des analyses plus constructives, précises et cohérentes.

10. Problèmes liées aux entrepôts de données

Bon nombre de problèmes existent concernant le développement et la gestion d'un entrepôt de données. Ces problèmes sont détaillés dans [Connoly et al 05], on les résume dans les points suivants :

- La sous-estimation des ressources nécessaires au chargement des données ;
- Des problèmes masqués dus aux systèmes sources ;
- Les données requises ne sont pas toutes collectées ;
- L'augmentation permanente des demandes des utilisateurs ;
- L'homogénéisation des données : le concepteur de l'entrepôt peut être tenté d'accentuer les similitudes au lieu des différences des données ;
- La forte sollicitation des ressources ;
- La propriété des données : parfois, l'entreposage des données risque de changer l'attitude des utilisateurs finaux. Les données *propres* à un département ou à un service spécifique sont accessibles par un autre service.
- La forte maintenance ;
- Les projets de longue durée ;
- La complexité de l'intégration.

11. Conclusion

Tout au long de ce chapitre, nous avons synthétisé les différents concepts liés aux entrepôts de données. Pour cela, nous avons défini ce qu'est un entrepôt de données ainsi que son architecture. Nous avons introduit aussi la notion des mini-entrepôts de données (Datamarts). Par la suite, nous avons présenté un concept important utilisé pour la modélisation des données au niveau d'un entrepôt, c'est la modélisation multidimensionnelle. La maintenance d'un ED et son évolution ont été présentées aussi dans ce chapitre. Nous avons cité également quelques outils liés aux entrepôts. En dernier, nous avons mentionné les avantages d'une solution d'ED ainsi que les problèmes liés au développement et à la gestion d'un ED.

Après avoir cerné tous les concepts et les notions liés aux ED, il est intéressant de mettre l'accent sur les différentes approches de construction d'ED vu que notre objectif principal est de proposer une démarche pour la conception d'ED. Ces approches seront présentées dans le chapitre suivant.

Chapitre 2

Approches de conception d'entrepôts de données

1. Introduction

Depuis les années 90, une nouvelle voie de recherche portant sur la conception des projets d'entreposage a été ouverte. En effet, Comme pour tout système informatique, cette conception passe généralement par les trois niveaux de conception habituels: conceptuel, logique et physique [Khouri 09]. Les premiers travaux de projets d'entreposage généraient directement le schéma logique de l'entrepôt sans passer par une étape de modélisation conceptuelle. Cette conception s'effectue soit selon la vision d'Inmon [Inmon 92], en construisant le schéma logique ensuite physique de l'ED à partir des *sources de données* par une approche itérative ; soit selon la vision de Kimball [Kimball et al 96] à partir des *besoins des décideurs et utilisateurs* de l'ED.

La modélisation conceptuelle a longtemps été négligée dans les projets de développement des EDs. Cependant, le passage direct aux phases logiques et physiques présente plusieurs inconvénients, cette manière de faire ne permet pas par exemple de :

- Etablir une communication entre les concepteurs de l'entrepôt et les utilisateurs de ce dernier (décideur), ce qui conduit dans la plupart du temps au rejet du projet d'entreposage ;

- Détecter les éventuelles erreurs de conception qui doivent être décelées et corrigées dès le départ avant le passage au modèle logique ou physique.

L'axe de recherche de « conception des EDs » a commencé à présenter un certain intérêt lorsque des études ont montré que les principales causes possibles d'échec des projets d'entreposage à constituer un support au processus de prise de décision, est l'absence d'une

vue globale du processus ou *l'absence d'une méthodologie de conception de l'entrepôt de données* [Golfarelli et al 98].

Dans ce chapitre, nous nous intéressons aux méthodes de conception des EDs. Pour cela, nous commençons par décrire les différentes phases du cycle de vie d'un ED. Ensuite nous détaillerons les différentes étapes de la phase de conception. Et enfin nous présentons les différentes approches de conception d'un ED

2. Cycle de vie d'un entrepôt de données

Le cycle de vie d'un ED peut être structuré en trois principales phases [Golfarilli 09]:

a - **Planification** : cette phase vise à préparer le terrain pour le développement de l'entrepôt. Elle inclut les tâches suivantes :

- Déterminer l'étendue du projet ainsi que les buts et objectifs de l'entrepôt à développer
- Evaluer la faisabilité technique et économique de l'entrepôt
- Identifier les futurs utilisateurs de l'entrepôt

b - **Conception et implémentation** : cette phase consiste à développer le schéma de l'entrepôt et à mettre en place toutes les ressources nécessaires à son implémentation et à son déploiement. Cette conception comporte cinq principales étapes qui seront détaillées dans le point suivant.

c - **Maintenance et évolution** : la maintenance de l'entrepôt implique l'optimisation de ses performances périodiquement. L'évolution de l'entrepôt concerne la mise à jour de son schéma en fonction des différents changements survenant au niveau des sources ou des besoins des utilisateurs.

Pour les besoins de notre problématique qui consiste à proposer une démarche de conception d'un ED, nous détaillons davantage la deuxième phase « conception et implémentation ».

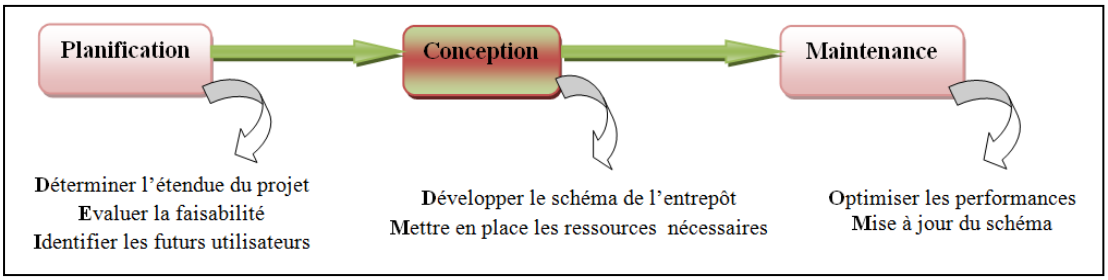


Figure 2.1. *Cycle de vie d'un ED et tâches effectuées*

3. Conception d'un entrepôt de données

Même sil n'existe actuellement aucun consensus sur une méthode de conception précise, la plupart des travaux s'accordent à distinguer les étapes suivantes pour la conception d'un ED :

- analyse des besoins,
- modélisation conceptuelle,
- modélisation logique,
- processus ETL et
- modélisation physique.

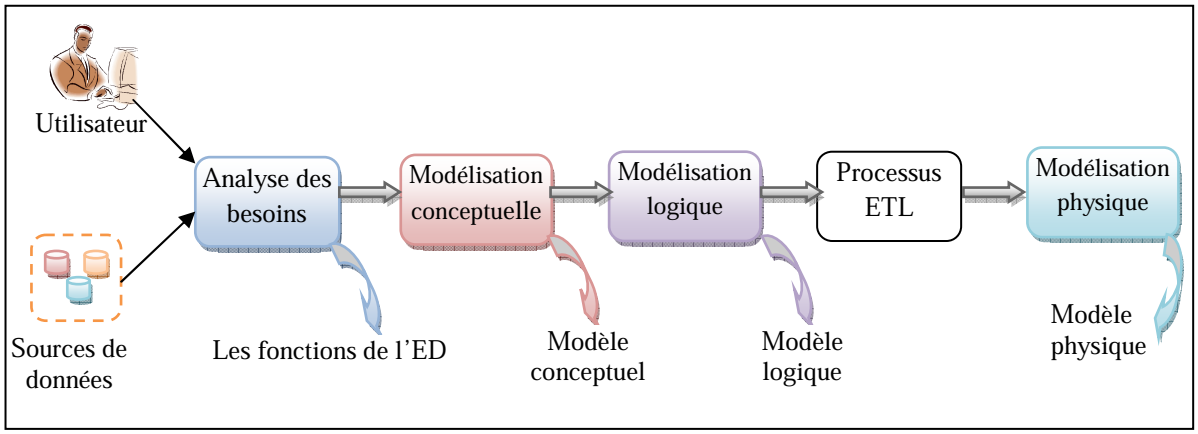


Figure 2.2. *Etapes de conception d'un entrepôt de données*

3.1 Analyse des besoins

L'analyse des besoins joue un rôle clé dans tout projet logiciel en permettant de réduire significativement le risque d'échec du projet [Rizzi et al 06]. L'ingénierie des besoins a été définie comme le processus de développement des besoins selon un processus itératif et coopératif d'analyse du problème, de documentation des observations résultantes dans divers formats de représentation et de vérification des résultats obtenus [Bubenko et al 94]. La spécification des besoins d'un projet d'entrepôt de données permet de déterminer quelles seront les différentes fonctions de l'entrepôt ainsi que l'ensemble des informations requises que ce dernier doit couvrir (quelles sont les données qui doivent être accessibles, comment ces données sont transformées, organisées, calculées et agrégées) [Bruckner et al 01], [Shiefer et al 02].

Nous présentons dans les paragraphes suivants les méthodes utilisées pour la détermination des besoins ainsi que quelques travaux réalisés dans ce cadre.

- **Méthodes de collection des besoins**

Deux méthodes sont utilisées pour collecter les besoins : une *collecte orientée source* et une *collecte orientée utilisateur*. Une approche orientée source se limite à identifier les besoins de l'ED à partir de l'ensemble des données disponibles au niveau des sources. La collecte orientée utilisateurs identifie les besoins des utilisateurs et décideurs de l'entreprise et présente l'avantage principal de se concentrer à fournir un modèle répondant à ce qui est exigé plutôt que ce qui est disponible. Un ED, par essence, doit être construit à partir des sources de données. Une collecte orientée sources est par conséquent indispensable. Les besoins utilisateurs n'ont cependant pas toujours été considérés lors du développement des EDs. Ce sont généralement les travaux récents qui prennent en compte les besoins utilisateurs. Même si les méthodologies de développement d'un ED diffèrent généralement, elles

reconnaissent de plus en plus la nécessité d'identifier et de modéliser les besoins des utilisateurs [Sen et al 05].

La spécification des besoins (collectés à partir des sources et/ou des utilisateurs) passe par les activités suivantes : La collecte des besoins, l'analyse des besoins, la validation des besoins et la modélisation des besoins [Bellard et al 98].

- La *collecte des besoins* consiste à collecter les besoins souvent selon des approches itératives. La collecte à partir des sources passe par une analyse détaillée des sources de données. la collecte orientée utilisateur passe par des techniques de réunions, d'étude de la documentation existante, d'interviews et de brainstorming. Ces besoins sont ensuite documentés (généralement de manière informelle). La collecte a pour but de comprendre le domaine qui devra être modélisé.

- L'*analyse des besoins* consiste à étudier les besoins collectés, où des modèles initiaux peuvent être construits.

- La *validation des besoins* consiste à valider les modèles initiaux construits lors de l'analyse des besoins, avec les utilisateurs ainsi qu'avec les sources de données existantes. L'entrepôt est ainsi construit de manière itérative, ce qui garantit selon [Bellard et al 98] un développement rapide. L'expérience a montré que même après implémentation de l'entrepôt de données, un retour vers la phase d'identification des besoins est possible. L'utilisation et la maintenance du modèle de données est effectué continuellement durant tout le cycle de vie de l'entrepôt.

- **Travaux liés à l'analyse des besoins**

Les travaux sur l'analyse des besoins en ED restent assez limités. Certains travaux suggèrent l'utilisation des techniques utilisées pour l'analyse des besoins en BD, d'autres estiment qu'il faut des méthodes adaptées aux spécificités des EDs. On retrouve quelques travaux portant sur des *méthodologies* présentant des directives générales permettant de mener à bien l'analyse des besoins dans un projet d'ED. Nous présentons deux travaux (*Rilston et al., Winter et al.*) souvent référencés : [Rilston et al 03] et [Winter et al 04].

Rilston et al. [Rilston et al 03] proposent une méthodologie comprenant quatre phases :

- *Phase d'initialisation* qui consiste à identifier les utilisateurs cibles ainsi que le type d'application d'analyse le plus dominant ce qui permettra de choisir les modèles de données à utiliser, les sources de données pertinentes et les processus de décisions devant être considérés.
- Une deuxième *phase d'analyse* des sources de données existantes et les rapports utilisés régulièrement par les utilisateurs, et à développer des schémas de données agrégés.
- Une troisième phase qui consiste à *établir des priorités* entre les besoins en informations.

- Une quatrième phase consistant à *concevoir un modèle initial* des besoins servant de base au modèle conceptuel.

Winter et al. [Winter et al 04] proposent également une méthodologie comprenant quatre phases :

- *Planning de gestion des besoins* : consiste à définir les objectifs du projet par les utilisateurs et les concepteurs, définir les règles d'intégration des sources de données et à établir un planning de gestion des besoins du projet (contraintes de temps, délimiter les frontières du projet…).
- *Spécification des besoins* : s'effectue par un processus itératif d'acquisition, représentation et validation des besoins. La spécification s'effectue à travers plusieurs (sous) processus en plusieurs itérations en utilisant un modèle en spirale, où chaque itération produit un ensemble de spécifications plus raffiné.
- *Validation des besoins* par l'équipe de développement, les utilisateurs, et les experts du domaine. L'équipe de validation attache une liste d'actions à chaque problème identifié et des retours arrière vers les phases précédentes peuvent être envisagés.
- *Suivi et gestion de l'évolution des besoins* : la gestion doit être effectuée à deux niveaux : une gestion de l'évolution des besoins utilisateurs, et une gestion de l'évolution de l'architecture des bases de données sources afin d'étudier l'impact d'éventuels changements sur les modèles suivants (conceptuels, logiques et physiques).

3.2 Modélisation conceptuelle

Cette étape consiste à établir une méthode permettant de définir le schéma conceptuel d'un entrepôt, annoté par les concepts multidimensionnels (faits, mesures et dimensions). Le but de cette modélisation est de fournir une représentation abstraite de la situation en cours d'étude indépendamment de tout logiciel technique. Un modèle conceptuel est caractérisé par le domaine concerné, le formalisme utilisé pour modéliser le schéma conceptuel, et le point de vue correspondant aux besoins des utilisateurs [Fankam et al 09]. Le modèle conceptuel obtenu doit se conformer à la modélisation multidimensionnelle adaptée aux modèles d'EDs permettant d'organiser les données entreposées de manière à faciliter leur analyse décisionnelle. Une synthèse des travaux proposant des méthodes de conception de projets d'entreposage est présentée dans le paragraphe 3 de ce chapitre.

3.3 Modélisation logique

Cette étape consiste à dériver le schéma logique de l'entrepôt à partir de son schéma conceptuel, et à l'adapter aux particularités du serveur de l'entrepôt choisi (ROLAP, MOLAP ou HOLAP). Ce passage implique la gestion de certains aspects techniques comme la fragmentation des dimensions en hiérarchies, la gestion de l'agrégation des données, la gestion des dimensions génériques et des mesures non additives, etc.

3.4 Processus ETL (Extract-Transform-Load)

Cette étape consiste à effectuer les transformations nécessaires au chargement des données des sources au niveau du schéma logique de l'entrepôt. Ceci se fait en trois étapes [Marhoumi 06] :

- Extraction : consiste à récupérer les données à partir des systèmes sources. Cette étape nécessite de gérer la synchronisation des processus d'extraction afin d'assurer l'intégrité des données chargées.

- Transformation : consiste en une série de règles permettant le formatage des données extraites selon le schéma cible de l'entrepôt, comme par exemple assigner de la sémantique aux données sources et associer les champs sources aux champs cibles.

- Chargement : permet l'alimentation de l'entrepôt par les données en respectant les contraintes du SGBD cible. Cette étape doit être validée afin de détecter et corriger toute erreur ayant survenue durant le chargement.

3.5 Modélisation physique

Cette étape consiste à implémenter physiquement le schéma de l'entrepôt, et à spécifier les techniques et schémas d'optimisation de l'entrepôt. Les techniques d'optimisation peuvent être [Bellatreche 00] :

- les vues matérialisées : une vue matérialisée est une table contenant les résultats d'une requête.
- les techniques d'indexation : représentées par des structures permettant d'associer à une clé d'un n-uplet l'adresse relative de ce n-uplet. Plusieurs types d'index ont été proposés, on distingue les index définis sur une seule table ou vue (index en B-arbre, index de hachage et index bitmap) et les index de jointures définis sur plusieurs tables dans un schéma en étoile.
- la fragmentation : technique consistant à partitionner le schéma d'une base en plusieurs sous-schémas pour réduire le temps d'exécution des requêtes. On distingue la fragmentation horizontale (en fonction des tuples) de la fragmentation verticale (en fonction des attributs).

4. Approches de conception des projets d'entreposage

Un retour sur l'historique de conception des projets d'entreposage a permis de mieux analyser les différentes approches et méthodes de conception proposées. Cet historique remonte aux années 90, peu après la naissance du concept d'ED. La notion d'entrepôt de données étant issue de l'industrie, les premières approches de conception généraient directement le schéma logique de l'entrepôt sans passer par une phase de modélisation conceptuelle. La mise en place d'une vraie phase de conception pour le développement des entrepôts s'est imposée par la suite afin de réduire le risque d'échec du projet d'entreposage. Deux principales catégories de méthodes ont ainsi été proposées : une première catégorie se

basant uniquement sur les sources de données et une deuxième catégorie reconnaissant la nécessité d'inclure les besoins des utilisateurs de l'entrepôt lors de sa conception. Toutes ces méthodes ont cependant ignoré une problématique primordiale qui est l'intégration des données dues à l'hétérogénéité des sources.

Une toute nouvelle génération d'approches proposant la conception des entrepôts en utilisant des *ontologies* se met en place. Différentes études ont prouvé l'efficacité des ontologies pour assurer une intégration sémantique des données de manière automatique. Cette section, s'appuyant sur l'historique de conception des projets d'EDs, propose de classifier les méthodes de conception en trois grandes générations d'approches : approches directes (passage direct au modèle logique), approches traditionnelles (mise en place d'une vraie phase de conception), et les approches ontologiques (conception à base d'ontologies).

Pour chaque type d'approche, nous commençons par analyser son principe ainsi que les différentes méthodes de conception proposées dans la littérature. Par la suite, nous présentons les principales difficultés liées à ces approches.

4.1 Les approches directes

4.1.1 Principe et travaux connexes

Les approches de conception directes définissent le schéma logique d'un ED sans passer par une phase de modélisation conceptuelle. Cette manière de faire s'explique par le fait que les entrepôts soient issus originellement de l'industrie qui s'intéresse aux aspects pratiques et ne donne pas suffisamment d'importance aux problèmes de conceptualisation, mais également par le fait que les modèles logique et physique d'un entrepôt, jouent un rôle important dans l'amélioration et l'optimisation des performances du système décisionnel final, ce qui représente une des principales préoccupations des entrepôts. Ce schéma logique est défini à partir des sources de données (selon Inmon) [Inmon 92] ou des besoins organisationnels ou des deux (selon Kimball) [Kimball et al 96].

Selon la vision d'Inmon [Inmon 92], un schéma de données de l'entrepôt est construit à partir des sources de données selon un processus itératif. L'analyse des sources s'effectue selon les étapes suivantes :

- Trouver les données
- Accéder aux données
- Intégrer les données
- Fusionner les données

Un modèle de données est généré à chaque itération, qui servira comme schéma de base pour l'itération suivante. A chaque itération des tables de l'entrepôt sont conçues et permettent de construire une version de l'entrepôt de données. La mise en place de cet entrepôt auprès des utilisateurs permet de mieux cerner les besoins en informations. L'itération qui suit alimente l'entrepôt de l'itération précédente par de nouvelles données et organise davantage ses données. Ce développement itératif peut s'étendre sur plusieurs années.

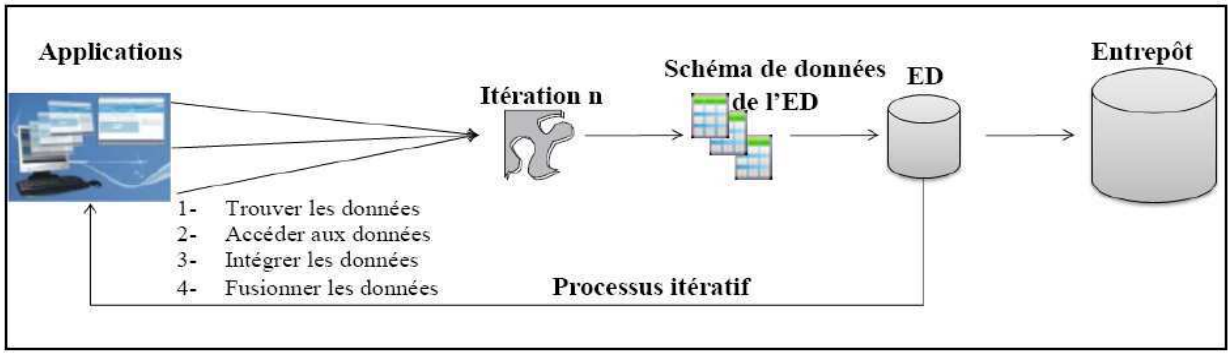


Figure 2.3. *Conception d'un ED à partir des sources (selon Inmon)* [Khouri09]

Selon la vision de Kimball [Kimball et al 96], un schéma en étoile de l'entrepôt est construit en analysant les besoins de l'entreprise. Ceci s'effectue en identifiant les besoins organisationnels, en déterminant la granularité des données à représenter dans le schéma, et en identifiant les faits et dimensions de ce schéma.

Selon cette vision, les décideurs et analystes d'une organisation ont une connaissance sur les faits qu'ils souhaitent analyser, ainsi que les dimensions selon lesquelles ils souhaitent analyser ces faits. Ils communiquent ces informations au concepteur, qui pioche ensuite dans les sources afin d'identifier les données qui peuvent correspondre aux faits et aux dimensions communiqués.

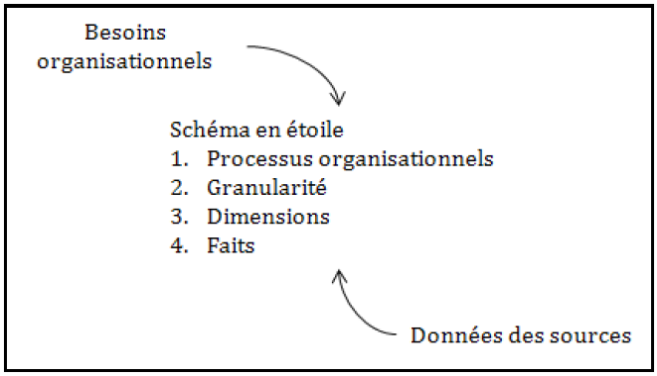


Figure 2.4. *Conception d'un ED à partir des besoins organisationnels (selon Kimball*

Les travaux s'inscrivant dans ce type d'approches, génèrent généralement le schéma logique de l'ED à partir des sources comme *Boehnlein* et *Moody*, ou à partir des besoins comme *Kimball*.

Boehnlein [Boehnlein et al 99] dérivent un schéma en étoile de l'entrepôt à partir des schémas E/A des BDDs sources, où les attributs quantitatifs sont considérés comme mesures potentielles. Les dimensions sont choisies parmi les entités liées aux mesures et dépendant d'elles.

Moody [Moody et al 00] propose de développer un schéma multidimensionnel à partir des schémas E/A à travers les étapes suivantes : **(1)** classer les entités en entités faits (ce sont les entités enregistrant des détails concernant des évènements particuliers) et en entités dimensions (ce sont les entités directement reliées aux entités faits par une relation (1,n)), **(2)**

identifier les hiérarchies qui correspondent à toute séquence d'entités reliées entre elles par des relations (1,n), et raffiner le modèle obtenu.

Kimball [Kimball et al 96] propose une méthode appuyant sa vision. Le schéma en étoile de l'entrepôt est construit en analysant les besoins identifiés à partir des processus organisationnels, selon les étapes suivantes :

- Sélectionner les processus organisationnels à analyser : un processus est défini comme activité naturelle effectuée dans l'organisation qui est généralement appuyée par une source de données d'un système transactionnel.
- Déterminer la granularité des données à prendre en compte, i.e le niveau de détail des données à analyser qui doit être représenté dans le schéma en étoile. Il est recommandé de considérer les données de granularité très fine qui sont très détaillées et ne peuvent être divisées. Ces données dites « atomiques » peuvent être agrégées de différentes manières.
- Choisir les dimensions ou axes d'analyse possibles, en répondant à la question : Comment les analystes décrivent-ils les données résultant des processus organisationnels sélectionnés ?
- Déterminer les tables de faits représentant l'objet de l'analyse pour chaque processus en répondant à la question : « que doit-on mesurer ? »

4.1.2 Analyse

Cette façon de faire des approches directes implique un important travail d'abstraction, et aussi une connaissance préalable du modèle multidimensionnel (faits et dimensions) par les décideurs (pour les méthodes se basant sur la vision de Kimball). De plus, le passage direct au schéma logique présente certaines limites. En effet, ces schémas :

- N'offrent pas un niveau d'abstraction indépendant de toute problématique d'implémentation : ils impliquent le choix d'une implémentation relationnelle et la prise de décisions concernant le niveau physique (normalisation des tables).
- Ne permettent pas d'instaurer une communication entre les concepteurs et les utilisateurs, ce qui est reconnu par plusieurs travaux et études comme un risque important de rejet du projet final [Gam et al 06].
- Omettent la détection des éventuelles erreurs de conception qui doivent être décelées et corrigées dès le départ avant le passage au modèle logique ou physique.

Un autre problème important concerne l'hétérogénéité des schémas des sources de données. Le concepteur doit analyser les sources de données afin de construire le schéma logique. Il se trouve ainsi devant une tâche complexe, où il doit gérer les différents conflits (syntaxiques et sémantiques) entre les données des sources. Imaginons par exemple que le décideur souhaite analyser le prix de vente d'un produit selon certaines périodes, et qu'au niveau des sources, on trouve les attributs '*Prix'* dans source 1, '*Price'* dans source 2 et '*Prix de vente*' dans source n. le concepteur se retrouve confronté à des questions du style :

l'attribut '*Prix'* désigne-t-il le prix de vente ou le prix de production ? l'attribut '*Price'* est-il un synonyme de l'attribut '*Prix'*, l'attribut '*Prix de vente*' représente-t-il le prix hors taxe ou le prix TTC ? …

Un travail important doit être fait par le concepteur pour assigner une sémantique à ces différents concepts. Ce travail se complique davantage lorsqu'on a affaire à un grand nombre de sources de données et qu'il n'existe aucune documentation disponible sur les sources de données.

4.2 Les approches traditionnelles

Les approches traditionnelles de conception ont été proposées lorsque différentes études ont montré que le manque ou l'absence d'une phase de modélisation conceptuelle est une cause possible d'échec des projets d'entreposage [Golfarelli et al 98]. Ces approches s'intéressent à mettre en place une vraie phase de conception lors du développement d'un ED. Les premiers travaux ont proposé de concevoir un ED principalement à partir des sources de données. D'autres travaux ont suivi, s'inspirant de la conception des BDs, et ont mis le point sur la nécessité d'inclure les besoins des utilisateurs, en plus des sources de données. Diverses études ont même été menées prouvant l'importance d'impliquer les besoins des utilisateurs dans la conception d'un ED [Gam et al 06]. On distingue ainsi deux catégories de méthodes dans les approches traditionnelles : les méthodes orientées sources (ou approches ascendantes) et les méthodes orientées besoins (ou approches descendantes). La mise en place d'une phase conceptuelle a pour but de :

- Instaurer un dialogue entre le concepteur et les futurs utilisateurs de l'entrepôt (spécialement pour les méthodes orientées besoins où les besoins utilisateurs sont modélisés et formalisés).
- Fournir une base stable pour la phase de modélisation logique.
- Fournir une documentation expressive et non ambiguë du processus de conception.

4.2.1 Les méthodes orientées sources

Les premières méthodes découlent de la définition d'Inmon qui indique que le développement d'un ED repose sur les données des sources par opposition aux méthodes de développement des bases de données traditionnelles qui reposent sur les besoins des utilisateurs. Ces méthodes étendent les schémas E/A par des faits et des dimensions [Tryfona et al 99], ou bien dérivent un modèle multidimensionnel de l'entrepôt à partir des schémas E/A ou des schémas relationnels des sources en utilisant des règles de transformation [Golfarelli et al 98], [Husseman et al 00].

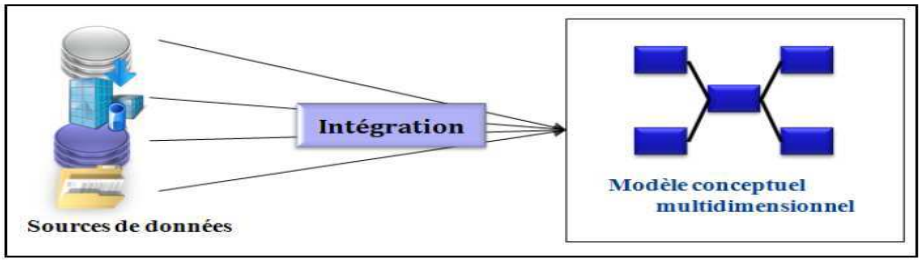


Figure 2.5. *Méthodes orientées sources*

Tryfona et al. [Tryfona et al 99] proposent un modèle conceptuel d'un ED nommé "starER" qui consiste à étendre un schéma E/A aux concepts de faits et de dimensions inhérents aux modèles multidimensionnels, ainsi que les différentes relations entre les niveaux de dimensions et les faits comme la spécialisation, la généralisation et l'agrégation. Ce modèle est accompagné d'une représentation graphique. Les faits sont représentés graphiquement par un cercle et correspondent aux sujets analysés. Les entités correspondant à un ensemble d'objets du monde réel partageant les mêmes propriétés, sont représentées par un rectangle. Les relations entre les entités et les faits sont représentées par un diamant. Les cardinalités possibles de ces associations ne doivent être que de type (n,m), (m,1) ou (1,m). Des attributs peuvent être associés aux entités faits et sont représentés par des ovales.

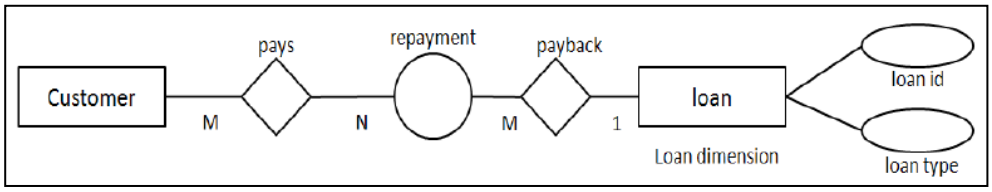


Figure 2.6. *Exemple d'un schéma StarER* [Khouri09]

Golfarelli et al. [Golfarelli et al 98] proposent une méthode semi automatisée, souvent référencée par d'autres travaux, pour concevoir un modèle conceptuel d'un ED baptisé Dimensional Fact model (DF model) à partir des modèles E/A décrivant les BDDs sources. Le modèle conceptuel proposé consiste en un ensemble de schémas de faits (fact schemes) représentés graphiquement sous forme d'un arbre dont la racine est le fait.

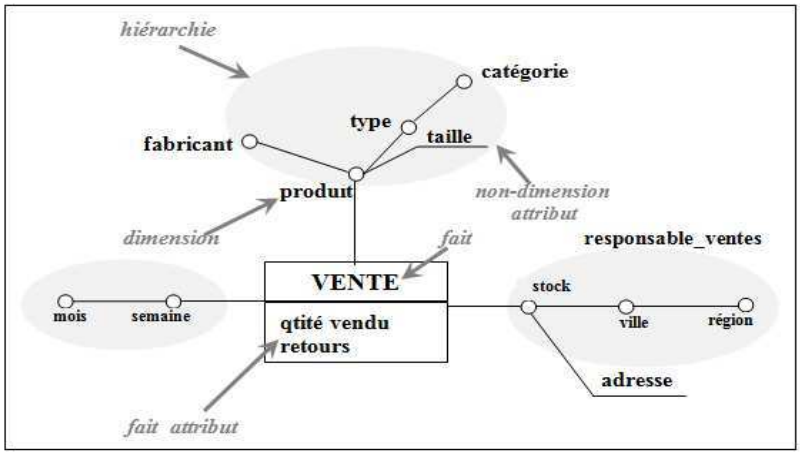


Figure 2.7. *Schéma de fait et de dimension*

La construction du modèle DF est constituée de six étapes : **(1)** Définir les faits (les entités et relations ayant subi le plus de modifications et de mises à jour sont considérés comme de bons candidats). Un fait peut être représenté par une ou plusieurs entités. **(2)** Pour chaque fait défini, construire l'arbre des attributs (les attributs des entités sélectionnées comme Fait), dont la racine est l'identifiant du fait et les noeuds ses attributs, **(3)** affiner l'arbre en éliminant les attributs les moins intéressants pour l'utilisateur, **(4)** définir les dimensions qui correspondent aux nœuds enfants de la racine de l'arbre, **(5)** définir les mesures des faits (généralement c'est

le nombre d'instances du fait, ou bien une expression du type sum/ average/ maximum/ minimum évalués sur les attributs de l'arbre), **(6)** et finalement définir les hiérarchies entre attributs.

Husemann et al. [Husseman et al 00] présentent une méthode permettant de générer un schéma conceptuel d'un entrepôt à partir des BDDs existantes. Le processus de conception s'effectue en analysant les attributs pertinents des bases pour spécifier leur rôle (fait, mesure ou dimension). Un fait représente un élément d'information atomique. Les mesures sont représentées par des éléments quantifiant le fait, et les dimensions comme des éléments le qualifiant. Les mesures doivent respectées les contraintes d'agrégation. Les hiérarchies de dimension sont définies en déterminant les dépendances fonctionnelles entre les différents niveaux de la hiérarchie.

Un modèle conceptuel défini uniquement à partir des sources de données risque de ne pas être en concordance avec les attentes des décideurs d'une organisation. Impliquer les décideurs qui sont les futurs utilisateurs de l'entrepôt est une étape primordiale. Des travaux se sont ainsi inspirés de la philosophie des bases de données, et se sont intéressés à développer des méthodes de conception d'EDs prenant en compte les besoins décisionnels.

4.2.2 Les méthodes orientées besoins utilisateurs

De plus en plus de travaux comme [Gam et al 06], [Giorgini et al 05] reconnaissent la nécessité d'identifier et de modéliser les besoins des utilisateurs pour la conception d'un ED. Différentes études ont en effet démontré qu'1/3 des projets d'entrepôt de données échouent à atteindre leurs objectifs, principalement à cause du manque de communication entre les concepteurs et les utilisateurs (décideurs) et à cause d'une mauvaise gestion du projet [Gam et al 06].

4.2.2.1 Analyse des besoins utilisateurs dans un projet d'ED

La modélisation des besoins des utilisateurs s'effectue soit en utilisant des modèles spécifiques proposés par les auteurs mêmes (exemple : le modèle MAP [Gam et al 06]), en utilisant les modèles fournis par le framework i* comme [Giorgini et al 05], ou en utilisant une modélisation objet comme [Firestone97], [List et al 00], [Mazon et al 08].

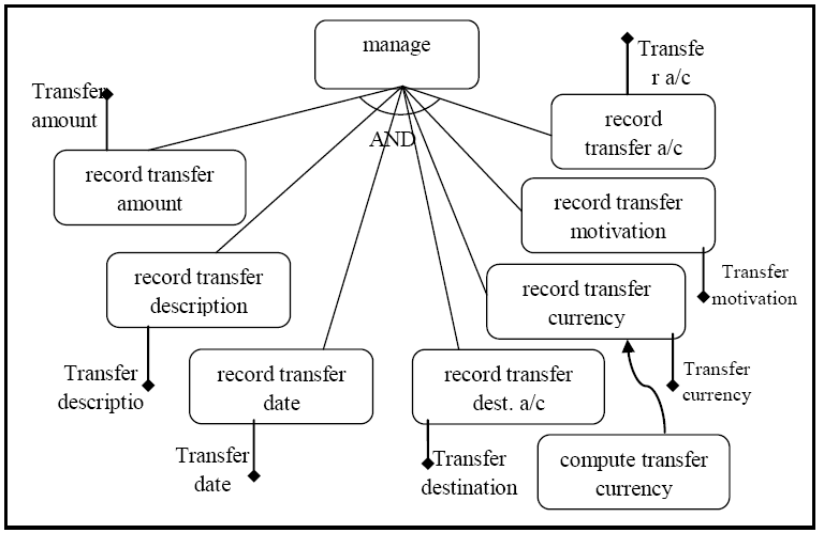


Figure 2.8. *Exemple d'un diagramme rationnel du framework i**

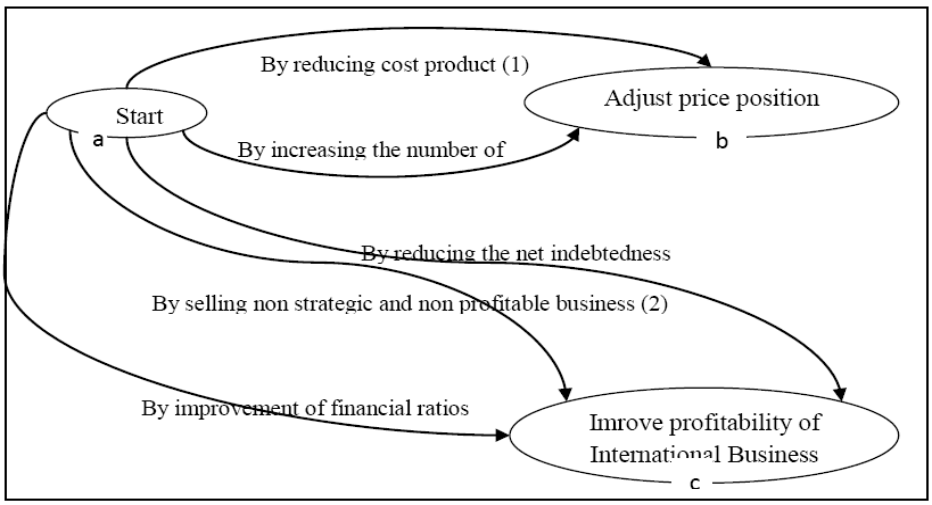


Figure 2.9. *Exemple d'un modèle MAP [Gam et al 06]*

Ces besoins peuvent être exprimés sous différentes formes. L'objet d'analyse de ces besoins peut être les *buts* de l'organisation, les *processus* de l'organisation ou les besoins de chaque *utilisateur* cible.

L'analyse orientée ***processus*** (travaux de [List et al 00], [Bruckner et al 01], [Shiefer et al 02]) analyse les besoins en identifiant les processus métiers de l'organisation. Un processus est décrit dans ce cas comme un ensemble structuré et mesuré d'activités, permettant de fournir un produit spécifique pour un client ou un marché particulier. Un processus métier est décrit comme un ensemble d'activités ordonnées dans l'espace et le temps (s'effectuant dans un intervalle précis de temps) ayant des entrées et des sorties bien définies. Les auteurs mettent cependant le point sur le fait que l'analyse des processus clés n'est pas toujours faisable. La raison est que ces processus de décision se constituent souvent de tâches non structurées, de plus les décideurs refusent généralement de divulguer ce type d'informations.

L'analyse orientée ***but*** (travaux de [Firestone97], [Gam et al 06], [Frendi et al 03], [Bonifati et al 01], [Prakash et al 03], [Giorgini et al 05], [Mazon et al 08]) analyse les besoins en identifiant les buts et objectifs de l'organisation. Les buts peuvent être définis comme des objectifs de haut niveau de l'organisation ou du système. Ils guident les diverses décisions de différents niveaux de l'entreprise.

L'analyse orientée ***utilisateurs*** (travaux de [Burmester et al 06]) s'intéresse à identifier les utilisateurs cibles et à spécifier et analyser leur besoins individuels pour les intégrer dans un modèle de besoins unifié.

Les méthodes basées sur les besoins permettent de générer un schéma conceptuel multidimensionnel de l'entrepôt en prenant en compte les besoins des décisionnels de l'organisation. Les sources de données restent cependant le cœur de construction de l'ED. Un mapping entre les sources et les besoins doit être effectué, ce mapping nécessite une étape d'intégration.

4.2.2.2 Mapping sources / Besoins

Le mapping Sources/Besoins peut s'effectuer de trois manières, classées comme suit :

a- Approche postérieure

Dans cette approche, des schémas conceptuels multidimensionnels sont définis à partir des sources de données. D'autres schémas conceptuels multidimensionnels sont générés à partir de l'analyse des besoins. Un mapping entre ces deux ensembles de schémas est ensuite effectué afin de générer le schéma conceptuel final de l'ED. Ce type de méthodes présente deux principaux inconvénients :

- Le premier réside dans la possibilité de générer un nombre de schémas inutiles qui ne seront pas exploités.
- Le deuxième inconvénient se présente dans la difficulté d'effectuer un mapping entre les schémas. Les problèmes d'hétérogénéité et donc d'intégration apparaissent à deux niveaux : une intégration des sources lors de la génération du premier ensemble de schémas conceptuels ; et une intégration lors du mapping entre les schémas obtenus à partir des sources et à partir des besoins.

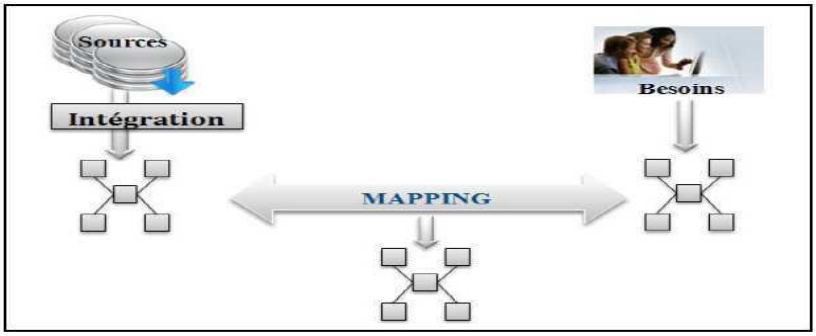


Figure 2.10. *Mapping Sources/Besoins selon une approche postérieure*

Bonifati et al. [Bonifati et al 01] proposent une méthode de conception orientée besoins, où le mapping Sources/Besoins se fait selon une approche postérieure. Un ensemble de schémas

multidimensionnels sont dérivés à partir des buts organisationnels (en adaptant une technique GQM). Un deuxième ensemble de schémas multidimensionnels est dérivé à partir de schémas E/A des BDDs sources à travers un algorithme qui identifie les faits comme les entités ayant des attributs numériques, et les dimensions comme les entités liées aux entités Fait par une relation (1,1) ou (1,n). Une dernière étape consiste à intégrer les deux ensembles de schémas multidimensionnels en un schéma unifié. Ce mapping est effectué manuellement par le concepteur.

b. Approche médiane

Dans cette approche, un schéma conceptuel est généré à partir des sources (resp. besoins) et validé par les besoins (resp. sources). Dans ces deux cas, un inconvénient majeur réside dans l'exploitation minime d'une des sources d'informations (sources de données ou besoins utilisateurs). Dans le premier cas (exp : travaux de *Phipps et al.*) [Phipps et al 02], un schéma conceptuel est défini à partir de l'analyse des sources. Une étape d'intégration doit être effectuée à ce niveau là. Ce schéma est ensuite validé par les besoins afin de définir le schéma conceptuel final. Cette validation est généralement effectuée manuellement par le concepteur en fin de conception.
Dans le deuxième cas (exp : travaux de *Giorgini et al.*) [Giorgini et al 05], un schéma conceptuel est défini à partir des besoins et est ensuite validé en analysant les sources. Cette validation nécessite une étape d'intégration lors du matching entre les concepts du premier modèle conceptuel (généré à partir des besoins) et les données se trouvant au niveau des sources.

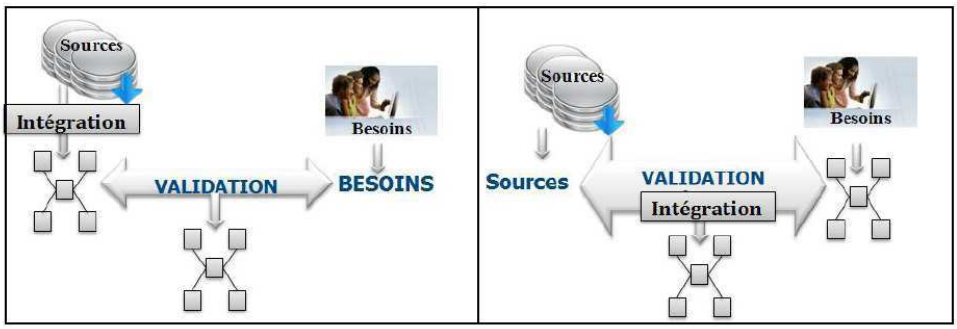


Figure 2.11. *Mapping Sources/Besoins selon une approche médiane*

Phipps et al. [Phipps et al 02] proposent de définir le schéma conceptuel de l'ED où le mapping sources/besoins se fait selon une approche médiane. Un premier modèle multidimensionnel est généré à partir des sources de données, où les entités ayant des attributs numériques représentent des faits, leurs attributs numériques représentent des mesures, et les autres attributs (attribut non numérique, non clé et ne représentant pas une date) représentent les dimensions de ce fait. Une dimension 'Temps' est associée à toutes les entités faits. Un ensemble de schémas multidimensionnels est donc généré. Ces schémas sont ensuite validés et raffinés par les besoins utilisateurs formalisés par des requêtes SQL. La pertinence des schémas est évaluée selon les règles suivantes :

- Si un schéma ne contient pas de tables se trouvant dans la clause From d'une requête, alors ce schéma est considéré comme non pertinent pour cette requête.
- Un schéma est considéré comme pertinent si les attributs numériques de son fait se trouvent dans la clause Select.

Giorgini et al. [Giorgini et al 05] proposent également de définir le schéma conceptuel de l'ED où le mapping sources/besoins se fait selon une approche médiane. Un schéma conceptuel multidimensionnel de l'entrepôt est défini à partir de l'analyse des besoins (qui s'effectue selon une approche orientée but). Les concepts multidimensionnels du schéma conceptuel sont ensuite mappés avec les entités des schémas relationnels des sources de données afin de les valider et de déterminer les hiérarchies de dimensions. Ce mapping est effectué manuellement par le concepteur en associant chaque fait, dimension ou mesure du schéma conceptuel à une table ou à un attribut dans les sources. Le schéma est ensuite présenté aux utilisateurs pour une deuxième validation.

c. Approche antérieure

Dans cette approche, un mapping entre les sources et les besoins est effectué en début de conception avant la génération d'un schéma conceptuel intermédiaire. [Annoni et al 06] recommande ce type de mapping. Confronter les besoins aux sources de données nécessite une étape d'intégration, dans laquelle il faudra identifier au niveau des sources, les concepts correspondants aux concepts utilisés pour exprimer les besoins.

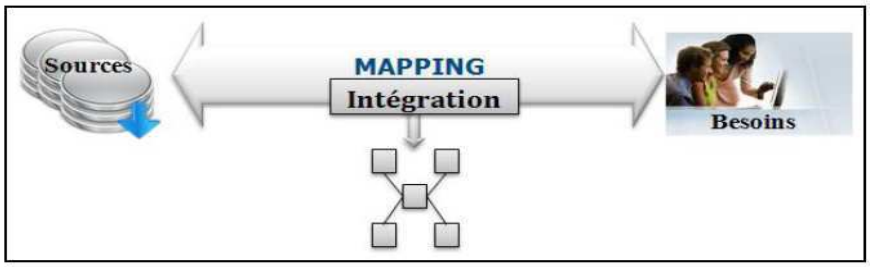


Figure 2.12. *Mapping Sources/Besoins selon une approche antérieure*

Romero et al. [Romero et al 06] proposent de générer un schéma conceptuel de l'ED selon cette approche. Les besoins utilisateurs sont exprimés sous forme de requêtes Sql sur des schémas de données relationnels. Un ensemble de schémas multidimensionnels modélisés sous forme de graphes sont générés en analysant les classes et attributs des requêtes Sql.

Selon [Khouri09], la confrontation entre les sources et les besoins utilisateurs par une approche antérieure est préférable aux autres approches. Les deux sources d'informations (sources et besoins) sont autant exploitées l'une que l'autre, de plus cette confrontation présente un gain en temps important en évitant de générer des schémas intermédiaires possiblement inutiles.

On remarque bien que pour ces approches traditionnelles, le problème d'intégration est toujours présent à différents stages de la conception quelque soit la méthode de conception (en début de conception pour les méthodes basées sur les sources ; et en début de conception ou pendant le mapping sources/besoins pour les méthodes impliquant les besoins). Ce problème d'intégration des schémas des sources peut cacher d'autres difficultés de conception comme l'effort d'un travail de rétro-conception (dû au fait que les bases de données sources soient stockées selon leur modèle logique).

Pour résoudre ce problème d'intégration inhérent à la conception d'un ED, un nouveau type d'approche a été proposé : il s'agit des approches *ontologiques*.

4.3 Les approches ontologiques

Les approches ontologiques sont de bonnes candidates pour la résolution du problème d'intégration rencontré dans les approches précédentes. A notre connaissance, peu de travaux ont été réalisés dans ce contexte [Romero et al 07], [Khouri 09], [Nabli et al, 09]. Dans la suite, nous décrivons les deux premiers travaux qui se distinguent par la prise en compte des besoins utilisateurs ou non, lors de la phase d'analyse.

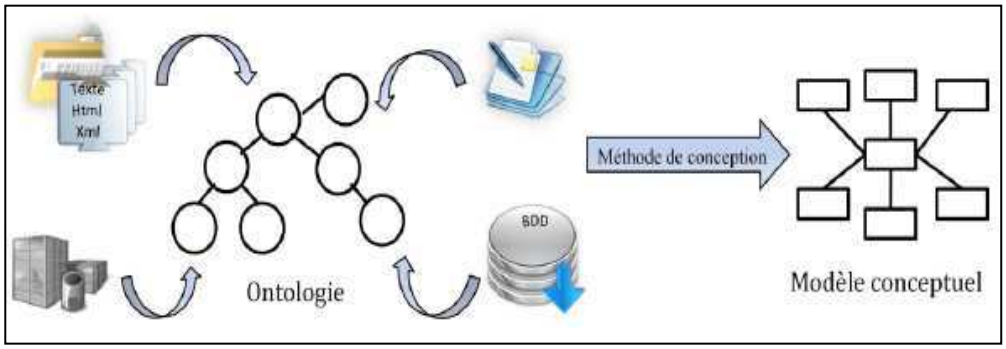


Figure 2.13. *Approche ontologique*

Un premier travail rentrant dans le cadre des approches ontologique à été réalisé par *Romeo et al* [Romero et al 07] qui présentent une méthode semi automatique de conception multidimensionnelle d'un ED, basée sur une ontologie représentant le domaine d'intérêt de l'ED. Cette méthode étudie les multiplicités entre les concepts de l'ontologie pour définir les concepts multidimensionnels '*faits*' et '*dimensions*' du schéma conceptuel. Ainsi, un concept de l'ontologie est considéré comme un fait potentiel s'il est relié à autant de dimensions et mesures que possible. Les mesures sont désignées comme les attributs numériques permettant l'agrégation des données. Ils sont reliés au concept représentant le fait par une relation (1,1). Un concept de l'ontologie est considéré comme dimension potentielle s'il est rattaché à un fait par une relation (1,n). Les hiérarchies de dimensions sont déterminées en recherchant les relations (n,1) entre les concepts identifiés comme dimensions. Cette identification peut se faire de manière automatique. Les résultats sont ensuite présentés au concepteur qui doit les valider.

Un deuxième travail qui traite les approches ontologiques est celui de [Khouri09]. En effet, les auteurs ont proposé une approche s'appuyant sur les approches précédentes mais en écartant certaines limites rencontrées dans les approches antérieures. Ces limites se résument dans les points suivants :

- ***Difficulté de conception*** : cette difficulté de conception s'exprime dans toutes les approches de conception. Dans les approches directes, un effort d'abstraction important doit être effectué par le concepteur durant le cycle de développement de l'entrepôt. Dans les approches traditionnelles, pour les méthodes orientées sources, une analyse détaillée et exhaustive des sources ne peut être possible que dans le cas où les sources sont détaillées, présentées dans des schémas de données suffisamment clairs et compréhensibles, spécialement si la méthode de génération du schéma est manuelle. On retrouve ce même problème dans les méthodes orientées besoins, en plus de la difficulté d'effectuer des

correspondances entre les sources et les besoins. Ce mapping Source/Besoins est généralement fait manuellement par le concepteur. Certains travaux suggèrent l'utilisation d'un thésaurus afin de guider le concepteur mais sans détailler davantage. Certaines méthodes présentent de plus des directives ou des étapes difficilement applicables comme par exemple 'un fait est une entité enregistrant des détails concernant des évènements particuliers' [Moody et al 00].

- ***Automatisation de la démarche de construction****:* même si on estime qu'une méthode de conception ne doit pas être complètement automatique (une méthode de conception n'a pas pour but de remplacer un concepteur mais d'interagir avec ce dernier afin de profiter de son expertise et de son savoir faire), apporter une part d'automatisation peut s'avérer indispensable. Dans toute méthode de conception, l'analyse des sources de données passe nécessairement par une étape d'intégration sémantique de ces sources. L'automatisation de l'intégration des sources devient nécessaire lorsqu'on a affaire à un grand nombre de sources de données. une autre conséquence de cette automatisation est de faciliter l'évolution de l'ED par l'ajout de nouvelles sources de données.

- ***Difficulté d'utilisation ultérieure de l'entrepôt de données généré*** *:* le modèle conceptuel d'un ED doit permettre d'exprimer à la fois les besoins applicatifs et la connaissance du domaine sous une forme intelligible pour un utilisateur ultérieur. Malheureusement, c'est le modèle logique (issu de diverses règles de transformations comme la normalisation des tables de dimensions) qui est exploité, et qui est en général très différent du modèle conceptuel. Une part importante de la sémantique du modèle conceptuel est perdue lors de sa traduction en un modèle logique. Ainsi, l'absence de représentation du modèle conceptuel dans l'ED rend l'interrogation de celui-ci très difficile même pour les utilisateurs ayant une bonne connaissance du domaine d'application.

- ***L'intégration des données*** *:* la problématique la plus importante dans toutes les méthodes citées reste l'intégration des données. De toutes les caractéristiques d'un ED, l'intégration est l'aspect le plus important [Inmon 92]. Un ED, par définition, intègre des données hétérogènes provenant de sources réparties. L'intégration de ces données pose une problématique importante due aux conflits syntaxiques et sémantiques survenant entre les données. on a vu tout au long de ce chapitre que cette problématique est présente dans toutes les approches et méthodes de conception existantes. Ces méthodes de conception, pour la plupart, ne peuvent être appliquées que sur un schéma intégré des sources de données. Dans toutes ces méthodes, cette forte hypothèse d'existence d'un schéma intégré est soit complètement ignorée (approches directes, et approches traditionnelles), soit non étudiée (approche ontologique).

Dans [Khouri09], les auteurs ont proposé une nouvelle méthode de conception d'un ED à base ontologique qui permet de prendre en compte les sources de données et les besoins des utilisateurs, tous deux représentés au niveau ontologique ; afin de fournir un modèle conceptuel représenté au niveau sémantique.

Les principales étapes constituant l'approche proposée sont :

- Définition d'une ontologie intégrant les sources de données au niveau sémantique.

- Génération automatique d'une ontologie « locale » à partir de l'ontologie intégrante représentant les besoins décisionnels.
- Définition du schéma conceptuel multidimensionnel de l'entrepôt de données à partir de l'ontologie locale.
- Validation du schéma conceptuel par un ensemble de recommandations proposées et manuellement par le concepteur.

La figure 2.14 montre un schéma de l'architecture de la méthode proposée.

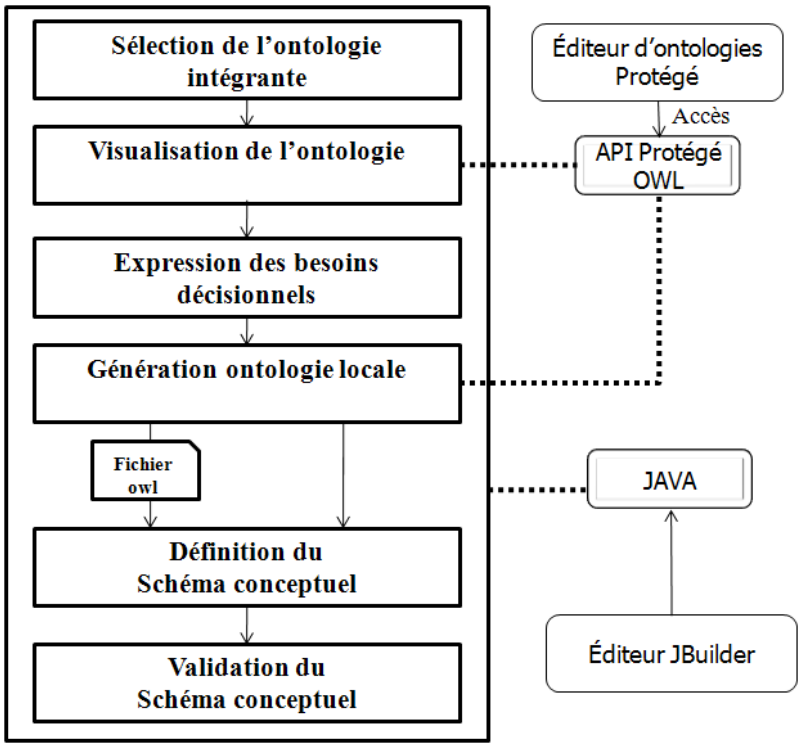


Figure 2.14. *Architecture d'implémentation de la méthode proposée* [Khouri 09]

L'apport de cette méthode par rapport aux limites citées ci-dessus se récapitule dans les points suivants [Khouri09]:

- ***Simplification de la tâche de conception :*** en considérant l'ontologie intégrante comme base et référence de conception, le concepteur gagne en temps et en efforts pour maîtriser le domaine de l'entrepôt à développer. L'ontologie facilite aussi l'intégration des données au niveau sémantique et de manière automatique, mais également le chargement des données au niveau de l'entrepôt lors du processus ETL.

- ***Interaction et autonomie de conception :*** l'objectif de cette méthode de conception est d'offrir un cadre de conception permettant de guider le concepteur tout au long de sa tâche en interagissant avec lui, en profitant de son expertise et en automatisant les étapes fastidieuses qui peuvent être automatisées, tout en offrant au concepteur une réelle autonomie dans cette tâche de conception.

- ***Sémantique du modèle conceptuel :*** un modèle conceptuel a pour but d'expliciter les besoins applicatifs et la connaissance du domaine une forme intelligible pour un utilisateur ultérieur. Le passage du niveau conceptuel au niveau logique cause nécessairement une perte de la sémantique du modèle conceptuel dûe aux différentes règles de transformation (normalisation, …). La présence d'une dimension sémantique au niveau du modèle conceptuel permet d'assurer une représentation de ce modèle au niveau de l'entrepôt final lors des passages aux niveaux logiques et physiques ; ce qui permet de faciliter interrogation de l'entrepôt par les utilisateurs finaux.

5 Conclusion

Nous avons présenté dans ce chapitre les différentes approches de conception d'un ED. L'étude de l'ensemble des approches a permis de les classer en trois catégories : *directes*, *traditionnelles* et *ontologiques*. Une problématique essentielle qui revient souvent concerne l'intégration des données due aux différents conflits syntaxiques et sémantiques. Les approches ontologiques sont apparues pour remédier à ce problème. Nous nous sommes basés sur deux travaux réalisés dans le cadre des approches ontologiques : le premier [Romero et al 07] est basé sur les sources de données et le second [Khouri 09] prend en compte, en plus des sources de données, les besoins des utilisateurs.

L'approche que nous proposons dans notre travail s'appuie essentiellement sur le travail réalisé dans [Khouri09] tout en prenant en compte les différentes caractéristiques des données utilisées dans notre cas à savoir les données multimédias. Les notions et les spécificités liées à ce type de données constitueront l'objet du prochain chapitre.

Chapitre 3

Modélisation des données multimédia

1. Introduction

De nos jours, la majorité des données circulant au niveau des entreprises ou utilisées par les individus sont de types complexes, elles sont qualifiées de données multimédia. Le multimédia permet de combiner des données de différents types (texte, image, audio, vidéo) à l'intérieur d'un même document numérique.

Ces dernières années ont vu un regain d'attention au niveau de la communauté informatique, chercheurs et praticiens, pour le traitement de types d'information tels que l'image et le son, qui traditionnellement étaient laissés à part. Ce traitement est fait par les systèmes d'information multimédia. Mais pour être satisfaisants du point de vue de l'utilisateur, les systèmes d'information multimédia doivent réussir à relever certains défis liés à la manipulation de ces nouveaux types de données.

En effet, pour être utilisées au niveau des entreprises et par les systèmes d'information, les données multimédia doivent être présentées sous une forme exploitable, d'où la nécessité de disposer de formats et de descripteurs dédiés à la représentation de données multimédia.

Dans ce chapitre, nous commençons par définir la représentation des données multimédia d'une manière générale (i.e. sur un dispositif de stockage), on parle de la numérisation des données. Par la suite nous passons à la modélisation des données multimédia dans les BD où nous mettons l'accent sur les SGBD multimédia. Nous présentons à titre d'exemple, la prise en compte des données multimédias par deux SGBD: Oracle inter-média et MySQL.

2. Les données multimédias

2.1 Le Multimédia

Le multimédia est l'intégration au sein d'un même système d'information de données textuelles, d'images fixes ou animées [web3]. Son histoire est très récente, depuis 1990 environ, et va de pair avec la tendance prononcée à la numérisation de l'information, la production de petits systèmes informatiques à possibilités très performantes et au développement de l'interconnexion de réseaux, notamment dans ce dernier cas, Internet et le World Wide Web.

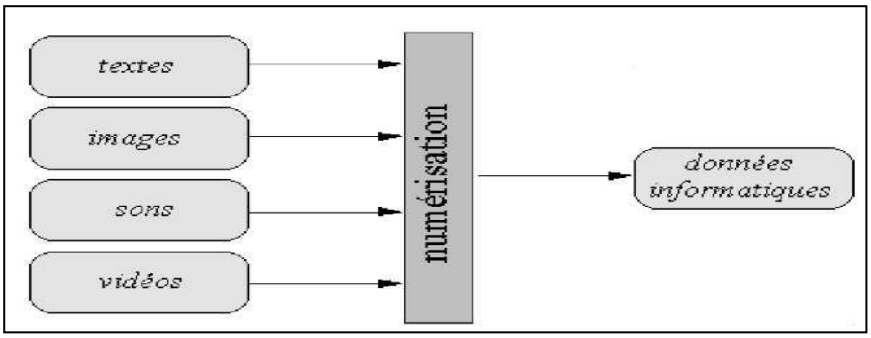


Figure 3.1. *La numérisation des données multimédias*

Selon les points de vue, le multimédia recouvre un marché en pleine expansion, un environnement de travail et de loisir, des applications professionnelles ou grand public, une industrie portant sur la production de matériels et logiciels. Une définition possible est : le multimédia est l'exploitation simultanée de données sonores, visuelles, informatiques, l'ensemble des techniques de création, de stockage, de transmission, de restitution, et aussi l'ensemble des données permettant cette exploitation simultanée [Terrasson, 92].

2.2 Utilité du multimédia

Un premier intérêt du multimédia est l'intégration généralisée des données et des outils de manipulation de ces données [web 3]. Avant le multimédia, des appareillages divers étaient nécessaires : un ordinateur pour les données informatiques, un magnétophone pour les cassettes de sons, un magnétoscope pour les vidéos, un téléphone pour le transport de la voix, du papier pour les images, etc. L'utilisation simultanée de données de natures différentes nécessitait donc une juxtaposition inconfortable de ces outils. L'utilisation de l'ordinateur comme outil unique est un progrès sensible. Par ailleurs, les données multimédias sont manipulées comme des données informatiques puisque codées sous forme de 0 et de 1 et peuvent donc, en particulier, être stockées de manière unique et transmises de manière unique ce qui constitue un second intérêt.

Cette intégration se fait cependant avec quelques problèmes :

- le volume de données numérisées est généralement important, ce qui nécessite des techniques de compression/décompression.
- Les traitements sont plus complexes et nécessitent des matériels et logiciels de plus en plus performants

- la transmission de données volumineuses nécessite des débits binaires sur les réseaux qui commencent seulement à apparaître.

Néanmoins cette tendance semble irréversible et entraîne une reconfiguration du paysage autrefois audio-visuel en paysage totalement numérique.

2.3 Types de données multimédias

La plupart des informations sont distribuées sous forme de textes, d'images fixes, de sons, de vidéo. Actuellement, il est possible de numériser ces médias (il n'est cependant pas encore possible de numériser les odeurs, les goûts et les sensations de toucher) [web 3].

2.3.1 Les textes

Ils sont composés de caractères dits d'imprimerie, la numérisation s'opère simplement par codage de chaque caractère en une suite de 0 et de 1. Le code ASCII (American Standard Code for Information Interchange) sur 7 bits permet de coder 128 caractères usuels. Comme les ordinateurs travaillent usuellement sur des mots qui sont des multiples de mots de 8 bits (octets), on peut rajouter un 0 devant le code ASCII, ce qui correspond au code normalisé. Ainsi le caractère "A" est codé 01000010. On peut aussi rajouter un 1 et définir un second jeu de 128 caractères (code ASCII étendu) ; malheureusement ce second jeu n'est pas normalisé et varie d'une plate forme informatique à l'autre.

Il est également possible de coder la mise en forme du texte : gras, italique, souligné, la taille des caractères, l'alignement, etc. Là encore, il n'y a pas de normalisation et les logiciels manipulant du texte possèdent encore leur propre système. Pour un texte comportant 2000 caractères (espaces compris), sans style, la taille du fichier numérisé sera donc de 2000 octets ce qui correspond à un volume faible.

2.3.2 Les images

Pour les applications multimédias, les images sont converties en matrices de points. Chaque point est codé suivant sa "couleur" et les codes résultants sont placés séquentiellement dans un fichier, ligne par ligne, colonne par colonne (figure 3.2).

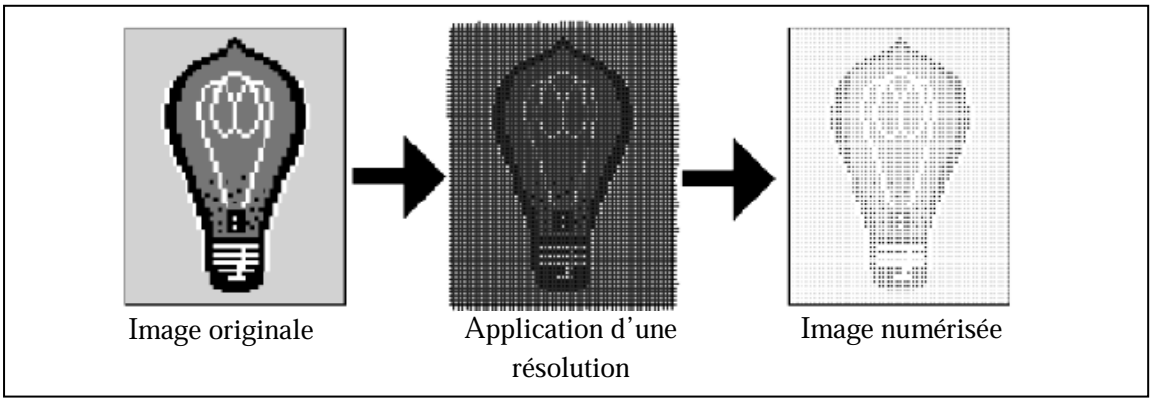


Figure 3.2. *Conversion de l'image*

Pour une image binaire, c'est à dire "bicolore", noir et blanc, chaque point est codé sur 2 bits (0 pour noir, 1 pour blanc. Pour une image avec 256 nuances de gris allant du noir total

au blanc total, chaque point sera codé sur 8 bits (de 00000000 pour noir à 11111111 pour blanc). Pour une image couleur, on exprime cette couleur comme une superposition d'une dose de Rouge, d'une dose de Vert et d'une dose de Bleu (système RVB) ; si on permet une variation de 0 à 255 pour une dose, une couleur quelconque sera codée comme la juxtaposition de trois octets, soit 24 bits (et donc 16 777 216 couleurs possibles).

Considérons une image pour laquelle la grille de codage se compose de 640*480 points. Si l'image est binaire, sa taille est de 38 400 octets, si l'image est à 256 nuances de gris, sa taille est de 307 200 octets ; si l'image possède des couleurs codées sur 24 bits, sa taille est de 921 600 octets. Par ailleurs, des techniques de compression permettent de gagner de la place mais au prix d'une décompression au moment de l'affichage des images ; ce que l'on gagne en volume est perdu en vitesse d'affichage.

La taille du fichier obtenu après numérisation et avant compression est fortement influencée par le pas de la grille de codage qui est appelé résolution et exprimé usuellement en points par pouce (dpi : dots per inch). Pour une même image une résolution de 600 dpi donnera un fichier image 36 fois plus volumineux que le fichier image résultant d'une numérisation à 100 dpi.

Les techniques de compression conduisent à des formats d'images, soit normalisés, soit imposés par les industriels. Deux formats d'image sont couramment employés pour les applications multimédias en ligne : le format GIF qui correspond à des images compressées sans perte d'information avec 256 nuances de couleurs et le format JPEG qui correspond à des images compressées avec perte d'information avec 16 millions de couleurs. Ces deux formats ont l'immense avantage d'être interprétés par la plupart des navigateurs du Web.

2.3.3 Les sons

Les technologies d'acquisition du son ont été longtemps analogiques : le son était représenté par les variations d'une grandeur physique, une tension électrique par exemple. Les techniques actuelles permettent d'obtenir directement un son numérisé : c'est notamment le cas des magnétophones produisant un enregistrement sur cassettes D.A.T. (Digital Audio Tapes, digitalisation du son à une fréquence de 48 KHz).

Pour numériser un son enregistré de manière analogique, on procède en trois étapes (figure 3.3):

- Echantillonnage : l'amplitude du signal analogique est mesurée à une fréquence d'échantillonnage f. On obtient ainsi une collection de mesures.

- Quantification : une échelle arbitraire allant de 0 à 2n-1 est employée pour convertir les mesures précédentes. Une approximation est faite de manière à ce que chaque mesure coïncide avec une graduation de l'échelle (cette approximation, qui modifie légèrement le signal, est appelée bruit de quantification).

- Codage : suivant sa grandeur dans cette nouvelle échelle, chaque mesure est codée sur n bits et placée séquentiellement dans un fichier binaire.

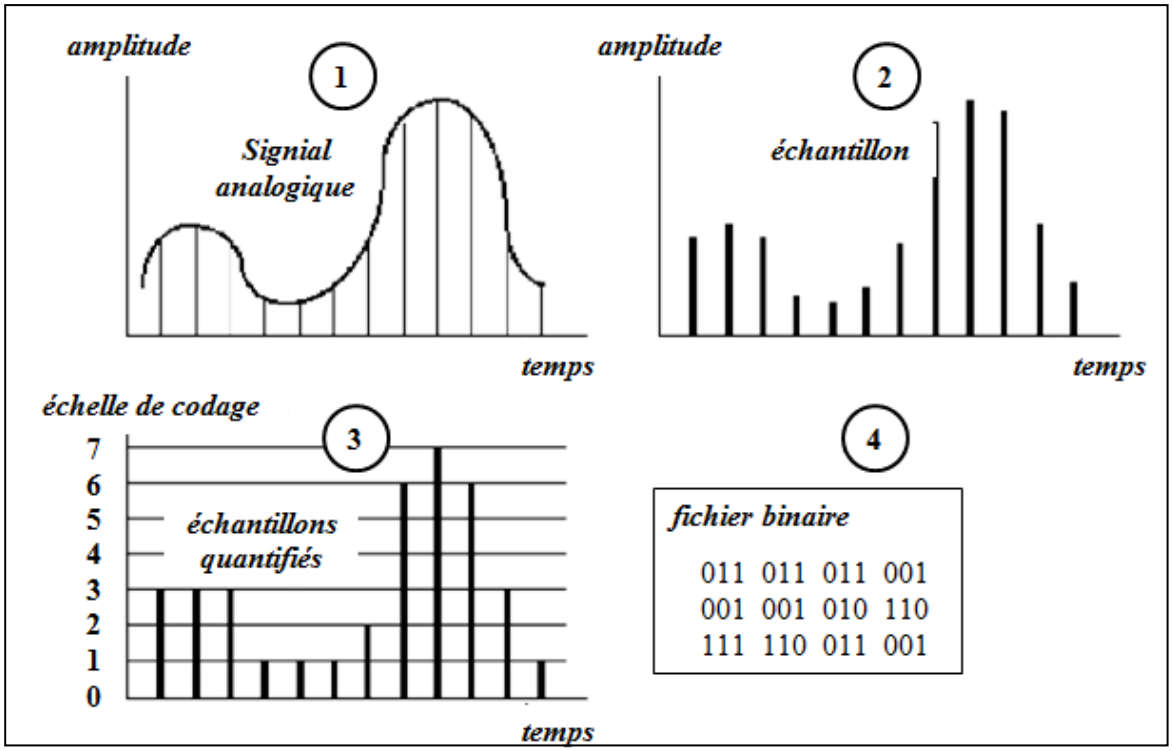


Figure 3.3. *La numérisation du son*

On comprendra que les valeurs de f et de n sont critiques pour la taille du fichier résultant. Usuellement 3 qualités de numérisation sont employées :

- la qualité Hifi ou CD audio : f=44 KHz, n=16 bits, stéreo (2 signaux sonores)
- la qualité "radio" : f= 22 KHz, n=8 bits, mono ou stéréo
- la qualité "téléphonique" : f=11 KHz, n=8 bits, mono

Ainsi, 1 heure de son correspondra à des fichiers de 630 Mo, 158 Mo et 40 Mo pour les qualités CD, radio et téléphonique respectivement. Nous verrons plus loin que les standards de CD ROM correspondent à une capacité d'environ 640 Mo. Les normes de codage numérique du son portent sur l'enregistrement non compressé des mesures quantifiées (PCM : Pulse Code Modulation), sur l'enregistrement des différences entre deux mesures successives, ce qui permet une réduction de volume de données (DPCM : Delta PCM), ou encore sur l'utilisation de techniques de prédiction des mesures (ADPCM : Adaptative Differential PCM).

2.3.4 La vidéo

Comme pour le son, la vidéo s'exprime par des signaux de nature analogique qu'il faut numériser pour une utilisation multimédia. Il existe ici quatre signaux à numériser : trois pour l'image et un pour le son. Les signaux d'image se répartissent en signal de luminance Y et deux signaux de chrominance Db et Dr, reliés aux composantes R,V,B (Rouge, Vert, Bleu) par les relations linéaires suivantes:

$$Y = 0.30R + 0.59V + 0.11B$$
$$Db = B - Y$$
$$Dr = R - Y.$$

La composante Y est échantillonnée à une fréquence de 13,5 MHz, tandis que les composantes de chrominance sont échantillonnées à une fréquence de 6,75 MHz. La quantification pour les trois signaux d'image s'opère sur 8 bits.
Les données peuvent être simultanément ou ultérieurement compressées. Plusieurs systèmes sont actuellement utilisés. La compression Indéo d'Intel est utilisée dans le logiciel Vidéo For Windows (le résultat de la numérisation et de la compression conduit à des fichiers vidéo AVI).

Pour des plates-formes Macintosh, mais aussi Windows, le logiciel Quick Time d'Apple joue un rôle analogue à partir de fichiers vidéo MOOV ; un grand intérêt de Quick Time est sa possibilité d'intégration de données initiales diverses (images, sons, textes) et de proposer plusieurs types de compression. Les logiciels Video For Windows et Quick Time réalisent également la synchronisation du son avec les images.

La norme MPEG, spécialement élaborée pour la compression de la vidéo, tend actuellement à s'imposer. Le principe de compression s'appuie sur trois types d'images : les images "intra" sont des images peu compressées qui servent de repère (une image intra pour 10 images successives) ; les images "prédites" sont des images obtenues par codage et compression des différences avec les images intra ou prédites précédentes (une image prédite toutes les trois images) ; les images "interpolées" sont calculées comme images intermédiaires entre les précédentes.

L'utilisation de vidéos numériques MPEG nécessite la présence d'une carte de décompression dans le micro-ordinateur d'exploitation.

2.4 Spécificité des supports

Les supports des données ou des applications multimédia doivent satisfaire deux contraintes essentielles [web 3]:

- posséder des capacités importantes de stockage
- posséder des taux d'entrée/sortie suffisants pour l'exploitation en temps réel.

La figure 3.4 montre quelques exemples sur l'espace de stockage nécessaire pour différents type de données multimédias.

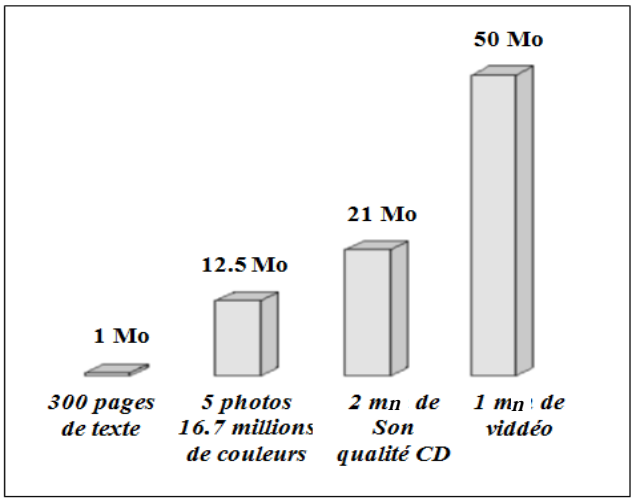


Figure 3.4. *Exemples sur l'espace de stockage*

Les disques durs magnétiques pourraient être de bons candidats pour l'hébergement d'une application multimédia de taille raisonnable ; en particulier, leur débit d'entrée/sortie est satisfaisant ; mais leurs coûts et leurs technologies qui les rendent indissociables de l'ordinateur (peu adapté au stockage) les éliminent, quant il s'agit de supports de diffusion commerciale, au profit de la famille des CD-ROM qui, née avec le multimédia, est actuellement le support quasi-universel des applications et des données multimédias distribués off line.

Par contre, pour des applications mettant en jeu des bases de données, accessibles en local ou à distance, l'utilisation des disques durs est encore la meilleure solution, sauf, peut-être, pour des données dont les modifications sont rares ; dans ce dernier cas, l'utilisation de dispositifs de type juke-box de CD-ROM peut être une solution.

2.5 Spécificité des outils et méthodes de développement

Le développement d'une application multimédia implique des outils spécifiques adaptés au type des données à traiter ou à leur intégration.

Les outils qui sont liés au type de donnée sont [web 3]:

- pour l'image : outils d'acquisition, de dessin, d'édition et de retouche ;
- pour le son : outils de numérisation, d'édition et de retouche ;
- pour la vidéo : outils de numérisation, d'édition et de retouche, d'animation ;
- pour le texte : outils d'édition, de reconnaissance optique de caractères.

Suivant les cas, ces outils sont des composants matériels et/ou logiciels. En effet, les outils d'intégration sont logiciels et correspondent soit à la structuration des données, soit à leur manipulation et en particulier à la réalisation de l'interface utilisateur. Les outils de structuration des données sont principalement des systèmes de gestion de bases de données dotés d'extensions permettant la gestion de données multimédias. Les outils de manipulation peuvent être des langages orientés multimédias (ils réalisent alors directement l'intégration de données multimédias de types différents) ou des langages de programmation usuels (c'est au programmeur de réaliser tous les traitements).

2.6 Spécificité des outils de traitement

Le multimédia nécessite, pour son exploitation, des ressources supplémentaires par rapport à celles des ordinateurs classiques. La station multimédia est un ordinateur, le plus souvent micro-ordinateur, doté:

- de composants matériels performants: processeurs rapides (200 MHz), mémoire vive de grande capacité (32 Mo), écran 640x480 au moins ;
- d'extensions spécifiques : carte son, baffles ou casque, microphone, carte vidéo, carte réseau, modem, lecteur de CD-ROM,… ;
- de modules logiciels particuliers pour l'exploitation du son, de la vidéo, de l'image, de la communication (notamment navigateur www).

3. Problématique de la modélisation des données multimédia

Les systèmes d'information multimédia traitent non seulement des données alphanumériques conventionnelles, mais aussi du son (voix ou autres), des graphiques et des images (fixes ou animées). Ainsi, l'utilisateur peut désormais manipuler non seulement les informations du type textuel traditionnel, mais aussi les informations de type graphique : image, son ou une combinaison quelconque de ces formes de données. En plus de ses sens de vue et de toucher, il peut donc utiliser son sens d'ouïe et sa voix pour communiquer avec son système, d'où un niveau d'interaction plus élevé conduisant potentiellement à un niveau de satisfaction et d'utilité plus élevé. Mais avant de réaliser pleinement ces bénéfices, les systèmes d'information multimédia doivent relever plusieurs défis [Berra et al 93], [Little et al 93], [Little et al 91]. Parmi ces défis on peut citer: l'acquisition, la compression et le stockage des données multimédia, l'accès aux données stockées, et leur présentation.

3.1 Acquisition

Cette étape consiste à saisir les données à partir de diverses sources. Les moyens et techniques employés pour faire cette acquisition dépendent naturellement du type d'objet à saisir et du médium utilisé. Contrairement aux systèmes traditionnels, pour les systèmes multimédia, on peut imaginer différents types de dispositifs d'entrée (camera, vidéo, lecteur optique ...), selon le media.

3.2 Compression et stockage

Le stockage de certains types de données multimédia nécessite d'énormes quantités d'espace. En effet, les données de type audio, les graphiques, les images et plus particulièrement les données de type vidéo sont très volumineux, d'où l'importance de la compression. Cette compression est requise notamment avant le stockage ou le transfert des données multimédia. Plusieurs méthodes et algorithmes de compression de données ont été proposés [Fox 91].
Même après compression, le volume des données multimédia reste appréciable. En outre, les supports de stockage et les matériels de traitement de ces données sont très hétérogènes.
Le stockage de tels objets soulève ainsi de nouveaux défis concernant la gestion, la distribution et le transfert des données multimédia, et le choix des modèles conceptuels et logiques de données.

3.3 Accès

Les données multimédia peuvent être consultées de diverses manières dont les plus courantes sont: la recherche par identificateur où l'on accède aux objets multimédia par un attribut spécifique prédéfini comme étant l'identificateur de l'objet; la recherche par instructions conditionnelles qui se fait sur la base d'expressions booléennes sur des attributs des objets stockés; la recherche par similarité qui consiste à rechercher un objet en regard de critères de similarité préalablement définis; et la recherche par contenu ou recherche sémantique qui se fait en tenant compte du sens que l'on accorde aux objets. Le choix d'un mode de consultation dépend du contexte de l'application et du type de données consultées.

3.4 Composition et présentation

Les données de type audio et vidéo contiennent, outre leur composante spatiale, une composante temporelle. La composition spatiale est la représentation à un moment donné dans le temps de l'assemblage des objets dans l'espace. La composition temporelle consiste à faire l'évaluation de la relation entre les différents éléments et à faire le chargement et la présentation de ces éléments en conséquence. Cette composante temporelle constitue une différence fondamentale entre systèmes d'information multimédia et les autres types de systèmes [Berra et al 93] et il est primordial de bien l'intégrer à la composante spatiale. I1 convient donc de bien modéliser ces deux aspects des données multimédia de façon à faciliter la manipulation tant au niveau conceptuel qu'au niveau physique. De la qualité des modèles dépendra, au moins en partie, la performance du système, en particulier au niveau de la synchronisation des objets. Cette synchronisation des objets au moment de la présentation est un point critique des systèmes d'information multimédia.

4. XML et les données multimédias

Le but de cette section n'est pas de donner tous les détails sur XML, mais c'est plutôt de montrer comment utiliser XML pour la description des données multimédias. Cependant, avant de montrer son utilisation avec ce type de données, nous présentons brièvement XML à travers sa définition, ses avantages, sa manière d'utilisation et son application.

4.1 Qu'est ce que XML et pourquoi l'utiliser?

XML *(eXtensible Markup Language)* est un format universel de documents textuels et graphique fondé sur des balises comme HTML, mais il décrit le contenu plutôt que la présentation du contenu. Les balises peuvent avoir aussi des attributs, et cela fait penser aux langages de programmation à objets. En fait XML peut être utilisé pour "sérialiser" (stocker) des classes de langages de programmation [web 5].

XML tend à se généraliser comme format de document, c'est dorénavant le format de fichiers de Office et Open Office.org. Comme son nom l'indique, XML est extensible et il sert de base à de nombreux langages descriptifs comme on le verra plus loin. La première recommandation par le W3C, XML 1.0 date du 16 août 2006 mais son histoire remonte à 1996 comme l'indique l'historique du W3C.

4.2 Avantages de XML

Tout type de données peut être décrit par XML pourvu que l'on fournisse une grammaire de la structure (donc des balises). Sa structure arborescente et extensible permet de décrire n'importe quoi. Il est ainsi facile à *parser* par les logiciels tout en étant lisible par les humains. Son universalité permet de l'utiliser dans tout contexte, sur tout système. Puisque les balises décrivent les données brutes, il est facile d'y faire des recherches.

4.3 Comment utiliser XML

Pour utiliser un document XML, on a besoin d'un parseur qui reconnait la syntaxe et charge le document en mémoire. Il existe deux sortes de parseurs [web 5] :

- Dans le premier cas, le document est transformé en une arborescence en mémoire, dont le contenu est accessible selon les méthodes du DOM (Document Object Model).
- Il est également possible d'associer des fonctions à des balises avec les parseurs événementiels de type "sax".

XML, comme HTML a ses feuilles de style, nommées XSL (XML StyleSheet), qui fournissent des règles pour transformer un document XML dans un autre format XML ou non XML (XHTML par exemple).

Des outils ont été standardisés (par le W3C) pour accéder au contenu des documents XML [web 5]:

- ***XLink*** *(XML Linking Language) :* Un langage (avec la syntaxe XML), qui peut insérer des ressources pour décrire des hyperliens unidirectionnels (comme ceux de HTML) ou des liens plus complexes. Xlink peut être utilisé avec XPointer.
- ***XPath*** *:* indique comment atteindre des éléments spécifiques.
- ***XPointer*** *(XML Pointer Language) :* Un langage qui peut addresser des éléments sélectionnés dans la structure des documents. Il est basé sur XPath.
- ***DOM*** *(Document Object Model) :* Un objet utilisable notamment sur les navigateur permet d'accéder dynamiquement et changer soit le contenu, soit la structure de documents XML. Il propose une interface indépendante des plate-formes et s'applique à HTML autant qu'à XML.

4.4 Applications de XML

La puissance de XML va au-delà du simple stockage de données, de nombreuses utilisations émergent [web 5]:

- **SVG** définit des éléments graphiques pour produire des images redimensionnables, et peut remplacer le format Flash.
- **XUL** est un ensemble de balises prédéfinies pour construire des interfaces utilisateur graphiques, il est utilisé par le navigateur Firefox.
- **XAML** est un format similaire pour la plateforme .NET et il s'emploie dans *Silverlight*, le framework d'applications Web.
- **XHTML** est une version de HTML compatible avec XML.
- **HTML 5** est une révision majeure de HTML qui étend ce dernier pour les applications web. Le travail a été lancé par le W3C (World Wide Web Consortium) en mars 2007 visant la clôture des ajouts de fonctionnalités le 22 mai 2011 et une finalisation de la spécification en 2014.

4.5 Description des données multimédias avec XML

Dans ce qui suit, nous montrons comment utiliser XML pour décrire une donnée multimédia qui peut être soit une image, soit une vidéo ou encore du son.
Il existe deux manières de décrire une donnée multimédia avec XML :

- La première concerne la description du contenu : c'est-à-dire qu'on décrit tous les détails liés à la donnée. Dans le cas d'une image par exemple, on découpe cette dernière en zones et on décrit chaque zone à l'aide d'un fichier XML.
- La deuxième manière concerne la description de la donnée à travers ses informations : il s'agit des *métadonnées*.

Il est clair que la deuxième manière est plus simple par rapport à la première qui nécessite de décrire le moindre détail. Pour illustrer ces deux façons de description, nous présentons quelques travaux qui cernent ce domaine.

4.5.1 XML et le format SVG

La distribution d'images sur Internet passe actuellement principalement par des formats bitmaps tels que GIF, JPEG ou encore PNG [Jolif06]. Certains sites Web font aussi appel à des formats vectoriels, propriétaires, comme le format SWF de Macromedia ou VML de Microsoft. Ces formats, malgré leur diffusion importante, possèdent certaines limitations. Afin d'améliorer la diffusion d'images sur Internet, le World Wide Web Consortium (W3C) a mis en place en 1998 un groupe de travail chargé de définir un nouveau format d'images vectorielles pour pallier les limitations des formats existants et servir de « standard » sur le Web. Ce groupe de travail a donné naissance au format SVG *(scalable vector graphic)* et à une spécification le définissant. Celle-ci a abouti en 2001 à une recommandation définitive du W3C qui incita les acteurs d'Internet à l'adopter.

4.5.1.1 Objectifs de SVG

À la lumière des limitations des formats existants, on peut déterminer les principaux objectifs généraux de SVG [Jolif06] :

- réduire la taille de la majorité des images visualisées sur le Web ;
- distinguer dans une image les différents objets qui la composent, et ainsi rendre possible l'interaction et l'animation sur ces objets ;
- permettre un affichage optimal de l'image et un comportement correct lors des changements d'échelle ;
- être un « standard » ouvert supporté par un maximum d'applications et de navigateurs Internet, tout en obtenant des résultats dans la mesure du possible identiques et performants sur l'ensemble des plates-formes ;
- permettre une édition facile et rapide des images ;
- bien s'intégrer dans les contextes d'internationalisation (langues orientales...) et permettre l'accès aux documents SVG dans des contextes d'accessibilité réduite (personnes malvoyantes...).

4.5.1.2 Définition du format

Les objectifs précédemment décrits ont conduit à faire un certain nombre de choix sur la définition du format. Ce dernier a été choisi de telle façon à prendre en compte les points suivants [Jolif 06] :

- **Des images alliant *vectoriel* et *bitmap* :** SVG sert principalement à décrire des images de façon vectorielle. Cependant, il existe des images difficilement définissables ainsi. C'est pourquoi le format – devant être générique – prévoit également la possibilité d'intégrer des images bitmaps dans un document.
- **Un format basé sur XML :** Dans l'optique de permettre une édition facile ainsi qu'une grande interopérabilité, le format SVG est basé sur XML. Ainsi, le fichier SVG pourra être interprété dans de nombreux contextes.
- **Un modèle de document associé à SVG :** Les afficheurs SVG, étant destinés à permettre l'interaction avec les différents éléments d'une image SVG, doivent implémenter un modèle de document associé à chaque image SVG.
- **Compatibilité avec d'autres standards :** Afin de ne pas redéfinir des fonctionnalités déjà étudiées par ailleurs et de permettre aux concepteurs des images une meilleure réutilisation de ces dernières, SVG peut ou doit s'appuyer sur des standards existants pour effectuer certaines tâches *(XPointer, XLink, CSS, RDF, et SMIL).* De plus, les caractère des textes SVG doivent être compatibles avec Unicode.
- **Comparaison avec les autres formats :** SVG est un format possédant de nombreux avantages par rapport aux formats existants.

Quelques détails sur la syntaxe d'un document SVG ainsi que des exemples illustratifs de description d'images sont donnés en annexe A.

4.5.2 Utilisation de XML avec Flash pour représenter des mouvements

Une interpolation de mouvement peut être décrite à l'aide de code XML et des classes *ActionScript* dans le package fl.motion [web 1]. Adobe Flash Professional dispose d'une commande *« Copier le mouvement au format ActionScript »,* qui génère du code XML et ActionScript basé sur une interpolation de mouvement du scénario et qu'on peut utiliser dans d'autres symboles ou d'autres projets. On peut également adapter notre propre interpolation de mouvement. Tant qu'on a les classes fl.motion dans notre chemin de classe au moment de compiler, l'interpolation de mouvement sera appliquée à notre objet d'affichage spécifié.

La hiérarchie des éléments XML utilisés pour la description d'une interpolation de mouvement est la suivante :

```
<Motion>
 <Source>
      <dimensions/>
          <geom:Rectangle />
      </dimensions>
      <transformationPoint>
          <geom:Point />
      </transformationPoint>
  </Source>
  <Keyframe>
    <color>
        <Color />
    </color>
    <tweens>
        <SimpleEase />
        <CustomEase>
            <BezierControl />
            <BezierNode />
        </CustomEase>
    </tweens>
    <filters>
      <filters />
    </filters>
  </Keyframe>
</Motion>
```

Hiérarchie des éléments pour la description d'une interpolation

Chaque élément peut contenir ses propres attributs permettant de donner un sens à l'interpolation de mouvement en question. La liste des éléments et des attributs qui peuvent être affectés à un objet de mouvement est donnée dans un grand tableau dans [web 1]. Un exemple d'interpolation décrite avec Flash est donné en annexe A

4.5.3 Le standard SMIL

Le standard SMIL (Synchronised Multimedia Integration Language) est un langage de spécification de présentations *multimédia* basé sur XML. Tel qu'il a été défini par le W3C, c'est un langage de description de documents multimédias synchronisés. L'aspect temporel est donc un point crucial pour de telles présentations. Ce langage permet aussi d'exprimer d'autres informations: des informations spatiales, des informations hypermédias, et des méta-informations [Yang, 2000].

Dans le cadre de l'hypermédia, l'apport majeur du langage SMIL est l'introduction de la notion de lien temporel i.e. un lien dépendant d'un intervalle de validité temporelle. Par conséquent, en plus des deux dimensions temporelle et hypermédia, un autre aspect découle des liens temporels: les relations hyper-temporelles. Un modèle temporel pour le langage SMIL devrait permettre de représenter, en plus des informations de synchronisation du scénario temporel, les liens temporels associés aux éléments de la présentation.

Une autre particularité de SMIL vient du fait qu'un élément peut avoir trois états possibles: l'état *inactif*, l'état *actif*, et l'état *suspendu* ou pause. L'introduction de ce dernier état s'avère indispensable à la modélisation de certains comportements induits par les liens temporels dans le langage SMIL.

L'élément "media object" (<img>, <video>, <audio> and <text>... etc) permet l'intégration d'objets média au sein d'une présentation SMIL par référence à leurs URI (Uniform Resource Identifiers).

Des attributs de synchronization *begin, dur* et *end* peuvent être associés aux éléments de synchronisation

- L' attribut *begin* spécifie le début explicite d'un élément
- L' attribut *end* spécifie la fin explicite d'un élément
- L' attribut *dur* spécifie la durée explicite d'un élément

Les conteneurs de temps (time containers) *<seq>* and *<par>* basés sur le modèle temporel de SMIL constituent les éléments temporels les plus usuels :
<seq> décrit la séquence des éléments d'une présentation (l'un après l'autre)
<par> décrit une présentation de plusieurs éléments simultanément.
Trois sémantiques sont définies pour la terminaison d'un élément <par> élément, selon la valeur de l'attribut "*endsync*":

- *Last*: l'élément <par> se termine lorsque tous ses élément fils se terminent,
- *First*: l'élément <par> se termine dès que l'un de ses éléments fils se termine,
- *Master*: l'élément <par> se termine dès que l'élément maître (défini par l'attribut ID) se termine,

```
<seq>
     <img src="Titre.jpg" dur="4s"/>
        <par>
            <audio src="Discours.au" dur="14s"/>
                   <seq>
                   <video src="Auteur.avi" dur="6s"/>
                   <video src="Demo.avi" dur="10s"/>
                   </seq>
       </par>
</seq>
```

Spécification d'un document SMIL

5. La norme MPEG-7

De plus en plus d'informations audiovisuelles sont accessibles sous forme numérique, en tout endroit à travers le monde et de plus en plus de gens souhaitent les exploiter [Danjean 05]. Mais avant que quiconque ne puisse utiliser de telles informations, il sera nécessaire de commencer par les localiser. Au même moment, l'augmentation du nombre d'informations potentiellement intéressantes rend la recherche de plus en plus difficile.

Des solutions sont proposées pour faciliter la recherche d'information de type texte. Il n'est cependant pas pour autant possible de chercher des informations sur un contenu audiovisuel, puisqu'il n'existe en général pas de descriptions reconnaissables de ce type d'information. Dans certains cas particuliers, des solutions existent cependant. Des bases de données multimédias permettent aujourd'hui de chercher sur le marché des images à partir de certaines caractéristiques comme la couleur, la texture ou la forme d'objet dans l'image.

MPEG a commencé à mettre au point un nouvel outil de travail pour répondre au problème décrit précédemment [Danjean 05]. Ce nouveau membre de la famille MPEG, appelé "Multimédia Content Description Interface"(MPEG-7) étendra les capacités de recherche

limitées d'aujourd'hui pour inclure d'autres types d'informations. En d'autres termes, MPEG-7 va spécifier une description standard de différents types d'informations multimédia. Cette description devra être associée au contenu lui-même pour permettre la recherche rapide et efficace des informations qui intéressent l'utilisateur. Ces informations incluent; images, graphiques, audio, vidéo et de l'information sur comment ces éléments sont combinés dans une présentation multimédia (scénario).

5.1 Principe de description MPEG-7

MPEG-7 définit un ensemble de descripteurs de base qui peuvent être utilisés pour caractériser les contenus multimédias [Tola04]. Certains sont très proches du niveau physique (échantillonnage du signal, histogramme, …) mais d'autres offrent plutôt une sémantique élevée (titre, mots-clefs, commentaires, …).

MPEG-7 spécifie également des schémas de description (DTD) qui consistent en des structures prédéfinies regroupant un ensemble de descripteurs ainsi que leurs relations.

Une description MPEG-7 est fortement basée sur la syntaxe du langage XML et est de plus, complètement indépendante de la nature du contenu multimédia qui garde sa nature initiale : analogique, numérique…

Le format MPEG-7 est complètement indépendant de la technique de codage ou de stockage du contenu du document multimédia. On peut établir une description MPEG-7 d'un fichier MPEG-2 ou MPEG-4 bien sûr, mais on peut faire de même avec un film analogique ou un journal papier. Il s'agit uniquement d'un standard de représentation du contenu des documents. L'utilisation principale de MPEG-7 concernera évidemment les documents multimédia (contenant à la fois vidéo et audio). Les informations qui apparaîtront dans un document MPEG-7 seront de 5 natures différentes, résumées dans le tableau suivant :

Ensemble des éléments	Fonctionnalité
Création et production	Des méta-informations qui décrivent la création et la production du contenu, elles décrivent le titre, le créateur, le but de la création.
Utilisation	Des méta-informations reliées à l'utilisation du contenu: elles comportent les droits d'accès, des informations financières, des droits de publication. Ces informations peuvent faire l'objet de changement durant la durée de vie du contenu audio-visuel.
Média	Ces informations décrivent les caractéristiques de stockage: Format, éléments pour identifier le média.
Aspects structurels	Des descriptions d'un point de vue contenu: Ces informations décrivent les segments qui peuvent représenter des composantes spatiales, temporelles ou spatio-temporelles du contenu audio-visuel. Chaque segment peut être décrit par les caractéristiques suivantes (la couleur, la texture, la forme, la motion, d'autres caractéristiques audio...) et quelques informations sémantiques élémentaires.
Aspect Conceptuels	Des descriptions du contenu audio-visuel d'un point de vue conceptuel. Ces informations ne sont pas indiquées dans les documents techniques du standard MPEG, car elles sont en cours de standardisation.

Tableau 3.1. *Différentes natures des informations apparaissant dans un document MPEG-7* [Danjean 05]

Des exemples illustrant la description d'images et de séquence vidéo au format MPEG-7 sont fournis en annexe A.

6. Modélisation des données dans les BD multimédia

6.1 Caractéristiques d'un SGBD multimédia

Une base de données multimédia contient des informations conservées sous différents formats multimédias : fichiers sons, fichiers photo, séquence vidéo, données textes, etc. Le volume constitue une des principales caractéristiques de ces données qui peuvent atteindre quelques giga-octets.
A la lumière des caractéristiques que nous venons de citer, un SGBD multimédia est un SGBD permettant de gérer efficacement les différents types de données multimédia ; il doit pouvoir :

- Stocker des objets de gros volumes de données.
- Offrir des outils de modélisation (modèle, langage de définition de données) pour prendre en compte la nature spécifique des objets et les opérations associées.
- Mettre à la disposition des usagers des langages et des interfaces spécifiques pour la manipulation et l'interrogation de ces données.
- Optimiser les accès aux données en offrant des outils d'indexation spécialisés.

6.2 Architecture fonctionnelle d'un SGBD multimédia

La figure 3.5 présente une architecture fonctionnelle en couches pour un SGBD Multimédia [Djema et al, 07]. Au plus bas niveau figure un serveur d'objets qui prend en compte les aspects structurels et le volume des données. Généralement un SGBD classique peut servir à réaliser cette couche mais il doit être étendu si l'on veut gérer les différents aspects des données multimédias qui ont été décrits plus haut. Cette extension constitue une couche logicielle qui se place au dessus du serveur d'objets et qui est spécialement conçue pour traiter les aspects spatiaux et temporels, par exemple. Ainsi elle permet la composition et la présentation d'objets qui sont gérés de manière naturelle au niveau de l'interface.

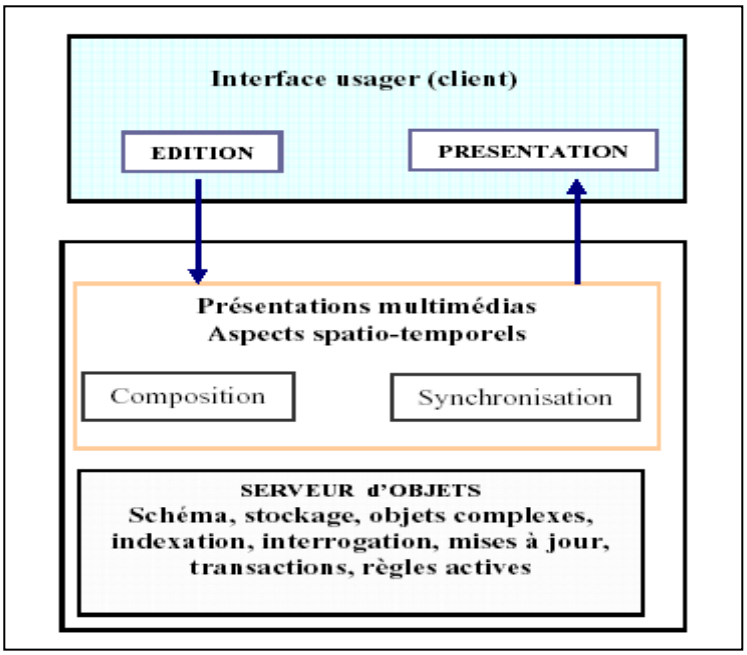


Figure 3.5. *Architecture fonctionnelle d'un SGBD Multimédia [Djema et al, 07]*

6.3 Gestion de données multimédia avec Oracle 8

Les données multimédia Oracle sont gérées par une des extensions de la base de données Oracle*8i* qui permet de faire de la recherche de documents au travers des interrogations plus ou moins complexes, selon le genre de documents. Oracle *inter*Media est capable aussi de gérer des documents textes, des sons, des images et des vidéos. L'ensemble des documents peuvent être stockés et diffusés, les images et les textes disposant de plus de fonctionnalités, via leurs index [Meylan 01].

Pour permettre au système de gestion de base de données de traiter les documents efficacement, ceux-ci doivent subir une indexation documentaire. Cette indexation permet de dégager l'information nécessaire pour leur repérage des listes, tables ou index.
L'indexation des documents textes génère une structure complexe de données, alors que celle des image consiste en la génération d'une signature binaire C'est la méthode *analyze()* qui s'occupe de générer la signature d'image, composée d' informations relatives aux couleurs, à la structure et à la texture de l'image. La comparaison d'image se fera alors uniquement à partir de cette signature.

Nous présentons dans ce qui suit quelques fonctionnalités offertes par Oracle concernant le stockage et le chargement de données multimédia.

6.3.1 Stockage des données

Les différents documents peuvent être stockés soit dans la base de données ou seulement référencés par celle-ci. On parle du stockage interne ou du stockage externe respectivement. Le choix du type de stockage dépend de l'application [Meylan 01].

6.3.1.1 Stockage interne des données

Les documents sont stockés dans des LOBs. Un LOB (Large OBject) est un type de données qui permet de stocker dans un champ un gros volume d'information allant jusqu'à 4 giga-octets. Il existe trois types de LOB qui sont utilisés selon la nature des données, il s'agit de :

- **BLOB (Binary LOB) :** Objet contenant des données binaires non structurées ;
- **CLOB (Caracter LOB) :** Objet contenant des données de types caractère avec de simples octets ;
- **NCLOB (National CLOB) :** Objet contenant des caractères de taille fixe d'un jeu de caractères international.

Les documents peuvent également être stockés des classes plus complexes (package ORDSYS) qui font appel à un LOB.

6.3.1.2 Stockage externe des données

Le document n'est alors que référencé par la base de données. Il est accessible par le SGBD. La signature et l'index du document sont toujours à l'intérieur de la base. Le document peut être alors un fichier BFILE (Binary File : Fichier binaire sur le système d'exploitation) ou un URL.

6.3.2 Chargement des données

Différentes méthodes et utilitaires existent pour réaliser un chargement de fichiers dans la base de données, qu'ils soient internes ou externes à celle-ci.

a. Ctxload

Il s'agit d'un exécutable et offre la possibilité d'effectuer les opérations suivantes :

- Chargement de texte dans des champs LONG ou LONG RAW.
- Mise à jour et exportation de documents, spécifiquement les colonnes de type LONG, LONG RAW, BLOB et CLOB.
- Importation et exportation de thésaurus contenus dans un fichier plat ASCII.

b. SQL*LOADER

Utilitaire historique de chargement. Il permet de charger les données selon les caractéristiques d'un fichier de contrôle en produisant des rapports d'activité (fichier log). Il permet de charger des documents dans tous les types de colonnes (LONG, LONG RAW, BLOB, BFILE, etc.). Les fichiers de contrôle servent à charger les fichiers dans la base ou de les référencer.

c. DBMS_LOB.LOADFROMFILE()

Cette procédure permet de charge un fichier externe dans la base de données. Elle est utile si l'on doit automatiser le chargement depuis la base de données via du code PL/SQL.

Les figures 3.11 et 3.12 représentent deux exemples qui montrent l'utilisation de certains types pour le chargement de données multimédia.

```
LOAD DATA
INFILE *
INTO TABLE T1
FIELDS TERMINATED BY ','
(id SEQUENCE(MAX,1),
descr CHAR(30),
nomFichier FILLER CHAR(30),
colonne_image BFILE(CONSTANT "c:\img", nomFichier))
BEGINDATA
Mer Méditerranée,image1.jpg,
Coucher de soleil,image2.jpg,
```

***Listing 3.1**- Exemple d'utilisation du type BFILE*

```
LOAD DATA
INFILE c:\fichier_controle.txt
INTO TABLE T2
APPEND
FIELDS TERMINATED BY ','
(id SEQUENCE(MAX, 1),
date_publication SYSDATE,
titre,
nomFichier FILLER CHAR(80),
texte LOBFILE(nomFichier) TERMINATED BY EOF)
fichier_controle.txt
Mer Méditerranée,image1.jpg,
Coucher de soleil,image2.jpg,
```

***Listing 3.2.** Exemple d'utilisation du type CLOB*

6.3.3 Gestion des images avec Oracle

La gestion des images stockées dans des champs de type ORDImage est simple (on compare les images sur la base de leurs attributs) et par conséquent limitée. Mais si l'on étend l'image aux possibilités fournies par VIR, cela devient plus intéressant.
La principale différence entre les images ORDImage et ORDVir est que ce dernier permet l'indexation des images au travers de leur signature et permet aussi des recherches basées sur le contenu des images (couleur, structure et texture) **[Meylan01]**.

6.3.3.1 Comparaison d'images

Pour utiliser le produit VIR, Il faut générer la signature (méthode analyze()) des images, basée sur les couleurs locales et globales, la structure et la texture de l'image. Ces critères sont en fait des valeurs que l'on attribue aux opérateurs VIRScore() et VIRSimilar() pour effectuer une comparaison. Ces valeurs vont de 0.0 (valeur par défaut) à 1.0 ou de 0 à 100 selon votre choix.

Pour mieux illustrer la manière de comparaison, nous prenons quelques exemples cités dans [Meylan 01]. Le premier exemple concerne la comparaison des couleurs. La figure 3.6 représente deux images portant les mêmes couleurs.

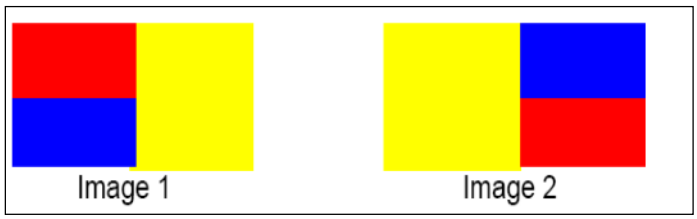


Figure 3.6. *Exemple de comparaison de deux images selon la couleur*

Les deux images ont la même taille et chaque couleur occupe le même pourcentage de l'image, d'une image à l'autre, à savoir rouge 25%, bleu 25% et jaune 50%.
Si les deux images sont comparées uniquement sur la couleur globale, on obtiendra une parfaite similarité entre les deux images (score = 0.0) parce que chaque couleur occupe le même pourcentage sur une image comme sur l'autre. Par contre, si on compare les images uniquement sur la couleur locale, il n'y a aucune similarité entre elles (score = 100) parce qu'aucune des couleurs de la première image ne se chevauche avec celles se trouvant sur l'autre image.

Le deuxième exemple permet de faire une comparaison des images selon la structure. La comparaison s'effectue sur les deux images de la figure 3.7

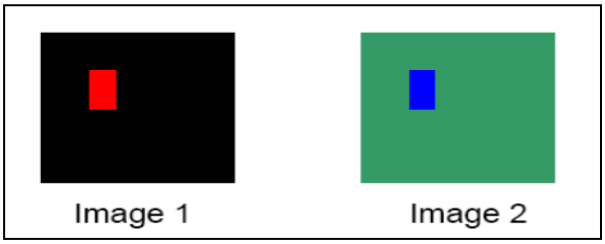
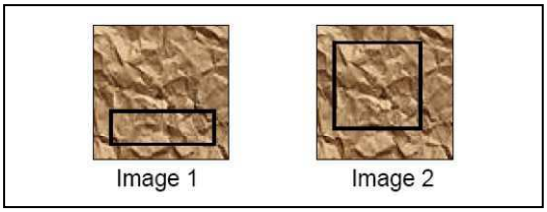


Figure 3.7. *Exemple de comparaison de deux images selon la structure*

Si on compare ces deux images entre elles uniquement sur la structure, les deux images sont identiques (score = 0.0), en considérant qu'il n'y a que les couleurs qui changent, la taille du petit rectangle étant la même. Par contre, on obtient un score de 100 si on les comparait sur la couleur (globale ou locale).

La structure est utile pour les requêtes sur les formes simples (cercles, polygones ou lignes diagonales) particulièrement quand les images sont dessinées en ne tenant pas compte des couleurs. Et également pour comparer des objets comme les lignes d'horizon dans les paysages, les structures rectangulaires dans les bâtiments et les structures organiques comme les arbres.

Le dernier exemple porte sur la comparaison de la texture. Cette comparaison est faite sur les deux images de la figure 3.8.



F***igure 3.8.*** *Exemple de comparaison de deux images selon la texture*

Si on compare ces deux images entre elles sur la texture uniquement, on obtiendra une parfaite similarité. Si les objets ont la même taille (ou position), le score diminuera.

La texture est utile pour comparer des images entièrement remplies de la même texture, comme les graines, le marbre, le sable ou les pierres. Ces images sont en général assez difficiles à catégoriser de part le manque de vocabulaire sur les textures, ce qui rend les recherches difficiles à formuler.

Pour comparer des images, on se base sur deux valeurs principales: le poids et le score

- **Le poids :** c'est une valeur qui représente à combien devra correspondre le degré de similarité d'un critère de recherche. Sa valeur doit être comprise entre 0.0 et 1.0. Par exemple, si on désire que la couleur globale soit totalement ignorée lors de la comparaison, on y assignera un poids de 0.0. Dans ce cas, chaque similarité ou différence entre les couleurs n'aura aucun impact dans la correspondance. Ou alors si la couleur globale a une grande importance pour la recherche, il faudra y affecter un poids plus important que pour les autres critères. Au moins un des quatre attributs visuel (couleur local, couleur globale, structure, texture) devra avoir un poids plus grand que zéro.

- **Le score :** La mesure de similarité pour chaque attribut visuel est calculée comme le *score* ou la *distance* entre deux images en prenant en compte leurs attributs. La valeur du score est comprise entre 0 (pas de différence) et 100 (aucune similarité possible). Plus la similarité entre deux images (en tenant compte de leurs attributs) est grande et plus le score sera petit.

L'exemple suivant illustre comment se calcule la distance. La figure 3.9 contient un graphique qui montre trois images comparées sur deux critères, la couleur globale (axe X) et la structure (axe Y).

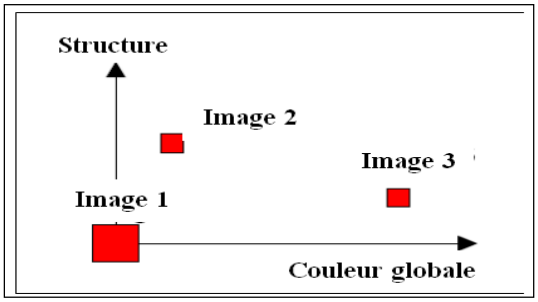


Figure 3.9. *Comparaison de trois images selon deux critères*

Pour la comparaison, l'image 1 est l'image de base et les deux autres sont les images à comparer avec l'image 1. Si l'on regarde la couleur globale, la distance entre l'image 1 et 2 n'est pas grande (disons 15). La distance entre l'image 1 et 3 est plus importante (disons 75). Si l'on compare sur la structure, l'image 3 aura un score de 25 et l'image 2 un score de 50 par exemple.

L'étape suivante consiste à choisir le critère auquel on désire attribuer le plus d'importance. Si on décide que ce critère soit 'la couleur globale' alors on doit lui affecter le poids le plus important, et ce sera l'image 2 qui sera la plus ressemblante. L'inverse se produit si l'on met plus de poids à la structure. Les résultats dépendent donc du poids de chaque attribut visuel. D'où l'importance de les attribuer correctement.

En dernière étape, lorsque deux images correspondent, le degré de similarité dépend d'un "poids additionné" qui comprend le poids et la distance des quatre attributs visuels de l'image que l'on compare. C'est ce poids qui détermine le degré de similarité entre deux images. Il est calculé selon la formule suivante:

```
poids_additionne = poids_couleur_globale * distance_couleur_globale +
                   poids_couleur_locale * distance_couleur_locale +
                   poids_texture * distance_texture +
                   poids_structure * distance_structure
```

Formule de calcul du poids additionné.

L'exemple suivant illustre la manière de calculer le poids additionnée qui correspond à la comparaison de l'image 2 à l'image 1. Le poids et la distance sont donnés dans les tableaux suivants :

Attribut visuel	Distance
Couleur globale	15
Couleur locale	90
Texture	5
Structure	50

Distances entre les deux images.

Attribut visuel	Distance
Couleur globale	0.1
Couleur locale	0.6
Texture	0.2
Structure	0.1

Poids pour les images comparées.

Figure 3.10 . *Comparaison entre deux images*

Dans cet exemple, les images sont le plus similaires sur la texture et ont le plus de différence selon la couleur locale.
La valeur du poids additionné est donc : 0.1*15+0.6*90+0.2*5+0.1*50 = 61.5.

Pour pouvoir interpréter la valeur obtenue du poids additionné, on la compare avec une valeur seuil (threshold). C'est une valeur comprise entre 0 et 100 qui sert de filtre. Si le poids additionné est plus petit ou égal au seuil, les images correspondent. Plus le seuil est petit et moins il y aura d'images qui correspondront et vice-versa.

6.3.3.2 Quelques opérateurs utilisés pour la comparaison d'images

Pour assurer une meilleure gestion de données multimédia, Oracle inter-média dispose d'un ensemble d'opérateurs facilitant la manipulation et le traitement de ces données. Nous citons dans cette section deux types d'opérateurs permettant la comparaison des images [Meylan 01].

a. Opérateur de similarité VIRSimilar

C'est un opérateur qui retourne si une image requête est similaire ou non à une image source, en se basant sur les quatre attributs visuels de l'image et le threshold. Cet opérateur fonctionne de la manière suivante : l'opérateur compare les signatures de deux images, calcule le poids de chaque attribut visuel, compare les poids additionné avec le threshold et retourne 1 si le poids additionné est plus petit ou égal au threshold (il existe une ressemblance). Sinon, l'opérateur retourne 0 (pas de ressemblance).

La syntaxe correspondante est la suivante :

```
VIRSimilar ( signature IN RAW,
                 querysignature IN RAW,
                 weightstring IN VARCHAR2,
                 threshold IN FLOAT
                 [referencetoScore IN NUMBER]) ;
```

- **signature** : signature de l'image (source) qui va être comparée.
- **querysignature** : signature de l'image à comparer
- **weightstring** : 'globalcolor="val" localcolor="val" texture="val" structure="val"' ou 'facial=1' pour indiquer que l'on travaille avec Viisage.
- **threshold** : seuil auquel le poids additionné de l'image doit correspondre
- **referencetoScore** : paramètre optionnel utilisé lorsqu'on utilise l'opérateur VIRScore dans la requête. La valeur de ce paramètre doit être la même que celle utilisée avec VIRScore.

b. Opérateur VIRScore

Avant d'utiliser cet opérateur, les signatures des images doivent avoir été générées (VIR.analyze()) et un index de type ORDVIRIDX sur les signatures doit exister. Cet opérateur est utile si une application veut faire une distinction plus fine qu'un simple "oui ou non" retourné par l'opérateur VIRSimilar().
VIRScore retourne un poids compris entre 0.0 (images identiques) et 100.0 (images totalement différentes).

6.4 Gestion de données multimédia avec MySQL

Le but de cette section est de montrer les principes de manipulation de fichiers multimédia dans une table MySQL. Nous allons voir principalement comment stocker ces fichiers dans une table MySQL.

- On va partir d'un fichier existant, avec comme exemple un fichier gif.
- On suppose le fichier déposé à un endroit accessible; ici il sera dans le dossier du script php.

6.4.1 Stockage d'un fichier dans une table mysql

Pour le stockage des données multimédia, on utilise les types de donnés vues dans la section précédente (les BLOBS (binary large objects)).

a. Création de la table

On va créer la table « fichiers » ou seront stockés les fichiers; il faut exécuter la commande SQL de création de la table suivante [Messin06] :

```
CREATE TABLE `fichiers` (
     `Id` int(11) NOT NULL auto_increment,
     `Titre` varchar(50) default NULL,
     `Donnees` longblob,
     `Type` varchar(50) default NULL,
     PRIMARY KEY (`Id`),
     UNIQUE KEY `id` (`Id`)
   ) ;
```

La table ainsi créée comporte un identifiant Id, un titre pour le fichier stocké (qui peut également être le nom du fichier), les données du fichier, ainsi que le type de données.
Le type *blob* des données indique que ces données sont stockées sans modification (liés à une table de caractères, suppression des blancs ou autre), contrairement à ce qui peut se passer pour des types du genre *varchar* ou *text*. Le type *longblob* indique aussi la taille maximum des données qui peuvent être stockées (tinyblob <256 octets, blob<65536, longblob <4294967296!).
Il faut noter que les données sont stockées non compressées, et demande donc l'espace disque adéquat. De plus, il faut tenir compte des capacités d'échanges entre le client et le serveur. Enfin, il faut prendre en considération les temps de transmission vers les utilisateurs. Pour de petites images, le type blob peut-être suffisant.

b. Stockage du fichier dans la table

On suppose la connexion avec la base de données établie, on peut manipuler les données.

- **Préparation des données**

Cette étape consiste à lire le fichier dans une variable.

```
$titre = 'Fichier test';
   $type = 'image/gif';
   if(!$r=@fopen(FICHIER,"r"))die("Erreur d'accès au fichier ".FICHIER."\n") ;
   $size=filesize(FICHIER);
   $donnees = addslashes(fread($r, $size));
```

Après avoir défini le titre et le type du fichier, on ouvre le fichier par l'instruction *fopen* qui retourne une ressource de gestion de fichier; on détermine ensuite la taille du fichier. Pour le *fopen*, on utilise aussi le @ pour éviter les messages d'erreurs éventuels.

- **Insertion dans la table**

Il ne reste plus qu'à former et exécuter la requête SQL d'insertion dans la table.

```
$req= "INSERT INTO ".TABLE." ( Id, Titre, Donnees, Type )".
" VALUES ('$id','$titre','$donnees','$type')";
mysql_query($req) or die("Pas moyen d'ajouter le fichier à la table !!!");
```

A ce stade, on a réussi à stocker un fichier dans une table de la base MySql.
On peut penser à récupérer l'identifiant pour une utilisation ultérieure, puisque la fonction d'auto-incrément de mysql génère automatiquement un nouvel Id à chaque insertion.

```
$id= mysql_insert_id();
print "Numéro d'identifiant dans la table: $id<br>\n";
```

7. Conclusion

Dans ce chapitre nous avons mis l'accent sur un type particulier de données circulant au niveau des entreprises ou même chez les individus : les données *multimédias*. Pour cela, nous avons introduit le multimédia à travers sa définition, son utilité et les différents types de données (texte, image, son et vidéo). Nous avons vu aussi les spécificités des données multimédias telles que les spécificités des supports ainsi que celles des outils de traitement et de développement. La problématique de la modélisation des données multimédias a été posée aussi. Nous avons introduit également le langage XML et nous avons montré son utilisation pour la description des données multimédia à travers un ensemble d'outils tels que SVG et MPEG. Nous avons enchainé par la modélisation des données multimédias dans les BDs où nous avons entamé le côté technique à travers la description de manipulation de données multimédia dans Oracle Inter-Média et MySQL.

Dans notre démarche de conception d'ED, nous prenons en compte des données multimédia, ainsi nous avons constaté qu'il est nécessaire de faire une synthèse sur les différents travaux liés aux ED et traitant des données multimédia. Le prochain chapitre résume un ensemble de travaux et contributions visant à développer et à améliorer ce domaine de recherche.

Chapitre 4

Les Entrepôts de données multimédia

1. Introduction

Avec l'apparition de l'Internet et le développement des différentes représentations et formats des différents documents, les données peuvent être *hétérogènes* et classées en plusieurs catégories [Hamdoun et al 07]: *structurées* (données relationnelles, données objet), *semi-structurées* (HTML, XML, graphes) ou même *non structurées* (texte, images, son). Dans un tel contexte, le besoin d'intégration se fait de plus en plus sentir. Cependant, pour répondre à ce besoin, le développement des applications d'intégration se voit contraint de composer avec la répartition des sources, *l'hétérogénéité* de leurs structures et la complexité des données. La figure 4.1 montre l'hétérogénéité des sources d'un entrepôt de données [Hamdoun et al 07].

Nous introduisons dans ce chapitre les entrepôts de données multimédia (EDM). Pour cela, nous commençons par présenter les différents concepts et notions liés à ce type d'entrepôts, puis nous citons quelques travaux réalisés dans ce contexte. Ces travaux visent à assurer un ensemble de tâches telles que la construction, la maintenance et la gestion des processus d'aide à la décision.

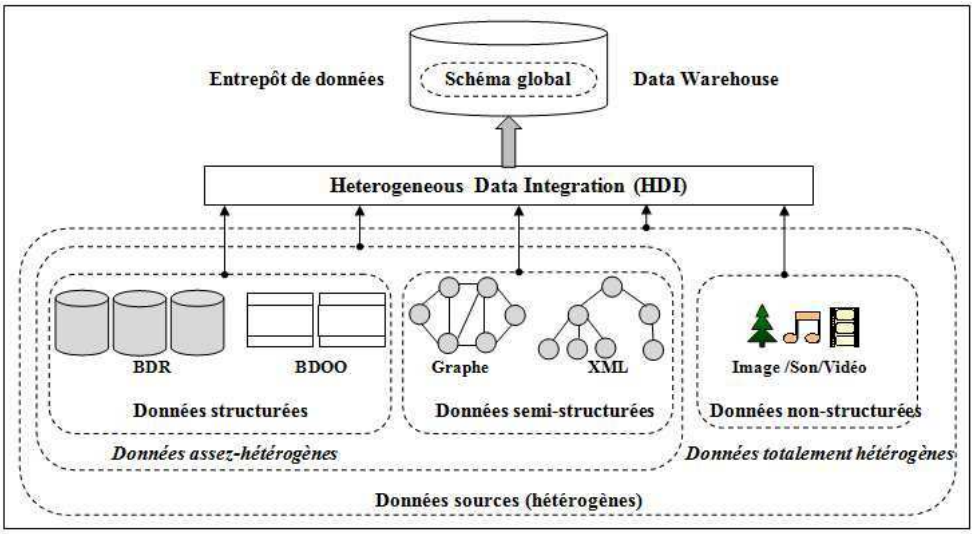


Figure 4.1. *Hétérogénéité des sources d'un entrepôt de données*

2. Qu'est ce qu'un entrepôt de données multimédia ?

Un entrepôt de données multimédia est un entrepôt qui s'alimente de données provenant de diverses sources au niveau desquelles les données sont sous forme de texte, image, son, ou même vidéo. Ces types de données sont appelées '*données multimédias*', certaines d'entre elles peuvent être stockées dans des bases de données traditionnelles (relationnelles, orientées objet), et par conséquent on peut les exploiter à l'aide d'outils d'un entrepôt classique. Mais, il existe d'autres données multimédia qui doivent être stockées dans des formats spécifiques, autres que des bases de données traditionnelles, (ex : MPEG-7) [Kim et al 03]. Ces dernières ne peuvent pas être exploitées avec les outils classiques. On parle alors des entrepôts de données multimédias.

3. Architecture d'un entrepôt de données multimédia

La figure 4.2 illustre l'architecture d'un entrepôt de données multimédia proposée dans une étude faite dans [Kim et al 03][Kim et al 04] qui consiste en la construction d'un ED dédié au Web. Au centre de la figure apparaît l'entrepôt de données multimédia, en bas de la figure on observe les sources de données multimédia hétérogènes, et en haut, on trouve les clients qui permettent aux utilisateurs d'interagir avec l'entrepôt de données multimédia via le Web.
Une description de ces composants sera donnée à la suite de cette figure.

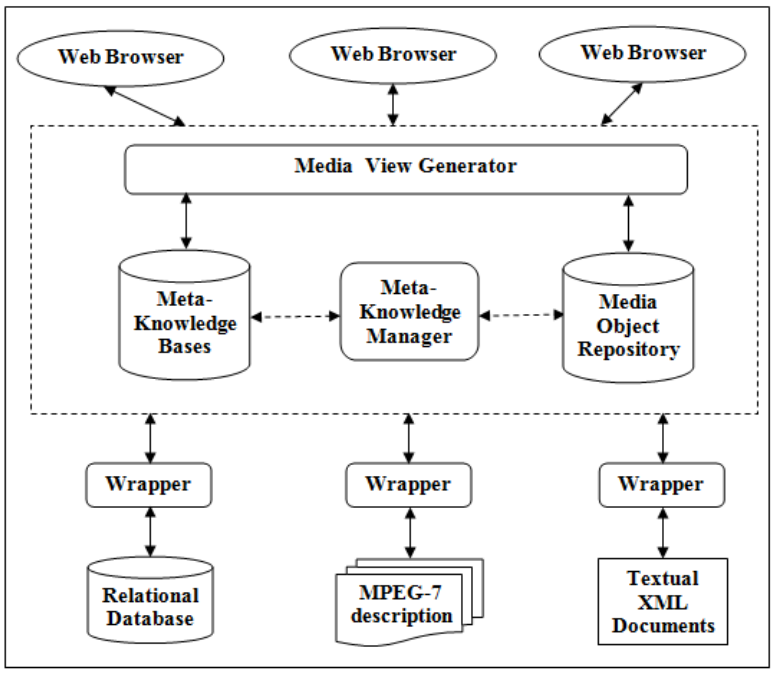


Figure 4.2. *Architecture de base d'un entrepôt de données multimédia*

L'EDM est constitué de quatre composants :

(1) Média object repository : c'est le lieu de stockage de différents types d'objets média.

(2) Média View generator : c'est un moteur de requêtes qui génère des *média-vues* qui peuvent être vues comme des vues matérialisées d'objets médias. Ce moteur génère deux types de média-vues :
- Les intra-média vues qui intègrent des objets médias homogènes.
- Les inter-média vues qui intègrent des objets médias hétérogènes.

(3) Meta-knowledge bases et (4) meta-knowlage manager: ces deux composants sont chargés de la gestion des changements des sources de données.

Les sources de données multimedia peuvent être soit stockées dans des bases de données relationnelles ou orientées-objets, soit dans des formats de descripteur MPEG-7, soit dans des textes de données semi-structurés (document XML).
Au dessus de chaque source de données, on trouve des *wrappers* qui se chargent de la transformation des données et de la communication avec l'entrepôt. Les *wrappers* transforment les différents types de données multimédias représentées par différentes structures en un modèle commun de l'entrepôt.

4. Domaines d'utilisation des entrepôts de données multimédias

Les entrepôts de données multimédias peuvent être utilisés dans presque tous les domaines qui génèrent et traitent des données multimédias. En effet, la numérisation de l'information multimédia a permis l'apparition de nouveaux équipements et de nouvelles applications (enseignement à distance, télé-médecine, surveillance électronique, etc.) [Hacid04]. De ce fait, les secteurs modernes produisent des quantités de plus en plus

volumineuses de données complexes électroniques qui contiennent, éventuellement, des données multimédias.
A titre d'exemple, nous présentons ici quelques domaines utilisant les entrepôts de données multimédias.

a- La médecine : dans le secteur médical, des données administratives conventionnelles, des données thérapeutiques et des diagnostics sont maintenant remplies avec des données complexes de type multimédia telles que les images de rayon X, l'échographie, l'électrocardiogramme, etc [Arigon et al 06]. Dans un domaine tel que la cardiologie, les données, qui sont employées pour des études cliniques, ne sont pas seulement alphanumériques, mais elles peuvent également être composées d'images ou de signaux. Ces données sont capturées par des dispositifs médicaux électroniques. Ce rapport de fait a soulevé l'idée d'extraire et de rassembler ces données afin de tirer des informations utiles. En effet, la recherche médicale nécessite de grands ensembles de données, provenant de diverses sources, récoltées dans le but d'analyser et d'extraire de l'information. Ces bases de données sont souvent organisées autour d'une pathologie ou d'une classe de pathologies, et sont employées pour valider des hypothèses de recherches et pour établir des connaissances au niveau des cliniques.

Avant d'être introduites dans la base de données, les données doivent être validées par des experts afin de garantir la qualité et la cohérence de l'unité. Dans ce cas, la constitution d'une telle base de données est très coûteuse en temps et demande un investissement humain. Il est seulement normal que les chercheurs au niveau des cliniques soient intéressés par l'entreposage des données et par le processus d'analyse en ligne (OLAP) afin de constituer et contrôler des données de haute qualité. Avec ces technologies, ils ont également la capacité de naviguer dans de grands ensembles de données selon leurs besoins (études épidémiologiques, suivi de population, évaluation des indicateurs,…).

b- Le Web : Des données non structurées (multimédias) telles que des emails, des textes HTML, des images, des vidéos et des documents de bureau sont de plus en plus accumulées au niveau des ordinateurs personnels. Ceci est dû d'une part, au traitement de textes et d'une autre part au Web. Ces données ont la particularité d'être volumineuses et précieuses au sein de l'entreprise. Nous avons donc besoin d'un entrepôt de données où les documents multimédia seront analysés et gérés pour partager et utiliser les informations à l'échelle de l'entreprise [Gong99]. Comme les données qui sont manipulées sont de type multimédia, alors l'entrepôt en question est évidemment un entrepôt multimédia.

Les entreprises qui utilisent le Web (souvent pour l'alimentation) doivent disposer d'un entrepôt de données multimédia.

Un entrepôt de données nécessite fréquemment des sources de données externes. Par exemple, une entreprise qui souhaite faire de la veille concurrentielle ne pourra se contenter d'intégrer dans son entrepôt uniquement des données provenant de ses bases de production. Une source de données primordiale dans un tel contexte est le web [Boussaid et al, 07]. Cependant, les données diffusées sur ce média sont très hétérogènes contenant

ainsi des données multimédias, et pour les utiliser il est inévitable de disposer d'un entrepôt de données multimédia.

Les entrepôts de données multimédias peuvent être liés au Web d'une autre façon. En effet, il existe un outil permettant de gérer la publication des pages multimédia sur le Web à l'aide d'un procédé nouveau de gestion du contenu inspiré de l'architecture multidimensionnelle pour les données (les entrepôts de données). L'outil en question est l'éditeur *InStranet* [web2]. Ce dernier, relié à une base de données spécifiquement consacrée au site, peut non seulement l'alimenter en fonction des changements intervenus dans la publication, mais aussi exécuter des requêtes suivant différents axes et présenter le contenu à l'écran. Ce contenu peut être du texte, des images, des sons, etc.

5. Problèmes liés aux entrepôts de données multimédias

La production croissante de données multimédias numérisées amplifie les problèmes classiques de gestion de données multimédias et en crée de nouveaux tels que l'accès par le contenu, la personnalisation des contenus, l'accès à partir d'appareils mobiles, etc.

Les problèmes les plus rencontrés sont recensés dans [Hacid04][Gong99][Arigon et al 05] et se résument en :

- Stockage de données volumineuses : il est connu que les données multimédias occupent beaucoup d'espace mémoire, ce qui rend aussi la gestion de l'entrepôt plus difficile.
- Fonctions d'agrégat à redéfinir : il faut définir des fonctions d'agrégat spécifiques, telles que des listes et des fusions de données multimédias ;
- Utilisation de descripteurs pour former les dimensions du modèle : en effet, les données sont décrites par différents types de descripteurs (descripteurs textuels, descripteurs sur le contenu, etc.) ;
- L'indexation physique des données multimédias ;
- L'intégration des données multimédias ;
- Le traitement des requêtes sur les données multimédias ;
- Etc.

6. Modélisation des données dans les EDM

La modélisation d'une donnée multimédia dans un entrepôt de données ressemble à celle d'une donnée de type simple, seulement, il faut effectuer des modifications concernant les tables de *faits* et les tables de *dimensions*.

Dans ce qui suit, nous commençons en premier lieu par présenter un état de l'art concernant des travaux liés à la modélisation multidimensionnelle multimédia. Par la suite, nous mettons l'accent sur les modifications apportées sur un cube de données classique dédié à la modélisation des données multimédias.

6.1 Travaux liés à la modélisation multidimensionnelle multimédia

Plusieurs travaux concernant la conception de cubes de données multimédias ont été entamés afin d'impliquer des analyses multidimensionnelles concernant de grandes bases de données multimédias.

Dans [You et al 01], les auteurs cherchent à étendre le concept des entrepôts de données traditionnelles ainsi que celui des bases de données multimédias de telle sorte à stocker et représenter les données multimédias.

Le système d'analyse *MultiMédiaMiner* utilise un cube de données multimédia en rendant possible le stockage des données multidimensionnelles à différentes granularités [Zaiane et al 98]. Dans le domaine médical, l'utilisation des traitements pour l'analyse et l'exploration de grandes quantités de données multimédia est une tâche critique. Par exemple, on peut citer une étude entamée dans le domaine de la détection du cancer du sein dans laquelle le but est le développement d'un entrepôt de données numériques pour la mammographie représentant ainsi un outil d'aide au diagnostic [Zhang et al 01].

Une autre étude [Tikekar et al 95] traite le problème de stockage et de restitution des données médicales *images* des entrepôts de données.

D'autres études sont faites autour de la maladie du cancer consistant à faire une analyse multidimensionnelle sur un ensemble de données épidémiologiques Spatio-temporelles [Kamp et al 97].

Tous ces travaux s'appuient sur un *même* modèle multidimensionnel de base. Cependant, de nouvelles fonctionnalités peuvent être rajoutées à ce modèle afin d'offrir une meilleure utilisation.

6.2 Cube de données multimédias

Les travaux cités ci-dessus sont basés sur les bases de données multimédias et sur les *descripteurs* qui définissent les données. En effet, la conception des cubes de données multimédias est basée sur la conception des cubes de données traditionnelles. Par conséquent, le modèle en *étoile* ou le modèle en *flocon de neige* est utilisé pour modéliser un entrepôt de données multimédias de telle façon que la table de *faits* contienne la donnée multimédia sous forme de *liens*, et les dimensions représentent les *descripteurs* de cette donnée [Arigon et al 06].

Les descripteurs qui sont utilisés pour caractériser les faits des données multimédias sont classés en deux types : les descripteurs *basés sur le contenu* et les descripteurs *textuels*. Les descripteurs textuels représentent les informations extrinsèques ou externes (ex : mots clés, date d'acquisition, auteur, thème, etc.). Les descripteurs basés sur le contenu concernent les informations intrinsèques (nécessaires) des données (ex : couleur, texture, forme, etc.) qui sont généralement automatiquement extraites des données [Han et al 01]. Par exemple, dans un entrepôt de données, ayant comme *fait* une donnée image, les dimensions pour un descripteur basé sur le contenu peuvent être la couleur et la texture, et les dimensions pour un descripteur textuel peuvent être le thème et les mots clés.

Toutes les études citées précédemment utilisent des dimensions basées sur des descripteurs de données calculés au moment du chargement.

Comme déjà mentionné, la conception des cubes de données multimédias suit les paradigmes de la conception des cubes de données traditionnelles. En effet, les membres d'une dimension qui correspondent aux valeurs de descripteurs de données sont calculés en amont durant le processus ETL (*Extraction Transformation Loading*) et deviennent statiques une fois stockés dans les tables de dimensions.

7. Quelques travaux liés aux entrepôts de données multimédias

Dans ce paragraphe, nous allons présenter quelques travaux réalisés autour des entrepôts de données multimédias. Certains visent à améliorer la qualité des entrepôts de données multimédias, et par conséquent ils s'intéressent à améliorer des tâches importantes telles que l'intégration, la modélisation, l'interrogation, etc. d'autres travaux n'apportent aucun plus, ils utilisent seulement la technologie des entrepôts multimédias pour réaliser leurs buts.

7.1 XML et les entrepôts de données multimédia

Dans [Kim et al 03], les auteurs ont présenté un environnement d'entreposage de données multimédia intégrant des données multimédia provenant de sources diverses. Afin de gagner en flexibilité et en interopérabilité, la technologie XML a été utilisée pour la représentation des données. L'architecture de l'entrepôt utilisée comprend les quatre composants : *Média object repository, Média View generator, Meta-knowledge bases et meta-knowlage manager* précédemment décrits dans la figure 4.2.

Dans la suite, nous explicitons le 'générateur de vues' *(media view generartor)*, ensuite nous expliquons ce que c'est le gestionnaire '*meta-knowledge manager'*, et nous termininons par décrire les '*wrappers'*.

7.1.1 Le générateur *Media View Generator*

Ce générateur crée un ensemble de vues matérialisées des sources de données en question, ces vues sont appelées *'médias-vues'*. Les chercheurs ont adopté XML, qui est accepté comme un format standard pour l'échange de données sur le Web.
La seule insuffisance de XML comme un modèle d'entrepôts multimédias est le manque de gestion des relations telles que la contrainte de clé étrangère dans les modèles relationnels. Cependant, pour la modélisation des contenus multimédias, la gestion des relations entre les objets est importante. C'est pourquoi les chercheurs ont étendu XML en se basant sur les concepts de la modélisation orientée objets pour définir un nouveau modèle XML/M (M pour multimédia).

- **Le modèle XML/M**

L'unité de base de ce modèle est l'*objet média*. Pour modéliser les objets médias, les concepteurs ont utilisé deux fonctionnalités principales de la modélisation orientée objets : *l'identité d'objet* et *la nidification d'objet.* Chaque objet média possède un identifiant d'objet unique, et son arbre de structure décrit les contenus médias. Le modèle XML/M ne nécessite pas des schémas fixes, à cause de son auto-description (les données multimédias ne sont pas décrites selon les mêmes critères). Par exemple, un objet image peut être décrit par une

couleur, la texture, une forme, etc., tandis qu'un objet vidéo peut être décrit par une scène significative, une description de scène, un cadre de début, un cadre de fin, etc.

Les objets médias avec des relations sémantiques peuvent être intégrés dans un objet multimédia qui peut être considéré comme un objet complexe. Ce dernier est généré en se basant sur les *graphes objet-relation (ORG :Object-relation graph)* spécifiant les relation sémantiques entre les objets médias.

- **Le langage XQuery/M**

Le but principal de ce langage de requêtes est de fournir un mécanisme pour générer des médias-vues. Les concepteurs ont adopté le XQuery, qui est un langage standard du W3C pour le langage XML. Ses opérations sont classifiées en opérations d'accès aux objets médias et en opérations de création de nouveaux objets médias.
L'accès aux objets médias a été réalisé en utilisant l'expression FLWOR [XQuery], qui utilise l'expression du chemin et les motifs. Pour agréger des objets médias avec des relations sémantiques, des opérateurs spéciaux ont été développés tels *que CopyTree, SelectSubTree* et *MergTree*.

7.1.2 Le gestionnaire *Meta-Knowledge Manager*

Ce gestionnaire contrôle les modifications des sources de données. En effet, les vues matérialisées sont mises à jour suite à ces modifications. Pour cela, les auteurs ont développé une technique de maintenance de vues basée sur des règles. Les *médias-vues* étant divisées en deux catégories : les *intra-médias-vues* et les *inter-médias-vues*, cependant, trois groupes de règles ont été distingués : des règles pour la maintenance des *intra-médias-vues*, des règles pour la maintenance des *inter-médias-vues*, et des règles pour la maintenance des contenus.

- **Règles pour la maintenance des intra-médias vues**

Si les modifications consistent en une création (ou suppression) d'un objet média, et que cet objet appartient à une intra-média vue, alors les opérations *Add* (ou *Remove*) sont exécutées contre la vue en question. L'opération Add ajoute le OID (*Object Identifier*), et l'opération *Remove* le supprime.

- **Règles pour la maintenance des inter-médias vues**

Dans ce cas, les modifications affectent l'ORG. Cependant, l'ORG lié aux changements est mis à jour, ensuite les inter-médias vues qui réfèrent à cet ORG sont recalculées.

- **Règles pour la maintenance des contenus**

Les modifications des contenus concernent seulement les vues inter-médias dans lesquelles les contenus multimédias sont représentés par des arbres XML, et par conséquent leurs modifications sont gérées par des algorithmes de comparaison d'arbres.

7.1.3 Les *Wrappers*

Un *wrapper* fournit deux services majeurs pour cet environnement d'entreposage de données multimédia.
Tout d'abord, il transforme les sources de données en un modèle d'entrepôt. Et dans ce cadre, les auteurs ont développé deux types de wrappers : le wrapper relationnel et le wrapper MPEG-7. Beaucoup de bases de données relationnelles supportent des vues XML de données relationnelles. Le wrapper relationnel restructure ces documents XML en XML/M en utilisant XSLT. Puisque la description MPEG-7 est représentée par XML, le wrapper MPEG-7 transforme les documents XML de la même manière.

En deuxième lieu, le wrapper supporte la communication avec l'entrepôt de données multimédia en utilisant le protocole SOAP[Kim et al 03]. En effet, si les données changent, les wrappers génèrent un message SOAP et l'envoient vers l'entrepôt. Ensuite, si le gestionnaire *Meta-Knowledge Manager* reçoit ce message de notification, il envoie un message de demande au wrapper. En troisième étape, les wrappers envoient un message contenant l'information modifiée au Meta-*Knowledge Manager.* En dernière étape, le Meta-Knowledge Manager extrait l'information du message SOAP, et la reflète vers les média-vues en utilisant l'approche basée sur les règles vues précédemment.

La figure 4.3 montre les deux tâches effectuées par un *wrapper*.

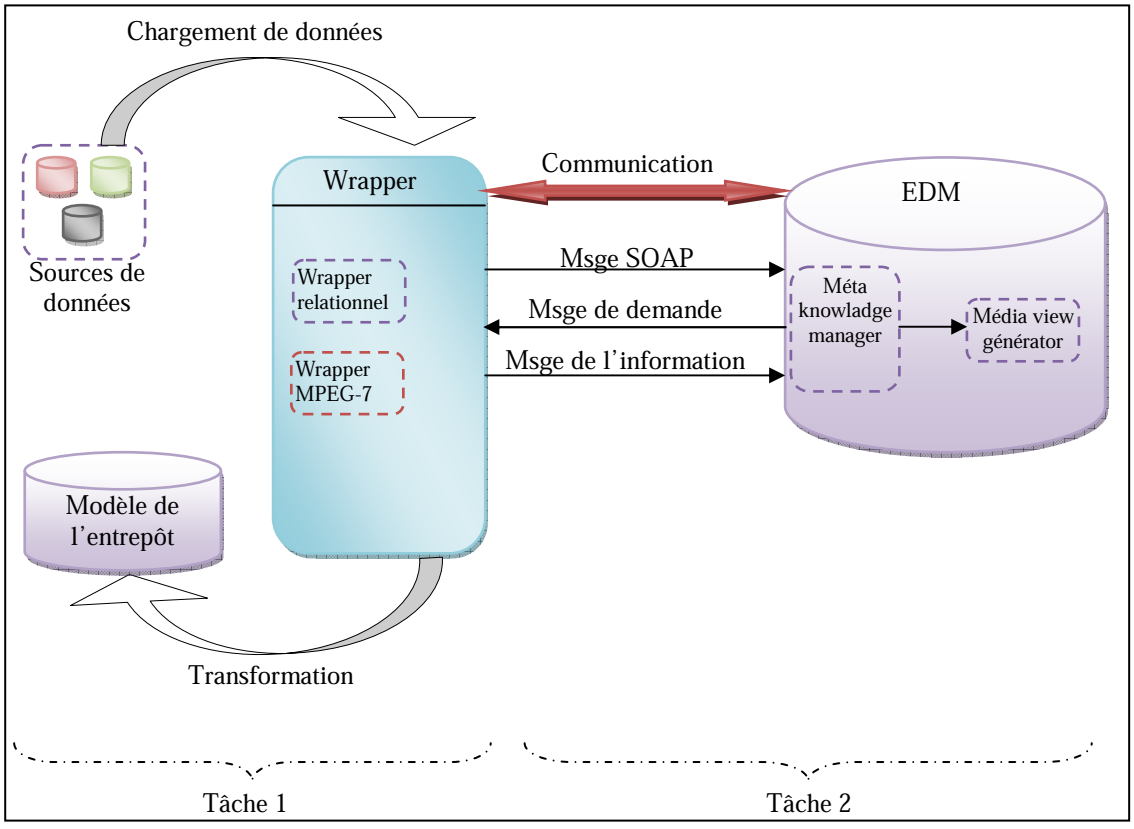


Figure 4.3. *Tâches réalisée par un wrapper*

7.2 Aspect sémantique dans les entrepôts de données multimédias

Ce travail est une suite du précédent [Kim et al 04]. En effet, les auteurs ont développé une technique de gestion pour les objets multimédias sans perdre la sémantique des informations.

Une technique basée sur la sémantique a été proposée pour gérer les objets multimédias dans l'environnement décrit dans [Kim et al 04]. Les auteurs ont souligné que les objets multimédias doivent être gérés tout en préservant les contenus multimédias et les relations entre ces objets.

Pour cela une nouvelle notion a été définie : il s'agit de la relation entre les objets (on définit l'objet *relation*). Par la suite, nous décrivons le versionnement d'objets, en cas de modifications, basé sur la sémantique. Les notions déjà décrites précédemment restent valables ici.

7.2.1 Relations entre les objets

La seule insuffisance de XML comme un modèle d'entrepôts multimédias est le manque de gestion des relations [Kim et al 04]. Par conséquent, il est nécessaire de gérer les relations indépendamment.

Les objets médias représentent le contenu multimédia, tandis que les objets relations spécifient différents types de relations sémantiques.

Les auteurs ont classifié les relations qui sont nécessaires pour les applications multimédias en : relations temporelles, spatiales et sémantiques (tableau 4.1).

Les relations sémantiques sont définies pour capturer le concept d'abstraction des contenus multimédias. Ces relations sont au nombre de cinq:

Relation Is-A: (est un): un objet Media M1 est un objet media M2 signifie que le contenu de M2 est une généralisation du contenu de M1, et le contenu de M1 est une spécialisation du contenu de M2. Par exemple, un objet vidéo contenant des scènes d'un jeu de football *est un* objet vidéo contenant des jeux sportifs.

Relation Is-Part-Of: (est une partie de) : un objet Media M1 *est une partie d'un* objet media M2 signifie que le contenu de M2 inclut le contenu de M1. Par exemple, un objet vidéo contenant des scènes de score fait partie d'un objet vidéo contenant des scènes de football

Relation Is-the Same-As: (est le même que) un objet Media M1 est *le même qu'un* objet media M2 signifie que le contenu de M2 et le contenu de M1 sont les mêmes. Par exemple, un objet audio A1 est le même qu'un objet audio A2 si A1 et A2 représentent la même musique, mais sont enregistrés par différents orchestres.

Relation Is-Involved-In: (est impliqué dans) : un objet Media M1 *est impliqué dans* un objet media M2 consiste en l'association de M1et M2 avec le même concept. Par exemple, l'objet image I1 contenant l'actrice "Audrey Hepburn" est impliqué dans l'objet audio A1 contenant la chanson "Moon River" avec le concept "Film: Breakfast at Tiffany".

Relation Appears-In: (apparaît dans) : un objet Media M1 *apparaît dans* un objet media M2 signifie que le contenu de M1 apparaît dans M2. Par exemple, l'objet image I1 contenant l'actrice "Audrey Hepburn" apparaît dans l'objet vidéo V1 contenant le vidéo clip de "Film: Breakfast at Tiffany".

Spatial Relations Objects O_1 O_2		Temporal Relations Interval T_1 T_2		Semantic Relations Content Abstraction
Relation Name	Semantics	Relation Name	Semantics	Relation Name
Disjoint		Before		Is-A
Equals		Equals		Is-Part-Of
Meets		Contains		Is-the Same-As
Overlaps with Disjoint B.		Overlaps		Is-Involved-In
Overlaps with Intersecting B.		Meets		Appears-In
Cover /Covered-by		Starts		
Contains/ Contained-by		Finishes		

Tableau 4.1. *Classification de relations pour les applications multimédias*

Formellement, l'objet relation consiste en une paire (XID, T), où XID est un objet identifiant de l'objet relation, et T est un arbre constitué soit des feuilles, soit des paires *label/arbre* où : Type T = (Relation spatiale | Relation temporelle| Relation sémantique)

Relation spatiale = (nom relation, 1 objet média ou plus)
Relation temporelle = (nom relation, 1 objet média ou plus)
Relation sémantique = (nom relation, 1 objet média ou plus)

L'arbre T est décrit par une relation spatiale, temporelle ou sémantique, et chaque relation est décrite par le nom de la relation et les objets participant à la relation. La figure 4.4 permet d'illustrer cette définition :

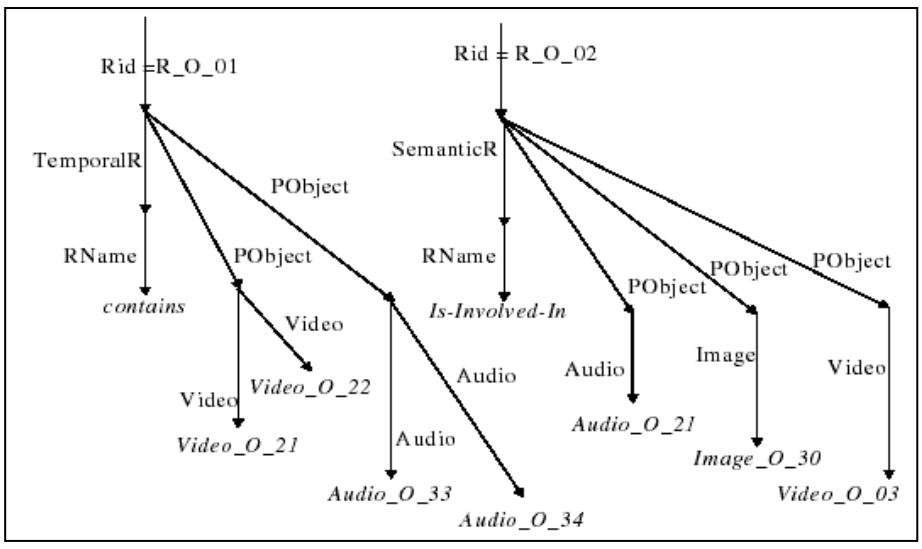


Figure 4.4. *Exemple de l'objet 'relation'*

La partie gauche de la figure montre un objet relation avec "R_O_01" comme objet identifiant, spécifiant une relation binaire. Les objets sources *Vidéo_O_21* et *Vidéo_O_22,* et les objets cibles *Audio_O_33* et *Audio_O_34* ont une relation temporelle '*contains'*.

La partie droite montre un objet relation avec "R_O_02" comme objet identifiant, spécifiant une relation 3-aire. Les objets participants sont : *Audio_O_21*, *Image_O_03*, and *Video_O_03*, avec une relation sémantique "*Is_Involved_In*".

7.2.2 Versionnement de schémas basé sur la sémantique

Puisque le modèle défini par les auteurs est basé sur XML, le problème de versionnement des objets multimédias dans cet environnement converge vers un versionnement de documents XML. Plusieurs recherches ont été faites dans ce cadre, on peut citer [Kim et al 04]: le système *Xyleme,* l'approche basée-référence et l'approche basée-copie.

Les utilisateurs d'un entrepôt de données multimédia peuvent être intéressés par les objets '*vues'*, plutôt que de s'intéresser aux modifications des objets de la vue.

Puisque le contenu multimédia est représenté par une structure d'arbre XML, on doit préserver cette structure pour garder la sémantique de l'information. En tenant compte de cette hypothèse, l'approche basée-copie convient pour gérer les versions des vues. Cependant, puisque cette approche nécessite plus d'espace, on a besoin d'augmenter les performances de stockage[Kim et al 04]. Par conséquent, on utilise l'approche basée-copie pour la partie modifiée du sous arbre et l'approche basée-référence pour l'autre partie.

Puisque les vues inter et intra médias sont composées de deux sous arbres, on divise les vues en deux parties. Si l'utilisateur modifie un des sous arbres, alors la partie du sous arbre est considérée comme un objet versionnable. On crée une nouvelle version pour cet objet avec un système de génération de nombre de versions. Le sous arbre inchangé, correspondant à un objet non versionnable, est stocké une fois et possède un lien avec la partie versionnable. Lorsque les utilisateurs veulent récupérer l'ensemble des objets, l'ensemble des vues est récupéré en fusionnant la partie versionnable et celle non versionnable.

Pour effectuer ces modifications, les fonctionnalités de contrôle de changements sont nécessaires. L'approche de versionnement décrite ici est conçue pour préserver la sémantique des objets multimédias. En effet, elle réduit le stockage sans perdre la sémantique des objets.

7.3 Traitement des requêtes dans les entrepôts de données multimédias

Dans un environnement d'entrepôts de données traditionnelles, les requêtes sont généralement effectuées sur des vues matérialisées stockées dans l'entrepôt. Par contre, dans un environnement d'entrepôts multimédias, à cause de l'énorme espace de stockage nécessaire, il est impossible de stocker les vues matérialisées pour répondre à toutes les requêtes [Perrizo et al 97]. La solution proposée est d'utiliser une *instanciation à la demande* (IOD, *Instantiation-On-Demand*). Pour cela, on doit assurer un traitement rapide de l'IOD pour les vues. Un travail à été réalisé dans [Perrizo et al 97] s'appuyant sur la technique d'instanciation à la demande appliquée aux vues. Ce dernier consiste en l'utilisation des structures de *vecteur de domaine* (DV, Domain Vector) pour accélérer les requêtes.

Les auteurs de [Perrizo et al 97] décrivent dans leur travail une technique d'indexation similaire à l'indexation *BitMap* où ils ont montré comment les vecteurs de domaine (DV) peuvent être utilisés pour accélérer des requêtes dans des entrepôts de données multimédias.

7.3.1 Utilisation du vecteur de domaine

En général, les requêtes dans un entrepôt de données multimédia peuvent être classifiées en quatre catégories [Perrizo et al 97] :

- des requêtes qui peuvent être résolues par des vues matérialisées.
- des requêtes qui peuvent être résolues par la jointure de deux relations de sources différentes (lointaines).
- des requêtes qui peuvent être résolues par la jointure d'une vue matérialisée avec une relation dans une autre source (lointaine).
- des requêtes qui peuvent être résolues par la jointure de deux vues matérialisées.

Concernant la première catégorie, afin d'accélérer les requêtes, on utilise les index de vecteur de domaine (*DVIs, Domain Vector Indexes*). Pour les autres catégories, on utilise des algorithmes de jointures qui se basent sur les vecteurs de domaine.

7.3.2 Avantages apportés par la structure du vecteur de domaine

Les DVIs sont utilisées principalement pour accélérer les jointures dans un environnement multimédia ; la structure du VD présente les avantages suivants [Perrizo et al 97] :

- une taille plus petite du vecteur lorsque la cardinalité est grande ;
- facile à étendre pour des domaines plus grands ;
- prêt à prendre en compte les OID ;
- de nouvelles techniques pour améliorer les performances d'accélération avec les vecteurs de domaine.

8. Conclusion

Le but de ce chapitre était de cerner la technologie des entrepôts de données multimédias qui deviennent indispensables dans le monde des entreprises, notamment dans les domaines de la médecine, la biologie, l'enseignement à distance, etc. Au début de ce chapitre, nous avons présenté une architecture d'un EDM, ainsi que quelques notions liées à cette catégorie d'entrepôts. Par la suite, nous avons mis l'accent sur quelques travaux réalisés autour des EDM qui se résument en trois travaux. Le premier concerne l'extension de XML pour prendre en compte les objets multimédia. Le second vise la prise en compte de l'aspect sémantique dans les EDM. Et le troisième a pour but le traitement des requêtes dans les EDM.

La particularité des EDM est qu'ils traitent principalement des données multimédia de natures très variées. Cependant, nous voyons qu'il est très intéressant de prendre en compte l'aspect sémantique lié à ce type de données. C'est pourquoi nous nous sommes penchés sur l'utilisation des *ontologies* dans notre approche vu que cette structure est d'un grand apport pour la gestion des conflits sémantiques entre les données au moment de leur intégration au niveau de l'entrepôt.

Le chapitre suivant présente une introduction aux ontologies en mettant l'accent sur les différents concepts et notions liées à ces structures. Une partie du chapitre est consacrée à l'utilisation des ontologies dans les applications multimédia.

Chapitre 5

Les Ontologies : Concepts, Principes et Applications

1. Introduction

Les *Systèmes d'Information des Entreprises (SIE)* sont largement utilisés par les entreprises modernes. Les SIE sont par définition un ensemble de systèmes qui couvrent l'infrastructure d'information de l'entreprise, améliorent la productivité des employés à travers un ensemble d'outils et offrent aussi des services aux utilisateurs [Rifaieh04].

En raison de leur complexité, ces systèmes sont amenés à coopérer et à échanger des données pour répondre à certains besoins. Il faut alors pouvoir exprimer la sémantique des concepts utilisés par ces systèmes de façon explicite. Pour résoudre ce problème d'hétérogénéité sémantique, les *ontologies* sont apparues.

Les ontologies sont à l'heure actuelle au cœur des travaux menés en Ingénierie des Connaissances (IC). Ces travaux visent l'établissement de représentations à travers lesquelles les machines puissent manipuler la sémantique des informations. Les ontologies apparaissent comme des composants logiciels s'insérant dans les SI en leur apportant une dimension sémantique qui leur faisait défaut jusqu'ici.

Le champ d'application des ontologies ne cesse de s'élargir et couvre les systèmes conseillers (systèmes d'aide à la décision, systèmes d'enseignement assisté par ordinateur), les systèmes de résolution de problèmes ou les systèmes de gestion de connaissances. Plus précisément, les ontologies ont donné une grande satisfaction pour résoudre les conflits sémantiques et structurels entre les différentes sources de données lors d'un processus d'intégration de BD [Ruzzi04], [Bellatrech06].

Quelques méthodologies ont été proposées pour soutenir le développement des ontologies et les mettre en place dans un système d'intégration [Ruzzi04], [Xuan et al 06].

Nous présentons dans ce chapitre un panorama sur les ontologies où nous décrivons leur principe de base, les concepts qui y sont liés ainsi que leurs applications. Une bonne partie du chapitre est consacrée à l'utilisation des ontologies dans le domaine multimédia, notamment l'imagerie médicale et satellitaire.

2. Motivation

L'humain est capable par la force de son langage de référencer deux contextes différents, en utilisant le même terme, ou l'utilisation de plusieurs termes pour référencer la même idée. Or, l'automatisation n'est pas apte à faire ceci. Cela serait bien, si la machine pouvait comprendre le sens des termes, même s'ils émanent de plusieurs bases de données. Comment régler le cas où un terme prend deux significations différentes, comment lever toute ambiguïté ? Ce problème est traité par la notion d'***ontologie***.

Une ontologie est définie comme la conceptualisation des objets reconnus comme existant dans un domaine, de leurs propriétés et des relations les reliant. Elle définit un vocabulaire commun pour les chercheurs qui ont besoin de partager l'information dans un domaine. Elle inclut des définitions lisibles en machine des concepts de base de ce domaine et de leurs relations.

La notion d'ontologie est devenue un élément clé dans toute une gamme d'applications faisant appel à des connaissances. Un des plus grands projets basés sur l'utilisation d'ontologies est le Web sémantique, qui consiste à ajouter au Web actuel une véritable couche de connaissances, permettant des recherches d'information au niveau sémantique et non plus syntaxique. Les ontologies sont devenues donc très courantes dans le World-Wide-Web. Le champ des ontologies dans le web varie de taxonomies larges servant à catégoriser les sites Web (tels que dans Yahoo!) aux catégorisations de produits destinés à la vente et de leurs caractéristiques (tel que dans Amazon.com).

Le Consortium *W3C* a développé le « *Resource Description Framework* » (*RDF*), un langage d'encodage de la connaissance sur les pages Web pour rendre cette connaissance compréhensible aux agents électroniques qui effectuent des recherches d'information [Noy et al 00]. L'organisme DARPA « Defence Advances Research Projects Agency », conjointement avec le W3C, a développé le DARPA Agent Markup Language (DAML) en élargissant RDF avec d'autres constructions expressives qui visent à faciliter l'interaction des agents sur le Web.

Plusieurs ontologies normalisées sont utilisables par les experts de domaines dans différentes disciplines. La médecine par exemple, a produit de vastes vocabulaires normalisés structurés tels que SNOMED et le réseau sémantique du « Unified Medical Language System ». De même naissent de larges ontologies universelles : par exemple le Programme des Nations Unies pour le développement et Dun & Bradstreet ont uni leurs efforts pour développer l'ontologie UNSPSC qui fournit une terminologie pour les produits et les services. (www.unspsc.org).

Cependant, l'exploitation des ontologies nécessite une bonne compréhension sémantique, une bonne conception, une bonne définition des concepts et des relations à représenter, ainsi qu'une bonne gestion de leur évolution, tout au long d'un processus de développement. On développe une ontologie pour des raisons multiples, en voici quelques-unes :

- Partager la compréhension commune de la structure de l'information entre les personnes ou les fabricants de logiciels.
- Permettre la réutilisation du savoir sur un domaine
- Expliciter ce qui est considéré comme implicite sur un domaine
- Distinguer le savoir sur un domaine du savoir opérationnel
- Analyser le savoir sur un domaine

3. Définitions

La littérature d'Intelligence Artificielle contient plusieurs définitions pour une ontologie ; un bon nombre d'entre elles sont même contradictoires. Voici quelques définitions proposées :

Définition 1 : une *ontologie* est une description formelle explicite des concepts dans un domaine du discours (*classes*, appelées parfois *concepts*), des propriétés de chaque concept décrivant des caractéristiques et attributs du concept (*attributs*, appelés parfois *rôles* ou *propriétés*) et des restrictions sur les attributs (*facettes*, appelées parfois *restrictions de rôles*) [Noy et al 00].

Une ontologie ainsi que l'ensemble des *instances* individuelles des classes constituent une *base de connaissances*. Il y a en réalité une frontière subtile qui marque la fin d'une ontologie et le début d'une base de connaissances [Noy et al 00]. En effet, Le but d'une ontologie est de définir un vocabulaire pour décrire un domaine, si possible de manière complète ; ni plus, ni moins. Contrairement aux bases de connaissances par exemple, on n'attend pas d'une ontologie qu'elle soit en mesure de *fournir systématiquement une réponse à une question arbitraire sur le domaine*. Une ontologie est la théorie la plus faible couvrant un domaine ; elle ne définit que les termes nécessaires pour partager la connaissance liée à ce domaine

Définition 2 : Une ontologie est une structure de donnée opérationnelle, qui rend compte des concepts et des relations entre ceux-ci, pour modéliser un domaine donné [Djezzar et al 07].

Définition 3 : Une des plus simples et des plus populaires définitions est celle de M. Tom Gruber [Grubert 93]: «Une ontologie est une spécification formelle, explicite d'une conceptualisation partagée ». Les attributs "formelle" et "explicite" signifient qu'une ontologie permet l'interprétation automatisée de la conceptualisation par la machine.

Autrement dit, une ontologie définit un vocabulaire commun pour les chercheurs qui ont besoin de partager l'information dans un domaine. Elle inclue des définitions lisibles en machine des concepts de base de ce domaine et de leurs relations.

4. Eléments de base d'une ontologie

Les connaissances traduites par une ontologie sont véhiculées à l'aide des éléments suivants :

Les concepts : correspondent aux abstractions pertinentes d'un segment de la réalité (le domaine du problème), retenues en fonction des objectifs qu'on se donne et de l'application envisagée pour l'ontologie. Un concept peut se définir comme une entité composée de trois éléments distincts :

- *Le **terme*** : exprimant le concept en langage naturel.
- ***Notion*** ou ***intension*** du concept : la signification du concept.
- *Les **objets*** dénotés par le concept, appelés également « *réalisations* » ou « *extensions* » du concept.

Prenons l'exemple du concept ORDINATEUR. Le terme de ce concept est le nom commun *ordinateur.* Sa notion est d'*être une machine de traitement de données.* Son extension est l'*ensemble des marques d'ordinateur existantes (dell, hp, msi …)*.

Les classes : elles représentent le centre d'intérêt de l'ontologie et décrivent les concepts d'un domaine. Une classe peut avoir des sous-classes qui représentent des concepts plus spécifiques que la super classe (ou classe supérieure).
Une classe peut avoir des instances. Ces instances sont des entités réelles de cette classe, elles sont une représentation des extensions du concept. Exemple : *hp* est une instance de la classe '*ordinateur'*. Il est à noter qu'une ontologie ainsi que l'ensemble des instances de toutes ses classes constituent une base de connaissances.

Les attributs : Les attributs décrivent les propriétés des classes et des instances de l'ontologie.
Exemple : pour la classe *'ordinateur'*, on peut citer les attributs : *année de production, lieu de production, quantité produite, etc.*

Les relations : désignent les associations prédéfinies qui existent entre les concepts de l'ontologie. A titre d'exemple, on trouve les relations incluant les associations suivantes :

- Sous classe de (généralisation spécialisation) ;
- Partie de (agrégation ou composition) ;
- Associée à ;
- Instance de ;
- etc.

Les axiomes : désignent les assertions acceptées comme vraies dans le domaine étudié. Les axiomes et les règles permettent de vérifier la cohérence d'une ontologie, et aussi d'inférer de nouvelles connaissances.
Exemple : « Si deux personnes sont frères, alors il existe quelqu'un qui est la mère de chacun d'eux ».

Les fonctions **:** constituent des cas particuliers des relations, dans lesquelles un élément de la relation, le nième est définit en fonction des n-1 éléments précédents.

5. Construction d'une ontologie

En réalité, il n'existe pas qu'un seul mode ou qu'une seule méthodologie *correcte* pour développer une ontologie. En effet, au fur et à mesure, des méthodologies de construction sont apparues [Djezzar et al 07]. Une véritable ingénierie s'est constituée autour des ontologies, ayant pour but leur construction, mais plus largement leur gestion tout au long d'un cycle de vie. La construction des ontologies demande une étude des connaissances humaines, la définition de langages de représentation et la réalisation de systèmes pour les manipuler. Depuis la naissance de l'intelligence artificielle, plusieurs paradigmes de

représentation de connaissances ont été développés, permettant de formaliser les connaissances d'un domaine, puis de mettre en œuvre des raisonnements sur ces représentations. Parmi ces paradigmes, trois grands modèles sont distingués : les langages de frames, les graphes conceptuels et les logiques de description [Djezzar et al 07].

Nous décrivons dans ce qui suit une approche générale pour la construction d'une ontologie contenant les différentes étapes à suivre [Noy et al 00]. Nous présentons également quelques méthodologies proposées par des chercheurs dans certains travaux.

5.1 Approche générale

La figure 5.1 montre les principales étapes de l'approche générale de construction d'une ontologie:

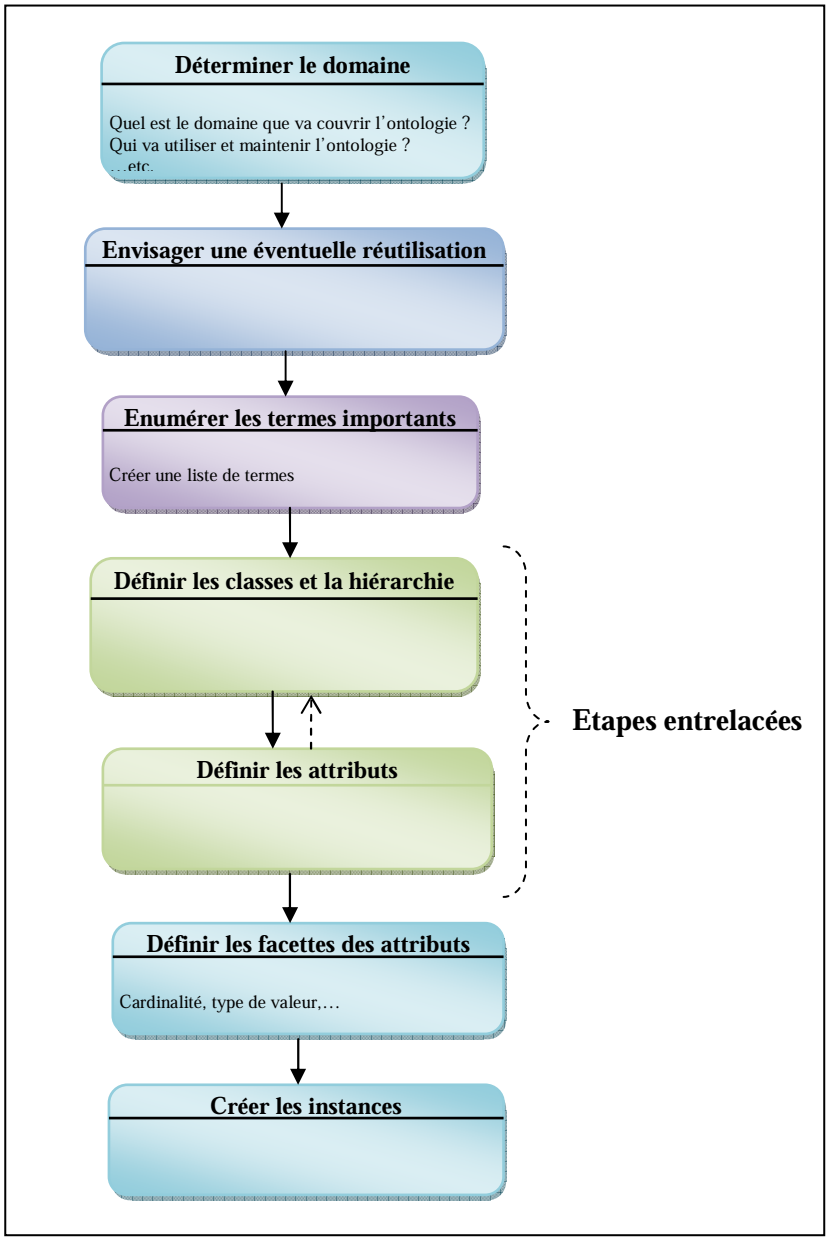


Figure 5.1. *Etapes générales pour la construction d'une ontologie*

Etape 1 ***: Détermination du domaine et son étendue :***

Pour déterminer le domaine et son étendue, il s'agit de répondre aux questions suivantes:

- Quel est le domaine que va couvrir l'ontologie ?
- Dans quel but utiliserons-nous l'ontologie ?
- A quels types de questions l'ontologie devra-t-elle fournir des réponses ?
- Qui va utiliser et maintenir l'ontologie ?

Les réponses à ces questions peuvent varier au cours du processus de conception de l'ontologie, mais à chaque moment, elles aident à limiter la portée du modèle.

Etape 2 ***: Envisager une éventuelle réutilisation des ontologies existantes***

Il est toujours utile de prendre en considération ce que d'autres personnes ont fait et d'examiner si nous pouvons élargir des sources existantes et les affiner pour répondre aux besoins de notre domaine ou de notre tâche particulière.

Réutiliser des ontologies existantes peut même constituer une exigence si le système a besoin d'interagir avec d'autres applications qui utilisent déjà des ontologies spécifiques ou des vocabulaires contrôlés. Il existe des bibliothèques d'ontologies réutilisables sur le Web et dans la littérature. Par exemple, la bibliothèque des ontologies Ontolingua (http://www.ksl.stanford.edu/software/ontolingua/) ou bien la bibliothèque des ontologies DAML (http://www.daml.org/ontologies/). Un certain nombre d'ontologies commerciales sont également disponibles pour le tout public (ex. UNSPSC (http://www.unspc.org).

Etape 3 ***: Enumérer les termes importants dans l'ontologie***

Il est utile de noter sous forme de liste tous les termes à traiter ou à expliquer à un utilisateur. Sur quels termes souhaiterions-nous discuter ? Quelles sont les propriétés de ces termes ? Que veut-on dire par ces termes ?

Il est important d'établir une liste exhaustive de termes et de ne pas se soucier de l'éventuel chevauchement entre les concepts qu'ils représentent, les relations entre les termes ou toute autre propriété des concepts, ni si ces concepts sont des classes ou des propriétés.

Remarque : Les deux étapes suivantes *« développement de la hiérarchie des classes »* et *« définition des propriétés des concepts »* sont étroitement entrelacées. Il est difficile de le faire en ordre linéaire strict. En général, il faut entamer quelques définitions de concepts dans la hiérarchie, ensuite procéder à la description des propriétés de ces concepts et ainsi de suite. Ces deux étapes sont aussi les plus importantes dans le processus de construction d'une ontologie.

Etape 4 ***: Définir les classes et la hiérarchie des classes***

Il existe un certain nombre d'approches possibles pour développer une hiérarchie de classes :

- Un procédé de développement *de haut en bas* : commence par la définition des concepts les plus généraux du domaine et se poursuit par la spécialisation des concepts.

- Un procédé de développement *de bas en haut* : commence par la définition des classes les plus spécifiques, les feuilles d'une hiérarchie, et se poursuit avec le regroupement de ces classes en concepts plus généraux.

- Un procédé *combiné* de développement est une combinaison des deux approches, de haut en bas et de bas en haut. Au tout début, les concepts les plus saillants sont définis, ensuite ils sont généralisés ou spécialisés, suivant le cas.

Aucune de ces trois méthodes n'est fondamentalement meilleure que les autres. L'approche à adopter dépend fortement du point de vue personnel sur le domaine. L'approche combinée est souvent, la plus facile à utiliser pour la plupart des développeurs d'ontologies, étant donné que les concepts « du milieu » ont tendance à être les concepts les plus descriptifs du domaine.

Quelle que soit l'approche choisie, habituellement il faut définir les *classes*. Dans la liste créée lors de la troisième étape, on sélectionne les termes qui décrivent des objets ayant une existence indépendante plutôt que les termes qui décrivent ces objets. Ces termes constitueront les classes dans l'ontologie et deviendront des points d'ancrage dans la hiérarchie des classes.

La figure 5.2 montre une représentation graphique de la hiérarchie des classes d'une ontologie, à l'aide de l'outil *Jambalaya* sous l'éditeur d'ontologies *Protégé*.

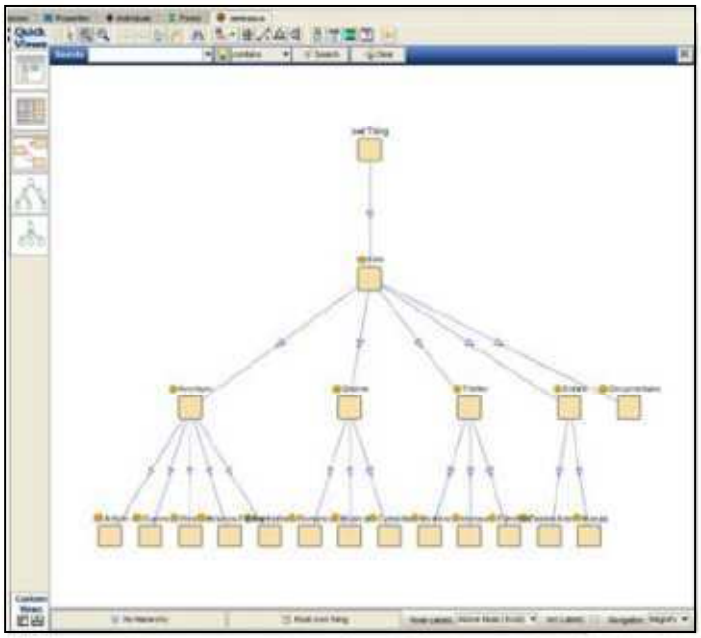


Figure 5.2. *Exemple d'une représentation hiérarchique d'une l'ontologie*

<u>Etape 5</u> ***: Définir les propriétés des classes – Les attributs***

Les classes seules ne fourniront pas assez d'information pour répondre aux questions de compétence de l'étape 1. Après avoir défini quelques classes, on doit décrire la structure interne des concepts tels que : le type, la cardinalité, etc.

Étape 6. ***Définir les facettes des attributs***

Les attributs peuvent avoir plusieurs facettes décrivant la valeur du type, les valeurs autorisées, le nombre de valeurs (cardinalité), et d'autres caractéristiques de valeurs que les attributs peuvent avoir.

a. La cardinalité des attributs

La cardinalité des attributs définit le nombre de valeurs qu'un attribut peut avoir. Certains systèmes distinguent entre cardinalité unique (n'autorisant qu'une seule valeur) et cardinalité multiple (autorisant une ou plusieurs valeurs).

Certains systèmes permettent de spécifier une cardinalité minimale et maximale pour décrire plus précisément le nombre de valeurs d'un attribut. Cardinalité minimale N veut dire qu'un attribut doit avoir au moins N valeurs. Cardinalité maximale M veut dire qu'un attribut peut avoir au maximum M valeurs. Il est parfois utile de fixer la cardinalité maximale à 0. Cette définition voudrait dire que l'attribut ne peut pas avoir de valeurs pour une sous-classe particulière.

b. Le type de valeur des attributs

La facette « type de valeur » décrit les types de valeurs pouvant être affectés à l'attribut. Voici une liste des types de valeurs les plus typiques :

- **Chaîne de caractères** est le type de valeur le plus simple utilisé pour des attributs tels que nom : la valeur est une simple chaîne de caractères
- **Nombre** (quelquefois des types de valeurs Enveloppe et Entier plus spécifiques sont utilisés) décrit des attributs ayant des valeurs numériques. Par exemple, le prix d'une voiture peut avoir le type de valeur Enveloppe.
- Les attributs **Booléens** sont des icônes simples de type oui - non.
- Les attributs **Énumérés** précisent une liste de valeurs spécifiques autorisées pour l'attribut. Dans Protégé-2000 les attributs énumérés sont de type Symbol.
- Les attributs de type **Instance** permettent la définition des relations entre individus. Les attributs ayant pour type de valeur Instance, imposent aussi la définition d'une liste de classes autorisées desquelles les instances sont issues.

c. Le domaine et le rang d'un attribut

Les classes autorisées pour les attributs de type Instance sont souvent appelées **étendue** d'un attribut.

Les classes auxquelles un attribut est rattaché ou les classes dont l'attribut décrit les propriétés, sont appelées le domaine d'un attribut. Dans les systèmes où les attributs sont *rattachés* aux classes, les classes auxquelles l'attribut est rattaché constituent le domaine de cet attribut. Il n'y a pas besoin de spécifier le domaine séparément.

Les règles de base pour déterminer le domaine et l'étendue d'un attribut sont similaires :

- Pour définir le domaine ou l'étendue d'un attribut, chercher la ou les classes les plus générales qui peuvent être respectivement le domaine ou l'étendue pour les attributs.
- En revanche, ne pas définir un domaine ou une étendue trop généraux : toutes les classes du domaine d'un attribut doivent être décrites par l'attribut et les instances de toutes les classes de l'étendue d'un attribut doivent être des valeurs potentielles de remplissage pour l'attribut.

- Ne pas choisir une classe trop générale pour l'étendue (c'est à dire, il est inutile de créer une étendue CHOSE) mais il est possible de choisir une classe qui couvre toutes les valeurs de remplissage.

- **Etape 7 :** ***Créer les instances***

La dernière étape consiste à créer les instances des classes dans la hiérarchie. Définir une instance individuelle d'une classe exige de :

1- choisir une classe,
2- créer une instance individuelle de cette classe,
3- la renseigner avec les valeurs des attributs.

Exemple

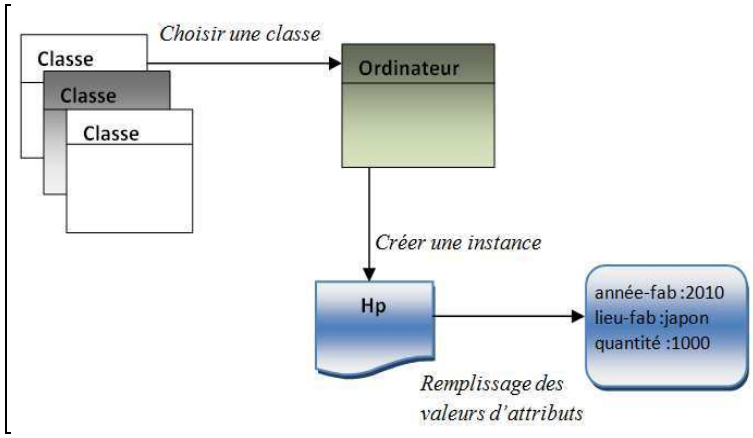


Figure 5.3. *Exemple de création d'une instance*

Remarque. Parfois, il est possible de modifier certaines étapes ou même d'en rajouter de nouvelles afin de mieux décrire le domaine en question. Ceci dépend des besoins des utilisateurs, de la complexité du domaine, des types de données, etc.

Dans la suite, nous décrivons les étapes d'un processus de développement d'une ontologie d'application incluant une nouvelle étape de formalisation basée sur la logique de description.

5.2 Processus de développement d'une ontologie d'application

Le besoin d'intégration des applications dans les entreprises (EAI) se fait de plus en plus sentir pour accroitre leur réactivité, pour faire face à de nouvelles contraintes,…etc. la refonte complète des systèmes d'information n'est plus au goût du jour. Désormais, le souci des entreprises est de faire relier leurs applications de manière à ce que les échanges inter-applications doivent porter une sémantique interprétable entre des entités communicantes.

Le travail réalisé dans [Guergour07] consiste en la construction d'une ontologie d'application dans le cadre de l'EAI *(Entreprise Application Integration)*. Le développement de cette ontologie doit suivre un processus de construction d'ontologies comptant un ensemble de phases spécifiées de façon très détaillée afin de cerner l'étendue et d'aboutir à une ontologie typique. Pour cela, les concepteurs se sont attachés à :

1. Proposer un processus de développement d'ontologies d'applications partant de connaissances brutes et arrivant à une ontologie d'application opérationnelle représentée par le langage OWL.
2. Construire l'ontologie d'application en suivant les étapes du processus proposé.

5.2.1 Description du Processus

Le processus de développement d'ontologie proposé dans [Guergour07] comporte cinq phases à savoir: **(1)** la spécification des besoins, **(2)** la conceptualisation, **(3)** la formalisation, **(4)** l'implémentation et **(5)** test et évolution de l'ontologie. La réalisation de ces différentes étapes, s'appuie sur les résultats des travaux réalisés en ingénierie des connaissances et en Intelligence Artificielle (IA). Ces travaux mettent l'accent sur l'aspect de représentation formelle et de raisonnement, et utilisent en particulier le formalisme de la logique de description qui possède une sémantique claire et procure des services d'inférences et des mécanismes de raisonnement puissants.

Phase 1 : Spécification

Cette phase consiste à établir un document formel de spécification des besoins représenté dans le langage RDF. Ce dernier permet de décrire l'ontologie à construire à travers les cinq aspects (i.e. propriétés) suivants :

a- **Le domaine de connaissance :** déterminer aussi précisément que possible le domaine que va couvrir l'ontologie.
b- **L'objectif :** le but de l'ontologie à créer pour le domaine considéré.
c- **Les utilisateurs :** identifier au maximum les futurs utilisateurs de l'ontologie à créer.
d- **Les sources d'information:** déterminer les sources d'informations d'où les connaissances seront obtenues, par exemple, les experts du domaine, les documents techniques, etc., sont des sources de connaissance.
e- **La portée de l'ontologie :** déterminer à priori la liste des termes (les plus importants) pour le domaine à représenter [Hamamet al 04].

Phase 2 : Conceptualisation

C'est l'étape la plus importante dans le processus de construction de l'ontologie. Elle est inspirée de la méthodologie METHONTOLOGY [Hamamet al 04] qui consiste à identifier et à structurer les connaissances du domaine, à partir des sources d'informations. Ces connaissances permettent d'aboutir à un ensemble de représentations intermédiaires semi-formelles indépendamment des langages de formalisation à utiliser pour représenter l'ontologie. A la fin de cette phase, on obtient une ontologie conceptuelle. Pour cela, on distingue les principales tâches suivantes :

a- Construction du glossaire de termes.
b- Construction du diagramme de relations binaires et de classification des concepts
c- Construction du dictionnaire de concepts (DC).
d- Décrire les relations dans une table de relations binaires.
e- Spécifier des contraintes sur les attributs dans une table d'attributs.
f- Spécifier des axiomes sur les concepts dans une table d'axiomes logiques.
g- Décrire les instances des concepts dans une table d'instances.

La description de chaque étape est définie au fur et à mesure de la construction de l'ontologie d'application.

Phase 3 : Formalisation

Cette phase consiste à formaliser l'ontologie conceptuelle obtenue dans la phase précédente afin de faciliter sa représentation ultérieure dans un langage complètement formel et opérationnel. Le formalisme utilisé est la *logique de description* basée sur le langage SHIQ qui présente une logique de description très expressive et qui offre un certain nombre de constructeurs pour décrire les concepts.

Syntaxe	Interprétation
T	Le concept le plus général (TOP)
⊥	Le concept le plus spécifique (Buttom)
C	Le nom d'un concept
R	Le nom d'un rôle (une relation binaire ou bien un attribut)
A	Le nom d'un individu
C1∩ C2	Conjonction de concepts pour définir de nouveaux concepts
C1∪ C2	Disjonction de concepts pour définir de nouveaux concepts
¬C	Utilisé pour définir le complément d'un concept
∀R.C	Définit le co-domaine du rôle R
∃ R.C	Il y a au moins un objet relié par le rôle R au concept C
(≥ n R.C) (≤ n R.C)	Cardinalité Minimum/Maximum (n est un nombre entier non négatif).
R^{-}	Le rôle inverse

Tableau 5.1. *Syntaxe du langage SHIQ*

La logique de description est constituée de deux parties : une partie *terminologique* (TBOX) permettant de décrire les concepts et les rôles et une partie *assertionnelle* (ABOX) décrivant les instances.

- **Partie terminologique TBOX**

La description des concepts et les rôles doit respecter la grammaire suivante:

< **Concept** >→ <concept-primitif> | <T> | <⊥> |
<Concept > ∩ < Concept > | < Concept > ∪ < Concept >|
¬< Concept >| ∀<rôle>.< Concept >| ∃ <rôle>. < Concept > |
≥ n <rôle>. < Concept> | ≤ n <rôle>. <concept> |
<rôle> → < rôle-primitif>
<rôle> → <rôle> ∩ <rôle>| <rôle> ∪ <rôle>

Ce qui montre dans la grammaire précédente que les concepts et les rôles sont exprimés de manière déclarative. Cette dernière comprend des définitions de concepts et de rôles ainsi que des inclusions de concepts.

Une définition de rôle peut être exprimée aussi en fonction des concepts qui l'utilisent.

R :(C1, C2) où R est un nom de rôle, C1 est un concept source et C2 est un concept cible
R :=R1 où R1 est le rôle inverse pour R

- **Partie assertionnelle ABOX**

Elle définit l'ensemble des axiomes utilisés pour décrire des situations concrètes. Pour chaque individu et pour chaque relation qui relie deux individus, on exprime les deux formes suivantes :

a : C où C est un concept défini et a est un individu.
(a1, a2) : R où R est un rôle défini et a1, a2 sont deux individus définis.

Phase 4 : Implémentation

L'ontologie obtenue dans la phase précédente est appelée une ontologie formelle. Le but de l'étape d'implémentation sera donc de coder l'ontologie formelle en OWL qui dispose de fonctionnalités sémantiques plus riches que ses prédécesseurs comme RDFS et DAML+OIL. A la fin ce cette phase, on aura une ontologie opérationnelle.

Afin de faciliter le processus de codification, les auteurs de [Hamamet al 04] ont utilisé PROTEGE OWL version 3.1.1 disposant d'une interface modulaire, développée au « Stanford Medical Informatics » de l'Université de Stanford7, permettant l'édition, la visualisation, le contrôle (vérification des contraintes) d'ontologies, issu du modèle de frames et contient des classes (concepts), des slots (propriétés) et des facettes (valeurs des propriétés et contraintes), ainsi que des instances des classes et des propriétés.

Phase 5 : Tests et évolution

Cette phase consiste à exploiter les services d'inférence fournis par la logique de description afin de supporter le processus de construction et d'améliorer la qualité de l'ontologie [Hamamet al 04]. Pour ce faire, les auteurs ont proposé l'utilisation de l'outil RACER, un système de la logique de description. Ce dernier, permet de lire un document au format OWL (ontologie OWL) et de le représenter sous forme d'une base de connaissances LD et de fournir des services d'inférence pour les niveaux TBOX et ABOX.

Cette phase sert aussi à suivre l'évolution de l'ontologie, c'est-à-dire les nouveaux concepts à ajouter dans la partie terminologique (TBOX) de l'ontologie. Une classification a lieu chaque fois qu'une définition de concept est nouvellement créée. Le mécanisme de raisonnement de base des logiques de description est la classification de concepts. Elle est réalisée par un algorithme de classification, appelé « le classifieur ». Le classifieur utilise la description d'un nouveau concept pour le placer à l'endroit correspondant dans la hiérarchie. Afin de trouver la place appropriée au nouveau concept, l'algorithme de classification détermine les relations de subsomption entre ce concept et les autres; Ces relations peuvent être spécifiées directement, trouvées par transitivité ou calculées à partir de la sémantique des conditions des rôles. La recherche de la place correcte pour le nouveau concept comporte trois étapes :

- La recherche des subsumants les plus spécifiques SPS (concepts qui subsument le concept à classer et dont les fils ne le subsument pas)

- La recherche des subsumés les plus généraux SPG (concepts subsumés par le concept à classer et dont les pères ne sont pas subsumés par lui).
- Insertion du nouveau concept dans la hiérarchie.

5.2.2 Limites de l'approche proposée

Vu que la spécification et la mise en œuvre d'une architecture basée sur les ontologies visant à intégrer les applications de l'entreprise est une tâche complexe, les auteurs de [Guergour07] se sont limités seulement, au développement de l'ontologie d'application qui s'inscrit dans le cadre de l'EAI. De ce fait, ils ont dégagé plusieurs limites et perspectives, en voici les plus importantes:

- **Scénarios de communication incomplets :** Les scénarios de communication proposés sont incomplets et dénués de toutes informations détaillées sur l'acheminement des messages entre les applications. L'objectif, donc, derrière la proposition des scénarios de communication est de montrer l'effet de l'ajout des ontologies au sein du système d'intégration et d'aider à dégager les termes de l'ontologie afin de faciliter sa construction. Par conséquent, des enrichissements et des améliorations doivent être apportés à ces scénarios afin de garantir la réactivité et la flexibilité du système d'intégration, et d'avoir une interopérabilité sémantique entre ses applications.

- **Etendre le processus de construction d'une ontologie d'application à d'autres usages** : se baser sur les phases du processus proposé afin d'aboutir à un autre (par adaptation) qui sera adéquat et conforme à différents domaines autres que l'EAI.

- **Suivre l'évolution de l'ontologie :** l'ontologie d'application est construite, il est nécessaire de suivre l'évolution des concepts c.-à-d. les nouveaux concepts à ajouter dans la partie terminologique (TBOX) de l'ontologie. Le résultat de cette étape sera une nouvelle ontologie avec une nouvelle hiérarchie de concepts. Cet aspect devra être étendu par l'utilisation du raisonnement par classification qui est l'un des mécanismes de raisonnement de base pour les logiques de description.

5.2.3 Discussion

Par analogie à l'approche générale citée précédemment, nous remarquons que les sept étapes du processus général se résument dans les deux premières phases du processus proposé dans [Guergour 07]. En effet, la phase de spécification du processus proposé englobe les deux premières étapes du processus général ; et la phase de conceptualisation du processus proposé regroupe les autres étapes du processus général.

Nous constatons ainsi que la phase de formalisation a été rajoutée dans le processus proposé afin de faciliter une représentation ultérieure de l'ontologie dans un langage complètement formel et opérationnel. Le choix a été porté sur le formalisme de représentation de la logique de description en s'appuyant sur sa syntaxe de type SHIQ qui présente une logique de description très expressive et qui offre un certain nombre de constructeurs pour décrire les concepts.

6 Ontologies et multimédia

En plus de leur construction à partir de texte, les ontologies sont aussi construites pour des données multimédia [Dalakleidi et al, 2011] telles que les images. Cette catégorie d'ontologie présente un grand avantage dans des domaines aussi variés que divers à savoir la médecine, la biologie [Campedel et al 08], la géographie, le web [Rouquet et al, 2010], etc. En effet, plusieurs travaux ont été effectués dans ce cadre ; à titre d'exemple, en biologie, les biologistes ne disposent pas à l'heure actuelle de protocole d'annotation, de classification ni d'archivage, certains travaux traitent ce problème. Dans la suite, nous nous intéressons aux données de type « image » afin de mettre en évidence leurs aspects de conceptualisation en vue de construire une ontologie « image ».

6.1 Analyse de l'image et conceptualisation

- **Les concepts**

Dans la littérature, loin des problématiques ontologiques, les experts en traitement d'images tentent de produire un codage du contenu sémantique de ces images [Campedel et al 08]. La méthode la plus courante de codage est celle utilisée dans [Barnard et al 03], [Duygulu et al 02] et [Jeon et al 03] : elle consiste à segmenter les images en régions (visual tokens), et à extraire pour chaque région, un vecteur de caractéristiques (spectrales, textuelles, formes, etc.) qui la représente. Ces représentations des régions sont ensuite quantifiées par un algorithme de clustering (en général les k-means) ; les groupes de régions (clusters) obtenus sont appelés *blobs*. Ainsi, chaque blob a une étiquette qui permet de l'identifier ; l'ensemble des étiquettes produites est assimilé à un dictionnaire visuel pouvant être utilisé pour décrire le contenu de l'image. Une variante de cette méthode de codage est de découper l'image en « *imagettes* » en utilisant une grille régulière, plutôt que de la segmenter. Ainsi nous voyons émerger des concepts possédant une description numérique et un identifiant (l'étiquette).

- **Les relations**

Caractériser ensuite les relations entre blobs (*mots visuels*) n'est pas simple. Le plus souvent, cela se fait au niveau sémantique, i.e. après association à des mots. Par exemple dans [Barnard et al 03] et [Glotin et al 05], les auteurs utilisent cette caractérisation de l'image pour annoter les images naturelles. L'image codée et le texte de l'annotation sont groupés pour produire une représentation caractérisée par une distribution jointe reliant les images et les mots. [Duygulu et al 02] utilisent un *modèle de traduction* pour associer un mot à une région individuelle de l'image. Ce modèle se comporte comme un *lexique* qui, sachant les mots dans une langue, les prédirait dans une autre langue [Campedel et al 08].

Il est cependant intéressant de tenir compte des dépendances entre les mots visuels. L'idée est alors d'exploiter les relations syntaxiques entre les mots visuels pour déterminer la sémantique de l'image. En effet, les relations spatiales constituent un élément essentiel des descriptions d'agencement entre les régions d'une scène et sont donc utiles à un grand nombre de tâches liées à la reconnaissance des formes, la vision par ordinateur, les systèmes d'information géographique (SIG), et plus particulièrement l'interprétation des scènes.

Freeman [Freemen 75] a étudié l'ensemble de ces relations pour la langue anglaise et a proposé une liste de treize relations spatiales de base pour décrire la position relative de deux objets dans un espace bidimensionnel. Dans la littérature, ces relations sont généralement classées en trois familles : relations *topologiques*, relations *métriques* (ou distances) et les relations *directions*. On rajoute souvent à ces trois familles les relations *morphologiques* ou encore les relations de *symétrie*. Pour représenter ces relations spatiales, deux types de modèles sont utilisés : les modèles de représentation **qualitative** souvent liés à la logique formelle ou à la classification des intervalles temporels proposée par Allen [Allen 83], et les modèles de représentation **quantitative** plutôt rattachés à la logique floue ou aux probabilités. Bloch [Bloch05] propose une présentation détaillée de ces approches floues. Plus encore, [Hudelot et al 06] décrit une ontologie des relations spatiales, exploitée en collaboration avec un formalisme flou.

6.2 Exploitation d'un consensus

L'idée proposée par les auteurs de [Campedel et al 08] est de démontrer qu'il est possible d'identifier des concepts sémantiquement pertinents, à partir de l'image et des attributs extraits. Le principe repose sur la production d'un consensus à partir de différents points-de-vue d'analyse de l'image. N'est ce pas exactement notre façon humaine de procéder ? Nous avons chacun une perception du monde différente et pourtant, par le biais d'une communication, nous confrontons nos vues et définissons les concepts, éléments stables, qui régissent notre monde. En pratique, ces *points- de- vue* sont produits par des classificateurs non supervisés : des algorithmes de clusterisation.

L'idée du consensus est d'identifier l'information commune portée par les différents résultats de classification pour ensuite en analyser la redondance et la complémentarité.

Combiner des classificateurs n'est pas nouveau, en particulier dans la communauté de la classification supervisée. Les objectifs sont généralement d'améliorer des performances de classification, d'identifier des résultats communs à plusieurs algorithmes, de classifier des données partiellement étiquetées ou des données distribuées, etc. Cependant, lorsqu'il s'agit de classification non supervisée le problème n'est pas simple (pas de classes prédéfinies ni mesures d'erreurs bien établies) et la littérature est bien moins fournie.

Pour mieux illustrer cette notion de classification, les auteurs de [Campedel et al 08] ont proposé une méthode de classification non supervisée et ont démontré son efficacité sur une image PELICAN de Toulouse à très haute résolution (à 1m de résolution). L'algorithme *LSEC* qu'ils ont utilisé opère en quatre étapes, après application de l'algorithme, ils ont montré comment effectuer une fouille de données et ils ont terminé par des résultats expérimentaux. Deux conclusions assez intéressantes ont été tirées : (1) les clusters produits étaient beaucoup plus proches d'une interprétation sémantique que les résultats des clusterisations individuelles ; et (2) les relations identifiées entre clusters étaient elles-aussi très informatives et sémantiquement pertinentes. Bien entendu, la qualité du résultat dépend des attributs extraits de l'image. Dans le cadre de l'imagerie à très haute résolution (inférieure au mètre), les *bonnes caractéristiques* ne sont pas vraiment connues ; elles reposent à priori plus sur des descripteurs géométriques et des relations spatiales structurantes.

6.3 Applications des ontologies multimédia

Dans cette section, nous illustrons l'utilisation des ontologies multimédia dans les domaines de la biologie et de l'imagerie satellitaire

6.3.1 Le domaine de la biologie : « Les réseaux de neurones »

Chez les biologistes, le besoin se fait de plus en plus ressentir de disposer de protocole d'annotation, de classification et d'archivage. Cependant, Il n'est pas toujours aisé de mettre au point des outils automatiques d'annotation d'images, notamment lorsque les images sont complexes et hétérogènes. En effet, les images sont stockées sous divers formats et sur divers supports (DVD, disque dur amovible, micro ordinateur, etc.). De ce fait, la seule façon d'accéder à une image particulière est de se rappeler où elle a été stockée ou de parcourir tous les supports. L'absence de classification entraîne une impossibilité d'effectuer une recherche automatique parmi les images (par exemple pour exhiber les images ayant un ensemble précis de caractéristiques). L'information contenue dans ces images n'est donc pas ou peu exploitable.

Le travail présenté dans [Da Costa et al 07] est destiné à la catégorie des biologistes qui ont pour objectif de comprendre la réorganisation des réseaux de neurones cérébraux au cours de la vie post natale. Ils utilisent pour cela le modèle biologique du Xénope. Cet amphibien est très utilisé dans le cadre des recherches dans le domaine de la biologie du développement. Les figures 5.4 et 5.5 représentent un exemple d'images collectées servant pour des études de recherche pour les biologistes.

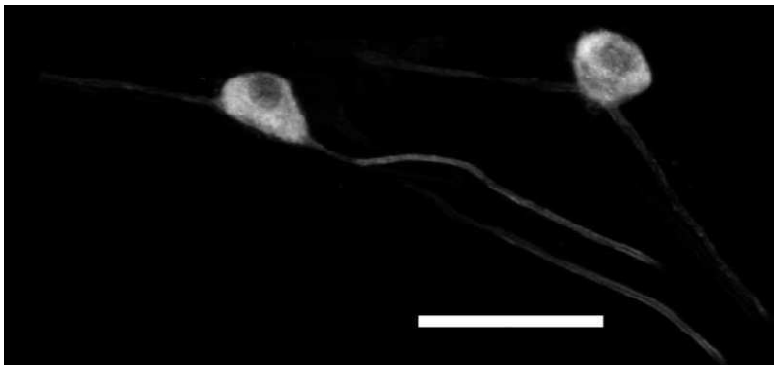


Figure 5.4 *Morphologie de deux neurones bipolaires dans le système olfactif trigéminal. Les neurones ont été marqués avec le carbocyanine DiI. Les cellulesdu corps de deux neurones ainsi que leur axones et dendrites sont distinguables[Gaudin04], [Gaudin et al 05].*

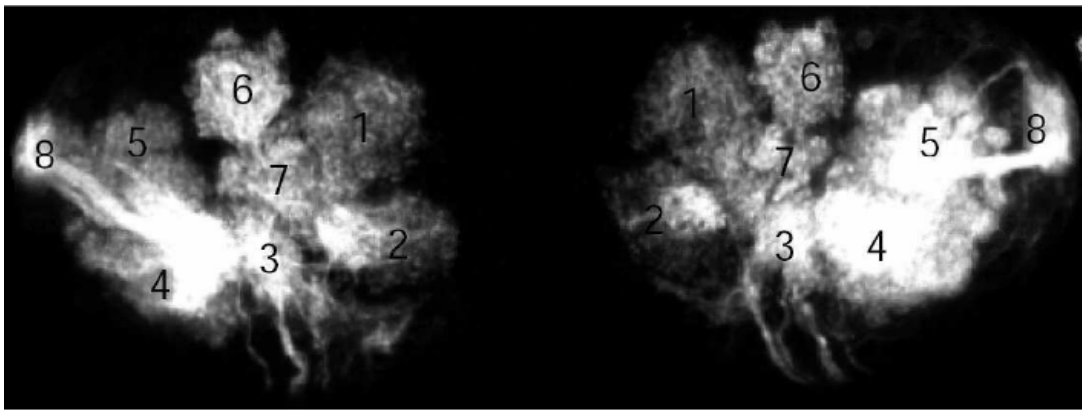


Figure 5.5 *Vue frontale de la projection axonale dans le bulbe olfactif au stade 47 (marqueur DiI) [Gaudin04], [Gaudin et al 05].*

Afin de construire une base de données images à partir de cette collection, il est nécessaire d'annoter les images. Pour cela, les auteurs de [Da Costa et al 07] ont envisagé des approches utilisant des algorithmes automatiques de segmentation pour classifier les images ou reconnaître des zones spécifiques [Lemaitre 07]. À cause de la proximité de certaines zones et de leur enchevêtrement (comme par exemple les zones 3 et 4 de la figure 5.5), les spécialistes en traitement du signal ont estimé que la résolution du problème de la reconnaissance automatique de ce type d'images pouvait prendre énormément de temps ; de la même façon, la reconnaissance de certains types de zones dans les images est un problème à part entière et ne peut pas être généralisé à la collection d'images du fait des problèmes d'hétérogénéité.

Par conséquent, les auteurs de [Da Costa et al 07] ont orienté leur recherche vers l'utilisation de toutes les informations disponibles pour permettre la construction à moindre coût de cette base d'images très spécialisée. L'annotation des images est faite par des techniciens, cela nécessite de pouvoir contrôler les annotations pour garantir qu'elles soient conformes à la connaissance des experts du domaine.

- **Méthodes de construction d'ontologies**

Un des problèmes majeurs qui peut survenir lorsque l'on souhaite représenter la connaissance d'un domaine est que cette connaissance peut ne pas être la même entre différents experts d'un domaine [Da Costa et al 07]. Par exemple, deux experts différents peuvent l'un considérer que la barbille d'un Xénope doit être complètement formée pour qu'elle soit considérée comme existante, l'autre considérer que dès qu'elle est discernable, elle existe.

Dans une optique de partage des connaissances, cela introduit une contrainte sérieuse dans la représentation des connaissances : pour être pertinente, l'ontologie ne peut pas être une représentation exhaustive de la connaissance spécifique d'un domaine [Da Costa et al 07]. Pour que les raisonnements automatiques s'appuyant sur l'ontologie soient pertinents, il est nécessaire que cette ontologie ne représente que la partie du domaine qui fait consensus [Furst02]. A partir de ce consensus, la construction d'une ontologie de domaine peut alors être mise en œuvre en deux étapes distinctes :

- l'étape d'*amorçage* qui détermine la méthodologie initiale permettant d'assembler les différents concepts de l'ontologie ;

- l'étape de *raffinement* qui détermine la méthodologie qui sera utilisée pour obtenir une ontologie qui sera considérée comme un consensus à la fois entre les experts du domaine d'une part, et entre les experts du domaine et les experts en représentation de connaissance, d'autre part.

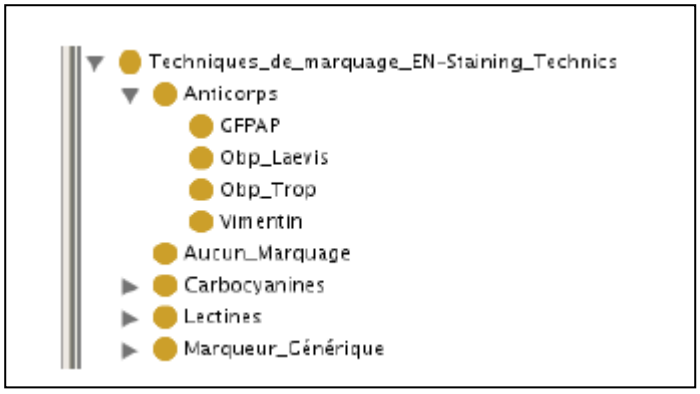


Figure 5.6. *Le concept Anticorps peut être raffiné*

Etape 1 : *Amorçage de la construction d'une ontologie*

Il existe trois familles d'approches présentées dans [Noy et al 00] pour amorcer la construction d'une ontologie :

a- les méthodes descendantes partent des concepts les plus généraux du domaine de discours pour les raffiner. Dans le domaine des Xénopes, une telle approche consiste à démarrer de concepts de haut niveau tels que "*techniques de marquage*", qui sont raffinés en concepts fils (tel que *Anticorps*), ces concepts fils pouvant eux même être raffinés [Da Costa et al 07], voir figure 5.6.

b- les méthodes ascendantes partent des concepts les plus spécifiques et ont pour principe de les regrouper sous des concepts plus généraux. Dans le domaine des Xénopes, cette approche revient à lister des concepts tels que *Ebop*, *Nerf Trigéminal*, *Nerf Olfactif*, *Buble Olfactif Principal*, puis à regrouper ces concepts sous des concepts plus généraux (les *nerfs*, le *bulbe olfactif*, etc.)

c- les méthodes combinées avec lesquelles, les concepts saillants du domaine de discours sont exhibés puis ils sont généralisés ou spécialisés selon les principes des méthodes ascendantes ou descendantes.

Etape 2 : *de l'amorçage à une ontologie opérationnelle : le raffinement*

L'ontologie telle qu'elle existe après une phase d'amorçage est perfectible [Da Costa et al 07]. En effet, il est possible que la connaissance du domaine exprimée par les experts du domaine ne soit pas reflétée de façon fidèle ni exhaustive (dans la limite du consensus sur le domaine). Il est donc nécessaire de raffiner cette ontologie, et de vérifier par des discussions entre les experts du domaine et les spécialistes en représentation des connaissances que l'ontologie décrit correctement le domaine. Cette vérification nécessite la mise en place de cycles de discussion entre les experts du domaine et les spécialistes en représentation de connaissances [Sure 02] ; l'ontologie initiale puis les versions modifiées servent de base à ces discussions.

- **Construction de l'ontologie pour la base de données images Xénope**

Les méthodes ascendantes et descendantes de construction d'ontologie présentent un inconvénient: elles limitent le travail à des opérations de généralisation ou de spécialisation. Le but des auteurs de [Da Costa et al 07] est de conserver une certaine souplesse et de pouvoir généraliser et spécialiser des termes durant la construction de l'ontologie ; c'est pourquoi leur approche pour le début de la construction de l'ontologie était une approche *combinée*.
Le premier pas de l'approche proposée est un travail avec les biologistes pour tenter de dégager les concepts saillants dans le domaine des Xénopes. À partir des informations recueillies et dont une partie est présentée dans le tableau 2, une liste d'une trentaine de concepts a été mise au point, ce qui a permis de proposer un premier niveau à la racine de l'ontologie.

Les barbilles sont présentes sur les têtards mais pas à tous les stades. (Les barbilles arrivent au stade 44 et disparaissent au stade 61)
à partir du Stade 47 : Cavité olfactive formée d'une seule cavité. Organe voméronasal observable Au niveau du bulbe on distingue 7 zones dans le bulbe olfactif principal tandis que la zone 8 constitue le bulbe olfactif accessoire. Les zones 1,5,6, sont en position positions dorsales, les zones 2,3,4 sont en position ventrales. La zone 8 est visible à la fois en position ventrale et dorsale. Toutes les zones ne sont visibles que selon l'axe d'observation antero-postérieur. La zone 7 n'est visible que dans cet axe d'observation. La zone 7 peut être subdivisée en 3 sous zones 7a, 7b, 7c, et la zone 3 en 2 sous zones 3e et 3i. Les zones SIAlpha, SIBeta, SIGama, et SIDelta sont également visibles. Les tailles des glomérules sont comprises entre 13 micromètre et 17 micromètre à partir du Stade 51 : Apparition de la cavité médiane (sytème olfactif aquatique)
à partir du Stade 52 : Les neurones de la cavité médiane atteignent le bulbe olfactif
à partir du Stade 59 : les axones de la cavité principale disparaissent du bulbe ventral
Adulte : stades 66 et ultérieurs

Tableau 5.2. *Première ébauche de règles du domaine par l'expert du domaine. Le découpage selon les stades exhibe déjà l'importance de délimiter chaque règle à un ensemble précis de stades. À ce stade, les règles sont encore exprimées avec des complexités hétérogènes.*

Pour ce premier niveau, l'idée est de dissocier les aspects de l'ontologie utiles pour les annotations des aspects représentant la connaissance du domaine. Après avoir cerné les concepts importants du domaine, l'expert en représentation des connaissances établit une grille que l'expert du domaine va remplir en indiquant quelles sont les implications des métadonnées des images, et dont un extrait est présenté dans le tableau 3.

	Concept			Implication			
	Stade	marqueur	moyen d'observation	EXISTE	NEXISTE PAS	VISIBLE	NON VISIBLE
1	47				zone 9		
2	47				glomérules dans zones 1,3,7		
3	47			zone 8			
4	47			zone 7a			
5	47			zone 7b			
6	47			zone 7c			
7	47			Si alpha			
8	51-57	SBA	confocal	zone 9			zone 9
9	57-59	SBA	confocal	zone 5		zone 5	
10	57-59	SBA	confocal	zone 3e		zone 3e	
11	57-59	SBA	confocal	EBOP		EBOP	

Tableau 5.3. *Certaines règles véhiculent une sémantique simple : La zone 9 n'existe pas au stade 47 ; d'autres sont plus complexes : dans les stades 51 à 57, même si la zone 9 existe, elle ne sera pas exhibée par un marqueur de type SBA en imagerie confocale.*

La construction de l'ontologie commence alors, avec deux types de branches. Les branches qui contiennent la connaissance du domaine et les branches qui contiennent la connaissance technique. La figure 5.7 montre la hiérarchie des branches.

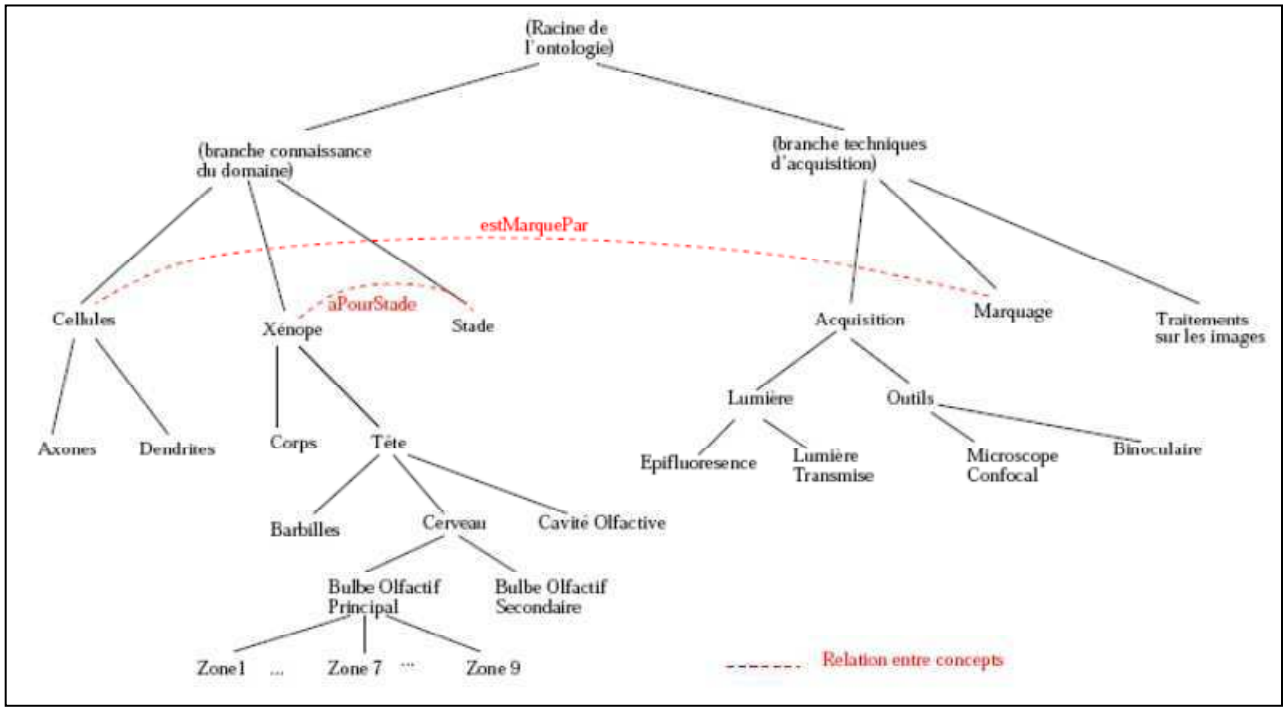


Figure 5.7. *Hiérarchie entre quelques concepts de l'ontologie et relations mises en place entre les différents concepts.*

Les branches qui contiennent la connaissance du domaine incluent les branches suivantes :

- *Cellules* : différents types de cellules connus dans le domaine des Xénopes (Axones, Cellules gliales, . . .) ;
- *Stades d'évolution* : stades d'évolution des Xénopes (Stade 1, Stade 2, . . .) ;
- *Xénopes*: structure biologique des Xénopes (Corps, Tête, Barbilles, Cerveau, Nerfs, etc.).

Les branches qui contiennent la connaissance technique englobent les branches suivantes :

- Techniques *d'acquisition* :
  différentes lumières (épifluoresence, lumière directe, etc.) ;
  différents outils (microsope confocal, loupe binoculaire, etc.).
- Techniques de *marquage* (Anticorps, Lectines, etc.) ;
- *Traitements* sur les images (Rotation, Reconstructions 3D, etc.) ;

Les concepts qui n'avaient pas leur place à la racine ont été placés dans les branches de l'ontologie qui les concernent ; par la suite, une première ossature est obtenue en raffinant et généralisant les concepts [Da Costa et al 07].

Des relations entre les branches de l'ontologie relatives aux techniques d'acquisition et les branches de l'ontologie relatives à la connaissance du domaine ont été mises en place. Par exemple la relation *estMarquePar* a pour domaine la branche *Marquage* et pour champ la branche *Cellules*. La figure 5.7 présente un extrait de l'ontologie Xénope et des relations qui nous ont permis de définir ensuite des classes (l'équivalent dans le langage d'implémentation utilisé - Java - des concepts de l'ontologie) qui seront utilisées lors de l'annotation : *Existe*, *NExistePas*, *Visible* et *NonVisible*.

Les auteurs ont évité de s'appuyer sur la théorie du monde clos. Ils se sont placés donc dans la théorie du monde ouvert basé sur le principe suivant : « si une assertion ne peut pas être prouvée, elle est considérée comme *inconnue* (au lieu de faux dans le *monde clos*) ». Cela évite qu'un élément qui n'est pas considéré comme *Visible* ne soit considéré comme *NonVisible* : il est possible que la connaissance du domaine ne permette pas de déterminer le caractère *Visible* ou *NonVisible* d'un élément. Avec la théorie du *monde ouvert*, chaque appartenance d'un élément à l'une des classes (*Existe*, *NExistePas*, *Visible* et *NonVisible)* ne peut pas être remise en question puisqu'elle dépend directement de la connaissance exprimée dans l'ontologie et des propriétés de l'élément en question.

En fonction des informations connues sur les différentes instances, on souhaite définir quand cela est possible l'appartenance à la classe *Visible* ou à la classe *NonVisible*. Comme un élément ne peut être considéré à la fois *visible* et *non visible*, ces deux classes sont définies comme étant disjointes l'une de l'autre. De la même façon, les classes *Existe*, *NExistePas* sont disjointes. Le tableau 4 récapitule les incompatibilités d'appartenance d'un élément aux classes correspondant aux différents concepts.

	EXISTE	NEXISTE PAS	VISIBLE	NON VISIBLE
EXISTE		X		
NEXISTE PAS	X		X	
VISIBLE		X		X
NON VISIBLE			X	

Tableau 5.4. *Incompatibilité entre les classes des différents concepts.*

A partir des concepts retenus du domaine et des règles de compatibilité, le spécialiste en représentation de connaissance établit un tableau afin que l'expert du domaine puisse représenter la connaissance de son domaine d'une manière plus formelle.
Un certain nombre de règles est présenté dans le tableau 5.2. L'ontologie comporte deux cent quatre-vingt-dix règles. Sur l'ensemble de ces règles seul un sous ensemble limité ne contient que des règles atomiques (i.e. : ne comportant qu'un seul critère). Le spécialiste en représentation des connaissances a donc guidé l'expert du domaine dans la rédaction des règles pour obtenir des règles ayant à la fois une réalité visuelle sur les images et une sémantique forte.

6.3.2 Les ontologies et l'imagerie satellitaire

L'objectif derrière l'utilisation des ontologies dans l'imagerie satellitaire est de permettre aux experts (photo-interprètes) d'exploiter le grand nombre des images satellitaires récoltées [Campedel et al 08]. En effet, le nombre et la résolution des images satellitaires ne cessent d'augmenter. Il en résulte un accroissement phénoménal de la quantité d'information disponible mais non exploitable en totalité par des photo-interprètes. Ces experts ont appris à exploiter des résultats d'analyse des images (appariement, segmentation, classification, plus ou moins automatiques ...) afin d'extraire une information utile dans un but applicatif bien défini : par exemple, l'urbaniste s'intéresse à la morphologie des villes (distribution des types d'habitation, disposition des réseaux de transport, espaces verts, ...) alors que l'agronome étudie l'évolution des cultures (état des cultures, suivi des tailles des parcelles exploitées, ...). Les images doivent être bien choisies pour permettre la mise en évidence des objets d'intérêt de l'application choisie. Il est donc nécessaire de développer des processus (semi-) automatiques facilitant l'indexation et la récupération du contenu de ces images, en termes de concepts et relations entre ces concepts. Cette indexation repose sur des processus d'analyse automatiques des images et des connaissances à priori sur leur domaine d'usage.

- **Les projets du domaine satellitaire**

Lorsque l'on considère la façon dont sont exploitées les images satellitaires, on observe un effort sémantique et ontologique important, en particulier cartographique. Mais les cartes ne sont qu'un type de représentation du contenu d'une image, valable à un temps donné.

Dans le domaine satellitaire, le projet Corine Land Cover [Campedel et al 08] avait pour objectif de produire un inventaire biophysique de l'occupation des sols européens. Des cartes de couverture des sols sont générées à partir d'images satellitaires (Landsat) qui sont interprétées visuellement par des experts, s'aidant de données exogènes (photographies aériennes, cartes topographiques, etc). Des centaines de milliers de km2 ont été décrits et les cartes sont disponibles gratuitement. Afin de rendre consistantes ces cartes, les experts disposent d'une nomenclature précise. Celle-ci est composée de 45 concepts organisés hiérarchiquement sur trois niveaux. Chaque concept est précisément décrit, ainsi que ses liens

et oppositions avec d'autres concepts (par exemple, le concept *"tissu urbain discontinu"* est défini en lui-même et par opposition au *"tissu urbain continu*"). Peut-on parler d'ontologie pour cette nomenclature détaillée ?

Dans différents domaines d'application, on retrouve cette envie de décrire les connaissances par un nombre fini de concepts décrits précisément et organisés. Parmi les projets marquants, on trouve le projet *ANR FoDoMUST* (Fouille de Données MUltiStratégies) et un projet de recherche européen et suisse appelé *Townontology*. Ces deux projets concernent la mise au point d'ontologies urbaines. Dans le premier cas, l'ontologie est conditionnée par les caractéristiques spectrales des images satellitaires et doit nourrir un SIG (Système d'Information Géographique) avec des résultats sur la localisation et l'identification des éléments du tissu urbain (surfaces minéralisées, végétation, eau). Dans le second cas, la description ontologique repose sur les objets physiques qui composent la matérialité de la ville ainsi que les événements historiques déterminant la composante temporelle des transformations du territoire ; il n'est pas fait référence aux images satellitaires. La question posée toujours par les auteurs de [Campedel et al 08] est évidente : combien d'ontologies différentes faut-il donc définir pour arriver à une conceptualisation acceptable par le plus grand nombre de la notion de "urbain" ?

- **Le projet DAFOE**

Afin de progresser dans l'élaboration d'une ontologie adaptée à leurs besoins, les auteurs de [Campedel et al 08] ont participé activement au projet DAFOE afin d'améliorer leur processus d'indexation des images satellitaires et les résultats obtenus serviront à valider la plate-forme développée dans le cadre du projet DAFOE [Charlet et al, 08].

Il s'agit d'un projet ANR[i] débuté en janvier 2007, sous le pilotage de JeanCharlet (INSERM). La plupart des outils développés autour des ontologies permettent de les construire en précisant comment représenter les concepts et formaliser leur sémantique, ils ne précisent pas comment trouver les concepts ni comment expliciter leur signification. L'objectif du projet est de proposer une méthode complète associée à une plateforme technique pour concevoir des ontologies, de la modélisation à partir du domaine à leur évolution en passant par leur formalisation et exploitation. S'appuyant sur les acquis de travaux antérieurs, à la fois issus des partenaires et de la littérature du domaine, le projet visait à prendre en charge la modélisation sémantique des concepts ontologiques pour motiver et justifier les représentations formelles qui seront utilisées et aussi pour en faciliter la révision.

Le projet DAFOE a permis aux chercheurs : (1) d'enrichir leurs annotations sémantiques sur les images manipulées, (2) de raisonner à partir de concepts sémantiques et non sémantiques et (3) d'offrir une interface d'interaction avec l'utilisateur pour apprendre et affiner ses propres centres d'intérêt.

En effet, afin d'exploiter au maximum les connaissances qu'ils peuvent acquérir à partir des images (même si elles ne peuvent être associées directement à une étiquette sémantique) et des usages faits de ces images, [Campedel et al 08] avaient défini formellement deux ontologies: l'une issue de l'analyse automatique des images et l'autre issue des besoins applicatifs des images satellitaires. A l'aide de techniques d'apprentissage, ces deux

ontologies étaient mises en relation pour faire le lien entre les experts du traitement automatique des images et les experts de leur interprétation visuelle.

6.4 Une méthode interactive et incrémentale de construction de concepts

Dans *l'Institut de Recherche en Informatique de Toulouse*, Alexandre NOUVEL et Patrice DALLE ont proposé une approche interactive pour la réalisation d'applications de traitement d'images en s'appuyant sur la définition d'une ontologie liée aux images [Nouvel et al 02]. Dans cette approche, l'utilisateur associe les résultats d'enchaînements d'opérateurs à des concepts du modèle de la scène, au lieu de procéder par cycles "essai-erreur" d'opérateurs (choix, paramétrage et exécution d'un opérateur, évaluation de ses effets, puis ajustement ou correction). La définition des concepts est conduite par l'utilisateur qui les manipule directement, de même que leurs instances (des entités produites par les opérateurs).

La définition incrémentale des concepts (par spécification, extraction et nomination d'entités ou par spécialisation et généralisation d'autres concepts) revient à construire une ontologie décrivant le domaine d'application. L'intérêt de cette démarche est de masquer les fonctionnalités du traitement d'images, de raisonner en termes de concepts du domaine d'application, et de se placer dans une logique d'interprétation. Cette approche offre une vision plus claire sur la définition d'une ontologie lorsqu'il s'agit de données multimédias (en particulier les images). Nous présentons dans ce qui suit quelques détails liés à la démarche proposée dans [Nouvel et al 02].

- **Description des bases de la démarche**

Un concept représente une notion visuelle pour l'utilisateur, celle-ci peut être liée ou non à des éléments du domaine des connaissances de l'utilisateur sur le contenu des images qu'il cherche à analyser. Le concept peut représenter une notion visuelle pertinente bien que sans rapport direct avec le contexte de l'application que l'utilisateur cherche à résoudre : par exemple, *« objet clair », « objet allongé »* ou *« objet clair allongé qui traverse l'image ».* Un tel concept est qualifié de "générique" [Nouvel et al 02].

Un concept peut aussi être associé à une notion du domaine de compétence de l'utilisateur. Ceci peut être fait de deux manières : par *renommage* d'un concept générique reconnu par l'utilisateur en tant que concept de son domaine (« objet clair allongé qui traverse l'image » pourra être renommé par exemple dans une image de cartes électroniques en « fil traversant l'image », ou bien en « autoroute » dans une image aérienne), ou par construction à partir d'autres concepts étant eux-mêmes génériques ou du domaine de l'utilisateur (par exemple, le concept « lacs » est défini par spécialisation du concept générique « régions de texture ondulée » par adjonction d'une propriété de surface faisant référence au concept du domaine « mer »).

Dans le système développé par [Nouvel et al 02], les données manipulées sont présentées à l'utilisateur sous forme de groupes d'entités. Chaque groupe d'entités est l'instance d'un concept que l'utilisateur peut nommer et réutiliser. Un groupe d'entités est soit le résultat de l'extraction d'informations d'une donnée suite à la définition d'un concept par l'utilisateur (par exemple, une partition en régions de l'image suite à la définition du concept « objets définis par leur texture »), soit le résultat de la sélection de certaines entités dans un autre groupe d'entités, selon une propriété spécifiée par l'utilisateur (par exemple, le sous ensemble des

régions dont le degré de compacité est compris dans un certain intervalle). Dans l'interface conçue dans [Nouvel et al 02], les concepts ne sont pas directement manipulables, sauf par le biais d'entités qui les instancient.

Définir un concept consiste donc pour l'utilisateur à caractériser un ensemble de transformations apportées par une chaîne d'opérateurs sur des données. Il doit indiquer pour cela les propriétés sur lesquelles les transformations doivent porter et les caractéristiques souhaitées pour la donnée à produire. Les chaînes d'opérateurs sont masquées à l'utilisateur, mais peuvent être réutilisées par le système pour instancier des concepts à partir de nouvelles données, afin de tester si une définition de concept correspond bien à la notion que l'utilisateur voulait représenter par ce concept.

Pour produire des entités, l'utilisateur dispose d'une liste des concepts déjà produits. Trois catégories d'opérations sont possibles sur un concept :

- faire produire par le système un ensemble de nouvelles entités par structuration ou déstructuration d'un groupe d'entités ;
- modifier la définition du concept en accédant aux propriétés ayant servi à spécifier ses caractéristiques : généralisation ou spécialisation du concept ;
- construire la définition du concept par union (disjonction) de concepts existants.
- Enfin, une opération de nommage de concept est proposée ; elle permet de construire le lexique associé au domaine et de réutiliser par la suite la définition du nom du concept ou des mesures portant sur les entités qui le composent

7. Conclusion

Tout au long de ce chapitre, nous avons étudié les différentes notions et concepts liés aux ontologies. Nous avons commencé par donner quelques définitions concernant les ontologies. Nous avons cité aussi les éléments de base sur lesquels s'appuie une ontologie. Nous avons présenté également l'approche générale de construction d'une ontologie. Cette approche, selon les besoins et le domaine en question, peut être améliorée à travers la modification des étapes qui la constituent ou même l'ajout de nouvelles étapes.

Un type particulier d'ontologie a été pris en compte aussi dans ce chapitre, il s'agit des ontologies *multimédias* traitant des données complexes, particulièrement les images. Des travaux réalisés pour ce type de données ont été décrits. Vu le type de données que nous traitons (données multimédias), ce type d'ontologies sera utilisé dans notre approche que nous décrivons dans le prochain chapitre.

[i] Agence Nationale de la Recherche

Chapitre 6

Démarche de conception d'EDM proposée

1. Introduction

La construction d'un entrepôt de données passe par trois phases principales à savoir : la planification, la modélisation conceptuelle et la maintenance. Cependant, la phase de conception est une phase très importante dans les projets d'entreposage de données. En effet, différentes études ont montré que le manque ou l'absence d'une phase de modélisation conceptuelle est une cause possible d'échec des projets d'entreposage. Pour remédier à ce problème, il est donc nécessaire d'utiliser une vraie phase de conception afin de réduire le risque de l'échec.

Plusieurs approches de conception ont été proposées. Ces dernières peuvent être classifiées en trois catégories : les approches directes, les approches traditionnelles et une nouvelle catégorie englobant les approches ontologiques. L'apport de cette dernière catégorie apparait dans l'utilisation des ontologies qui permettent d'assurer une *intégration* sémantique des données*, provenant de sources hétérogènes,* de manière automatique [Nabli et al, 09]. L'étude faite sur ces approches nous a fait ressentir un grand besoin de se pencher vers les approches de conception des systèmes d'intégration vu la grande ressemblance entre ces derniers et les EDs. De plus, ces approches d'intégration nous permettent de résoudre le problème d'intégration rencontré dans les approches de conception de projets d'entreposage.

D'après l'étude que nous avons faite, peu de travaux ont été réalisés dans ce contexte, nous citons [Romero et al 07] qui présente une méthode de conception dérivant le schéma conceptuel de l'entrepôt uniquement à partir des sources de données. Un autre travail [Khouri09] prend en compte les sources de données et les besoins des utilisateurs. En effet,

l'implication des besoins des utilisateurs dans la conception d'un ED est très importante ; elle a été prouvée par diverses études faite dans ce contexte.

Partant de cela, notre contribution consiste à proposer une approche de conception d'un ED prenant en compte les sources de données et les besoins des utilisateurs en même temps. L'apport de notre approche par rapport aux approches existantes réside dans l'utilisation de données de types complexes qui sont les données multimédia.

De ce fait, notre démarche se base sur les approches de conception des systèmes d'intégration [Khouri 09]. La démarche proposée consiste principalement à faire une analyse de ces approches par rapport aux problèmes rencontrés dans les projets d'entreposage ainsi que ceux liés à complexité des données multimédias. Notre but est d'éviter au maximum les problèmes en question.

2. Analyse des approches par rapport aux problèmes des EDs et aux données multimédias

Nous présentons ici une analyse des approches de conception d'ED présentées dans le chapitre 2 afin de tirer l'approche appropriée dans notre cas. Cette approche est définie de telle façon à éliminer ou à éviter le maximum de problèmes qui sont liés soit aux approches proposées, soit aux problèmes des EDs ou bien ceux des données multimédias.

D'après l'étude que nous avons faite, nous remarquons que dans le premier type d'approches, qui englobe les approches *directes*, on ne définit même pas une modélisation conceptuelle. Par conséquent, ce type d'approche n'est pas recommandé dans notre cas vu que notre objectif principal est de fournir une modélisation conceptuelle d'un ED multimédia.

De leur parts, les approches traditionnelles orientées sources sont à écarter aussi parce qu'elles ne prennent pas en compte les besoins des utilisateurs. Ceci conduit éventuellement au rejet du projet de l'entreposage.

Dans la suite, nous nous intéressons seulement aux approches traditionnelles orientées besoins (les sources de données restent dans tous les cas, le cœur de construction d'un ED) et aux approches ontologiques.

2.1 Analyse par rapport aux problèmes des EDs

Nous nous intéressons seulement aux problèmes qui peuvent influencer la conception d'un ED (le problème de la maintenance d'un ED par exemple n'est pas lié à la conception, donc nous ne le traitons pas). Les principaux problèmes qui nous intéressent sont :

- Des problèmes masqués dus aux systèmes sources ;
- Les données requises ne sont pas toutes collectées ;
- La complexité de l'opération d'intégration des données.

Le premier problème correspond aux problèmes *masqués dus aux systèmes sources*, cela signifie que des problèmes relatifs aux systèmes sources peuvent être cachés et n'être identifiés qu'après des années d'utilisation ou ne sont quelquefois même jamais détectés. Un simple exemple concerne les valeurs nulles. Supposons que le concepteur de l'entrepôt définit un champ qui accepte une valeur nulle et que l'utilisateur a besoin de faire une analyse sur les valeurs de ce champ ; ceci pose un problème.

Pour les approches traditionnelles orientées sources ce problème risque de conduire à des résultats non satisfaisants lors de l'interrogation de l'ED vu que ces approches sont basées essentiellement sur les sources de données.

Pour les approches traditionnelles orientées besoins, le problème peut être résolu grâce au mapping qu'on fait entre les sources et les besoins.

Concernant les approches ontologiques, le problème peut être aussi résolu. En effet, ces approches ont été introduites pour résoudre le problème d'intégration. Pour atteindre cet objectif, elles prennent souvent en compte les besoins des utilisateurs.

Ce premier problème peut être, éventuellement, résolu en rajoutant une analyse supplémentaire des sources par le développeur qui doit décider s'il faut régler le problème au niveau de l'entrepôt de données et (ou) au niveau des systèmes sources.

Concernant le deuxième problème qui est lié à la récolte des données. En effet, parfois les données requises ne sont pas toutes collectées. Si par exemple l'utilisateur désire faire une étude sur un critère et que ce dernier n'est même pas pris en compte par le concepteur, alors l'utilisateur sera bloqué.

Nous remarquons que ce problème ressemble au premier car tout deux concernent les sources de données. Donc les commentaires liés au premier point restent valables ici.

Un autre problème concerne la complexité de l'intégration qui est un problème assez difficile à résoudre et qui constitue en même temps la base d'un ED.

Nous avons vu que parmi les étapes des approches traditionnelles orientées besoins on trouve celle liée au mapping entre les sources et les besoins. Ce mapping nécessite forcément une intégration. Cette intégration est faite souvent par le concepteur pour une éventuelle validation. Cette tâche prend un temps considérablement élevé et un effort supplémentaire vu sa complexité. De ce fait, il est plus intéressant d'automatiser cette tâche.

Les approches ontologiques permettent de résoudre le problème d'intégration. Cependant, ces approches permettent d'assurer une intégration sémantique et d'une manière automatique ce qui représente un grand avantage.

2.2 Analyse par rapport aux données multimédias

Comme l'analyse précédente, cette analyse ne prend en compte que les problèmes qui peuvent influencer la phase de conception d'un ED. Ces problèmes sont :

- Stockage de données volumineuses ;
- L'intégration des données multimédias ;
- Le traitement des requêtes sur les données multimédias.

Le premier problème concerne donc le stockage des données multimédias. En effet, ces données sont volumineuses et occupent un énorme espace de stockage. A notre avis, ceci pose problème lorsque le matériel utilisé pour le stockage et pour le traitement des données est limité en capacité. Mais, de nos jours, le développement technologique a prouvé le contraire.

Ce problème est lié donc au matériel utilisé plutôt qu'aux approches proposées précédemment. L'utilisation d'un matériel sophistiqué permet d'éliminer ce problème.

Le deuxième problème est lié à la complexité de l'intégration que nous avons déjà entamée dans le paragraphe précédent. Cette tâche devient de plus en plus complexe avec les données multimédias. Les approches ontologiques qui sont jugées candidates pour la résolution de ce problème doivent de plus prendre en compte les données multimédias et définir des ontologies dédiées à ce type de données.

Un autre problème concerne le traitement des requêtes sur les données multimédias. Ce traitement nécessite des techniques et des méthodes efficaces.
Pour les approches orientées besoins, le mapping entre les sources et les données nécessite souvent une étape de formalisation des besoins voire une génération de requêtes. Ceci induit un temps de traitement et un effort supplémentaires.
Les approches ontologiques sont confrontées à ce problème aussi. Cependant, l'utilisation des ontologies apporte un gain en temps et en effort [Khouri 09].

2.3 Comment choisir l'approche appropriée ?

En se basant sur l'analyse faite dans la section précédente et sur des travaux effectués dans le domaine des entrepôts de données multimédia d'une part et les ontologies multimédia d'autre part, nous proposons notre approche de conception qui vise à éviter au maximum les problèmes rencontrés précédemment.

Ainsi, notre approche sera définie en s'appuyant sur deux idées principales : la première concerne l'utilisation d'une approche ontologique, et la deuxième consiste à considérer un ED de données comme étant un cas particulier d'un système d'intégration (SI).

En effet, d'après l'analyse précédemment effectuée, nous constatons que l'approche ontologique est la plus appropriée pour notre cas. Cependant, nous prenons aussi en compte les besoins des utilisateurs.

L'approche ontologique a été choisie parce qu'elle permet *la gestion des conflits sémantiques* au moment de l'intégration des données. Ceci présente un grand apport pour la conception d'ED vu que le problème d'intégration est toujours présent dans la conception de ce dernier.

Il existe plusieurs approches d'intégration. Nous ne pouvons pas juger si une approche est meilleure par rapport aux autres car cela dépend des besoins du projet en question.

Nous présentons dans ce qui suit les différentes approches d'intégration. Par la suite, nous analysons ces approches par rapport à nos besoins afin d'opter pour l'utilisation de l'une d'entre elles.

3. Approches d'intégration des données

Nous nous appuyons sur la classification des approches d'intégration proposée par [Nguyen06] qui classifie ces approches selon les trois critères orthogonaux suivants :

- La représentation des données intégrées ;
- Le sens de mise en correspondance entre le schéma global et les schémas locaux, et
- La nature du processus d'intégration.

Pour chaque critère on distingue un nombre d'approches. Nous présentons, dans ce qui suit le principe de chaque approche.

3.1 La représentation des données intégrées

Ce critère permet de distinguer les approches d'intégration selon la manière dont elles stockent les données. On distingue ainsi : l'approche *virtuelle* et l'approche *matérialisée*.

3.1.1 Approche virtuelle

L'approche *virtuelle* stocke les données uniquement au niveau des sources. Les systèmes d'intégration construits selon cette approche, appelés systèmes médiateurs, reposent sur deux composants : le médiateur dont le rôle est de localiser les données pertinentes à une requête d'utilisateur, et l'adaptateur dont le rôle est d'accéder au contenu de ces sources.

Une requête d'utilisateur est formulée selon le schéma global du médiateur. Des vues abstraites décrivent le contenu de chaque source dans les termes du médiateur. Le médiateur sélectionne les sources pertinentes pour la requête en utilisant ces vues. La requête doit être reformulée afin qu'elle puisse être évaluée sur les sources pertinentes (c'est ce qu'on appelle réécriture des requêtes). Le résultat de cette réécriture est un ensemble de sous requêtes posées sur les sources. Les adaptateurs, contenant les correspondances entre les données du schéma global et ceux des sources, traduisent les sous requêtes dans le langage spécifique de chaque source.

Plusieurs systèmes d'intégration ont été développés selon cette approche : projets Tsimmis [Chawathe et al 94], Projet Padoue [Lumineau05], Projet Picsel [Giraldo05].

3.1.2 Approche matérialisée

L'approche *matérialisée* transfère les données des sources au niveau d'un entrepôt (il ya donc duplication des données). L'interrogation des données se fait directement sur les données de l'entrepôt et non sur les sources d'origine. La conception d'un système d'intégration selon une architecture matérialisée est similaire à celle d'un entrepôt de données [Giraldo 05].

Certaines entreprises, pour des raisons de sécurité, préfèrent opter pour une solution d'ED, même si cela implique un coût de stockage et de maintenance des données. Quelques systèmes d'intégration ont été développés selon l'approche matérialisée comme le projet

américain WHIPS [Hammer et al 95], le projet européen DWQ (Data Warehouse Quality) [Jarke et al 97] et le projet français OntoDawa [Nguyen 06].

3.2 Le sens de mise en correspondance entre le schéma global et les schémas locaux

Ce critère permet de distinguer les approches d'intégration selon la liaison entre le schéma global et les schémas locaux des sources à intégrer. On distingue pour ce critère deux approches : *GaV* (Global as View) et *LaV* (Local as View).

3.2.1 Approche GaV

L'approche *GaV* définit le schéma global en fonction des schémas des sources (schéma locaux). Chaque schéma (local ou global) est défini comme étant un ensemble de relations. Les relations globales sont définies comme des vues sur les relations des schémas locaux.

Parmi les systèmes utilisant l'approche GaV, on peut citer les systèmes TSIMMIS [Chawathe et al 94], Momis [Beneventano et al 00], et Whips [Hammer et al 95].

3.2.2 Approche LaV

L'approche *LaV* est l'approche duale de GaV. Elle définit les schémas des sources à intégrer en fonction du schéma global. Les relations des schémas des sources (relations locales) sont définies comme des vues sur les relations globales.

Parmi les systèmes utilisant l'approche LaV, on peut citer les systèmes InfoMaster [Genesereth et al 97], Picsel [Giraldo05], et Information Manifold [Levy et al 96].

Ce deuxième critère est important pour deux problématiques: la *réécriture des requêtes* (réécriture des requêtes utilisateurs sur les sources et construction de la réponse) et la *scalabilité du système d'intégration* (ajout ou suppression de sources de données) [Nguyen06]. L'approche GaV facilite la réécriture des requêtes mais l'approche LaV est plus intéressante pour assurer la scalabilité du système d'intégration.

En effet, une requête utilisateur s'exprime en termes des relations du schéma global, sa réécriture en fonction des schémas des sources, dans une approche GaV, nécessite un simple dépliement des relations utilisées dans la requête par leurs définitions locales. Cette réécriture, dans une approche LaV, devient un problème complexe nécessitant des inférences [Belaid07].

D'autre part, l'ajout d'une nouvelle source de données, pouvant nécessiter une mise à jour du schéma global dans une approche GaV, est facilité dans une approche LaV. Seules les vues des nouvelles sources doivent être rajoutées (sous réserve que le schéma global ait été bien défini initialement).

3.3 La nature du processus d'intégration

Ce dernier critère permet de distinguer les approches d'intégration selon le degré d'automaticité du processus d'intégration. Notons que cette automaticité concerne la résolution ou l'élimination des conflits sémantiques entre les sources durant le processus d'intégration. On distingue les approches *manuelles*, *semi-automatiques* et *automatiques*.

3.3.1 Les approches manuelles

Les premières approches proposées étaient des approches *manuelles*. Ces approches permettent d'automatiser l'intégration des données au niveau syntaxique. Les conflits sémantiques sont gérés manuellement et nécessitent la présence d'un expert humain pour interpréter la sémantique des données. Plusieurs systèmes ont été développés selon cette approche comme les systèmes multi-bases de données, la fédération des bases de données, le système Tsimmis [Chawathe et al 94]. Ces approches manuelles deviennent impraticables lorsque le nombre de sources de données à intégrer est important, ou lorsque les sources évoluent fréquemment.

3.3.2 Les approches semi-automatiques

Afin d'apporter plus d'automatisation dans la résolution des conflits sémantiques, plusieurs travaux se sont tournés vers les ontologies.

Les approches *semi-automatiques* reposent sur des ontologies ***linguistiques*** et permettent d'automatiser partiellement la gestion des conflits sémantiques. Les ontologies linguistiques traitent des termes, et non des concepts. Ceci peut créer des conflits de noms.
Momis [Beneventano et al 00] est un exemple de projet reposant sur l'ontologie linguistique Wordnet (Wordnet est une base de données lexicale développée par des linguistes du laboratoire des sciences cognitives de l'université de Princeton) pour l'intégration des sources.

Notons qu'une ontologie linguistique définit le sens des mots utilisés dans un domaine d'étude. Ces mots sont essentiellement liés par des relations linguistiques telles que la synonymie, l'antonimie, l'hyperonymie, etc.

3.3.3 Les approches automatiques

Les approches *automatiques* consistent à associer aux données des sources une ontologie ***conceptuelle*** qui en définit le sens. Une ontologie conceptuelle traite les concepts d'un domaine donné. Elle est définit comme « *une spécification formelle et explicite d'une conceptualisation partagée d'un domaine de connaissance* » [Gruber 95]. La sémantique du domaine est ainsi spécifiée formellement à travers des concepts, leurs propriétés ainsi que les relations entre les concepts. La référence à une ontologie permet d'éliminer *automatiquement* les conflits sémantiques entre les sources. La connexion entre les sources et les ontologies peut se faire de plusieurs manières [Wache et al 01] : définition des termes de la BD par les concepts ontologiques (Projet Buster [Stuckenschmidt et al 00]), annotation des données (Projet SHOE [Heflin et al 99]), enrichissement de la source par des règles logiques (logique de description) (Projet Picsel [Goasdoué et al 99]), etc.

4. Analyse des approches par rapport à notre cas

Pour le premier critère qui correspond à la représentation des données, il apparaît clairement que le développement d'un entrepôt de données suit une approche *matérialisée*. Bien que cette approche nécessite un énorme espace de stockage vu que nous traitons des données multimédias. Cependant, ce problème peut être résolu car, de nos jours, la technologie ne cesse de nous offrir des dispositifs de stockage assez grand.
De plus, une approche virtuelle nécessite une réécriture de requêtes qui n'est pas facile à réaliser. Cette difficulté croit considérablement dans le cas des données multimédias.
De ce fait, nous nous positionnons pour ce premier critère dans une *approche matérialisée*.

En ce qui concerne la mise en correspondance entre le schéma global et les schémas locaux, nous optons pour une approche *LaV*. Nous avons choisi cette approche pour assurer la scalabilité du système d'intégration. En effet, l'ajout de nouvelles sources de données n'influe pas sur le schéma global. On doit juste définir le schéma local de la source ajoutée en fonction du schéma global. De plus, nous avons écarté l'approche *GaV* (qui facilite la réécriture des requêtes) pour être cohérent avec le premier critère où nous avons choisi une approche matérialisée qui ne nécessite pas la réécriture des requêtes. Donc, L'approche *LaV* convient bien à notre cas.

Pour le dernier critère, notre but est d'automatiser au maximum le processus d'intégration. De ce fait, nous écartons les approches manuelles parce qu'elles n'offrent que l'automatisation de l'intégration syntaxique, celle liée à la sémantique est faite manuellement. Les approches semi-automatiques sont à éloigner aussi car elles permettent d'automatiser partiellement la gestion des conflits sémantiques. De ce fait, nous optons pour une approche *automatique* qui est basée sur une ontologie conceptuelle qui permet d'éliminer *automatiquement* les conflits sémantiques entre les sources en définissant les relations entre les concepts manipulés dans l'ED.

En résumé, d'après l'étude que nous avons faite sur les approches d'intégration ainsi que l'analyse par rapport aux données multimédias, nous nous positionnons dans une approche *ED, LaV, automatique*.

5. Approche proposée

Nous décrivons dans ce qui suit l'architecture générale représentant l'approche proposée. Nous décrivons ensuite les étapes de l'approche. Enfin, nous soulignons les apports de l'approche par rapport aux limites des approches existantes.

5.1 Architecture de l'approche proposée

La figure 6.1 représente l'architecture globale de l'approche proposée. Elle contient les principales étapes qu'il faut suivre pour définir le modèle conceptuel de l'entrepôt.

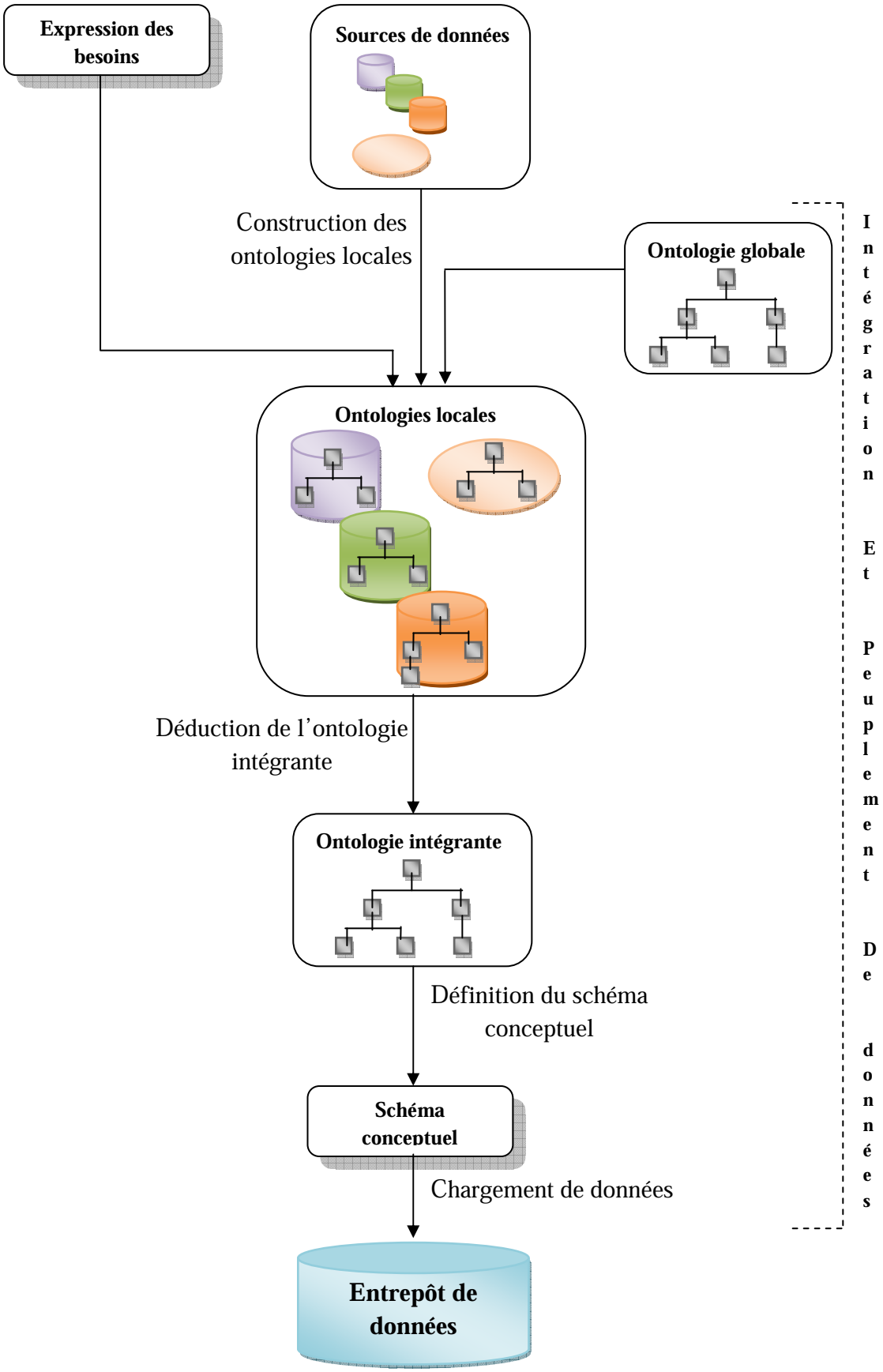


Figure 6.1. *Architecture de l'approche proposée*

5.2 Les étapes de la démarche proposée

Etape 1 : définir le domaine

Puisque notre approche est basée sur les ontologies, il est nécessaire de définir un domaine précis vu que les ontologies sont souvent liées à un domaine particulier. De ce fait, nous définissons dans cette première étape le domaine sur lequel on désire faire des études.

De plus, nous traitons dans notre approche un type particulier de données qui sont les données multimédia. Par conséquent, nous devons nous positionner dans un domaine traitant ce type de données, par exemple, le domaine médical. En effet, les applications informatiques médicales sont très connues et très utilisées aujourd'hui dans les hôpitaux du monde entier. Des applications qui gèrent le stockage et l'accès aux dossiers des patients, en passant par les applications d'analyse et visualisation d'images médicales, et jusqu'aux applications d'aide au diagnostic, à la décision et à la planification du traitement ou des gestes chirurgicaux, celles-ci ont de multiples fonctionnalités.

L'imagerie médicale est une source de données principale pour différents types d'applications. Les méthodes de segmentation d'images médicales et de visualisation, par exemple, fournissent une meilleure vue des images, facilitant ainsi la décision sur le traitement à appliquer. L'imagerie médicale regroupe ainsi plusieurs types d'images liées aux différents organes du corps humain.

Etape 2 : recenser les sources de données et les besoins des utilisateurs

Comme notre démarche est guidée par les besoins des utilisateurs et les sources de données, il est nécessaire de recenser les sources de données dont on dispose et d'étudier les besoins des utilisateurs de l'entrepôt qu'on désire construire.

a- Recenser les sources de données

Cette étape consiste à analyser les données suivant un certain nombre de critères à savoir la nature des données, leur provenance, les informations qu'elles fournissent, le modèle de données utilisé pour leur représentation, etc.

Ainsi, à titre d'exemple, on dispose dans le domaine médical d'un nombre de données de natures très variées qu'on peut regrouper dans les catégories suivantes :

- Les données médicales qui englobent les données pathologiques, les données épidémiologiques et les données cliniques.
- Les données médico-techniques qui sont liées aux données pharmaceutiques, les données des laboratoires et les données liées à l'imagerie médicale.
- Les données administratives liées à la gestion financière, la gestion des ressources humaines (GRH), etc.

D'après cette catégorisation, nous constatons que nous disposons d'une grande quantité de données de types divers (texte, image, etc.) dispersées sur différentes structures de santé dont les systèmes d'information sont souvent autonomes et hétérogènes. En effet, les sources de

données proviennent principalement des données opérationnelles des établissements de santé contenues dans des bases de données de type hiérarchique ou en réseau. Elles peuvent être fournies aussi par :

- Des données départementales contenues dans des systèmes de fichiers tels que VSAM et des SGBDR tels que ORACLE.
- Des données privées détenues dans des stations de travail et des serveurs privés.
- Des systèmes externes tels que des bases de données commerciales disponibles ou des bases de données détenues par les fournisseurs ou les clients de l'organisation.

Les données qui proviennent de différentes sources véhiculent généralement des informations très utiles pour les utilisateurs. Dans le domaine médical par exemple, les sources de données fournissent des informations sur [Assélé Kama et al, 2010] :

- La gestion des consultations,
- La gestion des RDV médicaux,
- L'état des patients à tout moment,
- L'évolution d'une pathologie,
- Les images médicales et radiologie,
- Les hébergements hospitaliers,
- Etc.

En plus de leur diversité de type, l'hétérogénéité des données apparait dans la manière dont ses données sont représentées. En effet, les données peuvent être soit structurées et regroupées dans des bases de données relationnelles ou objets, soit semi-structurées sous forme de graphes ou fichiers XML par exemple, soit non structurées et stockées dans des fichiers de formats divers (image, son, etc.).

Dans le domaine médical, on est souvent confronté à des données complexes telles que les images. Une manière de les manipuler consiste à les stocker dans des bases de données *images* en utilisant un SGBD dédié qui prend en compte des données multimédias. La particularité de ces BD et qu'elles soient volumineuses vu la taille des données en question. De plus, afin de pouvoir faire des études d'une manière régulière (l'évolution d'une maladie par exemple), on est souvent amené à accéder à des données anciennes qui doivent être stockées dans les BD, ce qui augmente encore la taille de ces dernières.

b- Etudier les besoins des utilisateurs

Notre démarche de conception d'ED est basée en premier lieu sur les sources de données qui sont à la base de tout projet d'entreposage. Cependant, la prise en compte des besoins des utilisateurs joue un grand rôle pour l'aboutissement des projets d'entreposage. Afin d'étudier les besoins des utilisateurs, nous devons :

- Définir les utilisateurs de l'ED en question,
- Dresser, pour chaque utilisateur, la liste des requêtes auxquelles il voudrait répondre pour la prise de décision.

Si l'on reste dans le domaine médical, les utilisateurs qu'on peut recenser sont les médecins, les infirmiers, les paramédicaux, les radiologues, les biologistes, etc. Le but étant de permettre à ces acteurs de capturer, partager et appliquer des connaissances collectives pour prendre des décisions optimales en temps réel.

Etape 3 : Filtrer les besoins utilisateurs et les sources

A partir des besoins utilisateurs et des sources de données disponibles, nous déterminons les sources de données utiles pour répondre aux requêtes des utilisateurs (décideurs). A ce niveau, deux cas se présentent :

1- On peut écarter des sources de données non pertinentes c'est-à-dire celles qui n'apportent pas d'informations utiles pour répondre aux requêtes.

2- Repérer les requêtes utilisateurs pour lesquelles il est nécessaire d'avoir d'autres sources de données.

Cette étape est assez importante puisqu'elle permet de détecter au préalable (avant de passer à l'implémentation) les sources de données non utiles pour l'ED en question et de rajouter, éventuellement, d'autres sources pour satisfaire un certain nombre de requêtes.

Etape 4 : définition et confrontation d'ontologies

Afin de cerner tous les concepts du domaine selon une vue donnée, les étapes à suivre sont :

- définir (ou utiliser) une ontologie globale (existante) comportant tous les concepts susceptibles d'être manipulées dans les sources de données.
- établir une ontologie locale pour chaque source de données, il s'agit de définir les concepts liées à chaque source examinée à part, ainsi que la hiérarchie entre ces concepts.
- confronter l'ontologie globale avec chaque ontologie locale afin de définir les correspondances entre les différents concepts.

Etape 5 : définir l'ontologie intégrante

Cette étape consiste à déduire l'ontologie intégrante qui nous sert dans l'étape d'intégration des données au niveau de l'ED. L'ontologie intégrante est définie suite à la confrontation des ontologies locales avec l'ontologie globale en établissant les équivalences entre les différents concepts.

Etape 6 : intégration des sources de données

A partir des concepts définis dans l'ontologie intégrante, on procède à la déduction les faits et des dimensions qui constituent le schéma conceptuel de l'entrepôt qu'on désire construire.

6. Exemple d'illustration : Classification des sujets (patients) sains ou atteints d'une pathologie du cerveau.

Cet exemple consiste à appliquer les étapes de la démarche proposée à un domaine particulier : c'est le domaine médical. Pour cela, nous expliquons l'application de chaque étape par rapport au domaine visé.

Etape 1 : Définir le domaine et cerner le contexte

Nous avons opté pour le domaine médical. Ainsi, on dispose d'un ensemble de données réparties sur un ensemble d'hôpitaux. Au niveau de ces derniers, on est confronté à une grande quantité de données de types divers. Nous nous intéressons principalement à un ensemble restreint de données visées par une pathologie particulière. Nous optons donc pour les données liées au cerveau humain et nous traitons principalement les *images* représentant l'anatomie du cerveau humain.

Etape 2 : Recenser les sources de données et les besoins des utilisateurs

Nous disposons d'une grande variété de données, entre autres : les données liées aux hôpitaux, les données liées aux médecins, les données liées aux patients, les données pathologiques, les données radiologiques, les données liées aux examens pathologiques, etc. nous résumons l'ensemble de ces données dans les points suivants :

- Source de données SD1 : représentant les BD décrivant les différents hôpitaux et centres de santé existants sur le territoire. Pour chaque hôpital ou centre de sante, on dispose des informations suivantes : *Raison social, Adresse, Coordonnées, Région, Capacité, Services, etc.*

- Source de données SD2 : liée à l'annuaire des médecins qui contient un ensemble d'informations relatives aux médecins telles que : le nom, la spécialité, l'expérience (en nombre d'année), l'hôpital auquel il appartient, etc.

- Source de données SD3 : contenant la BD décrivant les différentes pathologies qui existent. Les informations liées à cette source le nom, les signes, le traitement, etc.

- Source de données SD4 : représentés par la BD liée aux patients. Les informations fournies par cette source de données concerne le nom du patient, âge, adresse, sexe, antécédents familiaux, antécédents personnels, etc.

- Source de données SD5 : regroupant des images radiologiques liées à différentes maladies et différents patients.

Après avoir recensé les sources de données, nous devons définir les futurs utilisateurs et préciser leurs besoins. De ce fait, nous visons dans notre exemple une catégorie spécifique des acteurs du domaine médical, c'est celle des médecins s'intéressant à l'anatomie du cerveau

humain et nous proposons ainsi de construire un ED offrant une assistance à ses utilisateurs afin de pouvoir prendre des décisions efficacement et rapidement. En effet, un des besoins majeurs des médecins spécialistes dans le domaine neurologique est de pouvoir déduire, à partir d'une image, l'existence d'une anomalie au niveau du cerveau. Autrement dit, on doit comparer l'anatomie d'une image donnée avec celle d'une image *référence*. Pour cela, nous devons disposer d'une description de l'anatomie *référence* du cerveau.

Un autre besoin pourrait être lié aux services de santé désirant effectuer à grande échelle, une classification des sujets sains ou atteints d'une pathologie donnée du cerveau.

Pour répondre à ce besoin, l'entrepôt de données doit être conçu de telle sorte qu'il soit en mesure de distinguer entre les images normales et celles présentant des anomalies. Une solution pourrait être de les classifier dans deux catégories distinctes par exemple. Ainsi, la contribution de l'utilisateur (médecin spécialiste) se limite à une simple analyse et validation.

Etape 3 : Filtrer les sources de données

Cette étape consiste à filtrer les sources de données en fonction des besoins des utilisateurs. En effet, nous ne prenons pas en compte toutes les sources de données. Puisque notre objectif s'insère dans le cadre des données multimédias alors nous nous intéressons à la source SD5 relative aux images radiologiques et plus précisément aux images liées au cerveau humain.

Etape 4 : Définition et confrontation d'ontologies

Cette étape consiste à définir et à confronter les ontologies dont nous avons besoin. Pour illustrer les étapes de notre démarche qui sont liées aux ontologies, nous devons définir l'ontologie globale ainsi que les ontologies locales.

- **Ontologie globale**

On définit ici à titre d'exemple, une ontologie modélisant l'anatomie du cerveau humain. La hiérarchie des concepts participant à la description du cerveau est donnée dans la figure suivante :

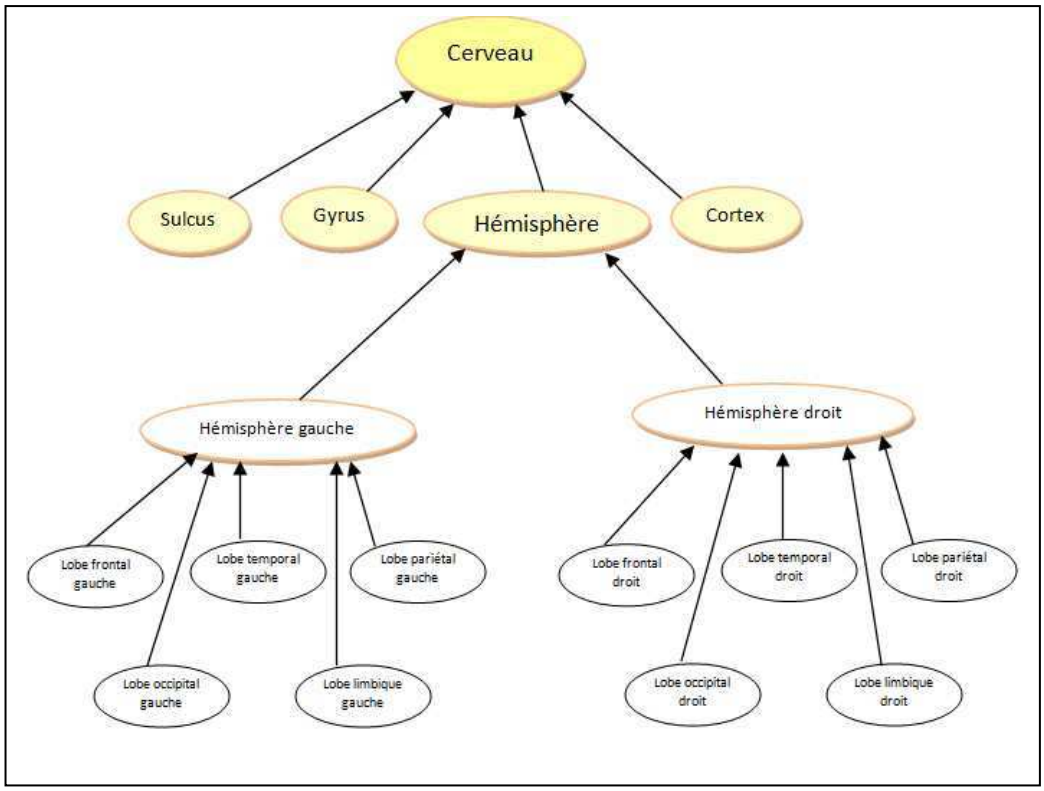


Figure 6.2. *Hiérarchie de concepts décrivant l'anatomie du cerveau humain*

Contrairement au domaine de l'analyse et de l'indexation de documents textuels pour lesquels les ontologies sont largement utilisées, dans le domaine de l'interprétation et de l'indexation sémantique d'images, la mise en correspondance du niveau perceptif (pixels ou voxels, groupes de pixels) et du niveau linguistique (concepts du domaine) de l'interprétation a souvent constitué une barrière à leur exploitation. Il s'agit du **fossé sémantique**, défini comme *le manque de concordance entre les informations perceptuelles que l'on peut extraire des images et l'interprétation qu'ont ces données pour un utilisateur dans une situation déterminée.* Cependant, les relations spatiales jouent un rôle important dans la description et la reconnaissance des objets : elles permettent en effet de lever l'ambiguïté entre des objets d'apparences similaires et sont souvent plus stables que les caractéristiques des objets eux-mêmes. Ces relations peuvent être de nature topologique (relations ensemblistes, adjacence) ou métriques (distances, directions relatives) comme le montre le schéma de la figure suivante :

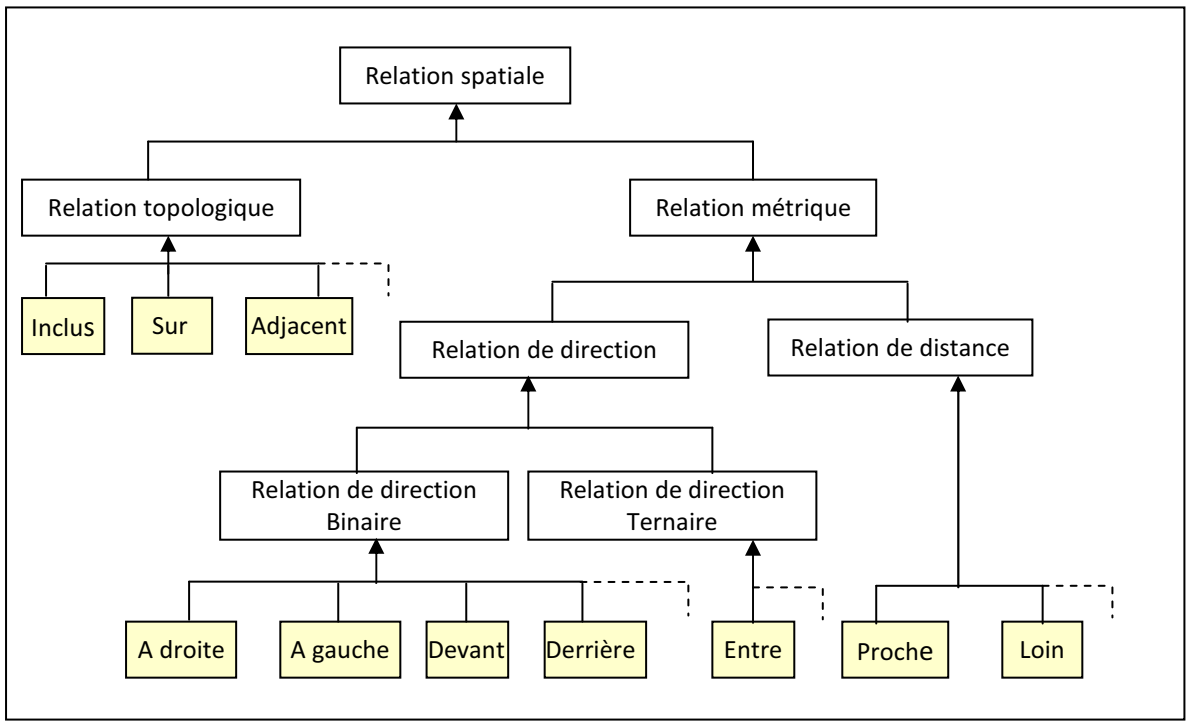


Figure 6.3. *Extrait de l'organisation hiérarchique des relations spatiales*

Pour illustrer la notion des relations spatiales nous prenons quelques exemples liés à l'anatomie du cerveau.

Ex1 : le thalamus droit est *à gauche* du troisième ventricule et *en-dessous* du ventricule latéral.

Ex2 : le lobe frontal se trouve *devant* le lobe pariétal.

Ex3 : le lobe pariétal se trouve *entre* le lobe frontal et le lobe occipital.

Ex4 : le gyrus occipital moyen est *adjacent* au gyrus angulaire.

Ces relations sont très importantes car elles fournissent des informations liées à la morphologie d'un cerveau humain sain. Cependant, l'apparence d'une anomalie au niveau de ces relations implique un cerveau atteint d'une maladie particulière. Parmi ces maladies on peut citer : Tumeur cérébrale, Accident vasculaire, traumatisme crânien, sclérose en plaques, maladie d'Alzheimer (MA), etc.

Dans le cas de la MA par exemple, il s'agit d'une pathologie neuro-dégénérative évolutive caractérisée par des lésions : dépôts amyloïdes, dégénérescences neuro-fibrillaires (DNF). Ces lésions se répandent dans le cerveau en parallèle d'une dégradation progressive des fonctions cognitives. Lorsque les DNF apparaissent au niveau du lobe temporal interne (cortex entorhinal et hippocampe) l'atteinte cognitive débute avec des trouble de la mémoire, qui resteront prédominants. Quand les DNF envahissent l'ensemble du cortex, les autres fonctions cognitives (visu-spatiales, langage,…) sont alors perturbées.

En effet, lors de l'apparition des DNF, les relations spatiales entre les différentes parties du cerveau se modifient. Ainsi, on peut constater l'existence d'une anomalie.

- **Définition des ontologies locales**

Les ontologies locales sont liées aux différentes sources de données. Par conséquent, des conflits peuvent exister entre ces ontologies. Pour un même concept par exemple, on peut avoir des appellations différentes. Or, comme c'est un domaine scientifique que nous traiton, alors les noms des concepts sont généralement identiques. Cependant, deux types de conflits se présentent : des conflits au niveau de la langue et des conflits au niveau des relations spatiales.

Les conflits au niveau de la langue apparaissent lors de l'utilisation de différentes langues pour exprimer un même concept. Par exemple, les concepts exprimés dans la figure 6.2 en langue française peuvent être exprimés en anglais dans une autre source de données figure 6.4.

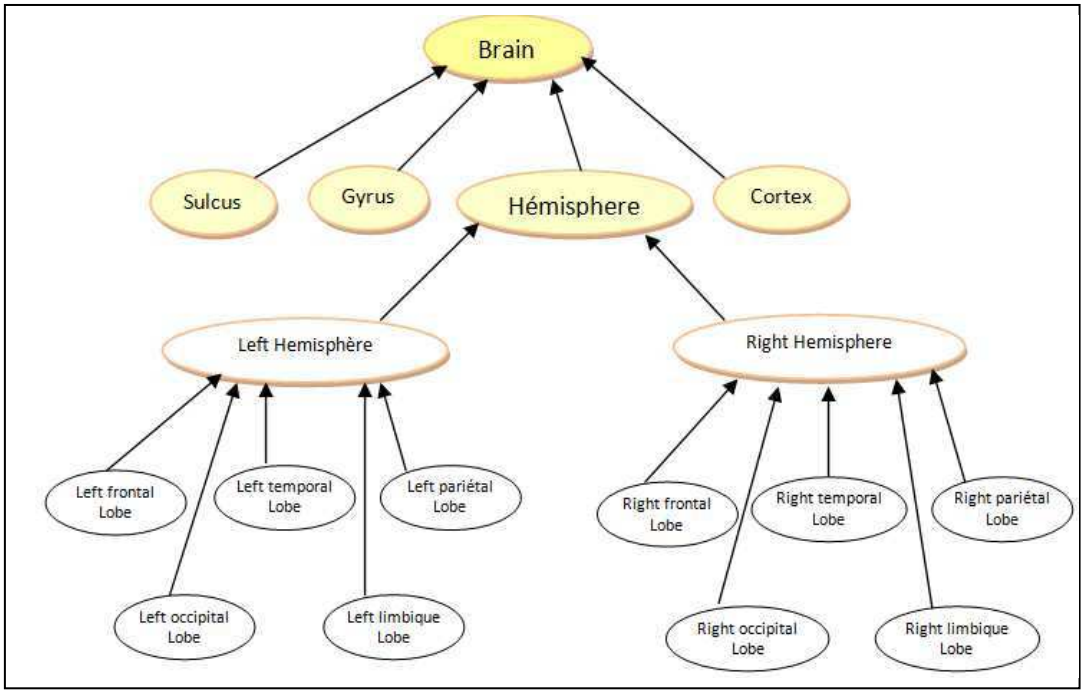


Figure 6.4. *Hiérarchie de concepts du cerveau en anglais*

En ce qui concerne les conflits au niveau des relations spatiales, des concepts peuvent être caractérisés par une relation spatiale dans une ontologie O1 liée à une source S1 alors qu'ils soient caractérisés par une autre relation dans une ontologie O2 liée à une source S2. Par exemple, dans l'ontologie O1 on peut se limiter à définir une relation *d'adjacence* entre le lobe frontale et le lobe pariétal (le lobe frontal est *adjacent* au lobe pariétal), alors que dans une ontologie O2 on définit une relation de *direction* (le lobe frontal se trouve devant le lobe pariétal).

L'étape suivante consiste à déduire l'ontologie *intégrante* que nous utilisons pour générer le schéma de l'entrepôt. Cette ontologie est obtenue par la confrontation de l'ontologie

globale avec les ontologies locales afin de régler les conflits sémantiques cités précédemment. En effet, nous utilisons l'ontologie globale comme référence permettant de définir une relation de similarité '*same as*' entre les concepts et/ou les relations spatiales présentant des conflits.

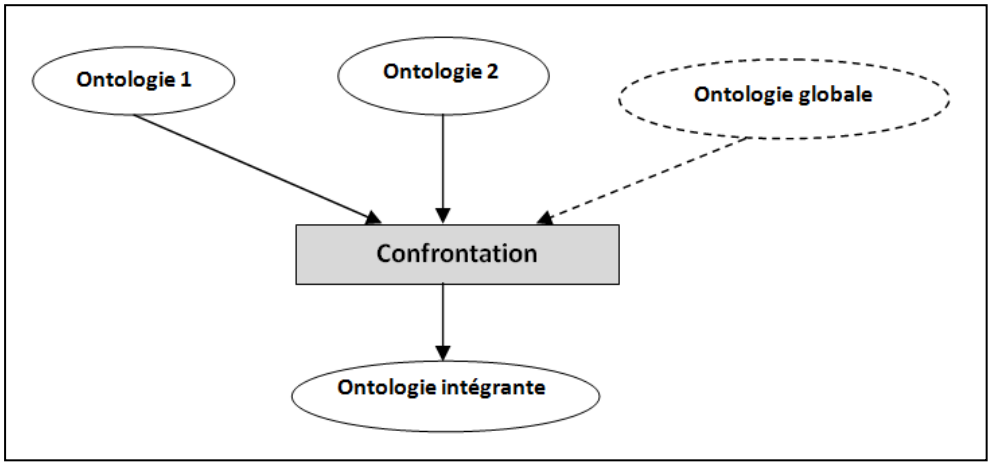


Figure 6.5. *Déduction de l'ontologie intégrante*

Etape 6 : définition du schéma de l'entrepôt

Cette étape consiste à concevoir le schéma de l'entrepôt en s'appuyant sur l'ontologie intégrante définie dans l'étape précédente. Dans notre exemple, nous pouvons définir le schéma de l'entrepôt à l'aide d'un modèle en *étoile* qui à comme fait *classification* et comme dimensions image, région, maladie et temps. La figure 6.6 montre le schéma en *étoile* en question.

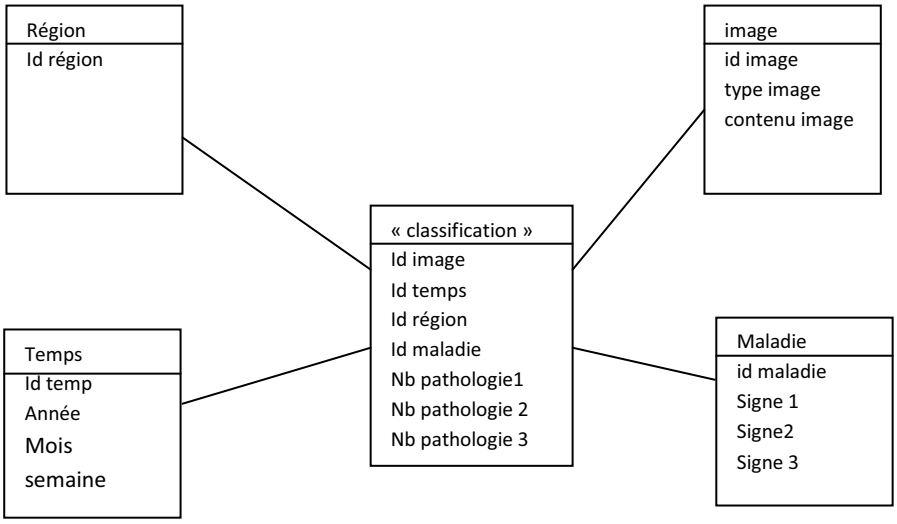


Figure 6.6. *Schéma en étoile pour le fait « classification »*

7. Conclusion

Dans ce chapitre, nous avons décrit l'approche de conception d'EDM que nous avons proposée en nous inspirant des différentes approches existantes. Nous avons commencé par cerner les problèmes pouvant être liés aux données multimédia et aux approches de construction d'ED. Cette analyse a permis de ressentir la grande ressemblance entre les ED et les systèmes d'intégration de données, c'est pourquoi nous nous sommes amenés à présenter un état de l'art sur les approches d'intégration. Par la suite, nous avons décrit les étapes générales de notre approche et nous avons terminé par donner un exemple illustrant les étapes de notre approche.

Conclusion générale

Le but principal de notre travail était de proposer une démarche pour la conception d'un entrepôt de données multimédias à base ontologique. Ceci nous a amenés à consacrer une grande partie de ce rapport pour la présentation de différents concepts et notions nécessaires à l'aboutissement de notre démarche. Ainsi, nous avons décomposé notre travail en deux parties : la première a consisté en un état de l'art et la seconde a été consacrée à la description de notre démarche. Nous avons jugé qu'il est très important et inévitable de bien aborder tous les volets rentrant dans la conception d'un entrepôt de données multimédia, c'est pourquoi nous avons accordé une grande partie du document pour présenter un état de l'art sur les différentes notions.

La partie attribuée à l'état de l'art nous a permis de récolter un grand ensemble d'information concernant les entrepôts de données, les données multimédias, et les ontologies. L'apport principal ressorti de cette étude concerne les méthodes de conception des projets d'entreposage, qui nous ont permis de voir la classification de ces méthodes en trois générations d'approches : les approches directes, les approches traditionnelles et les approches ontologiques. Les approches directes s'intéressent uniquement à la modélisation logique et physique de l'entrepôt. Les approches traditionnelles consistent en la mise en place d'une vraie phase de conception. Dans cette génération d'approches, on retrouve deux catégories de méthodes : des méthodes se basant uniquement sur les sources de données et des méthodes se basant sur les sources et sur les besoins des utilisateurs. En ce qui concerne les approches ontologiques, ces dernières permettent de gérer l'hétérogénéité des sources en exploitant une ou plusieurs ontologies.

L'analyse de ces approches nous a permis de conclure qu'elles souffrent d'un problème d'intégration, tâche très importante dans les projets d'entreposage. Ainsi, nous avons remarqué qu'il existe une grande similarité entre les systèmes d'intégration et les ED. De ce fait, nous nous sommes amenés à analyser les différentes approches d'intégration qui existent.

En nous basant sur ces approches, nous avons proposé notre démarche de conception d'entrepôt traitant des données multimédias. Notre démarche a été proposée après avoir effectué une analyse des différentes approches de telle façon à éviter au maximum les problèmes liés aux entrepôts de données ainsi que ceux liés aux données multimédias. La démarche proposée repose sur les sources de données et est *guidée par les besoins des utilisateurs*. Afin de pallier aux conflits sémantiques liés aux données multimédia (notamment

les images), nous avons adopté une approche *ontologique* basée sur la confrontation d'ontologies locales pour aboutir à une ontologie intégrante, en utilisant comme référence une ontologie globale du domaine étudié. La démarche proposée a été illustrée sur un exemple lié au domaine médical pour la classification des sujets atteints de maladies cérébrales.

La démarche proposée constitue un noyau pour la conception d'un entrepôt de données multimédia à base ontologique. Des extensions futures de ce travail pourraient être :

- L'amélioration de la démarche en définissant les formalismes adéquats pour la description et la formalisation de chacune de ses étapes comme par exemple, proposer une modélisation '*type'* pour la définition des besoins utilisateurs vu que ce dernier doit formuler ses besoins en requêtes SQL qui peuvent être très complexes. Ceci donne l'occasion même à des utilisateurs '*non experts'* de participer à la réalisation du projet d'entreposage.

- Etudier la faisabilité de l'approche proposée en considérant des données plus complexes que l'image tels que le son et la vidéo. En effet, à travers les exemples que nous avons considérés, nous nous sommes focalisés essentiellement sur des données de type image.

- Pour la mise en œuvre de la démarche, plusieurs technologies peuvent être utilisées telles que les *services web* qui permettent d'intégrer dans une même architecture, différents outils hétérogènes et d'obtenir des solutions flexibles et évolutives [Luo et al, 05], [Choi et al, 07]. Récemment, de nouvelles solutions offertes via internet, sont envisageables, il s'agit du « cloud computing » [Hennion et al, 2012] permettant aux entreprises de gagner en efficience et en croissance en déployant une solution qui contrôle au mieux les dépenses informatiques telle une solution de data warehouse [Ting et al, 2011] qui exige une architecture robuste et de nombreuses ressources. Bien sur, ces solutions exigent un grand degré de confidentialité et de sécurité des données mises à la disposition du fournisseur de service.

Bibliographie

[Allen 83] J. F. Allen. "Maintaining knowledge about temporal intervals", *Communications of the ACM 26*(11), 832 – 843, 1983

[Amarouche 06] AMAROUCHE Idir Amine, "Démarche de conception ascendante d'un entrepôt de données distribué dans un environnement inter-organisationnel", Mémoire de Magister, USTHB, 2006.

[Annoni et al 06] E. Annoni, F. Ravat, O. Teste, et G. Zurfluh "Méthode de Développement des Systèmes d'Information Décisionnels : Roue de Deming", *INFORSID* 657-672, 2006.

[Arigon et al 05] ANNE-MURIEL ARIGON , ANNE TCHOUNIKINE and MARYVONNE MIQUEL, « Intégration de versions fonctionnelles dans les entrepôts de données multimédias », LIRIS, 10 juin 2005.

[Arigon et al 06] ANNE-MURIEL ARIGON, ANNE TCHOUNIKINE and MARYVONNE MIQUEL, "Handling Multiple Points of View in a Multimedia Data Warehouse", ACM Transactions on Multimedia Computing, Communications and Applications, Vol. 2, No. 3, August 2006, Pages 199–218.

[Asleson et al 06] R. Asleson, et N. Schutta, « *Foundations of Ajax* ». Apress. 2006.

[Assélé Kama et al, 2010] Ariane Assélé Kama, Giovanni Mels, Rémy Choquet, Jean Charlet et Marie- Christine Jaulent, « Une approche ontologique pour l'exploitation de données cliniques ». IC 2010.

[Barnard et al 03] K. Barnard, , P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, et M. Jordan, « Matching words and pictures", *Journal of Machine Learning Research 3*, 1107 – 1135, 2003.

[Belaid07] N. Belaid, « SIDOBO : Une architecture diintégration de sources à base ontologique ». Rapport de master. Université de Poitiers, 2007.

[Bellard et al 98] C. Ballard, D. Herreman, D. Schau, R. Bell, E. Kim, A. Valencic "Data Modeling Techniques for Data Warehousing", International Technical Support Organization, IBM, February 1998, http://www.redbooks.ibm.com.

[Bellatreche 00] L. Bellatreche. "Utilisation des index et de la fragmentation dans la conception logique et physique d'un entrepôt de données", Thèse pour obtenir le grade de Docteur en Informatique, université de Clermond Ferrand II, 2000.

[Bellatreche 06] L. Bellatreche, D. Nguyen Xuan, G. Pierra, Hondjack D., "Contribution of Ontology-based Data Modeling to Automatic Integration of Electronic Catalogues within EngineeringDatabases" Computers in Industry Journal, 2006.

[Beneventano et al 00] D. Beneventano, S. Bergamaschi, S. Castano, A. Corni, R. Guidetti, G. Malvezzi, M. Melchiori, and M. Vincini. Information integration: The MOMIS project demonstration. In The VLDB Journal, pages 611–614, 2000.

[Berra et al 93] P. B. Berra, F. Golshani, R. Mehrotm, 0. Liu Sheng: "Multimedia Information Systems". *IEEE Truns. on Knowledge and Datu Engineering,* Vol. *5,* No 4, August 1993, pp. 545-549.

[Bloch05] I. Bloch, "Fuzzy Spatial Relationships for Image Processing and Interpretation" : A Review. *Image and Vision Computing 23*(2), 89 – 110, 2005.

[Boehnlein et al 99] M. Boehnlein, A. Ulbrich-vom Ende "Deriving the Initial Data Warehouse Structures from the Conceptual Data Models of the Underlying Operational Information Systems", DOLAP"99, USA, 1999.

[Bonifati et al 01] A. Bonifati, F. Cattaneo S. Ceri A. Fuggettad et S. Paraboschi "Designing Data Marts for Data Warehouses", ACM Transactions on Software Engineering and Methodology, Vol. 10, No. 4, Pages 452–483, 2001.

[Boussaid et al 07] Omar Boussaid, Fadila Bentayeb, A Tanasescu « Integration and dimensional modeling approaches for complex data warehousing». Journal of Global Optimization 37 (4), 571-591. 2007

[Bruckner et al 01] R. Bruckner, B.List, J. Schiefer "developing requirements for data warehouse systems with use case", Seventh Americas Conference on Information Systems, 2001.

[Bruno et al, 02] Emmanuel Bruno, Jacques Le Maitre, Elisabeth Murisasco, "Indexation et interrogation de photos de presse décrites en MPEG-7 et stockées dans une base de données XML", Ingénierie des Systèmes d'Information 7(5-6): 169-186 (2002).

[Bubenko et al 94] J. Bubenko, C. Rolland, P. Loucopoulos, V. De Antonellis, "Facilitating 'fuzzy to formal' requirements modeling". IEEE 1st Conference on Requirements Engineering, ICRE"94 pp. 154-158, 1994.

[Burmester et al 06] L. Burmester, M. Goeken, "Method for user oriented modelling of data warehouse systems", 2006.

[Campedel et al 08] Marine Campedel, Marie Lienou, Ivan Kyrgyzov, Henri Maître, « Vers la construction d'une ontologie appliquée à l'imagerie satellitaire », Atelier ECOI, Ecole centrale paris- GET-Télécom Paris, 2008.

[Capdevielle et al 95] O. Capdevielle, P. Dalle - Formulation d'objectifs de traitement d'images. IHM'95, 7èmes journées sur l'Ingénierie des Interfaces Homme-Machine, Toulouse, octobre 1995.

[Charlet et al, 08] Jean Charlet **,** Sylvie Szulman, Guy Pierra, "DAFOE: A Multimodel and Multimethod platform for Building Domain Ontologies", proceedings JFO - December, 2008, Lyon, France.

[Chawathe et al 94] S. S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. D. Ullman, and J. Widom, The tsimmis project : "Integration of heterogeneous information sources", Proceedings of the 10th Meeting of the Information Processing Society of Japan, pages 7 18, Mars 1994.

[Choi et al, 07] C. Choi, R. M¨unch, B. Bunk, J. Barthelmes, C. Ebeling , "Combination of a data warehouse concept with web services for the establishment of the *Pseudomonas* systems biology database SYSTOMONAS", Journal of Integrative Bioinformatics, 2007

[Connoly et al] Connolly, Begg, "Systèmes de bases de données : Approche pratique de la conception, de l'implémentation et de l'administration", Chapitre 31 ; concepts de l'entrepôt de données, éditions Renald Goulet inc, 1436 pages, ISBN : 2-89377-267-6.

[Cros 97] P. Cros – « Typologie des mesures de traitement d'images ». DEA Informatique de l'Image et du Langage, Université Paul Sabatier, Toulouse, juin 1997.

[Da Costa et al 07] Arnaud DA COSTA, Éric LECLERCQ , Arnaud GAUDIN, Jean GASCUEL, Marinette SAVONNET, Marie-NoëlleTERRASSE, « Ontologies et extraction de connaissances : une expérience dans le domaine des bases de données image pour la biologie », 1ères journées Francophones sur les Ontologies, 2007.

[Dalakleidi et al, 2011] : K. Dalakleidi, S. Dasiopoulou, G. Stoilos, V. zouvaras, G. Stamou*, et al. "*Semantic Representation of Multimedia Content », LNCS - Volume 6050, pp 18-49, 2011.

[Danjean 05] Daniel Jean**,** "La compression vidéo MPEG-7", http://danjean.developpez.com/video/mpeg-7/- 2005.

[Djema et al, 07]: Djema L, Boumghar F.O, Debiane S, « L'imagerie médicale dans une base de données distribuée multimédia sous Oracle 9i ». Proceedings SETIT'07, Tunisie, mars 2007.

[Djezzar et al 07] Meriem Djezzar, Zizette Boufaida, Mounir Hemam "Une méthode de construction d'ontologies en logique de description", 8th International Symposium on Programming and Systems – ISPS'2007.

[Douglas05] K. Douglas, "*PostgreSQL : the comprehensive guide to building, programming and administering PostgreSQL databases".* Sams, 2005.

[Duygulu et al 02] P. Duygulu, , K. Barnard, J. de Freitas, et D. Forsyth, "Object recognition as machine translation : learning a lexicon for a fixed image vocabulary", *Seventh European Conference on Computer Vision IV*, 97 – 112, 2002.

[Fankam et al 09] C. Fankam, S. Jean, G. Pierra, L. Belltreche, Y. Ait-Ameur "Towards Connecting Database Applications to Ontologies", First International Conference on Advances in Databases, Knowledge, and Data Applications, 2009.

[Ficet 96] V. Ficet, Construction interactive d'un modèle conceptuel d'applications de traitement d'image. Proc. Journées jeunes chercheurs en IA, Nantes, 1996.

[Firestone97] J.M. Firestone "Object-Oriented Data Warehousing", Technical Report, Executive Information Systems, White Paper No.5, 1997

[Fox 91] E.A. Fox, "Advances in Interactive Digital Multimedia Systems", IEEE, *Computer,* Octobre1991, p. 9-2 1.

[Freemen 75] J. Freeman, "The modelling of spatial relations". *Computer Graphics and Image Processing 4*(2), 156 – 171, 1975.

[Frendi et al 03] M. Frendi, C. Salinesi "Requirements Engineering for Data Warehousing", Requirements Engineering:Foundation for Software Quality (REFSQ), (in association with the CAISE'03 Conference), Velden, Austria, June 2003.

[Furst02] F.FÜRST, « L'Ingénierie Ontologique », rapport, 2002, Institut de recherche en Informatique de Nantes.

[Gam et al 06] I. Gam, C. Salinesi "A Requirement-driven Approach for Designing Data Warehouses". REFSQ'2006, Luxembourg, 2006

[Gaudin04] A. GAUDIN, « Analyse des réorganisations neuroanatomiques des projections olfactives au cours de la métamorphose chez le xénope et la recherche des mécanisme moléculaires sous-jacents », PhD thesis, Centre Européen des Sciences du Goût, 2004.

[Gaudin et al 05] A. GAUDIN, J. GASCUEL, « 3D Atlas Describing the Ontogenic Evolution of the Primary Olfactory Projections in the Olfactory Bulb of Xenopus Laevis », The Journal Of Comparative Neurology, 2005.

[Genesereth et al 97] M. R. Genesereth, A. M. Keller, and O. M. Duschka. Infomaster : an information integration system. In ACM SIGMOD International Conference on Management of Data, pages 539–542, 1997.

[Giorgini et al 05] P. Giorgini, S. Rizzi, M. Garzetti "Goal-Oriented Requirement Analysis for Data Warehouse Design" DOLAP"05, Bremen, Germany, 4–5 Novembre 2005.

[Giraldo05] GL. Giraldo Gomez "Construction automatisée de l'ontologie de systèmes médiateurs". Thèse de doctorat, université Paris XI Orsay, 2005.

[Glotin et al 05] H. Glotin, et S. Tollari, Fast image auto-annotation with visual vector approximation clusters. In *Proc. of Fourth International Workshop on Content-Based Multimedia Indexing, 2005.*

[Goasdoué et al 99] Goasdoué F., Lattes V., Rousset M. The use of carin language and algorithms for information integration: The picsel project. International Journal of Cooperative Information Systems (IJCIS), 9(4):383 – 401. 1999.

[Goh et al 99] C. Goh, S. Bressan, E. Madnick, and M. D. Siegel. Context interchange: "New features and formalisms for the intelligent integration of information", ACM Transactions on Information Systems, 17(3):270–293, 1999.

[Golfarelli et al 98] M. Golfarelli, D. Maio, S. Rizzi, "Conceptual Design of Data Warehouses from E/R Schemes", Published in the Proceedings of the Hawaii International Conference On System Sciences, Kona, Hawaii, January 6-9, 1998

[Golfarilli 09] M. Golfarelli "From user requirements to conceptual design in data warehouse design – a survey", 2009.

[Gong 99] Y. Gong, « Advancing content-based image retrieval by exploiting image colours and regions features », Multimedia Systems Vol. 7, No 6, November 1999, pp 449-457.

[Grubert 93] T. GRUBER, «What is an Ontology? », http://www-ksl.stanford.edu/kst/what-is-an-ontology.html 1993.

[Gruber 95] T. Gruber, "A translation approach to portable ontology specification. Knowledge Acquisition", 5(2): 199–220, 1995.

[Guergour 07] Habib-ELLAH GUERGOUR, « Construction d'une ontologie d'application dans le cadre de l'EAI », Mémoire de Magister - Génie logiciel et intelligence artificielle - Université Mentouri de Constantine, Algérie, 2007.

[Hacid 04] Mohand-Saïd Hacid et Chantal Reynaud, L'intégration de sources de données, Revue Information-Interaction-Intelligence, juin 2004.

[Hamam et al 04] M. Hemam, Z. Boufaida, "An Ontology Development Process for the Semantic Web," *EKAW'04 Workshop on Knoweledge Management and Semantic Web*, Whittlebury Hall, Northamptonshie, UK, 2004.

[Hamdoun et al 07] Sana Hamdoun, Mohamed Badri, Faouzi Boufarès, « Construction et maintenance des entrepôts de données hétérogènes », e-TI : la revue électronique des technologies de l'information, Numéro 4 : Génie logiciel et intelligence artificielle, 23 juin 2007.

[Hammer et al 95] J. Hammer, H. Garcia-Molina, J. Widom, W. Labio, and Y. Zhuge. The Stanford data warehousing project. IEEE Quarterly Bulletin on Data Engineering ; "Special Issue on Materialized Views and Data Warehousing", 18(2) :41–48, 1995.

[Han et al 01] J. HAN AND M. KAMBER, 2001. "*Data mining, concepts and techniques",* Morgan Kaufmann.

[Heflin et al 99] Heflin J., Hendler J., Luke S. Shoe: A knowledge representation language for internet applications. Technical CS-TR-4078, Institute for Advanced Computer Studies, University of Maryland. 1999.

[Hennion et al, 2012] R. Hennion , H. Tournier , E. Bourgeois, «Cloud computing : Decider – Concevoir – Piloter-Amélioer ». Eyrolles, 2012.

[Hudelot et al 06] C. Hudelot , J. Atif, et I. Bloch, « Ontologie de relations spatiales floues pour l'interprétation d'images », In *Rencontres francophones sur la Logique Floue et ses Applications, LFA 2006*, Toulouse, France, pp. 363–370, 2006.

[Husseman et al 00] B. Hüsseman, J. Lechtenbörger, G. Vossen "Conceptual Data Warehouse Design", Proceedings of the International Workshop on Design and Management of Data *Warehouses (DMDW'2000)*, Stockholm, Sweden, 5-6 Juin 2000.

[Inmon 92] WH. Inmon "Building the data warehouse", Wiley, 1992.

[Jarke et al 97] M. Jarke and Y. Vassiliou. Data warehouse quality design : A review of the dwq project. "In Invited Paper, 2nd Conference on Information Quality". Massachusetts Institute of Technology, Cambridge, 1997.

[Jean 04] S. Jean, « Langage d'exploitation de base de données ontologiques ». Mémoire pour l'obtention du DEA T3IA, Université de Poiters, juin 2004.

[Jeon et al 03] J. Jeon, , V. Lavrenko, et R. Manmatha, "Automatic image annotation and retrieval using cross-media relevance models", *Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, 119 – 126, 2003

[Jolif 06] Christophe JOLIF, "format d'image SVG *"*, 10 novembre 2006.

[Kamp et al 97] V. KAMP AND F. WIETEK, 1997, "Database system for multidimensional data analysis", In *Proceedings of the International Database Engineering and Application Symposium (IDEAS)*. Concordia University, Montreal, Candada, August 25–27, 1997.

[Khouri 09] S.Khouri "modélisation conceptuelle à base ontologique d'un entrepôt de données", mémoire de Magister, Ecole nationale Supérieure d'Informatique (E.S.I) Oued-Smar Alger, 2009.

[Kim et al 03] Hyon Hee Kim and Seung Soo Park; "Building a Web-Enabled Multimedia Data Warehouse", HSI 2003, LNCS 2713, pp. 594-600, 2003, Springer-Verlag Berlin Heidelberg 2003.

[Kim et al 04] Hyon Hee Kim and Seung Soo Park; "A Semantics-Based Versioning Scheme for Multimedia Data", DASFAA 2004, LNCS 2973, pp. 277–288, 2004, Springer-Verlag Berlin Heidelberg 2004.

[Kimball et al 96] R. Kimball, M. Ross "The data warehousing toolkit". John Wiley&Sons, New York, 1996.

[Lambollez et al 95] P.-Y. Lambollez, J.-P. Queille, J-F. Voidrot, C. Chrisment, "EXREP : un outil générique de réécriture pour l'extraction d'informations textuelles", Revue ISI, Vol. 3(4), 1995, pp 471-487.

[Lemaitre 07] C. LEMAITRE, J. MITERAN, J. MATAS, « Definition of a model-based detector of curvilinear regions », *International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2007.

[Levy et al 96] A. Y. Levy, A. Rajaraman, and J. J. Ordille. The world wide web as a collection of views : Query processing in the information manifold. Proceedings of the International Workshop on Materialized Views : Techniques and Applications (VIEW"1996), pages 43–55, June 1996.

[List et al 00] B. List, J. Schiefer, et A. M. Tjoa "Process-Oriented Requirement Analysis Supporting the Data Warehouse Design Process", A Use Case Driven Approach, Proceedings of the 11th International Conference on Database and Expert Systems Applications (DEXA 2000), 593-603, 2000.

[Little et al 91] Thomas D. *C.* Little, Arif Ghafoor: "Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks". *Computer,* October 1991, pp. 42-50.

[Little et al 93] Thomas D. C. Little, Arif Ghafoor: "Interval-Based Conceptual Models for Time-Dependent Multimedia Data". *IEEE Trans. on Knowledge and Datu Engineering,* Vol. 5, No 4, August 1993, pp. 55 1-563.

[Lumineau05] N. Lumineau. « Organisation et Localisation de Données Hétérogènes et Réparties sur un Réseau Pair-à-Pair ». Thèse PhD, Université Pierre et Marie Curie - Paris VI, 2005.

[Luo et al, 05] Z. Luo, Z. Kaisong, X. Hongxia, Z. Kaipeng, "The Data Warehouse Model Based on Web service Technology". Jan 2005, Volume 2, No.1 (Serial No.2). Journal of communication and Computer, ISSN1548-7709,USA

[Lutz06] M. Lutz, "*Programming Python"*. O'Reilly Media. 2006.

[Marhoumi06] F. E. Marhoumi, "Entrepôts de données XML : Développement d'un outil Extraction Transformation Load (ETL) ", Memoire de fin d'etudes en vue de l'obtention du grade d'Ingenieur Civil Informaticien en Sciences Appliquées, Université de Bruxelles, 2006.

[Mazon et al 08] J.Mazon, J. Pardillo, E. Soler, O. Glorio, et J. Trujillo "Applying the i* framework to the development of data warehouses", In The 3rd International i* Workshop - istar08, 79-82, (2008).

[Messin] Alain Messin, Stockage de fichiers dans des tables MYSQL avec PHP, Centre National de la Recherche Scientifique (CNRS), UMS 2202, Admin06, 30/06/2006

[Meylan] Eddy.Meylan"Introduction à la gestion de données multimédia avec Oracle8", Haute Ecole Spécialisée de Suisse Occidentale (Hes-SO), 2001.

[Moody et al 00] D.L. Moody, M. Kortnik: "From Enterprise Models to Dimensionals Models: A Methodology for Data Warehouse and Data Mart Design", Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2000), Sweden, 2000.

[Nabli et al, 09] Ahlem Nabli, Jamel Feki, Faïez Gargouri, "An Ontology Based Method for Normalisation of Multidimensional Terminology". Advanced Internet Based Systems and applications, pp 235-246, Springer Verlag, 2009.

[Nguyen06] D. Nguyen Xuan, "Intégration de bases de données hétérogènes par articulation à priori d"ontologies: Applications aux catalogues de composants industriels". Thèse de doctorat, Université de Poitiers, 2006.

[Nobécourt00] J. Nobécourt, A method to build formal ontologies from texts. EKAW'2000, Juan-les-Pins, 2 octobre 2000.

[Nouvel et al 02] Alexandre NOUVEL, Patrice DALLE ; « Une approche interactive de définition d'ontologies image », Institut de Recherche en Informatique de Toulouse, 2002

[Noy et al 00] F. Natalya Noy et L. Deborah McGuinness "Développement d'une ontologie 101 : Guide pour la création de votre première ontologie", 2000.

[Peguiron05] PEGUIRON Frédérique, "Entrepôt de données et profil utilisateur : évolution du système d'information vers un système d'information stratégique ".

[Perrizo et al 97] W. Perrizo, Z. Zhang and S. Krebsbach, "A Query Processing Method For Data Warehouses Which Contain Multimedia", ACM 0-89791-850-9 97 0002 3.50, 1997

[Phipps et al 02] C. Phipps, et K. Davis "Automating Data Warehouse Conceptual Schema Design and Evaluation", DMDW, 23-32, 2002.

[Prakash et al 03] N. Prakash, A.Gosain "Requirements Driven Data Warehouse Development", CAiSE Short Paper Proceedings, - ceur-ws.org, 2003.

[Rifaieh04] R. D. Rifaieh " Utilisation des ontologies contextuelles pour le partage sémantique entre les systèmes d'information dans l'entreprise", l'Institut National des Sciences Appliquées de Lyon, 2004.

[Rilston et al 03] F. Rilston, S. Paim, F. Jaelson, B. Castro. "DWARF: An Approach for Requirements Definition and Management of Data Warehouse Systems", Proceeding of the 11th IEEE International Requirements Engineering Conference, September 08 - 12 (2003), pages1090-1099.

[Rizzi et al 06] S. Rizzi, A. Abello, J. Lechtenborger, J. Trujillo "Research in data warehouse modeling and design: dead or alive?", In: DOLAP, 2006.

[Romero et al 06] O. Romero, et A. Abelló "Multidimensional Design by Examples", In Proc. of 8th Int. Conf. on Data Warehousing and Knowledge Discovery, volume 4081 of LNCS, pages 85–94. Springer, 2006.

[Romero et al 07] O. Romero, et A. Abelló "Automating Multidimensional Design from Ontologies", DOLAP"07, 1–8, (2007).

[Rouquet et al, 2010] D. Rouquet, A. Fallaise, D. Schwab, H. Blanchon, V. Bellynck, « Classifcation multilingue et multimédia pour la recherche d'images dans le projet OMNIA ». Atelier RISE. INFORSID, mai 2010

[Ruiloba 01] R. Ruiloba, Description de la structure temporelle de la vidéo : outils d'évaluation. ORASIS 2001, pp. 27-36, Cahors, 5 au 8 juin 2001.

[Ruzzi04] M. Ruzzi, "Data Integration : state of the art, new issues and research plan", 2004

[Schreiber et al 93] G. Schreiber , B. Wielinga, J. Breuker, KADS : a principled approach to knowledge-based system development. Academic Press, Londres, Royaume- Uni, 1993.

[Sen et al 05] A. Sen et A. P. Sinha "A Comparison of Data Warehousing Methodologies", Communications of the ACM. March 2005/Vol. 48, No. 3.

[Shiefer et al 02] J. Schiefer, B. List, R.M. Bruckner, "A Holistic Approach For Managing Requirements of Data Warehouse Systems", Eighth Americas Conference on Information Systems, 2002

[Stuckenschmidt et al 00] H. Stuckenschmidt, Wache H., Vogele T., Visser U. Enabling technologies for interoperability. In Ubbo Visser and Hardy Pundt, editors, Workshop on the 14th International Symposium of Computer Science for Environmental Protection, pages 35–46, Bonn, Germany. 2000.

[Sure02] Y. SURE, S. STAAB, R. STUDER, « Methodology for development and employment of ontology based knowledge management applications », *SIGMOD Rec.*, vol. 31, no 4, 2002, p.18–23, ACM Press.

[Terrasson92] J. Terrasson, « Les outils du multimédia », éditions Armand Colin, 1992.

[Tikekar et al 95] R. V. TIKEKAR, AND F. FOTOUHI, 1995, „Storage and retrieval of medical images from data warehouses"; *Digital Image Storage and Archiving Systems*.

[Ting et al, 2011] I-Hsien Ting, Chia-Hung Lin, Chen-Shu Wang, "Constructing a Cloud Computing Based Social Networks Data Warehousing and Analyzing System" in proceedings of Advances in Social Networks Analysis and Mining (ASONAM), 735 – 740, 2011

[Tola04] Martial Tola, "Présentation de l'outil IDXVIDEO", Institut des Sciences de l'Homme, CNRS Lyon, 22 juin 2004.

[Tryfona et al 99] N. Tryfona, F. Busborg, J. Christiansen "starER: A Conceptual Model for Data Warehouse Design", In DOLAP (Kansas City, Missouri, USA), November 1999.

[Verhaegen06] Boris Verhaegen, "Requêtes OLAP sur une base de données XML native", Mémoire présenté sous la direction du Prof. Esteban Zimanyi en vue de l'obtention du grade de Licencié en Informatique, 2005–2006.

[Wache et al 01] H. Wache, T. V¨ogele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. H¨ubner. Ontology-based integration of information - a survey of existing approaches. Proceedings of the International Workshop on Ontologies and Information Sharing, pages 108–117, August, 2001.

[Winter et al 04] R. Winter, B. Strauch "Information Requirements Engineering for Data Warehouse Systems". Proceedings of the 2004 ACM symposium on Applied computing, 2004.

[Xuan et al 06] D. Nguyen Xuan, L.Bellatreche, G. PIERRA, "Un modèle à base ontologique pour la gestion de l'évolution asynchrone des entrepôts de données", 6e Conférence Francophone de MOd´elisation et SIMulation - MOSIM'06 - du 3 au 5 avril 2006, Rabat Maroc.

[Yang 2000]: Yang C.C., "Detection of the Time Conflicts for SMIL-based Multimedia Presentations", Proceeding of 2000 International Computer Symposium (ICS2000)- Workshop on Computer Networks, Internet, and Multimedia, Chiayi, Taiwan, 2000.

[You et al 01] J. YOU, T. DILLON, J. LIU, AND E. PISSALOUX, 2001, « On hierarchical multimedia information retrieval", In *Proceedings of the International Conference on Image Processing*. Thessaloniki, Greece, October 7–10, 2001.

[Zaiane et al 98] O. R. ZAIANE, J. HAN, Z. H. LI, AND J. HOU, 1998, "Mining multimEdia data", In *Proceedings of CASCON, Meeting of Minds*, pp 83–96, Toronto, Canada, November 1998.

[Zhang et al 01] H. ZHANG, X. CAO, S. T. WONG, A. S. LOU, AND E. A. SICKLES, 2001,"Developing a digital mammography data warehouse", In *Proceedings of SPIE,* Vol. 4323, pp. 308–315, Medical Imaging 2001: PACS and Integrated Medical Information Systems: Design and Evaluation, Eliot L. Siegel, H. K. Huang, Eds.

[web 1] "Eléments XML de mouvement ", Guide de référence ActionScript 3.0 pour la plate-forme Adobe Flash,

http://help.adobe.com/fr_FR/FlashPlatform/reference/actionscript/3/motionXSD.html?filter_flash=cs5&filter_flashplayer=10.2&filter_air=2.6

[web 2] le journal informatique, Benchmark Group, « InStranet publie du multimédia sur le Web à partir d'entrepôts de données », Mercredi 26 avril 2000, www.journaldunet.com/solutions/00avril/000426**instranet**.shtml;

[web 3] Formation ouverte et à distance, "le multimédia et sa spécificité ", www.foad.refer.org/IMG/pdf/D229_Chapitre-1.pdf

[web 4] XQuery, http://www.w3c.org/TR/XQuery.

[web 5] "XML Format universel de documents textuels et graphiques ", http://www.xul.fr/xml.html

[web 6] Anne Doucet, "Intégration de données hétérogènes et réparties", Anne.Doucet@lip6.fr

ANNEXES

Annexes A : Compléments sur les standards de description de données multimédia et Exemples

1. Le standard SVG

1.1 Structure d'un fichier SVG

Un fichier SVG est un document structuré conforme à une grammaire XML **[Jolif06]**. Avant toute chose, il doit donc être un document XML valide et commencer par une en-tête précisant la version de XML utilisée. En l'occurrence, SVG s'appuie sur la version 1.0. Un fichier doit donc commencer par une en-tête ressemblant à :

```
<?xml version="1.0" standalone="yes"?>
```

Le deuxième attribut (*standalone*) spécifie si le document XML qui suit est un fichier XML à part (utilisation la plus générale dans le cas de SVG) ou un fragment de XML inclus dans un autre document.
L'ensemble des éléments d'un fichier SVG est décrit à l'aide de balises XML. Chaque balise doit être ouverte : <balise ...>, puis fermée : </balise> ; ou encore ouverte et fermée en son sein même : <balise ... />. Les balises reconnues par SVG sont définies dans un DTD *(document type definition).* Le fichier XML doit débuter par une référence vers ce DTD afin que l'afficheur SVG puisse vérifier la validité du document.

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20000802//EN"
"http://www.w3.org/TR/2000/CR-SVG-20000802/DTD/svg-20000802.dtd">
```

Pour mieux comprendre la structure d'un fichier SVG, nous donnons dans ce qui suit le contenu minimal d'un fichier SVG et nous citons par la suite les différentes actions qui peuvent être faites pour enrichir cette structure minimale. Ceci est illustré dans l'exemple 1.

***Exemple 1**: cet exemple représente la description d'une image (figure A.1) à l'aide d'un fichier SVG (hors en-tête).*

***Figure A.1.** Image à réaliser en SVG*

```
<svg width="640px" height="320px"
    <title>
        Carte du tour du monde en ballon
    </title>
    <desc>
        Villes étapes et trajet d'un tour du monde imaginaire en ballon
    </desc>
    <!-- Partie à remplir avec les éléments SVG à dessiner --->
</svg>
```

La première des balises SVG qui est l'élément de base de tout document est la balise <svg>. Comme tout élément XML, svg est principalement caractérisée par ses attributs et ses descendants possibles. Les attributs le plus communément définis sur un élément svg sont :

- **width** et **height** : attributs permettant d'indiquer à l'afficheur SVG la longueur et la hauteur en pixels du rectangle de dessin dans lequel l'image doit être affichée ;
- **viewBox** : si cet attribut qui définit un rectangle est présent, l'afficheur SVG doit faire en sorte que les éléments compris dans ce rectangle soient entièrement affichés dans le rectangle de dessin. Cela entraîne une éventuelle transformation supplémentaire sur les éléments pour arriver à ce but.

L'élément svg a ensuite pour descendants, l'ensemble des objets décrivant l'image. Il peut aussi avoir comme descendants un *titre* et une *description* qui ne sont pas dessinés à l'écran mais peuvent être utilisés différemment (par exemple dans une info-bulle). Sur ce fichier de base, on peut donc effectuer un ensemble d'actions :

- **Transformation des objets :** consiste à faire des modifications sur las objets (transformation affine, translation, agrandissement, rotation, etc.)
- **Objets graphiques :** Les principaux objets graphiques disponibles dans SVG sont : le rectangle avec des angles éventuellement arrondis, le cercle et l'ellipse (circle et ellipse), la ligne et la ligne multiple (line et polyline), le polygone (polygon), l'image (image), l'élément multiforme, le chemin (path), le texte (text).

Pour illustrer l'utilisation des objets graphiques dans un fichier SVG, nous reprenons l'exemple précédent (exemple 1) et nous rajoutons les éléments nécessaires pour la description de ces objets. Plusieurs éléments peuvent être regroupés sous un élément g en les plaçant comme descendants de cet élément particulier.

Dans l'exemple1 on dispose d'une image qui contient un fond de carte qui est une image JPEG. Les éléments SVG étant dessinés dans l'ordre dans lequel l'afficheur SVG les lit, il faut mettre la référence sur ce fond de carte en premier afin que tous les autres objets soient bien dessinés dessus. Une image prend comme attributs **x, y, width** et **height**, définissant le rectangle dans lequel sera affiché l'image et **xlink:href** pour spécifier l'endroit où aller chercher cette image. Le document (hors en-têtes) est donc maintenant :

```
<svg width="640px" height="320px">
      <!-- titre et description omis -->
      <image x="320" y="0" width="320" height="160"
       xlink:href="carte.jpg"/>
</svg>
```

Sur l'image de l'exemple, les villes sont représentées par un groupe d'éléments graphiques: un rectangle et une ellipse. Plutôt que de définir plusieurs fois cet ensemble, nous allons le définir une seule fois et le réutiliser. Cela se fait en SVG à travers l'élément *defs (pour définition)* descendant de *svg* .

Un rectangle prend comme attributs **x, y, width** et **height** pour définir sa taille. Une ellipse a pour attributs **cx** et **cy** pour définir son centre et **rx** et **ry** ses rayons. Nous avons donc sous l'élément *defs* la définition d'un groupe *g* les assemblant et portant un identifiant pour être réutilisable par la suite :

```
<svg width="640px" height="320px">
   <defs>
      <g id="ville">
         <rect x="0" y="0" width="10" height="8"/>
         <ellipse cx="5" cy="4" rx="5" ry="4"/>
     </g>
   </defs>
   <image x="320" y="0" width="320" height="160" xlink:href="carte.jpg"/>
   <use id="san francisco" x="374" y="46px" width="5px" height="4"
xlink:href="#ville"/>
   <use id="londres" x="476" y="31" width="5px" height="4px" xlink:href="#ville"/>
   <use id="paris" x="480" y="37" width="5px" height="4px" xlink:href="#ville"/>
   <use id="tokyo" x="601" y="46" width="5px" height="4px" xlink:href="#ville"/>
</svg>
```

Ce groupe est référencé en utilisant l'élément *use*. Cet élément prend comme attributs **xlink:href** pour pointer vers l'élément à dessiner à cet endroit et **x, y, width** et **height** pour définir sa taille. Les positions (x, y) sont définies en unités utilisateur, et les tailles (width, height) en pixels. Ainsi, lors d'un zoom logique (différent d'une loupe qui grossirait tout sans distinction), les symboles représentant les villes ne seront pas agrandis puisque définis en pixels.
Nous avons montré à travers cet exemple comment utiliser les éléments *rect* et *ellipse* pour définir des rectangles et des ellipses respectivement. Cependant, Il existe d'autres fonctionnalités qui peuvent être rajoutées telles que l'ajout de couleurs ou d'une animation.

2. Exemple d'une interpolation avec Flash

L'exemple suivant présente le code ActionScript avec XML intégré décrivant l'interpolation de mouvement d'une occurrence de clip moveShape pour le symbole myShape, lorsqu'il effectue une rotation, se déplace, utilise un paramètre d'accélération personnalisé et modifie les valeurs alpha sur dix images :

```
import fl.motion.Animator;
var moveShape_xml:XML = <Motion duration="10" xmlns="fl.motion.*"
xmlns:geom="flash.geom.*" xmlns:filters="flash.filters.*">
    <source>
        <Source frameRate="12" x="41.35" y="91.35" scaleX="1" scaleY="1"
          rotation="0" elementType="movie clip" instanceName="moveShape"
          symbolName="myShape">
            <dimensions>
                <geom:Rectangle left="-46.65" top="-61.95" width="133.05"
                    height="133.95"/>
            </dimensions>
            <transformationPoint>
               <geom:Point x="0.49981210071401727" y="0.4998133631952222"/>
            </transformationPoint>
        </Source>
    </source>
    <Keyframe index="0" rotateTimes="2">
      <tweens>
         <CustomEase>
           <BezierControl x="0.08650266979261687" y="0.14705453864744866"/>
           <BezierControl x="0.23675978562091857" y="0.28829454738109694"/>
           <BezierNode x="0.2689728109485753" y="0.49688733564952436"/>
           <BezierControl x="0.32093023255813957" y="0.8333333333333333"/>
           <BezierControl x="0.5988021982960045" y="1.034249160488573"/>
           <BezierNode x="0.7309082984924317" y="0.8685852488735627"/>
           <BezierControl x="0.8116279069767443" y="0.767361111111111"/>
```

```
          <BezierControl x="0.910302766164144" y="0.9730908298492431"/>
        </CustomEase>
      </tweens>
   </Keyframe>
   <Keyframe index="9" x="371.95" y="188">
        <color>
            <Color alphaMultiplier="0.4" alphaOffset="0"/>
        </color>
    </Keyframe>
</Motion>;
var moveShape_animator:Animator = new Animator(moveShape_xml, moveShape);
moveShape_animator.play();
```

3. Exemples de description d'images et de séquences vidéo avec MPEG-7

Exemple 1 : *description d'une image*

Cet exemple montre la description d'une image (figure A.2) d'une personne par le biais d'un document MPEG-7 **[Bruno et al, 02]**. Ce dernier est décomposé en deux parties :

- une première partie qui contient des informations sémantique (descripteur sémantiques) sur la photo telles que la référence de la photo, son auteur, la date et le lieu de la prise de la photo, les mots clés, etc.
- une deuxième partie qui contient des descripteurs extraits de la photo par analyse d'image (descripteurs visuels) tels que la couleur et la texture.

Figure A.2. *Image à décrire avec MPEG-7*

```
<DescriptionMetadata>
     <PublicIdentifier>
          <UniqueID>A.DEL.16</UniqueID>
     </PublicIdentifier>
     <PrivateIdentifier>lef0102013C</PrivateIdentifier>
     <CreationTime>2001-8-29</CreationTime>
</DescriptionMetadata>
<ContentDescription>
     <Title>municipales 2001 B.Delanoë présente la liste du18ème</Title>
     <Abstract>
          <KeywordAnnotation>
            <Keyword>POLITIQUE</Keyword>
            <Keyword>Paris (75)</Keyword>
          </KeywordAnnotation>
         <KeywordAnnotation>
            <Keyword>municipales</Keyword>
            <Keyword>2001</Keyword>
            <Keyword>Delanoë</Keyword>
            <Keyword>18ème</Keyword>
         </KeywordAnnotation>
```

```
      </Abstract>
      <Creator>
          <Agent>
             <Name>
                 <GivenName>Didier</GivenName>
                 <FamilyName abbrev="lef">Lefèvre</FamilyName>
             </Name>
          </Agent>
      </Creator>
      <CreationCoordinates>
          <CreationDate>
             <TimePoint>2001-02-11</TimePoint>
          </CreationDate>
      </CreationCoordinates>
      <MediaInstance>
           <LocationDescription>POLITIQUE 6</LocationDescription>
      </MediaInstance>
  </ContentDescription>
```

***Listing**- Identification et description sémantique de la photo en MPEG-7*

Cette première partie du document contient des informations qui décrivent le « contenant » : référence de la photo, son auteur, le lieu et la date de la prise de vue, le support, etc. et le « contenu » à l'aide de mots-clés.

```
<Descriptor>
     <DayNight>nuit</DayNight>
     <Orientation>horizontal</Orientation>
     <ShotType>général</ShotType>
     <IntExt>intérieur</IntExt>
     <ScalableColor numberOfCoefficients="63">
            <Coefficients> 1 1 0 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 1
                           1 0 0 1 0 0 1 0 0 0   0 1 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0
                           0 0 1 0 1 0 0 <Coefficients>
     </ScalableColor>
     <ColorLayout numOfYCoef="64">
            <Ycoeff>
                 <YDCCoef>13</YDCCoeff>
                     <YACCoef> 27 23 2 16 10 14 16 9 9 17 14 13 16 16 16 16 14 15
                               17 17 16 16 17 16 14 15 17 17 17 18 15 18 15 15 14
                               17 15 17 16 17 16 16 16 16 16 16 17 17 15 17 16 16 16
                               16 16 16 17 15 17 16 17 16 </YACCoef>
            </Ycoeff>
            <CbCoeff>
                    <CbDCCoef>0</CbDCCoef>
                    <CbACCoef> 17 17 14 17 16 15 15 16 18 16 16 18 15 17 16 16 17
                        16 16 14 16 14 16 16 16 16 16 16 16 16 16 16 16 16 16 17 17 16
                        17 16 16 16 16 16 16 16 16 16 16 16 16 17 16 16 16 16 16 16
                        16 16 16 16 16 16
                    </CbACCoef>
            </CbCoeff>
            <CrCoeff>
                    <CrDCCoef>0</CrDCCoef>
                    <CrACCoef> 15 18 23 13 15 17 17 21 12 12 18 12 17 15 16 17 14
                            17 15 16 14 19 16 17 15 17 16 16 16 16 16 17 18 14 16
                            15 17 15 16 17 16 17 16 16 16 16 16 16 15 17 15 16 16
                            16 16 16 16 17 17 16 16 16 16
                   </CrACCoef>
            </CrCoeff>
     </ColorLayout>
     <DominantColor size="8">
            <ColorSpace type="RGB"/>
            <SpatialCoherency>0.37222583333333336</SpatialCoherency>
            <Values>
                  <Percentage>0.0838</Percentage>
                  <ColorValueIndex>7 2 1</ColorValueIndex>
```

```
                <ColorVariance>23161.6189061638 36972.29892632513
             1056.2499991597056    </ ColorVariance>
          </Values>
                        …
    </DominantColor>
</Descriptor>
```

Listing *- Description visuelle d'une photo en MPEG-7*

Dans cette deuxième partie du document, sept descripteurs visuels ont été retenus : trois descripteurs de bas niveau caractérisant les couleurs d'une photo, et quatre descripteurs de haut niveau, indiquant pour chaque photo : si elle a été prise à l'intérieur ou à l'extérieur ou bien de jour ou de nuit, son orientation, et le cadrage des personnages qui y apparaissent.

Les descripteurs visuels de haut niveau sont directement interrogeables, alors que les descripteurs de bas niveau sont uniquement utilisés pour mesurer la similarité de deux photos.

Exemple 2 : annotation d'une séquence vidéo

Cet exemple est tiré d'un travail **[Tola04]** qui consiste à concevoir un outil d'indexation de vidéo basé sur la norme MPEG-7**.**

L'étape d'indexation permet de décomposer les médias en segments temporels sous forme hiérarchique, puis de les annoter textuellement en utilisant les formats « Free Text annotations», « Keyword annotations » et « Structured annotations » du type de données « Text Annotation » de la norme MPEG-7.

```
    <TextAnnotation>
            <FreeTextAnnotation>PDG</FreeTextAnnotation>
        </TextAnnotation>
        <TextAnnotation>
            <StructuredAnnotation>
            <Who>
            <Name>PDG</Name>
            </Who>
        <WhatObject>
            <Name></Name>
            </WhatObject>
            <WhatAction>
            <Name>parle</Name>
            </WhatAction>
            <Where>
            <Name>bureau</Name>
            </Where>
            <When>
            <Name></Name>
        </When>
            <Why>
            <Name></Name>
        </Why>
            <How>
            <Name></Name>
            </How>
            </StructuredAnnotation>
        </TextAnnotation>
        <TextAnnotation>
            <KeywordAnnotation>
            <Keyword>assis, chaise, bureau</Keyword>
            </KeywordAnnotation>
        </TextAnnotation>
```

Listing*- Exemple d'annotation d'une séquence vidéo utilisant les différents formats disponibles du type de données « Text Annotation » de la norme MPEG-7*

4. Exemples de présentations SMIL

Exemple 1 : *image et texte s'enchaînent*

```
<smil>
<head>
<meta name="title" content="Image et texte s'enchaînent..." />
<layout> <root-layout width="330" height="300" />
<region id="i"top="20" left="90" width="150" height="60" z-index="1"/>
<region id="t1" top="100" left="5" width="330" height="400" z-index="2"/>
<region id="t2" z-index="3"/>
</layout>
</head>
<body>
<seq>
<par>
<a href="http://real-and-smil.com" show="new">
<img src="http://monsite.com/monimage.jpg" region="i" fill="freeze" /></a>
<text src="texte.rt" region="t1" begin="3" fill="freeze"/>
</par>
<text src="texte2.rt" region="t2" begin="1" fill="freeze"/>
</seq>
</body>
</smil>
```

Le premier élément intéressant de ce fichier est de montrer qu'il est possible d'enchâsser les balises seq et par. C'est-à-dire que la présentation se découpe en deux grandes parties. Comme vous avez pu le voir une image est jouée avec un texte dessous puis une partie du code est présenté. On a donc entre les balises seq image+texte1 puis texte2. Mais image+texte 1 doivent être joué simultanément on les place entre des balises par. Ce principe n'a pas de limite, on pourrait imaginer que la partie parallèle se découpe elle même en sous-parties séquentielles.

Exemple 2 : *le son de la chanson pendant les paroles défilent*

```
<smil>
<head>
<meta name="title" content="Texte, image et les Greyhounds" />
<layout>
<root-layout width="260" height="400" />
<region id="i" top="5" left="70" width="115" height="120" z-index="1"/>
<region id="t1" top="130" left="5" width="240" height="265" z-index="2"/>
<region id="t2" z-index="3"/>
</layout>
</head>
</body>
<seq>
<par>
<img src="image.rp" region="i" fill="freeze" />
<audio src="http://monsite.com/ma_chanson.rm" />
<text src="texte.rt" region="t1" fill="freeze"/>
</par>
<text src="texte2.rt" region="t2" begin="3" fill="freeze"/>
</seq>
</seq>
</body>
</smil>
```

Sans définition de durée l'image et le texte ne disparaissent qu'après la durée du fichier son.
L'intérêt de cet exemple se cache donc dans le fichier rp. (realpix) qui sert à agencer les images.

```
<imfl>
<head timeformat="dd:hh:mm:ss.xyz"
duration="30.0"
bitrate="12000"
width="115"
height="120"
preroll="10.0"
/>
<image handle="1" name="image1.jpg" mime="image/jpeg"/>
<image handle="2" name="image2.jpg" mime="image/jpeg"/>
<image handle="3" name="image3.jpg" mime="image/jpeg"/>

<crossfade start="0.0" duration="1.0" target="1" dstx="0" dsty="0" dstw="320"
dsth="240" aspect="true" />
<crossfade start="7.0" duration="1.0" target="2" dstx="0" dsty="0" dstw="320"
dsth="240" aspect="true" />
<crossfade start="14.0" duration="1.0" target="3" dstx="0" dsty="0" dstw="320"
dsth="240" aspect="true" />
<crossfade start="21.0" duration="1.0" target="1" dstx="0" dsty="0" dstw="320"
dsth="240" aspect="true" />

</imfl>
```

Dans la partie *head* on retrouve les méta-informations, sur la durée, la taille, la fonction, ... Chaque image est définie par un identifiant unique. Start désigne évidement la durée qui sépare l'apparition de l'image du début de la présentation dstx, dsty sont des informations semblables à celle apportées par left, top etc dans l'exemple précédent. On remarquera qu'elles sont toutes identiques afin que les images soient jouées exactement au même endroit.

Il existe un grand nombre d'effets de transition et autre possible. Et ce d'autant plus depuis l'arrivée du smil2.0. Vous en trouverez une liste quasi exhaustive en parcourant les liens suivant :

sur le site de realnetwork et sur le site du wc3 (xml)

notes: cela implique l'ajout d'une ligne de code telle que celle-ci

<smil xmlns="http://www.w3.org/2001/SMIL20/Language">

Le tableau suivant donne la liste exhaustive des éléments qui peuvent être appelés.

Éléments	**Définition**
Text	texte statique (.txt).
textstream	texte streamé RealText clips (.rt).
Img	image : .jpg ou .gif (ne pas utiliser de JPEG progresif.
Audio	son: .rm, .wma, .mov et .mpeg.
Video	video :.rm, .rmvb, .avi, .wma, .mov et .mpeg.
animation	animation Shockwave Flash (.swf)
Ref	Tout se qui ne rentre pas dans les autre catégorie par exemple les fichier .rp

Annexes B : Langages et Outils d'Edition d'Ontologies

1. Langages de définition et de manipulation d'ontologies

Il existe actuellement de nombreux langages de définition des ontologies parmi lesquels nous citons [Guergour 07] :

- ***RDF (Resource Description Framework)*** **:** le W3C a établi un cadre général pour la définition de toutes sortes de méta données. Il s'agit de RDF : langage de description d'informations sur le web.

  Le modèle RDF définit trois types d'objets :

  - Des ressouces : représente tous les objets décrit par RDF, et qui sont identifiés par leur URI (Uniform Ressource Identifier) ;

  - Des propriétés : une propriété est un attribut, un aspect, une caractéristique qui s'applique à une ressource. Il peut également s'agir d'une mise en relation avec une autre ressource.

  - Des valeurs : ce sont les valeurs particulières que peuvent prendre les propriétés.

Ces trois types d'objets peuvent être mis en relation par des assertions, c'est-à-dire des triplets (ressource, propriété, valeur), ou encore (sujet, prédicat, objet). Une description RDF est une suite d'assertions.

L'exemple ci-dessous illustre une assertion RDF :

Supposons les trois objets RDF :
- Une ressource : http://www.limsi.dz/ ;
- Une propriété : date_de_modification ;
- Une valeur : 11-février-2003.

On peut alors construire l'assertion : (http//www.limsi.dz/, date_de_modification, 11- février -2003), ce qui signifie : la date de modification de la page web HMSC est le 11 février 2003, et qui peut être représentés par le graphe suivant :

- ***RDFS (pour RDF schéma)*** **:** a pour but d'étendre ce langage en décrivant plus précisément les ressources utilisées afin d'étiqueter les graphes, pour cela, il fournit un mécanisme permettant de spécifier les classes dont les ressources seront des instances, comme propriétés : RDFS s'écrit toujours à l'aide de triplets RDF, en définissant la sémantique de nouveaux mot clés.

- ***DAML + OIL (DARPA Agent Markup Language + Ontology Inference Layer)*** **:** la sémantique sur le web est très riche. Il en résulte que la sémantique des classes, sous classes, propriétés, et sous propriétés de RDF(S) ne suffit pas pour la modélisation dans le web sémantique. OIL (Ontology Inference Layer) se présente dans une approche en couche, un langage standard d'ontologie, où chaque couche supplémentaire ajoute des fonctionnalités à la couche précédente.

  DAML est conjointement utilisé avec OIL pour créer un langage d'ontologie pour le web sémantique avec plusieurs couches.

- ***OWL (Ontology Web Langage)*** **:** OWL est le dernier né du W3C dans le domaine de description d'ontologie pour le web, OWL est construit au dessous de RDF, un document OWL est donc composé de triplets RDF, qui peuvent être écrits dans la syntaxe RDF de son choix. Tous comme DAML+OIL, OWL permet de faire des raisonnements, des inférences sur les ontologies. Ainsi il permettra de conclure sur des faits implicites.

OWL propose trois sous langages, à savoir :

- OWL Lite : c'est une vraie version d'OWL aux fonctionnalités réduites, mais suffisantes pour certains usages, par exemple pour la constitution de thésaurus.

- OWL DL : il correspond exactement aux logiques de description. Ce langage est expressif, cependant les procédure d'inférences sont complètes, et effectuables en un temps fini.

- OWL Full : il est considéré comme le plus expressif des sous langages OWL, il n'est d'ailleurs pas possible de réaliser des procédures de raisonnements automatique sur OWL Full [KMN03], il permet à une ontologie d'augmenter la signification de son vocabulaire.

2 Le langage OWL

Web Ontology Language (OWL) doit son nom au terme « ontologie ».

2.1 Objectifs et motivation

OWL est, tout comme RDF, un langage XML profitant de l'universalité syntaxique de XML.
Fondé sur la syntaxe de RDF/XML, OWL offre un moyen d'écrire des ontologies web. OWL se différencie du couple RDF/RDFS en ceci que, contrairement à RDF, il est justement un langage d'ontologies. SI RDF et RDFS apportent à l'utilisateur la capacité de décrire des classes (ie. avec des contructeurs) et des propriétés, OWL intègre, en plus, des outils de comparaison des propriétés et des classes : identité, équivalence, contraire, cardinalité, symétrie, transitivité, disjonction, etc.
Ainsi, OWL offre aux machines une plus grande capacité d'interprétation du contenu web que RDF et RDFS, grace à un vocabulaire plus large et à une vraie sémantique formelle.

2.2 Document OWL

Tout comme RDF dispose d'une structure logique dont RDF/XML est la sérialisation, les ontologies OWL se présentent sous forme de fichiers texte, de « documents OWL ». La création d'un document OWL fait l'objet de diverses recommandations :

2.2.1 Espaces de nommage de OWL
L'intégralité du vocabulaire de OWL provient de l'espace de nom de OWL,

2.2.2 Type MIME
Il n'existe aucun type MIME spécifique à OWL, un document OWL étant un document RDF. Il est donc recommandé d'employer le type MIME lié à RDF, à savoir application/rdf+xml ou, à la rigueur, le type MIME de XML, application/xml.

2.3.3 Extension
La recommandation du groupe de travail WebOnt indique qu'il est préférable d'employer, comme suffixe de nom de fichier, les extensions « .rdf » ou « .owl ».

2.4 Structure d'une ontologie OWL

Cette partie indique les principaux éléments constituant une ontologie OWL. Avant de poursuivre, il est bien nécessaire de comprendre qu'OWL n'a pas été conçu dans un esprit fermé permettant de borner les frontières d'une ontologie. Au contraire, la conception d'OWL a pris en compte la nature distribuée du web sémantique et, de ce fait, a intégré la possibilité d'étendre des ontologies existantes, ou d'employer diverses ontologies existantes pour compléter la définition d'une nouvelle ontologie.

2.4.1 Espaces de nommage

Afin de pouvoir employer des termes dans une ontologie, il est nécessaire d'indiquer avec précision de quels vocabulaires ces termes proviennent. C'est la raison pour laquelle, comme tout autre document XML, une ontologie commence par une déclaration d'espace de nom (parfois appelée « de nommage ») contenue dans une balise rdf:RDF. Supposons que nous souhaitons écrire une ontologie sur une population de personnes ou, d'une manière plus générale, sur l'humanite. Voici la déclaration d'espace de nom qui pourrait être employée :

```
<rdf:RDF
xmlns = "http://domain.tld/path/humanite#"
xmlns:humanite= "http://domain.tld/path/humanite#"
xmlns:base = "http://domain.tld/path/humanite#"
xmlns:vivant = "http://otherdomain.tld/otherpath/vivant#"
xmlns:owl = "http://www.w3.org/2002/07/owl#"
xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
xmlns:xsd = "http://www.w3.org/2001/XMLSchema#">
```

Les deux premières déclarations identifient l'espace de nommage propre à l'ontologie que nous sommes en train d'écrire. La première déclaration d'espace de nom indique à quelle ontologie se rapporter en cas d'utilisation de noms sans préfixe dans la suite de l'ontologie. La troisième déclaration identifie l'URI de base de l'ontologie courante.

La quatrième déclaration signifie simplement que, au cours de la rédaction de l'ontologie humanité, on va employer des concepts développés dans une ontologie vivant, qui décrit ce qu'est un être vivant. Les quatre dernières déclaration introduisent le vocabulaire d'OWL et les objets définis dans l'espace de nommage de RDF, du schéma RDF et des types de données du Schéma XML.

2.4.2 En-têtes d'une ontologie

Tout comme il existe une section d'en-tête <head>..</head> en haut de tout document XHTML bien formé, on peut écrire, à la suite de la déclaration d'espaces de nom, un en-tête décrivant le contenu de l'ontologie courante. C'est la balise owl:Ontology qui permet d'indiquer ces informations :

```
<owl:Ontology rdf:about="">
<rdfs:comment>Ontologie décrivant l'humanité</rdfs:comment>
<owl:imports
rdf:resource="http://otherdomain.tld/otherpath/vivant"/>
<rdfs:label>Ontologie sur l'humanité</rdfs:label>
 ...
```

2.5 Eléments du langage

Cette partie ne va pas reprendre toutes les finesses de OWL Lite, OWL DL et OWL Full, mais uniquement les plus importantes.

2.5.1 Les classes

Une classe définit un groupe d'individus qui sont réunis parce qu'ils ont des caractéristiques similaires. L'ensemble des individus d'une classe est désigné par le terme « extension de classe », chacun de ces individus étant alors une « instance » de la classe. Les trois versions d'OWL comportent les mêmes mécanismes de classe, à ceci près que OWL FULL est la seule version à permettre qu'une classe soit l'instance d'une autre classe (d'une métaclasse). A l'inverse, OWL Lite et OWL DL n'autorisent pas qu'une instance de classe soit elle-même une classe.

2.5.1.1 Déclaration de classe

La déclaration d'une classe se fait par le biais du mécanisme de « description de classe », qui se présente sous diverses formes. Une classe peut ainsi se déclarer de six manières différentes :

• *l'indicateur de classe :*

La description de la classe se fait, dans ce cas, directement par le nommage de cette classe. Une classe « humain » se déclare de la manière suivante :

```
<owl:Class rdf:ID="Humain" />
```

Il est à noter que ce type de description de classe est le seul qui permette de nommer une classe. Dans les cinq autres cas, la description représente une classe dite « anonyme », crée en plaçant des contraintes sur son extension.

• ***l'énumération des individus composant la classe***

Ce type de description se fait en énumérant les instances de la classe, à l'aide de la propriété owl:oneOf :

```
<owl:Class>
<owl:oneOf rdf:parseType="Collection">
<owl:Thing rdf:about="#Damien" />
<owl:Thing rdf:about="#Olivier" />
<owl:Thing rdf:about="#Philippe" />
<owl:Thing rdf:about="#Xavier" />
<owl:Thing rdf:about="#Yves" />
</owl:oneOf>
</owl:Class>
```

• ***La restriction de propriétés***

La description par restriction de propriété permet de définir une classe anonyme composée de toutes les instances de owl:Thing qui satisfont une ou plusieurs propriétés.
Ces contraintes peuvent être de deux types : contrainte de valeur ou contrainte de cardinalité
Une contrainte de valeur s'exerce sur la valeur d'une certaine propriété de l'individu (par exemple, pour un individu de la classe Humain, sexe = Homme), tandis qu'une contrainte de cardinalité porte sur le nombre de valeurs que peut prendre une propriété (par exemple, pour un individu de la classe Humain, aPourFrere est une propriété qui peut ne pas avoir de valeur, ou avoir plusieurs valeurs, suivant le nombre de frères de l'individu. La contrainte de cardinalité portant sur aPourFrere restreindra donc la classe décrite aux individus pour lesquels la propriété aPourFrere apparaît un certain nombre de fois).
Il existe naturellement différents opérateurs de comparaison pour établir les contraintes. Pour plus de détails, se reporter à « OWL Web Ontology Language Reference »

• ***Intersection, union ou complémentaire de classes***

Enfin, les descriptions par intersection, union ou complémentaire permettent de décrire une classe par, comme leur nom l'indique, l'intersection, l'union ou le complémentaire d'autres classes déjà définies, ou dont la définition se fait au sein même de la définition de la classe courante :

```
<owl:Class>
<owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="#etudiantsENST" />
<owl:Restriction>
<owl:onProperty rdf:resource="#aPourFrere" />
<owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">
</owl:cardinality>
</owl:Restriction >
</owl:intersectionOf>
</owl:Class>
```

2.5.1.2 Héritage

Il existe dans toute ontologie OWL une superclasse, nommée Thing, dont toutes les autres classes sont des sous-classes. Ceci nous ammène directement au concept d'héritage, disponible à l'aide de la propriété subClassOf :

```
<owl:Class rdf:ID="Humain">
<rdfs:subClassOf rdf:resource="&etre;#EtreVivant" />
</owl:Class>
<owl:Class rdf:ID="Homme">
<rdfs:subClassOf rdf:resource="#Humain" />
</owl:Class>
<owl:Class rdf:ID="Femme">
<rdfs:subClassOf rdf:resource="#Humain" />
</owl:Class>
```

Enfin, il existe également une classe nommée noThing, qui est sous-classe de toutes les classes OWL. Cette classe ne peut avoir aucune instance.

2.5.2 Les instances de classe

2.5.2.1 Définition d'un individu

La définition d'un individu consiste à énoncer un « fait », encore appelé « axiome d'individu ».
On peut distinguer deux types de faits :

• ***les faits concernant l'appartenance à une classe***
La plupart des faits concerne généralement la déclaration de l'appartenance à une classe d'un individu et les valeurs de propriété de cet individu. Un fait s'exprime de la manière suivante :

```
<Humain rdf:ID="Pierre">
<aPourPere rdf:resource="#Jacques" />
<aPourFrere rdf:resource="#Paul" />
</Humain>
```

Le fait écrit dans cet exemple exprime l'existence d'un Humain nommé « Pierre » dont le père s'appelle « Jacques », et qu'il a un frère nommé « Paul ». On peut également instancier un individu anonyme en omettant son identifiant :

```
<Humain>
<aPourPere rdf:resource="#Jacques" />
<aPourFrere rdf:resource="#Paul" />
</Humain>
```

Ce fait décrit, dans ce cas, l'existence d'un Humain dont le père se nomme « Jacques » et qui a un frère nommé « Paul ».

• ***les faits concernant l'identité des individus***

Une difficulté qui peut éventuellement apparaître dans le nommage des individus
concerne la non-unicité éventuelle des noms attribués aux individus. Par exemple, un même individu pourrait être désigné de plusieurs façons différentes.
C'est la raison pour laquelle OWL propose un mécanisme permettant de lever cette ambiguité, à l'aide des propriétés owl:sameAs, owl:diffrentFrom et owl:allDifferent. L'exemple suivant permet de déclarer que les noms « Louis_XIV » et « Le_Roi_Soleil » désignent la même personne :

```
<rdf:Description rdf:about="#Louis_XIV">
<owl:sameAs rdf:resource="#Le_Roi_Soleil" />
</rdf:Description>
```

2.5.2.2 Affecter une propriété à un individu

Une fois que l'on sait écrire une classe en OWL, la création d'une ontologie se fait par l'écriture d'instances de ces objets, et la description des relations qui lient ces instances.

2.5.3 Les propriétés

Exprimer les faits au sujet des classes et de leurs instancespeut se faire par le biais de propriétés OWL. OWL fait la distinction entre deux types de propriétés :
• ***les propriétés d'objet*** permettent de relier des instances à d'autres instances
• ***les propriétés de type de donnée*** permettent de relier des individus à des valeurs de données.

Une propriété d'objet est une instance de la classe owl:ObjectProperty, une propriété de type de donnée étant une instance de la classe owl:DatatypeProperty. Ces deux classes sont elles-même sous-classes de la classe RDF rdf:Property.
La définition des caractéristiques d'une propriété se fait à l'aide d'un axiome de propriété qui, dans sa forme minimale, ne fait qu'affirmer l'existence de la propriété :
<owl:ObjectProperty rdf:ID="aPourParent" />
Cependant, il est possible de définir beaucoup d'autres caractéristiques d'une propriété dans un axiome de propriété.

2.5.3.1 Définition d'une propriété

Afin de spécifier une propriété, il existe différentes manières de restreindre la relation qu'elle
symbolise. Par exemple, si on considère que l'existence d'une propriété pour un individu donné de l'ontologie constitue une fonction faisant correspondre à cet individu un autre individu ou une valeur de donnée, alors on peut préciser le domaine et l'image de la propriété. Une propriété peut également être définie comme la spécialisation d'une autre propriété.

```
<owl:ObjectProperty rdf:ID="habite">
<rdfs:domain rdf:resource="#Humain" />
<rdfs:range rdf:resource="#Pays" />
</owl:ObjectProperty>
```

Par exemple, on peut définir la propriété de type de données anneeDeNaissance :

```
<owl:Class rdf:ID="dateDeNaissance" />
<owl:DatatypeProperty rdf:ID="anneeDeNaissance">
<rdfs:domain rdf:resource="#dateDeNaissance" />
<rdfs:range rdf:resource="&xsd;positiveInteger"/>
```

On peut également employer un mécanisme de hiérarchie entre les propriétés, exactement comme il existe un mécanisme d'héritage sur les classes :

```
<owl:Class rdf:ID="Humain" />
<owl:ObjectProperty rdf:ID="estDeLaFamilleDe">
<rdfs:domain rdf:resource="#Humain" />
<rdfs:range rdf:resource="#Humain" />
 </owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="aPourFrere">
<rdfs:subPropertyOf rdf:resource="#estDeLaFamilleDe" />
<rdfs:range rdf:resource="#Humain" />
...
</owl:ObjectProperty>
</owl:Class>
```

La propriété aPourFrere est une sous-propriété de estDeLaFamilleDe, ce qui signifie que toute entité ayant une propriété aPourFrere d'une certaine valeur a aussi une propriété estDeLaFamilleDe de même valeur.

2.5.3.2 Caractéristiques des propriétés

En plus de ce mécanisme d'héritage et de restriction du domaine et de l'image d'une propriété, il existe divers moyens d'attacher des caractéristiques aux propriétés, ce qui permet d'affiner grandement la qualité des raisonnements liés à cette propriété.
Parmi les caractéristiques de propriétés principales, on trouve la transitivité, la symétrie, la fonctionnalité, l'inverse, etc. L'ajout d'une caractéristique à une propriété de l'ontologie se fait par l'emploi de la balise OWL type :

```
<owl:ObjectProperty rdf:ID="aPourPere">
<rdfs:domain rdf:resource="#Humain" />
<rdfs:range rdf:resource="#Humain" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="aPourFrere">
<rdf:type rdf:resource="&owl;SymmetricProperty" />
<rdfs:domain rdf:resource="#Humain" />
<rdfs:range rdf:resource="#Humain" />
</owl:ObjectProperty>
<Humain rdf:ID="Pierre">
<aPourFrere rdf:resource="#Paul" />
<aPourPere rdf:resource="#Jacques" />
</Humain>
```

3. Outils d'édition d'ontologies

Il existe actuellement de nombreux outils d'édition d'ontologies. Les éditeurs d'ontologie suivants sont gratuits et téléchargeables :

- Protégé est le plus connu et le plus utilisé des éditeurs d'ontologie. Open-source, développé par l'université Stanford, il a évolué depuis ses premières versions (Protégé-2000) pour intégrer à partir de 2003 les standards du Web sémantique et notamment OWL. Il offre de nombreux composants optionnels : raisonneurs, interfaces graphiques.
- **(en)** SWOOP est un éditeur d'ontologie développé par l'Université du Maryland dans le cadre du projet MINDSWAP. Contrairement à Protégé, il a été développé de façon native sur les standards RDF et OWL, qu'il prend en charge dans leurs différentes syntaxes (pas seulement XML). C'est une application plus légère que Protégé, moins évoluée en termes d'interface, mais qui intègre aussi des outils de raisonnement.
- KMgen est un éditeur d'ontologie pour le langage KM (KM: The Knowledge Machine).

OntoEdit (Ontology Editor) **:** il permet l'édition des hiérarchies des concepts, des relations et les expressions d'axiomes algébriques portant sur les relations, et des propriétés telles que la généricité d'un concept. Il inclut plusieurs outils graphiques dédiés à la visualisation d'ontologies. OntoEdit intègre notamment un serveur destiné à l'édition d'une ontologie par plusieurs utilisateurs.

Protégé 2000 **:** est une interface modulaire permettant l'édition, la visualisation, le contrôle (vérification des contraintes) d'ontologies, l'extraction d'ontologies à partir de sources textuelles, et la fusion semi automatique d'ontologies. Le modèle de connaissance sous jacent à Protégé 2000 est issu du modèle des *frames* et contient des classes (concepts), des *slots* (propriétés) et des *facettes* (valeurs des propriétés et contraintes), ainsi que des instances des classes et des propriétés.

Le tableau ci-dessous illustre une comparaison entre les deux outils :

	Protégé 2000	OntoEdit
Produit	Libre.	La plupart des fonctionnalités sont sur version non libre.
Ontologie générique	Quelques une (Dublin Core), difficile de se baser sur eux pour construire une nouvelle ontologie.	Aucune ontologie générique pour la réutilisation.
Documentations.	Plusieurs guides indiquant : comment construire une ontologie dans le sens général.	Quelques exemples sont disponibles pour comprendre ses caractéristiques.
Assistants.	Il existe plusieurs assistant (protégé utilisateurs, protégé-discussion, protégé –owl) très utiles.	Guide et tutorial sont suffisantes pour comprendre les différentes fonctionnalités et leur utilisation.
Développement	International protégé Workshop chaque années, il rassemble de nombreux chercheurs pour développer et utiliser des méthodologies et outils de protégé.	Seulement la version non libre a pas mal de nouvelles fonctionnalités.
Synchronisation entre les éditeurs.	installé en local, pas de synchronisation.	Installé localement n'est pas synchrone entre deux éditeurs.
Outils d'importation / d'exportation	Suivant plusieurs format : fichier texte, Tableau JDBC, RDF, OWL, etc.	Permet d'importer des ontologie AML-OIL, RDF, et d'exporter sous plusieurs formats : RDF, DTD, XML schéma, etc.
En général	Usage facile pour les utilisateurs novices pour écrire la connaissance d'assertion pour la base des connaissances.	Difficile à étendre.
Outils de génération de la documentation	Il inclut un plug in permettant la génération automatiquement de documentation sur le format Html.	Non disponible.

Tableau B.1. *Comparaison entre les outils protégé et OntoEdit.*