Rapport de DEA

Intégration de versions fonctionnelles dans les entrepôts de données multimédias au sein des systèmes OLAP

Anne-Muriel ARIGON

LIRIS – INSA de Lyon
Bâtiment 501
69621 Villeurbanne, France

Encadré par :
Maryvonne Miquel
Anne Tchounikine

Résumé :
Les entrepôts de données et les systèmes OLAP sont de plus en plus utilisés car ils proposent des architectures et des outils pour organiser, analyser et exploiter de grands volumes de données et améliorer ainsi la prise de décision. Les données entreposées sont intégrées dans des modèles multidimensionnels organisés selon le sujet analysé, appelé fait, et des axes d'analyse, nommés dimensions. Les entrepôts classiques ont une structure statique où seuls les faits sont dynamiques. Ces entrepôts intègrent généralement des données alphanumériques. Nous nous intéressons plus particulièrement aux données multimédias souvent caractérisées par des descripteurs. Plusieurs problèmes se posent : le stockage de données particulièrement volumineuses et nécessitant des outils spécifiques de visualisation, la modélisation de faits multimédias et la définition de fonctions d'agrégat spécifiques, et enfin le calcul et la modélisation de descripteurs comme dimensions de l'entrepôt. Or ces descripteurs peuvent être obtenus par divers modes de calcul que nous définissons comme des « versions fonctionnelles » de descripteurs. Nous proposons un modèle multidimensionnel multiversion fonctionnelle appelé « modèle M2F » en intégrant notamment la notion de « version de dimension » qui représente des dimensions dont les membres sont calculés selon les différentes versions fonctionnelles des descripteurs. Cette nouvelle approche permet d'intégrer au modèle un choix de modes de calculs de ces descripteurs afin de permettre à l'utilisateur de choisir la représentation de données la plus adaptée à son analyse. Nous mettons en œuvre un entrepôt de données multimédias dans le domaine médical en intégrant à un modèle multidimensionnel les données multimédias d'un essai thérapeutique. Nous définissons un modèle conceptuel et présentons les modèles logiques et physiques pour l'implémentation de notre approche. Enfin nous décrivons le prototype réalisé et les possibilités de visualisation des données dans une interface OLAP.

Mots clefs :
Entrepôt de données, OLAP (On-Line Analytical Processing), Données multimédias, Mode de calcul, Version fonctionnelle de descripteur, Version de dimension, modèle conceptuel.

Sommaire

1. Introduction 3
2. Etat de l'art 4
2.1. LES ENTREPOTS DE DONNEES 4
2.2. LES ENTREPOTS DE DONNEES MULTIMEDIAS 5
2.3. LES VERSIONS TEMPORELLES DANS LES ENTREPOTS DE DONNEES 8
3. Travaux effectués 8
3.1. OBJECTIFS 8
3.2. MODELE CONCEPTUEL M2F 11
3.3. MODELE LOGIQUE 17
3.4. MODELE PHYSIQUE ET IMPLEMENTATION 18
3.5. MISE EN ŒUVRE SUR L'ETUDE EMIAT 21
3.6. L'INTERFACE OLAP 22
4. Discussion et conclusion 25
4.1. APPORTS DE NOTRE APPROCHE 25
4.2. LIMITES DE NOTRE MODELE 25
4.3. PERSPECTIVES 25
5. Références bibliographiques 27

1. Introduction

Les entrepôts de données sont devenus un sujet majeur aussi bien dans le monde de l'industrie que dans celui de la recherche. Les principales motivations sont de tirer profit du grand volume de données stocké dans les bases de données. Les entrepôts sont aujourd'hui de plus en plus utilisés pour exploiter et analyser une grande quantité de données et faciliter ainsi les processus de prises de décision. Ces entrepôts de données sont souvent utilisés avec des systèmes OLAP (On-Line Analytical Processing) qui s'opposent aux systèmes OLTP (On-Line Transactional Processing) utilisés avec les bases de données. Les systèmes OLTP sont fondés sur des modèles normalisés garantissant la non-redondance des données, la fiabilité, la cohérence et la performance du système. Les systèmes OLAP sont plus adaptés pour l'exploitation de données et l'analyse décisionnelle. Les processus OLAP proposent des fonctionnalités d'exploration des données et répondent parfaitement aux besoins spécifiques des analyses d'informations. Les données entreposées sont modélisées sous forme de structures multidimensionnelles ou "hypercubes" qui organisent les données selon des dimensions dans différentes granularités, et des faits soumis à des fonctions d'agrégation classiques (somme, moyenne, …) Ainsi, dans les entrepôts de données, les faits sont considérés comme la partie dynamique de l'entrepôt, puisqu'ils peuvent être recalculés à chaque nouvelle alimentation de l'entrepôt, alors que les dimensions sont statiques.

Généralement, les entrepôts de données mis en œuvre dans les projets industriels et dans le domaine de la recherche porte sur des données alphanumériques. La conception d'entrepôts de données multimédia est un sujet récent qui soulève de nombreuses problématiques. Tout d'abord les données multimédias sont particulièrement volumineuses et leur stockage nécessite un traitement spécifique et des outils de visualisation adaptés. Les fonctions d'agrégat portant non plus sur des données alphanumériques mais sur des faits multimédias doivent être redéfinies. Enfin les données multimédias sont généralement caractérisées par des descripteurs qui peuvent être extraits manuellement ou automatiquement : les entrepôts multimédias vont être construits en utilisant ces descripteurs pour former les dimensions du modèle. Or ces descripteurs peuvent être obtenus par divers modes de calcul. Nous considérons donc qu'il est intéressant d'intégrer au modèle un choix de modes de calcul pour chaque descripteur afin de permettre à l'utilisateur de définir la représentation des données qu'il préfère. Dans ce cas, l'entrepôt ne doit pas être statique et doit intégrer de multiples vues pour représenter les données.

Le but de notre étude est de concevoir un modèle multidimensionnel capable de gérer des données multimédias caractérisées par des descripteurs obtenus par différents modes de calcul. Ce modèle est illustré par une étude de cas portant sur un entrepôt de données multimédias médicales. Ces données médicales proviennent d'un essai thérapeutique dont les résultats sont des signaux représentant des électrocardiogrammes. Ces données multimédias sont liées à des descripteurs qui peuvent être calculés par différents modes de calcul tels des algorithmes ou des classifications. Nous définissons ces modes de calcul comme des "versions fonctionnelles" de descripteurs. L'intérêt est d'analyser ces données multimédias selon les différentes versions fonctionnelles des descripteurs. Nos travaux abordent donc le problème de multiversion fonctionnelle dans les entrepôts de données et a pour objectif de permettre à l'utilisateur de définir la représentation des données la plus adaptée à son analyse en choisissant les versions fonctionnelles des descripteurs à travers une navigation facile et adaptée. La notion de versions fonctionnelles doit donc être intégrée aux structures multidimensionnelles et notamment à la définition des dimensions dont les membres seront calculés selon les différentes versions de descripteurs. Nous nommerons ces dimensions « versions de dimension ». Ces dimensions peuvent en outre être organisés selon des hiérarchies complexes (multiples, non strictes, etc…). Le modèle multidimensionnel que nous définissons dans ce travail, appelé « modèle M2F », est un modèle multidimensionnel qui permet l'intégration de version de dimension et la prise en compte de hiérarchies complexes.

Ce rapport est organisé comme suit : tout d'abord, nous présentons les problèmes et les solutions existantes dans le domaine des données multimédias et des entrepôts de données. Même si la problématique des versions fonctionnelles n'a pas souvent été abordée dans la littérature, elle présente néanmoins des similarités avec celle des évolutions temporelles dans les entrepôts de données. Nous étudierons les solutions apportées dans ce domaine pouvant être adaptées à nos travaux. Puis nous présentons les travaux réalisés en décrivant l'étude de cas, les objectifs à atteindre et le modèle conceptuel « M2F » que nous développons pour répondre à ces besoins. Les modèles logiques et physiques sont ensuite présentés pour l'implémentation de notre approche. Ce modèle est mis en œuvre et nous décrivons le prototype réalisé ainsi que l'interface de navigation réalisée. Enfin nous analysons les apports de notre modèle, ses limites et les perspectives possibles.

2. Etat de l'art

2.1. Les entrepôts de données

Un entrepôt de données est défini comme « une collection de données intégrées, orientées sujet, non volatiles, historisées, résumées et disponibles pour l'interrogation et l'analyse » [INM02]. L'objectif est d'extraire des données pertinentes à partir de bases de données de production et de les organiser suivant un modèle adapté afin de faciliter des prises de décision. Les analyses décisionnelles sont basées sur des traitements OLAP (On-Line Analytical Processing) qui sont définis comme « l'analyse dynamique de l'entreprise nécessaire pour créer, manipuler, animer et synthétiser l'information » [COD93]. En effet les besoins utilisateurs se portent vers un système de requêtes devant s'exécuter le plus rapidement possible. Il faut alors dénormaliser les modèles, permettre une redondance des données, pré-calculer les requêtes et représenter les informations sous forme multidimensionnelle. Les modèles d'entrepôts de données doivent faciliter la compréhension et l'écriture de requêtes et optimiser les temps d'exécution des requêtes. Ces modèles sont appelés modèles multidimensionnels ou hypercube de données et ont été formalisés par [CAB98]. Dans ces modèles, le sujet analysé appelé aussi la mesure ou le fait est représenté dans un espace qui présente plusieurs axes d'analyse nommés dimensions. Par exemple, dans un entrepôt portant sur des études démographiques, le nombre de naissance peut être analysé par zone géographique, par période de temps et par sexe. Ces dimensions se présentent en différentes granularités afin d'affiner ou élargir l'analyse en utilisant des opérateurs de forage (roll-up, drill-down) pour la navigation [AGR95]. Une dimension est représentée par un schéma qui définit différents niveaux de granularité reliés par des liens hiérarchiques. Par exemple, la dimension *Localisation* peut avoir un schéma constitué des niveaux *Ville, Département*, *Région* et *Pays.* Ces niveaux sont reliés par des liens hiérarchiques tels que *Ville*, qui est le niveau de granularité le plus bas, soit relié à *Département* relié à *Région* relié à *Pays,* qui est le niveau de granularité le plus haut. (cf. figure 1). Chaque niveau de dimension est composé de membres qui représentent les entités de la dimension considérée. Ces membres sont également reliés par des liens hiérarchiques, cette structure hiérarchique est l'instance de la dimension considérée. Par exemple, les membres du niveau *Ville* sont *Lyon*, *Avignon, Orange et Marseille,* ceux du niveau région sont *Vaucluse*, *Bouches du Rhône* et *Rhône*. Les membres sont liés tels que *Orange* et *Avignon* soit relié à *Vaucluse*, puis *Marseille* à *Bouches du Rhône* et *Lyon* à *Rhône*, etc. (cf. figure 2).

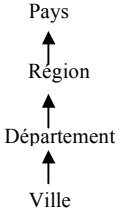


Figure 1 : schéma d'une dimension Localisation

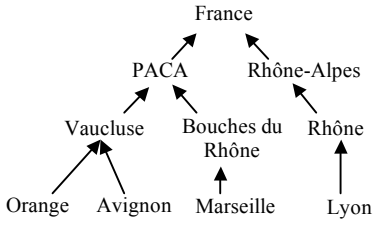


Figure 2 : Instance d'une dimension Localisation

Les données agrégées sont les faits calculés en utilisant des fonctions suivant des granularités différentes. Cinq fonctions classiques sont définies et permettent de compter le nombre de faits (« count »), de calculer la somme des valeurs des faits (« sum »), de retourner la valeur maximum ou minimum des faits (« min », « max ») ou de donner la moyenne des valeurs des faits (« avg »). Les agrégations de données sont calculées à partir de la table de fait et des liens entre les membres. Par exemple, la fonction d'agrégation « count » peut être utilisée pour compter le nombre de naissance par ville, par année et par sexe et à un niveau de granularité différente, cette fonction permet de compter le nombre de naissance par région, par année et par sexe.

Il existe plusieurs méthodes de représentation des structures multidimensionnelles. Les plus fréquentes sont les modèles en étoile, le modèle en flocon et le modèle en constellation [CAB98, TSO01]. Le schéma en étoile modélise les données comme un simple cube dans lequel les liens hiérarchiques des schémas des dimensions ne sont pas explicites mais encapsulés dans des attributs. Le schéma en flocon normalise les tables de dimension et représente les hiérarchies explicitement en identifiant chaque niveau d'une dimension dans des tables séparées. Enfin le modèle en constellation permet d'avoir plusieurs tables de fait en reliant plusieurs modèles en étoile.

Les entrepôts de données classiques traitent des données alphanumériques. Dans cette étude, nous nous interessons plus particulièrement à l'intégration de données multimédias dans les entrepôts de données.

2.2.Les entrepôts de données multimédias

2.2.1. Les données multimédias

Une base de données multimédias contient différents types de données complexes comme des textes, des graphiques, des images, des vidéos, des sons, de la musique ou d'autres formes d'information audio ou vidéo. Cette large variété de types de données nécessite de considérer l'organisation du contenu de toutes ces données multimédias. Il existe 3 types d'architecture pour représenter cette organisation. La première approche consiste à traiter les différents types de médias (vidéo, image, audio….) séparément en indexant et en structurant chaque type de données de façon optimale afin d'obtenir le meilleur accès possible aux données du type considéré. La deuxième approche réunit tous les types de données médias dans un même index où les données de tous les types ont la même structure et s'appuient sur le principe d'uniformité selon lequel du point de vue sémantique, le contenu des sources de données multimédias est souvent indépendant de la source elle-même. A chaque objet est associée une métadonnée représentative de ce que l'utilisateur demande et ces métadonnées sont indexées de manière à favoriser les accès des utilisateurs. La troisième approche utilise les deux précédentes en faisant une indexation et une structure communes pour certains types de données médias et des index et structures séparés pour d'autres types. [SUB98]

Dans la première architecture, chaque type de données est traité séparément et différemment. Les données multimédias sont généralement stockées en séquences de bits de longueurs différentes et ces segments de données sont reliés de manière à faciliter l'indexation. Pour cela, des descripteurs ou indicateurs sont extraits des données multimédias. Ils caractérisent la donnée et permettent de l'identifier. Deux familles de systèmes d'indexation et de recherche de documents multimédias existent et se basent sur des types de descripteurs de données multimédias différents. Le premier appelé « description-based retrieval system » utilise des descripteurs définis à partir de la description de la donnée (les descripteurs textuels) Le deuxième appelé « content-based retrieval system » se base sur des descripteurs représentant le contenu de la donnée et calculés directement sur la donnée (les descripteurs de contenu) [ZAI99, HAN01]. Par exemple, pour des données images, les descripteurs textuels peuvent être des mots-clé, la résolution, la légende de l'image et les descripteurs de contenu peuvent être la couleur, la texture, les formes. Dans l'exemple des vidéos, les descripteurs textuels

peuvent être la date, le réalisateur, le thème de la vidéo et les descripteurs de contenu sont le son, la qualité de l'image.

Ces descripteurs sont très diversifiés et un même descripteur peut être extrait de la donnée multimédia de diverses manières. Plusieurs algorithmes permettent de calculer un descripteur et plusieurs classifications permettent d'ordonner les valeurs d'un descripteur. Tous ces modes de calcul définissent des versions fonctionnelles du descripteur permettant de caractériser la donnée le plus précisément possible. Par exemple, si nous considérons la couleur d'une image, elle peut être calculée en utilisant différents algorithmes de traitement de l'image. De la même manière, les valeurs de la résolution d'une image peuvent être classées en utilisant des catégories (basse, moyenne, haute résolution…) ou en utilisant des tranches de résolution (2-3 megapixel, 3-4, 4-5,…)

Afin d'analyser et d'exploiter au mieux ces données multimédias, il est nécessaire de les intégrer à des entrepôts de données multimédias dont les modèles permettent des requêtes simples, efficaces et parfaitement adaptées à la gestion d'importants volumes de données.

2.2.2. Les entrepôts de données multimédias

Aujourd'hui les informations multimédias sont de plus en plus importantes et les bases de données multimédias se multiplient. Les données manipulées sont de plus en plus volumineuses et difficiles à gérer. L'utilisation d'entrepôts de données multimédias peut alors être plus efficace que celle des bases de données simples et permettre une exploitation et une meilleure recherche de ces données. Les axes d'analyse des données multimédias sont les descripteurs de ces données et les dimensions des entrepôts de données multimédias doivent être calculées à partir de ces descripteurs. Les faits sont les données multimédias elles-mêmes et il est nécessaire de trouver une méthode afin de stocker ces données dans la table de fait. Dans un entrepôt de données classique, les agrégations de données sont calculées grâce à des fonctions comme « count », « sum », « min », « max » et « avg ». Dans les entrepôts de données multimédias, les fonctions d'agrégations peuvent aussi être des fonctions spécifiques représentant des agrégats de données multimédias comme des listes, des moyennes ou des fusions de données multimédias. De même que les faits, les données agrégées demandent d'importantes espaces de stockage. Si les données agrégées sont calculées en temps réel, la navigation dans le cube de données devient difficile. Un compromis doit être trouvé entre le stockage des données agrégées et la rapidité de navigation.

La majorité des travaux sur les données multimédias a été faite dans le cadre de la gestion de données spatiales. La technique des entrepôts de données et les outils d'exploration et d'analyse OLAP permettent de manipuler les données spatiales et favorisent la prise de décision. La représentation cartographique des données est gérée par les SIG (Système d'Information Géographique). Il est nécessaire de disposer de systèmes permettant une navigation au sein des données spatiales ainsi qu'une interrogation claire et facile de ces données. La technologie SOLAP (Spatial OLAP) apporte des solutions en combinant les technologies OLAP et SIG. La conception d'entrepôts de données spatiales repose essentiellement sur le modèle en étoile. Plusieurs types de dimension sont définis [MIQ01] dont les dimensions spatiales géométriques où tous les niveaux des hiérarchies sont représentés cartographiquement et sont décrits par des objets géométriques (à l'inverse des dimensions spatiales non géométriques et spatiales mixtes). Il peut également y avoir plusieurs types de faits dont les faits spatiaux représentant des pointeurs sur des zones géographiques ou des régions. L'analyse de ces faits nécessite l'utilisation de fonctions d'agrégat spécifiques. Par exemple, il peut être intéressant de lister les régions répondants à des critères choisis et de les fusionner pour créer des regroupements de régions. Cependant, les données spatiales agrégées occupent en général un grand espace de stockage comme par exemple ces fusions de régions. Il n'est pas possible de stocker toutes les agrégations et plusieurs solutions peuvent être utilisés pour stocker des cubes précalculées. La première solution est le calcul de données approximées. Une région peut par exemple être mémorisée en l'approximant par un rectangle et en ne stockant que les deux points permettant sa construction (les points supérieur gauche et inférieur droit). La deuxième solution de stockage de cubes précalculés est le calcul de données sélectif. Des cubes sont réalisés uniquement

sur les données les plus accédées ou les plus susceptibles de l'être. Ces cubes peuvent être construits automatiquement en se basant sur des fréquences d'accès ou sur une approximation de la taille des cubes à générer [HAN01]. Une algèbre spécifique aux systèmes SOLAP est présentée afin de définir un ensemble d'opérateurs applicables aux données spatiales. Enfin, des outils de visualisation sont développés dans les SIG pour représenter les données spatiales.

Quelques travaux de construction de cubes de données multimédias ont été menés pour faciliter l'analyse multidimensionnelle de large base de données multimédias. Dans [YOU01], les auteurs cherchent à étendre le concept d'entrepôt de données classique et des bases de données multimédias afin de stocker et de représenter les données multimédias. L'entrepôt de données multimédias est un modèle en flocon qui facilite l'indexation, le traitement de requêtes dynamiques et la recherche hiérarchique pour la restitution de données multimédias. Dans ce modèle, la table de fait est constituée des données multimédias (de son contenu) et les dimensions correspondent aux différents types de données multimédias et à leurs descripteurs. Par exemple la dimension image est associée aux dimensions catégorie, type, caractéristique, source, date… et la dimension vidéo aux dimensions langages, source, date…

Un autre exemple est celui du système d'analyse de données multimédias MultiMediaMiner qui utilise un cube de données multimédias [ZAI98] permettant de stocker les données multidimensionnelles et de les agréger à des niveaux de granularité différents. Les données multimédias originales ne sont pas stockées directement. Les liens vers les données représentent les faits et les descripteurs des données représentent les dimensions. Le cube est constitué de plusieurs dimensions telles que la taille de l'image ou de la vidéo, la date de création de l'image ou de la vidéo, le type de format de l'image ou de la vidéo…

Dans le domaine médical, les problèmes d'exploitation et d'analyse de données volumineuses sont omniprésents et les données multimédias sont très utilisées. Nous pouvons citer par exemple une étude menée dans le cadre de la détection du cancer du sein. Cette étude a conduit au développement d'un entrepôt de données de mammographies numériques pour l'aide au diagnostique [ZHA01]. [RAH95] traite le problème du stockage et de la restitution de données d'images médicales à partir d'un entrepôt en comparant l'entrepôt à une pyramide de moyens de stockage où les informations les plus fréquemment utilisées sont stockées en haut de la pyramide, représenté par la mémoire vive, et les informations les plus détaillées, prenant le plus de place, en bas de la pyramide, représenté par des bandes magnétiques. D'autres études ont été menées notamment sur le cancer en faisant des analyses multidimensionnelles d'ensembles de données spatio-temporelles épidémiologiques [KAM97].

Ces travaux se basent sur la modélisation des bases de données multimédias et s'appuient sur des descripteurs qui définissent la donnée. La construction de cubes de données multimédias se fait d'une façon similaire à celle des cubes de données traditionnelles. Les entrepôts de données multimédias sont modélisés par des schémas en étoile ou en flocon. La table de fait rassemble les données multimédias dont, en général, seul les liens sont stockés et les dimensions représentent les descripteurs de ces données. Les données agrégées sont calculées grâce à des fonctions spécifiques et le stockage de ces agrégations nécessite un grand volume. Tous ces modèles sont statiques puisque les descripteurs sont figés c'est-à-dire calculés d'une manière unique au moment du chargement de l'entrepôt. Or, nous avons vu dans la partie précédente qu'un descripteur peut être calculé de diverses manières. Il peut être intéressant pour l'utilisateur de choisir le mode de calcul (ou la version fonctionnelle) de chaque descripteur afin de sélectionner la représentation ou la vue des données qu'il souhaite. Il est alors nécessaire d'intégrer la notion de version fonctionnelle de descripteurs de données aux entrepôts de données. Cette notion n'a pas souvent été abordée dans la littérature, mais nous constatons des similarités avec les évolutions temporelles des données stockées dans un entrepôt où chaque évolution est représentée par une version temporelle.

2.3. Les versions temporelles dans les entrepôts de données

Les données peuvent subir des changements au cours du temps (par exemple, le découpage d'une ville en district ou le découpage en service dans les hôpitaux). Ainsi, dans les modèles multidimensionnels, les dimensions et leurs attributs évoluent à travers le temps en même temps que la structure multidimensionnelle. La prise en compte des évolutions temporelles dans les entrepôts de données permet à l'utilisateur de choisir une version temporelle de dimensions pour la représentation des données.

Deux approches existent pour traiter l'évolution dans les structures multidimensionnelles : les modèles de mise à jour de l'entrepôt et les modèles prenant en compte l'historique des évolutions. Ce dernier modèle est particulièrement intéressant car seule la prise en compte de l'historique des évolutions permet d'analyser les données dans leurs différentes versions et d'orienter l'analyse sur leurs évolutions. Plusieurs auteurs ont proposé des modèles prenant en compte l'historique des évolutions en utilisant des notions de temps valide et temps de transaction [PED 01], de liens de transitions entre deux versions de dimensions [MEN 00] et de mode temporel de représentation des données [EDE01]. Le modèle multidimensionnel multiversion [BOD02a, BOD02b] intègre la notion de version dans les dimensions et rassemblent les données dans une table de fait appelée table de fait multiversion, selon différents modes temporels de présentation (MTP). (cf. figure 3) Il permet à l'utilisateur de choisir le mode temporel dans lequel il veut représenter ses données.

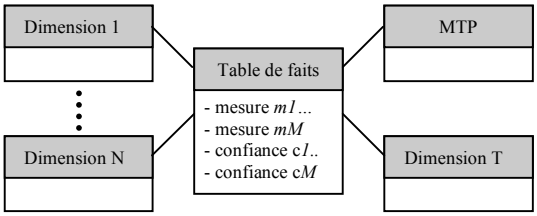


Figure 3 : Schéma logique

Ce modèle permet donc de définir la notion de version dans les modèles multidimensionnels. Il convient tout à fait au problème des évolutions temporelles dans les entrepôts où une évolution est représentée par un ensemble de versions de données. Cependant ce modèle permet une navigation assez limitée puisqu'en choisissant un mode temporel de représentation l'utilisateur sélectionne un ensemble fixe de dimensions correspondant aux versions des données du mode temporel de représentation choisi. Dans le cas des données multimédias, ceci ne convient pas puisque l'utilisateur peut être amené à choisir pour chaque descripteur la version fonctionnelle qu'il souhaite.

3. Travaux effectués

3.1. Objectifs

Ce travail s'effectue dans le cadre d'une collaboration avec une équipe de l'INSERM (ERM 107) spécialisée dans le domaine de la méthodologie de l'information en cardiologie qui a notamment travaillé sur les données d'un essai thérapeutique nommé étude EMIAT (European Myocardial Infarct Amiodarone Trial). Dans notre travail, nous mettons en œuvre un entrepôt de données multimédias dans le domaine médical en intégrant les données multimédias de cette étude à un modèle multidimensionnel. Afin d'expliquer plus clairement les objectifs de nos travaux, nous développons cette étude de cas en soulignant la nécessité des versions de dimension.

3.1.1. Etude de cas

L'étude EMIAT a été réalisée pour évaluer les effets de l'amiodarone comparée à un placebo chez des patients ayant survécu à un infarctus du myocarde. Le critère principal est de chercher une diminution de la mortalité totale des patients traités par amiodarone. Il s'agit d'attribuer aux patients de l'amiodarone ou un placebo aléatoirement et en double aveugle (aucune personne ne sait si elle a un médicament ou l'autre) et de suivre l'évolution de ces patients en étudiant leurs électrocardiogrammes (ECGs) et autres données médicales. L'étude EMIAT est multicentrique (plusieurs centres d'études), prospective (tend à améliorer le futur), randomisée (attribution aléatoire des types de médicaments), en double aveugle (aucune personne ne sait si elle a un médicament ou l'autre). Le suivi des patients est effectué pendant une durée moyenne de 21 mois après l'infarctus. Chaque patient passe des visites pendant la durée de l'expérience (sauf décès prématuré). Celles-ci ont lieu le jour de l'infarctus, deux semaines après, deux mois après et ensuite tous les quatre mois pendant deux ans. Lors de certaines visites, ont lieu des enregistrements Holter de l'ECG du patient pendant 24H sur 3 voies (dimensions de l'espace X, Y et Z) dont sont extraits six passages d'environ vingt minutes (répartis dans la journée et la nuit), des radiographies pulmonaires et des prélèvements de données (concentrations en potassium de créatine kinase, alamine aminotransférase, aspartate aminotransférase et thyrotrophine).

Cette étude fournit comme résultat un nombre important de données à exploiter et à analyser dont des données multimédias assez volumineuses de type signaux. Ces données multimédias sont les électrocardiogrammes ou ECGs des différents patients sur lesquels porte l'étude. A partir d'un ECG, plusieurs descripteurs ou indicateurs peuvent être calculés pour caractériser l'état de santé cardiaque d'un patient. Ainsi la durée du QT correspond à une durée type mesurée sur l'ECG, entre l'onde Q et l'onde T (cf. figure 4) et le niveau de bruit correspond aux interférences sur l'ECG au moment de sa prise. A ces ECGs sont associées d'autres informations telles que la pathologie du patient, l'heure de la visite...

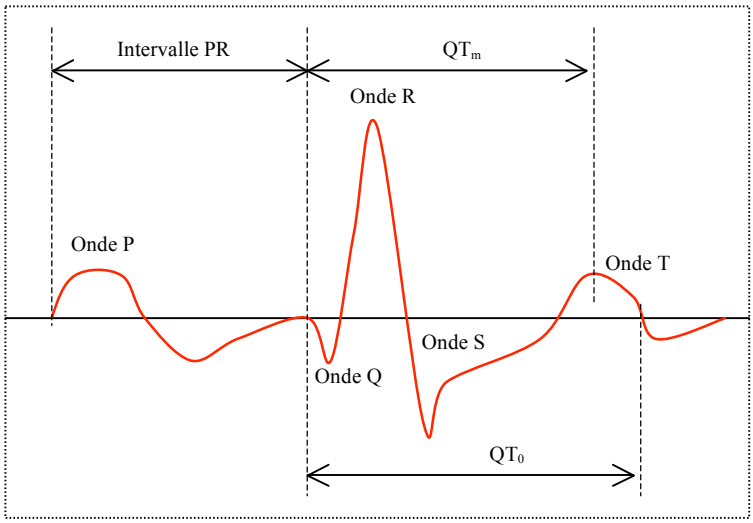


Figure 4 :
Schéma de la Y voix d'un ECG

Les ECGs sont donc caractérisés par deux types de descripteurs :

- des descripteurs textuels : pathologie principale, âge, sexe du patient, date et tranche horaire d'acquisition de l'ECG, technologie avec laquelle l'ECG est obtenu,…
- des descripteurs de contenu : la durée du QT, le niveau de bruit de l'ECG,…

Les faits sont des ECGs caractérisés par des descripteurs organisés en hiérarchies complexes. Par exemple, les tranches horaires peuvent être classées en heures puis en périodes (nuit, réveil, jour).

Certaines heures comme 6h peuvent appartenir à plusieurs périodes (réveil et nuit). Ce descripteur est alors organisé en hiérarchie non-stricte. Ces descripteurs peuvent être calculés par divers modes de calcul. Par exemple la durée du QT peut être obtenue grâce à plusieurs algorithmes. Nous allons donc traiter d'une part les hiérarchies complexes et d'autre part les versions de dimension que nous définissons comme des dimensions dont les membres sont calculés selon les différentes versions fonctionnelles de descripteurs.

3.1.2. Les hiérarchies complexes

Les descripteurs des données multimédias sont généralement organisés en hiérarchies complexes. Celles-ci peuvent être de différentes sortes :

- Hiérarchie explicite : hiérarchie définie explicitement par un schéma
- Hiérarchie multiple : plusieurs chemins différents dans le graphe des liens hiérarchiques entre niveaux c'est-à-dire que pour une dimension, les données peuvent être agrégées en utilisant différentes hiérarchies de niveaux. Considérons par exemple le profil des ECGs analysé selon la date d'enregistrement des ECGs. La dimension « *dateParSemaineOuMois* » est organisée en hiérarchie multiple (cf. figure 5) puisque les deux chemins possibles sont jours-mois-année et jours-semaine-année.
- Hiérarchie non-onto: déséquilibre entre les hiérarchies dans une dimension c'est-à-dire que la longueur des chemins entre la racine et les feuilles varie. Supposons par exemple que le profil des ECGs soit analysé selon la pathologie principale des patients. La dimension « *familleDePathologie* » est organisée en hiérarchie non-onto (cf. figures 6-7) puisque le membre « tachycardie » n'appartient pas au niveau de granularité le plus bas.

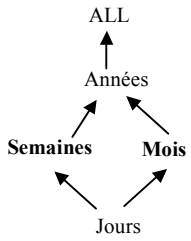


Figure 5
Schéma de la dimension à hiérarchie multiple

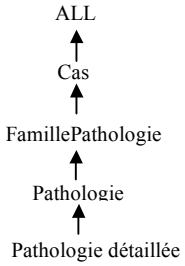


Figure 6
Schéma de la dimension à hiérarchie non-onto

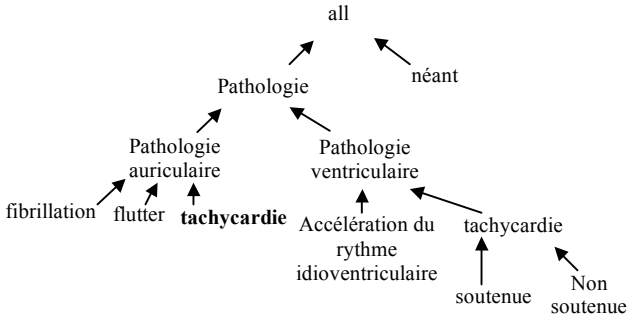


Figure 7
Instance de la dimension à hiérarchie non-onto

- Hiérarchie non-stricte: relation n-n entre les différents niveaux de dimension. Supposons cette fois-ci que les ECGs soient enregistrés à un moment précis de la journée, nous obtenons alors une autre dimension «*HoraireAcquisition* » organisée en hiérarchie non-stricte puisque le membre « *6* » est relié à deux membres parents. (cf.figures 8-9)
- Hiérarchie non-couvrante: dans l'instance d'une dimension, des niveaux peuvent être « sautés ». Considérons par exemple la durée du QT de ces électrocardiogrammes et la dimension « *QTalogo1* ». Cette dimension est organisée en hiérarchie non-couvrante puisque le membre « *0* » est relié directement au membre « *all* ». (cf. figures 10-11)

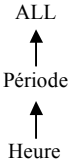


Figure 8
Schéma de la dimension à hiérarchie non-stricte

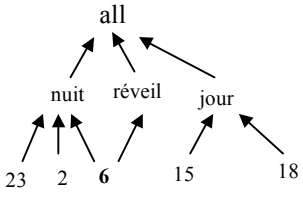


Figure 9
Instance de la dimension à hiérarchie non-stricte

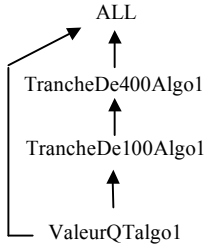


Figure 10
Schéma de la dimension à hiérarchie non-couvrante

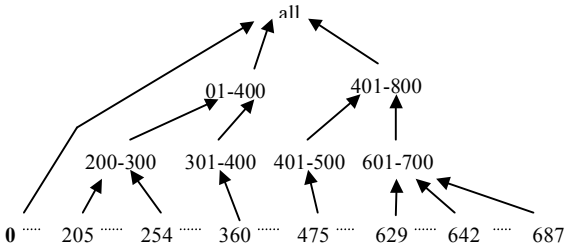


Figure 11
Instance de la dimension à hiérarchie non-couvrante

La plupart des modèles permettent de définir les hiérarchies explicites mais ne traitent que partiellement les hiérarchies complexes. Le modèle de Mendelson et Vaisman [MEN00] et celui de Hurtado et al. [HUR99a, HUR99b] tiennent compte des hiérarchies explicites et multiples ; celui de Agrawal et al. [AGR95] et celui de Kimball [KIM00] traitent les hiérarchies multiples; enfin celui de Jagadish et al. [JAG99] traite les hiérarchies explicites, non-onto et non-couvrantes. Quelques modèles couvrent toutes les caractéristiques des hiérarchies complexes [PED01, BOD02a, BOD02b]. Ils s'appuient sur des relations nommées parent-enfant c'est-à-dire des liens de filiations qui peuvent être multivalués, entre les niveaux et les membres de dimension. Un graphe définit l'instance de la dimension à partir de laquelle les relations « parent-enfant » sont définies en désignant l'enfant comme le membre d'un niveau et le parent le membre du niveau supérieur auquel il est relié [PED01]. Certains [BOD02a, BOD02b] déduisent le schéma de la dimension et sa hiérarchie à partir des instances et des liens qui existent entre les différents membres de dimension alors que d'autres [PED01] définissent les deux, c'est-à-dire le schéma de la dimension et l'instance de cette dimension correspondante. Cette représentation « parent-enfant » permet de supporter les hiérarchies complexes alors que les modèles en flocon et en étoile ne traitent pas les hiérarchies non-strictes et non-onto.

3.1.3. Versions de dimension

Certains descripteurs d'ECGs comme par exemple le QT ou l'âge peuvent être calculés ou classés de différentes manières. Ainsi, il peut exister plusieurs algorithmes de calcul de QT et l'analyste doit pouvoir analyser les données selon des valeurs de QT calculées par l'algorithme 1 et des valeurs de QT calculées par l'algorithme 2. De la même manière, nous pouvons définir plusieurs classifications pour l'âge comme par exemple un regroupement par classe et un regroupement par tranches d'âges. Nous définissons ces différents calculs ou classifications comme des versions fonctionnelles. Ainsi nous disposons de deux versions fonctionnelles pour le QT et de deux versions fonctionnelles pour l'âge. Il peut s'avérer intéressant d'avoir les ECGs d'une part en fonction d'un QT calculé par l'algorithme 1 et des âges classés par tranches d'âges et d'autre part selon un QT calculé par un algorithme 2 et des âges classés par tranches d'âges également. Toutes les versions fonctionnelles des descripteurs doivent pouvoir être combinées afin d'obtenir des vues différentes des données, de les comparer et de les analyser au mieux. Il s'agit donc d'intégrer les données multimédias de l'étude EMIAT à un entrepôt de données dans lequel les versions de dimension représentent des dimensions dont les membres sont calculés selon les versions fonctionnelles des descripteurs. Ainsi l'utilisateur peut visualiser ces données selon la combinaison de versions fonctionnelles qu'il souhaite.

Dans la suite du rapport, nous allons présenter le modèle conceptuel multidimensionnel multiversion fonctionnelle ou « modèle M2F ». Ce modèle intègre les dimensions à hiérarchies complexes et les versions de dimension afin de permettre à l'utilisateur de choisir les versions fonctionnelles des descripteurs en vue d'analyser au mieux les ECGs.

3.2. Modèle conceptuel M2F

3.2.1. Principe général

Comme la plupart des modèles multidimensionnels pour des entrepôts de données multimédias, notre approche est basée sur une table de fait regroupant l'ensemble des mesures qui représentent les données à analyser c'est-à-dire les données multimédias ou des pointeurs vers celles-ci et sur des dimensions qui constituent les axes d'analyse c'est-à-dire les descripteurs de ces données multimédias. Pour prendre en compte le problème de multiversion fonctionnelle, nous redéfinissons la structure multidimensionnelle en ajoutant la notion de version fonctionnelle. Ainsi nous introduisons les concepts de version de dimension, dimension multiversion, table de fait multiversion fonctionnelle et fonction de version de dimension. Une dimension multiversion est composée de

plusieurs versions de dimension, chacune étant une dimension pour une version donnée avec son propre schéma et sa propre instance. La table de fait multiversion fonctionnelle regroupe toutes les données en combinant les différentes versions de dimension d'une dimension multiversion avec les autres. Enfin, les fonctions de version de dimension sont les modes de calcul qui permettent d'obtenir les membres des versions de dimension.

Nous définissons les schémas des différentes dimensions en décrivant les niveaux et les liens hiérarchiques qui les lient. Nous décrivons également les instances des ces dimensions en décrivant l'ensemble des membres et des filiations. Notre approche permet donc d'avoir des dimensions explicites étant donné que les schémas de dimension sont définis explicitement et notre modèle supporte également les hiérarchies complexes (hiérarchie multiple, hiérarchie non-stricte, etc.) puisque les instances des dimensions sont construites à partir des membres et des liens hiérarchiques.

3.2.2. Définitions des concepts

Définition 1 (schéma de version de dimension)
Un schéma de version de dimension est un schéma de dimension pour une version donnée. Une version est un mode de calcul utilisé pour obtenir les membres d'une dimension. Le schéma $\boldsymbol{S_{VD}}$ de la version de dimension d'identifiant idVD est défini par le tuple $<\boldsymbol{idVD, \mathcal{N}, \sqsubset_{idVD}}>$ où :

- ***IdVD*** est l'identifiant de la version de dimension
- $\mathcal{N} = \{n_j , j=1,...,k\}$ est l'ensemble des niveaux de S_{VD} . Un niveau dans S_{VD} représente un ensemble de valeurs de même granularité associées à la même version de dimension. Un niveau n_j est défini par le tuple $<\boldsymbol{idNiveau_j, nomNiveau_j, [\mathcal{A}_j], [description_j]}>$ où :
  - $\boldsymbol{idNiveau_j}$ est l'identifiant du niveau de version de dimension
  - $\boldsymbol{nomNiveau_j}$ est le nom du niveau de version de dimension
  - $\boldsymbol{\mathcal{A}_j}$ est une propriété optionnelle qui représente l'ensemble des attributs descriptifs de ce niveau
  - $\boldsymbol{description_j}$ est une propriété optionnelle qui permet d'introduire des informations textuelles sur le niveau n_j
- $\sqsubset_{idVD}$ est un ordre partiel sur l'ensemble $\mathcal{N}$ qui définit les filiations entre les niveaux du schéma S_{VD}. Une filiation établit un lien hiérarchique entre deux niveaux de S_{VD}. L'ordre partiel $\sqsubset_{idVD}$ est défini tel que : $\forall (n_1,n_2) \in \mathcal{N}\times\mathcal{N}$, si $n_1 \sqsubset_{idVD} n_2$ alors n_1 a une granularité plus fine que n_2.

Un schéma de version de dimension peut donc être représenté par un graphe orienté dont les éléments de $\mathcal{N}$ sont les nœuds et $\sqsubset_{idVD}$ les arcs. Ce graphe doit être acyclique afin de permettre les agrégations des mesures vers les niveaux hiérarchiques supérieurs. On définit un niveau ALL comme étant la racine de la hiérarchie c'est à dire le niveau de granularité le plus haut.

Exemple 1 :
Supposons que l'on souhaite analyser l'influence de l'âge sur le profil des ECG d'un certain nombre de patients. L'âge est alors une dimension de l'entrepôt dont les membres peuvent être ordonnés de différentes manières. Les âges peuvent être classés par tranches d'âges par exemple des tranches de 5 ans, puis 10 ans, puis 50 ans. On peut également regrouper ces âges en classes d'âge (jeune enfant, enfant, adolescent, jeune adulte, adulte, senior) puis en catégories (mineur, majeur).

Soit le schéma $S_{âgeParTranche}$ de la version de dimension « *âgeParTranche* » dont *idVD = 1*. Le schéma de cette version de dimension est défini par :

$S_{âgeParTranche} = <1,\{n_1, n_2, n_3\},\sqsubset>$ avec
$n_1 = <1, « TrancheDe5 »>$
$n_2 = <2, « TrancheDe10 »>$
$n_3 = <3, « TrancheDe50 »>$
et l'ordonnancement suivant : $n_1 \sqsubset n_2$, $n_2 \sqsubset n_3$ et $n_3 \sqsubset ALL$
Le schéma peut être représenté par le graphe orienté de la figure 12.

Soit le schéma $S_{âgeParClasse}$ de la version de dimension « *âgeParClasse* » dont *idVD = 2*. Le schéma de cette version de dimension est défini par :

$S_{âgeParClasse} = <2,\{n_4, n_5\},\sqsubset>$ avec
$n_4 = <4, « ClassesAge »>$
$n_5 = <5, « Catégories »>$
et l'ordonnancement suivant : $n_4 \sqsubset n_5$ *et* $n_5 \sqsubset ALL$
On obtient le graphe orienté de la figure 13.

Considérons maintenant la durée du QT de ces électrocardiogrammes comme autre dimension de l'entrepôt. Cette durée du QT peut être calculée à l'aide de plusieurs algorithmes, par exemples algo1 et algo2. Le schéma de la dimension caractérisant la durée du QT a pour hiérarchie les valeurs de la durée du QT pour le niveau le plus fin, elles-mêmes regroupées en intervalle de 100ms, puis en intervalle de 400ms. Lors de l'utilisation de l'algo1, il arrive que la valeur de la durée du QT soit indéfinie. Dans ce cas, elle n'est pas classée dans la hiérarchie et est rattachée directement au niveau ALL.

Soit le schéma $S_{QTalgo1}$ de la version de dimension « *QTalgo1* » dont *idVD = 3*. On aura alors le schéma de cette version de dimension défini par :

$S_{QTalgo1} = <3,\{n_1, n_2, n_3\},\sqsubset>$ avec
$n'_1 = <1, « ValeurQTalgo1 »>$
$n'_2 = <2, « TrancheDe100Algo1 »>$
$n'_3 = <3, « TrancheDe400Algo1 »>$
et l'ordonnancement suivant : $n'_1 \sqsubset n'_2$, $n'_2 \sqsubset n'_3$, $n'_3 \sqsubset ALL$ *et* $n'_1 \sqsubset ALL$
On obtient le graphe orienté présenté à la figure 10 de la partie 3.1.2.

Soit le schéma $S_{Qtalgo2}$ de la version de dimension « *QTalgo2* » dont *idVD = 4*. Le schéma de cette version de dimension est défini par :

$S_{Qtalgo2} = <4,\{n_4, n_5, n_6\},\sqsubset>$ avec
$n'_4 = <4, « ValeurQTalgo2 »>$
$n'_5 = <5, « TrancheDe100Algo2 »>$
$n'_6 = <6, « TrancheDe400Algo2 »>$
et l'ordonnancement suivant : $n'_4 \sqsubset n'_5$, $n'_5 \sqsubset n'_6$ *et* $n'_6 \sqsubset ALL$
On obtient le graphe orienté de la figure 14.

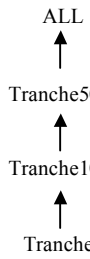


Figure 12
Schéma de la version de dimension « âgeParTranche »

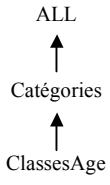


Figure 13
Schéma de la version de dimension « âgeParClasse »

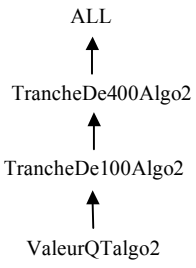


Figure 14
Schéma de la version de dimension « QTalgo2 »

Définition 2 (version de dimension)
Une version de dimension est une dimension pour une version donnée. La version de dimension ***VD*** de schéma $S_{VD} = <idVD, \mathcal{N}, \sqsubset_{nVD}>$ est définie par le tuple $<$***idVD,nomVD,*** $\mathcal{M}$***,*** $\sqsubset$***,[descriptionVD]***$>$ où :

- ***idVD*** est l'identifiant unique de la version de dimension
- ***nomVD*** est le nom de la version de dimension
- $\mathcal{M} = \{m_j, j=1...l\}$ est l'ensemble des membres de cette version de dimension. Un membre de version de dimension est un membre obtenu par le mode de calcul correspondant à la version de dimension. Il appartient à un des niveaux du schéma S_{VD}. On regroupe donc dans un niveau les membres de même granularité. Un membre m_j est représenté par un tuple $\langle id_j, val_j, [a_j], idNiveau_j \rangle$ où :
  - id_j est un identifiant unique pour ce membre de version de dimension
  - val_j est la valeur de ce membre de version de dimension
  - a_j est une propriété optionnelle qui contient l'ensemble des valeurs des attributs relatifs à ce membre (correspondant au niveau). Si cette propriété est définie pour le niveau correspondant au membre, alors elle doit l'être pour le membre.
  - $idNiveau_j$ est l'identifiant du niveau hiérarchique auquel appartient ce membre de version de dimension.
- $\sqsubset$ est un ordre partiel sur l'ensemble $\mathcal{M}$ qui définit les filiations entre les membres de *VD*. Une filiation établit un lien hiérarchique entre deux membres d'une même version de dimension. Pour chaque paire de niveaux (n_1,n_2), tel que $n_1 \sqsubset_{VD} n_2$, il existe au moins un couple $(m_1,m_2) \in \mathcal{M} \times \mathcal{M}$ tel que $m_1.idNiveau = n_1$ et $m_2.idNiveau = n_2$ et $m_1 \sqsubset m_2$. On dit alors que m_1 est de niveau inférieur à m_2 c'est-à-dire que m_1 a une granularité plus fine que m_2.
- ***descriptionVD*** est une propriété optionnelle contenant des commentaires éventuels sur la version de dimension

Une version de dimension peut donc être représentée par un graphe orienté dont les éléments de $\mathcal{M}$ sont les nœuds et $\sqsubset$ les arcs. Ce graphe doit être acyclique afin de permettre les agrégations des mesures vers les niveaux hiérarchiques supérieurs. La version de dimension ayant un schéma défini explicitement, on peut dire qu'elle est organisée en hiérarchie explicite. Dans la suite du rapport, nous désignerons par membre-feuille de version de dimension un membre d'une version de dimension n'ayant pas de fils. De plus, on définit le membre « all » comme l'unique membre contenu dans le niveau « ALL ». On note $\mathcal{MF}_{VD}$ l'ensemble des membres-feuilles de la version de dimension *VD*. Cet ensemble est défini par : $\mathcal{MF}_{VD} = \{m_j / m_j \in \mathcal{M} \text{ et } \neg \exists\, m_i \in \mathcal{M} \text{ tel que } (i \neq j \text{ et } m_i \sqsubset m_j)\}$

Exemple 2.
La version de dimension « *âgeParTranche* » dont le schéma $S_{âgeParTranche}$ est présenté dans l'exemple précédent est définie par :

$âgeParTranche = \langle 1, \text{« âgeParTranche »}, \{m_1,.....,m_7\}, \sqsubset \rangle$ avec
$m_1 = \langle 1, 0\text{-}5, 1 \rangle$
$m_2 = \langle 2, 6\text{-}10, 1 \rangle$
$m_3 = \langle 3, 11\text{-}15, 1 \rangle$
$m_4 = \langle 4, 16\text{-}20, 1 \rangle$
$m_5 = \langle 5, 0\text{-}10, 2 \rangle$
$m_6 = \langle 6, 11\text{-}20, 2 \rangle$
$m_7 = \langle 7, 0\text{-}50, 3 \rangle$
et l'ordonnancement suivant : $m_1 \sqsubset m_5$, $m_2 \sqsubset m_5$, $m_3 \sqsubset m_6$, $m_4 \sqsubset m_6$, $m_5 \sqsubset m_7$, $m_6 \sqsubset m_7$ *et* $m_7 \sqsubset all$

Les membres du niveau n_1 (« *TrancheDe5* ») sont donc $\{m1, m_2, m_3, m_4\}$, ceux du niveau n_2 (« *TrancheDe10* ») sont $\{m5, m_6\}$ et celui du niveau n_3 (« *TrancheDe50 »)* est $\{m_7\}$. L'ensemble $\mathcal{MF}_{âgeParTranche}$ est défini par : $\mathcal{MF}_{âgeParTranche} = \{m_1, m_2, m_3, m_4\}$
On obtindra le graphe orienté de la figure 15.

On définit de la même manière les versions de dimension « *âgeParClasse* », « *QTalgo1* » et « *QTalgo2* » dont les schémas sont $S_{âgeParClasse}$, $S_{QTalgo1}$, $S_{Qtalgo2}$ ainsi que les ensembles

$\mathcal{MF}_{âgeParClasse}$, $\mathcal{MF}_{QTalgo1}$ et $\mathcal{MF}_{Qtalgo2}$. On obtient respectivement les graphes orientés de la figure 16, la figure 11 présenté dans la partie 3.1.2 et la figure 17.

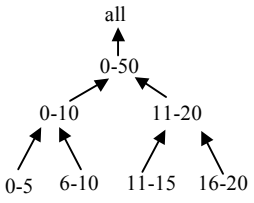


Figure 15
Version de dimension âgeParTranche

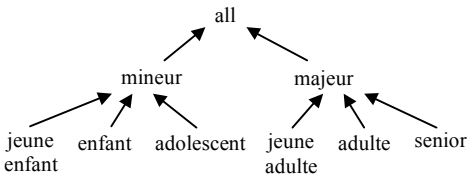


Figure 16
version de dimension « âgeParClasse »

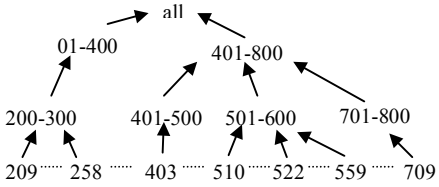


Figure 17
la version de dimension « QTalgo2 »

Définition 3 (dimension multiversion).
Une dimension multiversion ***DMV*** est une dimension qui contient 1 à n versions de dimension. Elle est définie par le tuple <***idDMV,nomDMV,*** $\mathcal{VD}$***,[descriptionDMV]***> où :

- ***idDMV*** est l'identifiant unique pour la dimension multiversion
- ***nomDMV*** est le nom de la dimension multiversion
- $\mathcal{VD} = \{VD_i, i=1,...,n\}$ est l'ensemble des versions de dimension associées à cette dimension multiversion
- ***DescriptionDMV*** est une propriété optionnelle contenant des informations textuelles sur la dimension multiversion

On note $\mathcal{MF}_{DMV}$ l'ensemble des membres-feuilles des versions de dimension contenues dans la dimension multiversion *DMV*. Cet ensemble est défini par : $\mathcal{MF}_{DMV} = \bigcup_{i=1}^{n} \mathcal{MF}_{VD_i}$ avec *n* le nombre de versions de dimension contenues dans la dimension multiversion *DMV*.

Exemple 3 :
Les versions de dimension « *âgeParTranche* » et « *âgeParClasse* » définies précédemment appartiennent à la dimension multiversion « *Age* » d'identifiant 1. Cette dimension multiversion est définie par :
Age = <1, « Age », {« ageParTranche », « ageParCatégorie »}>

Par souci de clarté, nous utiliserons dans la suite du texte les noms des membres de versions de dimension pour les identifier. On définit l'ensemble $\mathcal{MF}_{Age}$ par :
$\mathcal{MF}_{Age}$ *= {0-5, 6-10, 11-15, 16-20, jeune enfant, enfant, adolescent, jeune adulte, adulte, senior}*

Les versions de dimension « *QTalgo1* » et « *QTalgo2* » appartiennent à la dimension multiversion «*DuréeQT* » d'identifiant 2. Cette dimension multiversion est définie par :
DuréeQT = <2,QT, { QTalgo1, QTalgo2}>
On définit l'ensemble $\mathcal{MF}_{DuréeQT}$ par :
$\mathcal{MF}_{DuréeQT}$ *= {0,205, 230, 395, 403, 475, 512, 685, 709}*

Définition 4 (Table de fait multiversion fonctionnelle).
Une table de fait multiversion fonctionnelle fournit les mesures selon les différentes versions de dimension. Soit $\{\mu_i, i=1,...,m\}$ l'ensemble des mesures, une table de fait multiversion fonctionnelle ***tf*** est définie par une fonction telle que :

$$tf : DMV_1 \times DMV_2 \times ... \times DMV_n \rightarrow dom(\mu_1),...,dom(\mu_m)$$
$$m_1, m_2, ..., m_n \rightarrow v_1,...,v_m$$

où n est le nombre de dimensions multiversions de l'entrepôt, $m_i \in \mathcal{MF}_{DMV_i}$ avec $i=1,...,n$ et $dom(\mu_k)$ est le domaine des valeurs de la mesure μ_k. Cette fonction associe à un ensemble de membres feuille des versions de dimension de chaque dimension multiversion, l'ensemble des valeurs v_k des mesures μ_k.

Définition 5 (Fonction de version de dimension).
Les fonctions de version de dimension sont les modes de calcul qui permettent d'obtenir les membres d'une version de dimension *VD* à partir des données de la base de données de production. Une fonction de version de dimension $\boldsymbol{f_{VD}}$ est définie par le tuple $<\boldsymbol{idFonction_{VD}}, \boldsymbol{idVD}, \boldsymbol{nomFonction_{VD}}, \boldsymbol{énoncéFonction_{VD}}>$ où :

- $\boldsymbol{idFonction_{VD}}$ est l'identifiant de la fonction de version de dimension *VD*
- ***idVD*** est l'identifiant de la version de dimension *VD* dont les membres sont calculés en utilisant cette fonction de version de dimension
- $\boldsymbol{nomFonction_{VD}}$ est le nom de la fonction de version de dimension
- $\boldsymbol{énoncéFonction_{VD}}$ est l'énoncé de la fonction de version de dimension

Ces fonctions sont de la forme :

$$f_{VD} : \mathcal{BD}_f \rightarrow \mathcal{MF}_{\mathrm{VD}}$$
$$d \rightarrow m$$

où $\mathcal{BD}_f$ est l'ensemble des données de la base de données de production restreint à f_{VD} c'est-à-dire utilisé pour calculer les membres de *VD*. f_{VD} associe à une valeur de la base de données de production, un membre-feuille de la version de dimension correspondante *VD*.

Exemple 4 :
Supposons que dans la base de données de production, on ait un patient de *12 ans*. Soit la fonction $f_{âgeParClasse}$ définie pour la version de dimension « *âgeParClasse* » et la fonction $f_{âgeParTranche}$ définie pour la version de dimension « *âgeParTranche* ». On obtient alors respectivement pour les versions de dimension, les membres :

$f_{âgeParClasse}(12)$ = « jeune »
$f_{âgeParTranche}(12)$ = « 11-15»

De la même manière, supposons un électrocardiogramme « *ECG5* » de la base de données de production. Soit la fonction $f_{Qtalgo1}$ définie pour la version de dimension « *Qtalgo1*» et la fonction $f_{Qtalgo2}$ définie pour la version de dimension « *Qtalgo2*». On obtient alors alors respectivement pour les versions de dimension, les membres :

$f_{Qtalgo1}(ECG5) = 100$
$f_{Qtalgo2}(ECG5) = 110$

Définition 6 (structure multidimensionnelle multiversion fonctionnelle)
Une structure multidimensionnelle multiversion fonctionnelle ***M2F*** est définie par le tuple $\boldsymbol{<\mathcal{DMV}, tf, \mathcal{F}>}$ où :

- $\boldsymbol{\mathcal{DMV}} = \bigcup_{i=1}^{s} DMV_i$ est l'ensemble des dimensions multiversions
- ***tf*** est la table de fait multiversion fonctionnelle

- $\mathcal{F} = \bigcup_{j=1}^{r} f_{VD_j}$ est l'ensemble des fonctions des versions de dimension

Définition 7 (Agrégation de données).
Les agrégations de données peuvent être calculées à partir de la table de fait multiversion et des schémas des versions de dimension. Soit une fonction d'agrégation $\oplus_{\mu_k}$ pour chaque mesure μ_k , m un membre non-feuille de la version de dimension *VD* de la dimension multiversion DMV_l et $m_1^1, m_2^1, ..., m_J^1$ ses enfants (membres-feuilles) c'est-à-dire tels que :
$(m_1^l, m_2^l, ..., m_J^l) \in \mathcal{MF}_{VD} \times \ldots \times \mathcal{MF}_{VD}$
On a la relation suivante :

$$\forall j \in [1, J], tf(m_j^l, m_2, ..., m_n) = v_1^j, ..., v_m^j$$

avec n le nombre de dimensions multiversions de l'entrepôt.
Ainsi on obtiendra comme valeurs pour *m* :

$$tf(m, m_2, ..., m_n) = \oplus_{j=1}^{J} \mu_1 \, v_1^j, ..., \oplus_{j=1}^{J} \mu_m \, v_m^j$$

3.3. Modèle logique

Afin de permettre l'implémentation de notre approche sur les outils actuels, des adaptations du modèle conceptuel sont suggérées permettant de définir le modèle logique.

3.3.1. Dimensions multiversions

Une dimension multiversion contient un ensemble de versions de dimension. Elle peut être considérée comme une dimension classique dans laquelle les membres sont regroupés par version, chaque version ayant un schéma, une hiérarchie propre. Cependant une telle dimension ne peut avoir de membre all (niveau ALL). Les membres all de chaque version de dimension ne peuvent pas être regroupés en un seul membre puisqu'ils n'appartiennent pas à la même hiérarchie. (cf. figure 18)

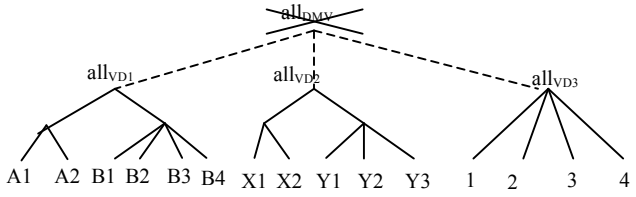
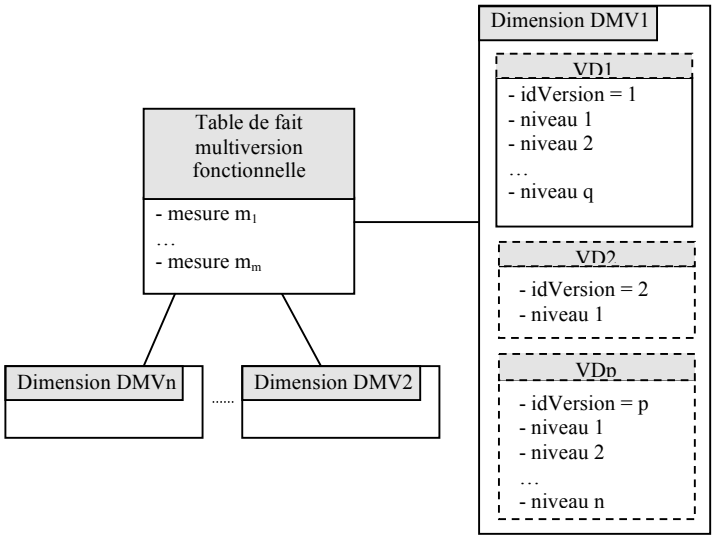


Figure18

3.3.2. Metadonnées

Les métadonnées regroupent toutes les informations à propos du cube multidimensionnel. Ces données ne sont pas intégrées au cube directement mais placées dans des tables relationnelles à part. Les informations à propos de la composition des dimensions multiversions et des schémas de chaque version de dimension sont présentes dans les métadonnées ce qui permet des visualisations du modèle au niveau logique. De plus, les informations sur les fonctions de versions de dimension facilitent le chargement de l'entrepôt.

3.3.3. Schéma du modèle logique



3.4. Modèle Physique et implémentation

Nous présentons dans cette partie le modèle physique et les choix que nous avons faits pour l'implémentation. Tout d'abord, nous présentons la structure multidimensionnelle multiversion fonctionnelle de l'entrepôt en décrivant la table de fait multiversion fonctionnelle et les dimensions multiversions. Puis nous décrivons les métadonnées. Enfin, nous développons l'architecture globale que nous avons adoptée.

3.4.1. Structure multidimensionnelle multiversion fonctionnelle de l'entrepôt

3.4.1.1. La table de fait multiversion fonctionnelle

La table de fait multiversion fonctionnelle est donc constituée des valeurs du fait pour toutes les combinaisons entre les versions de dimension de chaque dimension multiversion. A la construction, nous associons aux faits, des fonctions d'agrégation, nous permettant d'avoir des données agrégées suivant les hiérarchies des différentes versions de dimension.

3.4.1.2. Les dimensions multiversions

Une dimension multiversion regroupe dans une même table les membres de toutes les versions de dimension qu'elle contient. Les champs de la table sont fixes pour toutes les versions de dimension et les attributs des membres devront être identiques pour toutes les versions de dimension d'une même dimension multiversion.

La représentation « parent-enfant » s'adapte très bien à notre approche. Elle est notamment proposée par l'outil commercial Microsoft SQL Server 2000. Dans ce modèle, la dimension est stockée sur une seule table et chaque membre correspond à un tuple et a comme attribut la référence au membre de la table qui est son parent dans la hiérarchie. Dans ce cas, l'instance de la dimension est construite à partir des liens parents-enfants existants entre les membres. Ce modèle nous permet de traiter les hiérarchies des différentes versions de dimension différemment. Ce que les modèles en étoile ou en flocon ne nous permettent pas puisque dans de tels modèles les niveaux sont fixés pour une dimension. Au niveau physique, la représentation « parent-enfant » supporte les hiérarchies complexes mises à part les hiérarchies multiples (la clé de la table de dimension doit être l'identifiant du membre, ce qui ne permet pas à un membre d'avoir plusieurs parents). Cependant dans notre approche, nous utilisons, dans les tables des dimensions multiversions, un attribut donnant la version

de dimension à laquelle appartient le membre. Cet attribut, associé à la représentation parent-enfant, nous permet de modéliser les différentes versions de dimension et leur hiérarchie en les distinguant facilement ainsi que de modéliser toutes les hiérarchies complexes (hiérarchies multiples, non-strictes, non-onto,…).

Nous utilisons également la notion de « parent-enfant » pour les schémas des versions de dimensions. Ces schémas sont construits à partir des niveaux et des liens qu'il existe entre ces niveaux. Ces liens sont représentés par des relations parent-enfant dans une table contenant toutes les informations sur les niveaux. Nous détaillerons cette table dans les métadonnées (partie 3.4.2 du prapport).

Dans les hiérarchies multiples, les membres qui appartiennent au même niveau sont dupliqués et placés dans des versions de dimension différentes de la même dimension multiversion. Supposons par exemple que la dimension multiversion « *Date* » dont une version de dimension est « *dateParSemaineOuMois* » définie dans la partie 3.1.2 du rapport (figure 5) soit organisée en hiérarchie multiple. Dans notre modèle physique, nous transformons cette version de dimension en deux versions de dimension, en dupliquant les niveaux communs et leurs membres. (cf. figure 19). Ainsi les membres des niveaux « *Jours* » et « Années » sont dupliqués. De plus, afin de conserver la notion de hiérarchie multiple et de visualiser correctement la hiérarchie, nous utilisons une table des métadonnées qui conserve les liens pour recréer la hiérarchie multiple.

Les hiérarchies non-onto et non-couvrantes sont supportées en utilisant les relations parent-enfant entre les membres et entre les niveaux d'une version de dimension. Supposons par exemple la dimension multiversion « PathologiePrincipale » constituée d'une version de dimension « *familleDePathooigie* » soit organisée en hiérarchie non-onto et définie dans la partie 3.1.2 du rapport (figures 6 et 7). Considérons également que la dimension multiversion « *DuréeQT* » et la version de dimension « *QTalgo1* » soit organisée en hiérarchie non-couvrante et définie dans la partie 3.1.2 du rapport (figure 10 et 11). La représentation parent-enfant permet de supporter ces deux types de hiérarchies complexes puisque les schémas et les instances des versions de dimension sont construits respectivement à partir des liens de filiation qu'il existe entre les niveaux et entre les membres.

Enfin, en résonnant sur le même principe que les hiérarchies multiples mais en considérant les membres des versions de dimension et non plus les niveaux, notre approche permet de traiter les hiérarchies non-strictes. Considérons maintenant que la dimension multiversion « *HoraireAcquiqition* » dont l'une des versions de dimension «*TrancheHoraireAcquisition* » définie dans la partie 3.1.2 du rapport (figure 8 et 9) soit organisée en hiérarchie non-stricte. Dans ce cas, nous dédoublons le membre qui a plusieurs parents (ainsi que tous ses enfants) en établissant un lien de filiation entre chacun des membres obtenus et l'un ou l'autre des membres du niveau supérieur. (cf. figure 20). Pour ne pas perdre d'informations, une table dans les métadonnées conserve les liens permettant de recréer la hiérarchie non-stricte.

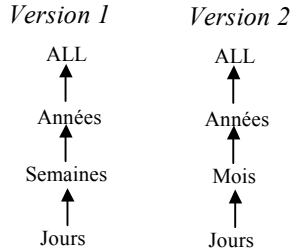


Figure 19
Exemple de traitement d'une hiérarchie multiple

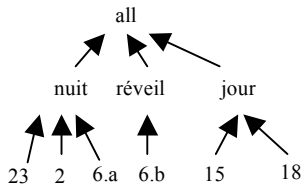


Figure 20
Exemple de traitement d'une hiérarchie non-stricte

3.4.2. Stockage des métadonnées

Les métadonnées sont stockées dans des tables relationnelles. Nous distinguons six tables relationnelles de métadonnées :

- Une table relative aux dimensions multiversions qui contient le nom, la description d'une version de dimension multiversion à partir de son identifiant.
- Une table relative aux versions de dimension qui renseigne sur le nom, la description d'une version de dimension à partir de son identifiant ainsi que l'identifiant de la dimension muliversion dans laquelle elle est contenue.
- Une table relative aux niveaux qui contient le nom, la description d'un niveau à partir de son identifiant ainsi que l'identifiant du niveau parent.
- Une table relative aux fonctions de version de dimension c'est-à-dire aux calculs des membres des versions de dimensions des dimensions multiversions.
- Une table relative aux hiérarchies multiples afin de ne pas perdre d'information et de conserver les liens permettant de recréer la hiérarchie multiple pour la visualiser correctement. Par exemple, pour la version de dimension « *dateParSemaineOuMois* » présentée précédemment, la table des métadonnées est la table 1 ci-dessous.

idVDA	Niveau A	idVDB	Niveau B
Version1	Jours	Version2	Jours
Version1	Année	Version2	Année

Table 1

Remarque : Dans le cas où nous avons plus de deux versions, nous créons une « chaîne » de liens comme illustré dans la table suivante (cf. table 2). Afin de retrouver la hiérarchie, nous cherchons si le « *niveau1* » de *V1* a une correspondance avec un niveau d'une autre version et si c'est le cas, nous cherchons alors si le niveau de cette autre version a une correspondance avec un niveau d'une autre version ainsi de suite jusqu'à trouver tous les équivalents.

idVDA	Niveau A	idVDB	Niveau B
Version1	Niveau1	Version 2	Niveau1
Version2	Niveau1	Version 3	Niveau1
Version1	Niveau2	Version 2	Niveau2
Version2	Niveau2	Version 3	Niveau2

Table 2

- Une table relative aux hiérarchies non-strictes, basée sur le même principe que précédemment mais en considérant les membres des versions de dimension et non les niveaux. Elle permet de ne pas perdre d'information et de conserver les liens afin de recréer la hiérarchie non stricte (permettant donc de visualiser l'instance de la hiérarchie). Par exemple, pour la version de dimension « *famillePathologie* » définie précédemment, la table des métadonnées correspond à la table 3:

idVD	Membre A	Membre B
V1	6.a	6.b

Table 3

Remarque : Nous effectuerons la même remarque sur le chaînage des données que précédemment.

Les trois premières tables des métadonnées permettent d'avoir des précisions sur les dimensions multiversions, les versions de dimension et les niveaux de hiérarchies des versions de dimension qui sont stockés en tant qu'attributs dans les tables des dimensions du cube lui-même. Il est possible d'obtenir des informations sur chaque dimension multiversion, sur les versions de dimension qui les composent et sur leurs membres ainsi que sur les niveaux de hiérarchie. Ces tables permettent de construire les schémas des versions de dimension et leur instance afin de les visualiser

La table suivante correspond aux fonctions de calcul des membres des versions de dimension. Ces fonctions ne figurent pas dans le cube lui-même. Ces données nous permettent de savoir comment les membres sont obtenus à partir des données de la base de données de production, ce qui permet un meilleur suivi et une meilleure analyse des résultats.

Les deux dernières tables permettent de conserver l'information selon laquelle les hiérarchies des versions de dimension sont soit multiples, soit non-strictes. Ainsi, les schémas et les instances des versions de dimensions organisées en hiérarchies multiples et non-strictes peuvent être correctement visualisés.

3.4.3. Architecture du système

Pour l'implémentation, nous utilisons l'outil SQL Server 2000 de Microsoft pour héberger les tables de la base de données de production ainsi que les tables de dimensions et les métadonnées.
L'outil Analysis Services de SQL Server permet de gérer les entrepôts de données OLAP comme nous le souhaitons. De plus, comme pour la plupart des entrepôts de données, nous adoptons une architecture 3-tiers présentée par la figure 21 et composée des trois parties suivantes :

- Entrepôt de données multimédias multiversion fonctionnelle
- Cube OLAP
- Outils de visualisation de données

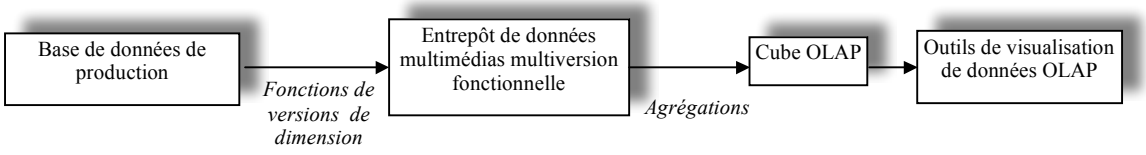


figure 21

3.5. Mise en œuvre sur l'étude EMIAT

En utilisant les données de l'étude EMIAT présentée en début de la partie 2 du rapport, nous développons un prototype mettant en oeuvre notre modèle multidimensionnel multiversion fonctionnelle. Les données de cette étude, données multimédias de type signaux caractérisées par des descripteurs sont intégrées à un entrepôt de données mutimédias multiversion fonctionnelle. Le prototype repose sur un cube OLAP construit à l'aide de Analysis Services et Visual Basic pour les fonctions d'agrégations et une interface Visual Basic utilisant des composants de Proclarity 4.0 pour la navigation.

L'entrepôt de données est composé d'une table de fait multiversion fonctionnelle et de huit dimensions multiversions. Le schéma de l'entrepôt est présenté en annexe (cf. annexe 1). Les faits sont les ECGs de l'étude EMIAT. Les dimensions multiversions représentent les descripteurs textuels et les descripteurs de contenu de ces ECGs. Parmi les dimensions multiversions, trois portent sur le patient, trois autres sur l'acquisition de l'ECG et deux autres sur le contenu de l'ECG. Les huit dimensions multiversions sont:

- La dimension multiversion « *PathologiePrincipale* » qui contient la version de dimension « *familleDePathologie* ». Celle-ci nous renseigne sur la pathologie principale du patient lors de l'enregistrement de l'ECG et propose différents niveaux de granularité comme pathologies détaillées, pathologies, familles de pathologies, etc.
- La dimension multiversion « *Age* » qui contient trois versions de dimensions. La première, nommée « *dateDeNaissance* », permet de connaître la date de naissance des patients à des niveaux de granularité tels que jours, mois ou année. La deuxième et la troisième, nommées « *âgeParClasse* » et « *âgeParTranche* » regroupent les âges des patients respectivement par

classes d'âges et par tranches d'âges. Ces deux versions de dimension sont présentées dans le modèle conceptuel.

- La dimension multiversion « *Sexe* » qui contient la version de dimension « *sexeParGenre* » avec un niveau de granularité unique permettant de connaître le sexe du patient.
- La dimension multiversion « *Date* » qui contient la version de dimension « *dateParSemaineOuMois* ». Elle donne la date d'enregistrement de l'ECG et est définie dans la partie 3.1.2 du rapport comme exemple de hiérarchie multiple.
- La dimension multiversion « *HoraireAcquisition* » qui contient la version de dimension « *trancheHoraireAcquisition* ». Cette version de dimension classe les horaires d'acquisition de l'ECG par tranches d'horaires ou périodes où l'ECG a été enregistré.
- La dimension multiversion « *Technologie* » qui contient la version de dimension « *typeTechnologie* ». Elle permet de connaître la technologie, l'appareil avec laquelle l'ECG a été enregistré et les regroupe en type de technologie dans un niveau de granularité supérieure.
- La dimension multiversion « *DuréeQT* » qui contient deux versions de dimension appelées respectivement « *QTalgo1* » et « *QTalgo2* ». Ces deux versions de dimension donnent les valeurs de la durée du QT d'un ECG calculées à l'aide de deux algorithmes. Elles sont définies dans le modèle conceptuel.
- La dimension multiversion « *NiveauDeBruit* » qui contient la version de dimension « *niveauBruitParCatégorie* ». Celle-ci nous renseigne sur le niveau de bruit de l'ECG et regroupe les différentes valeurs en catégorie dans un niveau de granularité supérieure.

Certains schémas et instances de versions de dimensions sont présentés en annexe (cf. annexe 2) sous forme de graphe.

Les agrégations de données sont calculées à partir de la table de fait multiversion fonctionnelle et des liens hiérarchiques entre les membres des versions de dimension. Les fonctions d'agrégation permettent de calculer des données agrégées suivant les niveaux de granularité des schémas des versions de dimension. Dans notre entrepôt de données, nous définissons les trois fonctions d'agrégat suivantes:

- nombreECG : cette fonction renseigne sur le nombre d'ECGs qui répondent aux caractéristiques choisies par l'utilisateur
- listeECG : cette fonction retourne la liste des identifiants des ECGs qui répondent aux caractéristiques choisies par l'utilisateur
- ECGmoyen : cette fonction donne l'identifiant de l'ECG moyen correspondant à la liste d'ECGs retournée par la fonction précédente. Cette ECG moyen est représenté par un ensemble de points calculés en faisant la moyenne entre les points de tous les ECGs de la liste. Il représente un agrégat de données multimédias.

La première fonction est une fonction d'agrégation classique « count ». Les deux suivantes sont des fonctions d'agrégation spécifiques qui ont été définies grâce à Visual Basic et intégrées à Analysis Services grâce à une de ses fonctionnalités « membre calculé ».

3.6.L'interface OLAP

L'application développée permet de visualiser les données de notre entrepôt de données multimédias multidimensionnel multiversion fonctionnelle. L'interface permet de visualiser les ECGs, les métadonnées et les hiérarchies des versions de dimensions.

3.6.1. Exploration des données multimédias

Notre interface permet d'explorer les données agrégées dans un tableau à deux entrées. Nous pouvons naviguer en choisissant l'agrégation de données (nombreEcg, listeECG ou ECGmoyen) à analyser et les dimensions multiversions selon lesquelles nous voulons explorer les données tout en fixant les niveaux des autres dimensions multiversions. Nous pouvons également sélectionner plusieurs dimensions pour une même entrée afin d'explorer les données selon plusieurs critères. Par exemple, nous pouvons choisir d'analyser les données selon l'âge par tranche, la pathologie et la date.

Il est possible de visualiser plusieurs versions de dimension d'une même dimension multiversion afin de faire des comparaisons. Un exemple est présenté dans la partie A et B de la figure 23 où les deux versions de dimension de la dimension multiversion « *DuréeQT* » sont sélectionnées pour une entrée. Sur l'autre entrée du tableau, la dimension « *PathologiePrincipale* » est sélectionnée permettant d'explorer les données c'est-à-dire ici, les liste d'ECGs selon les pathologies et les deux versions de dimensions « *QTalgo1* » et « *QTalgo2* ».

Enfin, les données multimédias peuvent être visualisées en sélectionnant un ECGmoyen ou en choisissant un ECG dans les données agrégées « listeEcg » obtenues. On voit dans la partie C de la figure 22, le schéma de la voix Y de l'ECG d'identifiant 6.

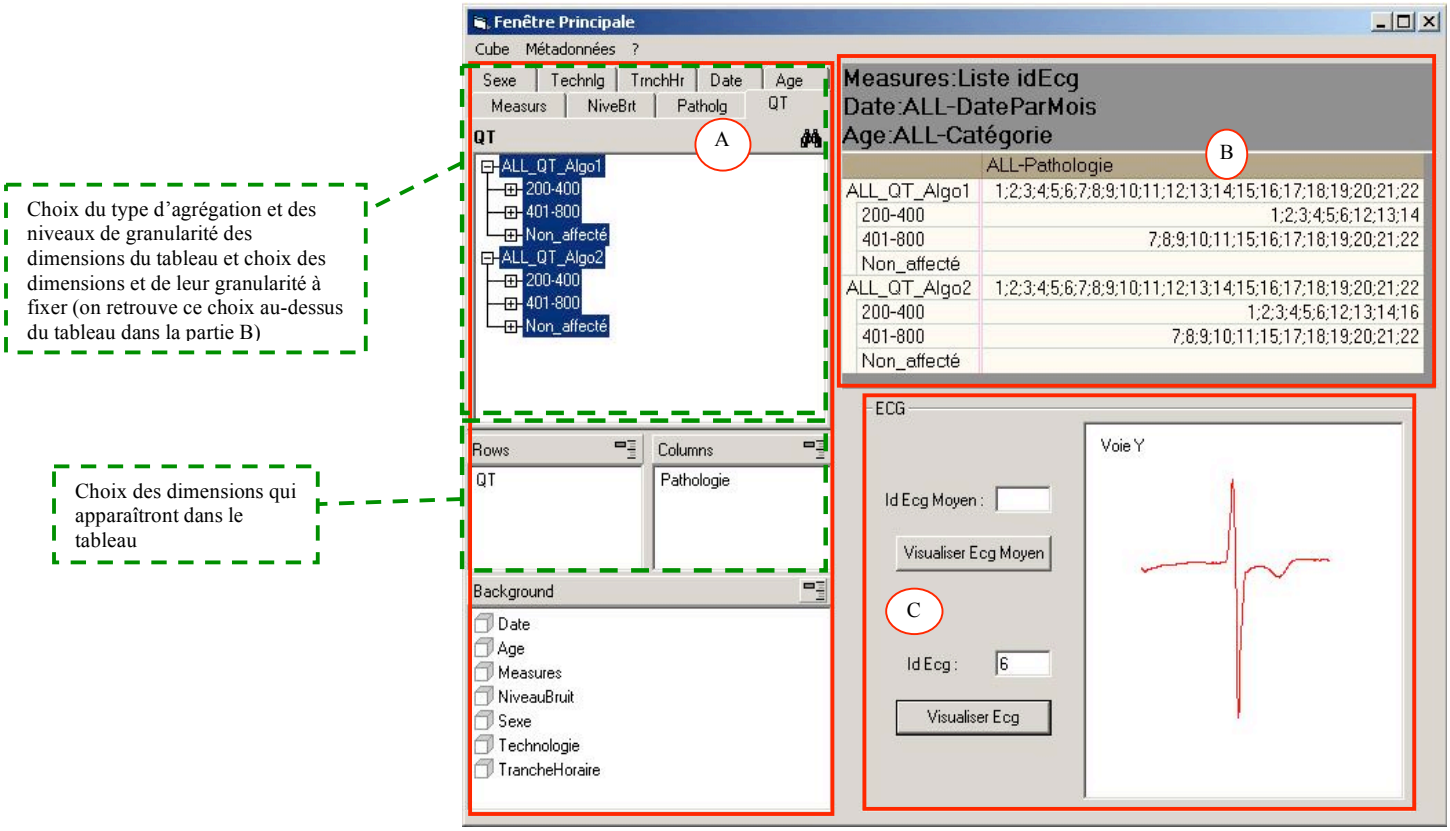


figure 22

A- Choix des dimensions et des agrégations de données: cette partie permet de sélectionner les valeurs voulues pour chaque dimension et le type d'agrégation de la mesure à analyser
B- Navigation dans le cube de données et exploration des données
C- Visualisation des données multimédias

3.6.2. Visualisation des métadonnées

Nous représentons les schémas et les instances de chaque version de dimension pour chaque dimension multiversion. Cela permet à l'utilisateur d'avoir une vue globale des différentes versions de dimension afin de naviguer plus facilement dans le cube. Par exemple, nous pouvons visualiser sur la figure 23 l'instance de la version de dimension « *QTalgo1* » de la dimension multiversion

« *DuréeQt* ». Il est possible de sélectionner la dimension multiversion et la version de dimension que nous souhaitons visualiser. Il est ensuite possible de choisir la représentation de la version de dimension c'est-à-dire le schéma (hiérarchie des niveaux de la version de dimension) ou l'instance (hiérarchie des membres de la version de dimension).

Nous permettons également de représenter les fonctions de versions de dimension afin d'améliorer l'analyse des résultats obtnus.

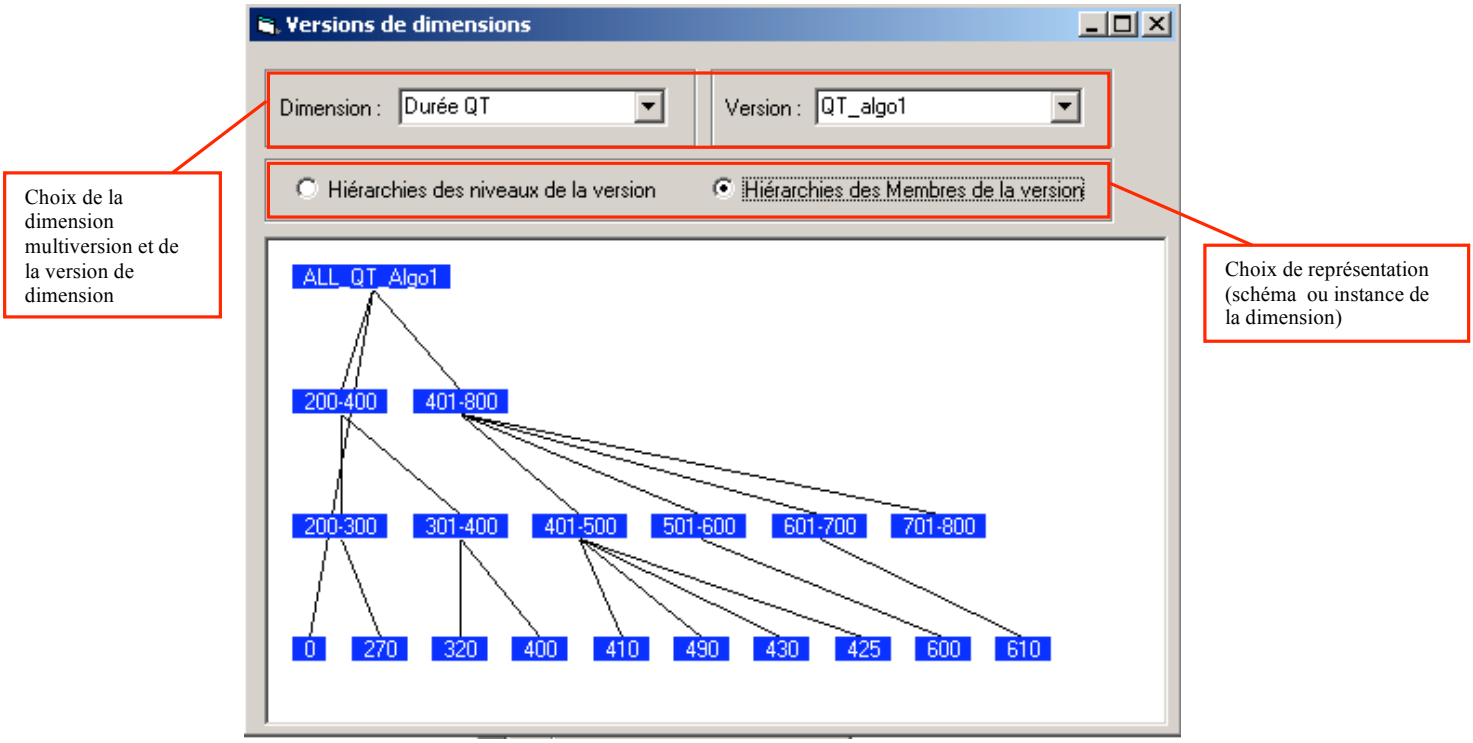


figure 23 :
instance de la version de dimension QT_algo1

4. Discussion et conclusion

4.1. Apports de notre approche

Nous avons présenté un nouveau modèle qui permet de gérer les données complexes comme les données multimédias en proposant à l'utilisateur de choisir différentes vues pour représenter les données par diverses versions fonctionnelles des descripteurs de ces données. Pour cela nous avons défini la notion de version et multiversion dans les dimensions et la table de fait pour pouvoir comparer les résultats obtenus en choisissant ces diverses versions. Nous avons également utilisé des fonctions spécifiques aux données multimédias que nous avons intégrées à l'entrepôt afin d'analyser au mieux ces données. Par ailleurs, de ce modèle conceptuel, nous avons dégagé un modèle logique et physique permettant l'implémentation de notre approche. Nous avons alors proposé un outil d'exploration de ces données complexes qui facilite la navigation dans le cube de données multidimensionnel. Nous permettons ainsi de visualiser les données suivant plusieurs versions des axes d'analyse et nous donnons la possibilité de visualiser la représentation de ces données multimédias.

4.2. Limites de notre modèle

Plusieurs limites peuvent exister dans notre approche. Certaines sont liées au modèle conceptuel, d'autres portent sur l'architecture logique et physique.

Dans l'approche conceptuelle, nous intégrons la notion de version dans les dimensions puisqu'il est nécessaire de prendre en compte les versions fonctionnelles des descripteurs. Cependant par manque de temps, nous n'avons pas considéré les versions de fait qui peuvent être intéressantes à traiter.

Des inconvénients apparaissent au niveau de l'implémentation pour le stockage des données. Tout d'abord nous avons défini un schéma par version de dimension. Or dans le cas où deux versions de dimension ont le même schéma, il y a redondance. Le traitement des hiérarchies multiples et non-strictes entraîne également des duplications de valeurs et une redondance de données. Le stockage de la table de fait peut être lui aussi optimisé puisque toutes les combinaisons entre les versions de dimension sont stockées dans la table de fait multiversion fonctionnelle. Pour toutes les versions d'une même dimension multiversion, les membres ont le même type d'attribut et les mêmes attributs puisqu'ils sont stockés dans la même table.

Enfin l'utilisation de la fonctionnalité « membre calculé » permet de calculer les agrégations en temps réel. Cependant les données multimédias sont volumineuses et difficiles à manipuler. Les temps de réponse deviennent plus longs et la navigation dans le cube de données se trouve limitée.

4.3. Perspectives

Au niveau du modèle conceptuel, il serait intéressant d'intégrer la notion de version aux faits, permettant ainsi d'avoir plusieurs versions d'un même fait. Pour cela, une dimension version de fait doit être ajoutée au modèle afin de permettre à l'utilisateur de choisir la version du fait qu'il souhaite analyser. Par exemple, supposons que les ECGs soient représentés de manière classique d'une part, comme nous l'avons vu dans notre modèle et d'autre part, par une représentation par transformée en ondelettes. Les données pourraient être analysées selon diverses vues au niveau des dimensions mais également diverses vues au niveau des faits.

Pour le stockage des données, ce modèle peut être amélioré afin d'obtenir de meilleures performances. En effet plusieurs optimisations sont envisageables. La première serait l'optimisation des requêtes en utilisant une fonctionnalité de cache pour ne pas recalculer les requêtes ayant déjà été calculées auparavant. La deuxième serait l'optimisation du stockage de l'entrepôt puisque si le nombre de dimensions multiversions et le nombre de versions de dimension sont très importants, l'entrepôt peut devenir difficile à stocker. Certaines versions de dimension pourraient être générées dynamiquement au moment de la navigation en utilisant des fonctions de mapping. Ces fonctions permettraient de calculer les membres d'une version de dimension à partir des membres d'une autre version de dimension de la même dimension multiversion. Ainsi ne seront stockées dans l'entrepôt que les versions de dimension les plus utilisées lors de la navigation dans le cube de données multimédias.

Enfin, une autre amélioration possible au niveau de la navigation dans le cube de données est le stockage des « membres calculés » pour accélérer cette navigation. Ces membres seraient calculés lors de la première navigation et enregistrés dans une table, puis simplement lus dans cette table lors des navigations suivantes. Nous pourrions également envisager de pré-calculer uniquement les données auxquelles les utilisateurs accèdent le plus fréquemment ou essayer de calculer des cubes approximatifs comme dans les systèmes SOLAP.

5. Références bibliographiques

[AGR95]
R. Agrawal, A. Gupta, S. Sarawagi. *Modeling Multidimensional Databases*. IBM Research Report. IBM Almaden Research Center, September 1995. 25p.

[BOD02a]
Body, M., Miquel M., Bédard Y., A. Tchounikine, *A multidimensional and multiversion structure for OLAP applications,* ACM Fifth International Workshop on DATA WAREHOUSING AND OLAP (DOLAP 2002), McLean, VA, USA, November 8th 2002.

[BOD02b]
Body, M., Miquel M., Bédard Y., A. Tchounikine, *Handling Evolutions in Multidimensional Structures,* ICDE 2002, the 19th International Conference on Data Engineering, Sponsored by the IEEE Computer Society, March 5 - March 8 2003, Bangalore, India.

[CAB98]
Luca Cabibbo, Riccardo Torlone, *A logical approach to multidimensional databases*, EDBT,1998.

[COD93]
E.F Codd, S.B Codd, C.T. Salley, P*roviding OLAP (on-line analytical processing) to user-analysts: An IT mandate*, Technical report, 1993.

[EDE01]
J. Eder, C. Koncilia, *Evolution of dimension data in temporal data warehouse*, In Proc. of the DaWak'01 Conference, Munich, Germany, 2001.

[HAN01]
Jiawei Han, Micheline Kamber, *Data mining, concepts and techniques*, Morgan Kaufmann Publishers, 2001.

[HUR99a]
Hurtado C., Mendelzon A., and Vaisman A., *Maintaining Data Cubes under Dimension Updates*, IEEE International Conference on Data Engineering, 1999.

[HUR99b]
Hurtado C., Mendelzon A., and Vaisman A., *Updating OLAP dimensions*, In Proc. of ACM second Int. Workshop on data warehousing and OLAP, USA, 1999.

[INM02]
W.H.INMON. *Building a data warehouse*. J. Wiley and Sons, 2002.

[JAG99]
H. V. Jagadish, Laks V. S. Lakshmanan , Divesh Srivastava, *What can hierarchies do for warehouses?*, In Proc. Of the 25th VLDB Conference, Edinburgh, Scotland, 1999.

[KAM97]
Vera Kamp, Frank Wietek, *Database system support for multidimensional data analysis in environmental Epidemiology*, IEEE Computer Society , 1997.

[KIM00]
Kimball R., *Is Your Dimensional Data Warehouse Expressive?*, Intelligent Enterprise, volume 3, no. 8, May 2000.

[MEN 00]
Mendelzon A., Vaisman A., *Temporal Queries in OLAP, VLDB 2000*: 242-253, 2000.

[MIQ01]
Miquel M., Y. Bédard & A. Brisebois, *Conception d'entrepôts de données géospatiales à partir de sources hétérogènes, exemple d'application en foresterie,* Ingénierie des Systèmes d'information-Networking and information Systems, Edition Hermes Science, Paris, 19 p. Paris, Vol.7, No 3, pp.89-111, 2002.

[PED 01]
Pedersen T.B., Jensen C., Dyreson C., *A foundation for capturing and querying complex multidimensional data,* Information Systems Special Issue: Data warehousing, vol 26, No 5, 2001.

[RAH95]
Rahul V. Tikekar, Farshad Fotouhi , *Storage and retrieval of medical images from data warehouses*, Digital Image Storage and Archiving Systems, 1995.

[SUB98]
V.S.Subrahmanian, *Principles of multimedia database systems*, Morgan Kaufmann Publishers Inc ,1998.

[TSO01]
Aris Tsois, Nikos Karayannidis, Timos Sellis, *MAC : conceptual data modeling for OLAP*, DMDW'2001.

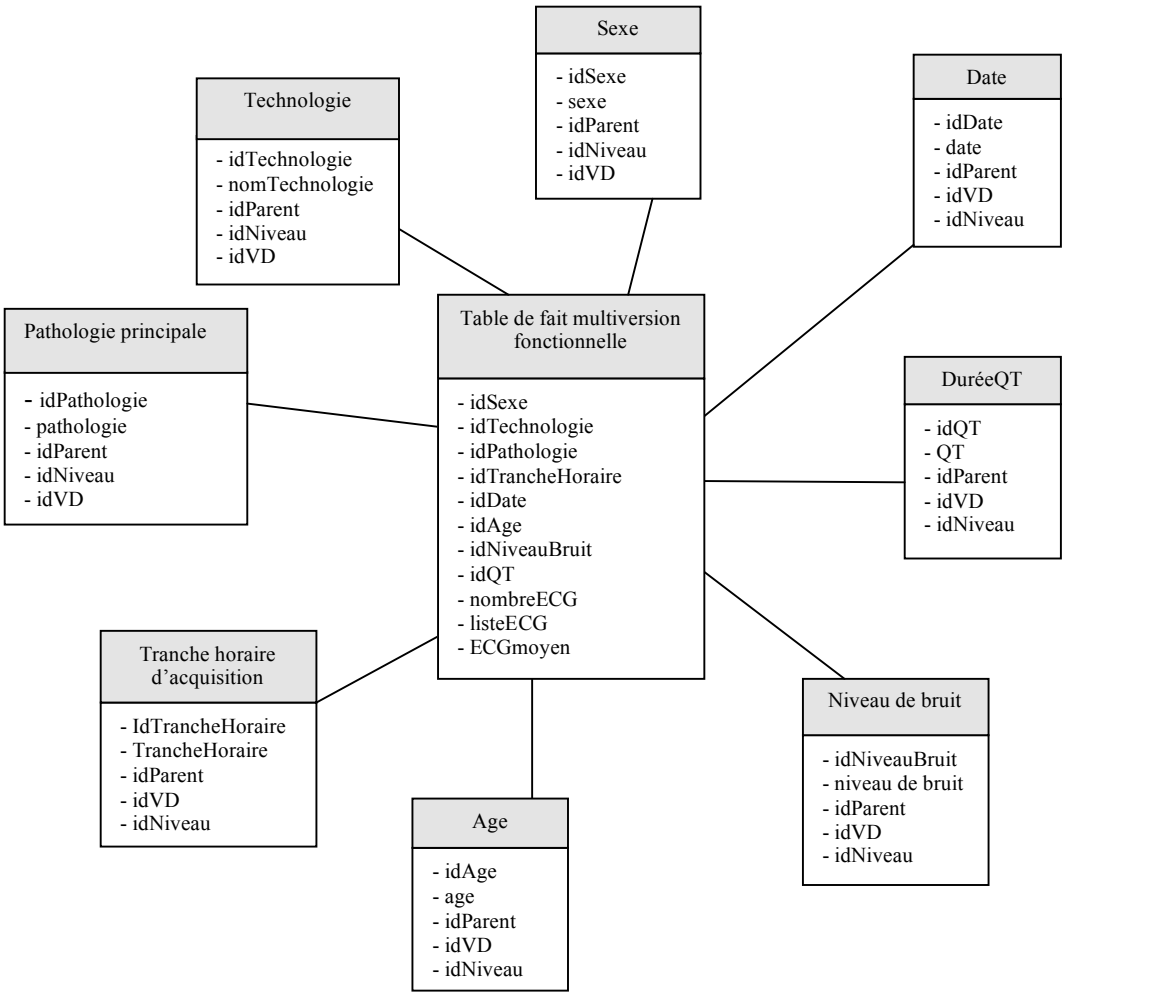
[YOU01]
Jane You and al., *On hierarchical multimedia information retrieval*, IEEE International Conference on Image Processing (ICIP), 2001.

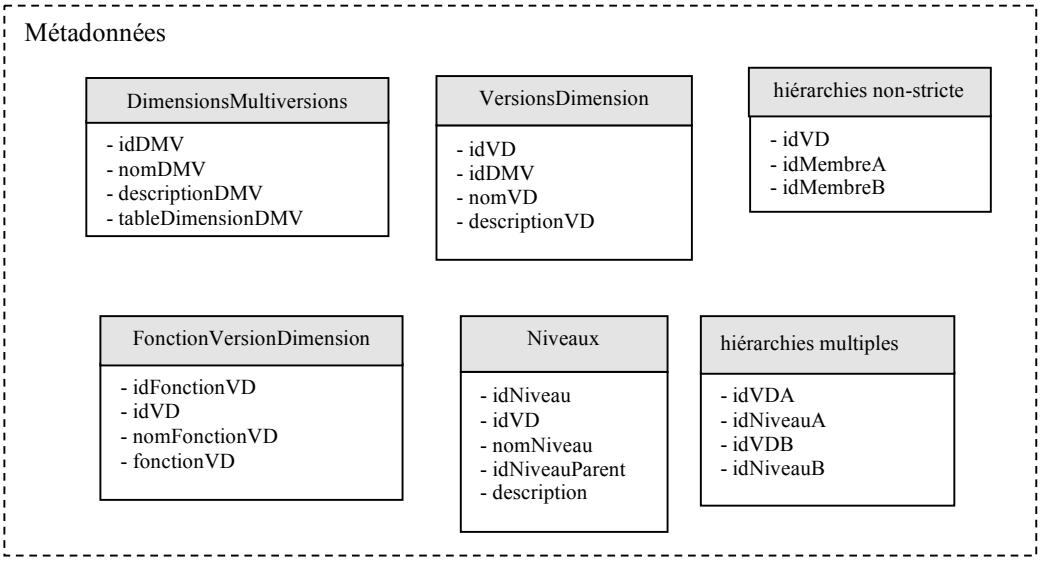
[ZAI98]
Osmar R. Zaïane, Jiawei Han, Ze-Nian Li, Jean Hou, *Mining Multimédia Data*, CASCON'98: Meeting of Minds, pp 83-96, Toronto, Canada, November 1998.

[ZAI99]
Osmar Rachid Zaïane, *Resource and knowledge discovery from the internet and multimedia repositories*, Ph.D., Simon Fraser University, 1999.

[ZHA01]
H. Zhang and al., *Developing a digital mammography data warehouse*, Medical Imaging 2001.

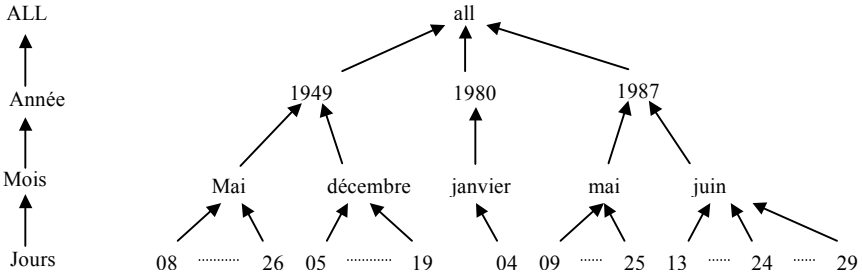
Annexe 1 : schéma de l'entrepôt de données multimédia du prototype





Annexe 2 : schéma et instances des dimensions du prototype

- DMV_1 : Age
  - VD_1 : dateNaissance

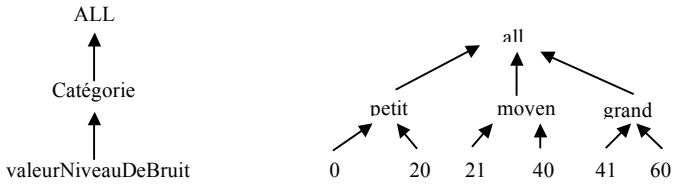


  - VD_2 : âgeParClasse (défini dans le modèle conceptuel)
  - VD_5 : âgeParTranche (défini dans le modèle conceptuel)

- DMV2 : Date (défini dans la partie 3.1.2 comme version de dimension organisée en hiérarchie multiple)

- DMV3 : QT
  - VD1 : QTalgo1 (défini dans la partie 3.1.2 comme version de dimension organisée en hiérarchie non-couvrante)
  - VD2 : QTalgo2 (défini dans le modèle conceptuel)

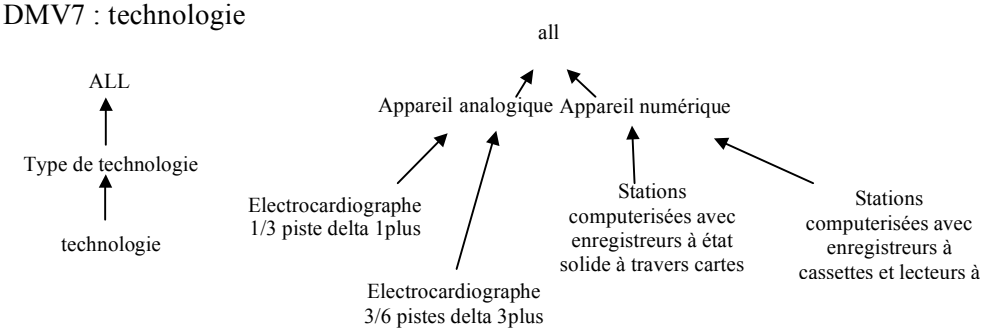
- DMV4 : Niveau de bruit (en microvolt)



- DMV5 : Pathologie principale (défini dans la partie 3.1.2 comme version de dimension organisée en hiérarchie non-onto)

- DMV6 : tranche horaire d'acquisition (défini dans la partie 3.1.2 comme version de dimension organisée en hiérarchie non-stricte)

- DMV7 : technologie



- DMV8 : sexe