Intégration de versions fonctionnelles dans les entrepôts de données multimédias

Anne-Muriel Arigon*,**, Maryvonne Miquel*, Anne Tchounikine*

*LIRIS (Laboratoire d'InfoRmatique en Images et Systèmes d'information)
UMR CNRS 5205
Bâtiment Blaise Pascal, INSA, 7 avenue Capelle, 69621 Villeurbanne Cedex, France
maryvonne.miquel@insa-lyon.fr, anne.tchounikine@insa-lyon.fr

**LBBE (Laboratoire de Biométrie et Biologie Evolutive) UMR CNRS 5558
Université Claude Bernard – Lyon 1, 43 bd. du 11 novembre 1918
69622 Villeurbanne Cedex, France
arigon@biomserv.univ-lyon1.fr

Résumé. Les modèles multidimensionnels ont une structure statique où les membres des dimensions sont calculés d'une manière unique. Cependant les données (particulièrement les données multimédias) sont souvent caractérisées par des descripteurs pouvant être obtenus par divers modes de calcul que nous définissons comme des "versions fonctionnelles" de descripteurs. Nous proposons un modèle multidimensionnel multiversion fonctionnelle ("modèle M2F") en intégrant notamment la notion de "version de dimension" qui représente des dimensions dont les membres sont calculés selon différentes versions fonctionnelles. Cette nouvelle approche permet d'intégrer au modèle un choix de modes de calcul de ces descripteurs afin de permettre à l'utilisateur de choisir la représentation de données la plus adaptée. 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 formellement un modèle conceptuel et présentons le prototype réalisé pour cette étude.

1. Introduction

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" [Inmon, 1996]. L'objectif est d'extraire les données pertinentes pour les prises de décision à partir de bases de données de production et de les organiser suivant un modèle adapté. Les analyses décisionnelles sont basées sur la technologie OLAP (On-Line Analytical Processing) comme définies dans [Chaudhuri et Dayal, 1997] [Vassiliadis et Sellis, 1999]. Dans cette approche, l'information est organisée autour de sujets majeurs et est modélisée afin de permettre un pré-calcul et un accès facile et rapide aux données agrégées. Les traitements OLAP font référence à des fonctionnalités d'analyse utilisées pour explorer les données. Selon Kimball [Kimball, 1996], le paradigme de la modélisation de données pour un entrepôt de données doit répondre à des besoins qui sont très différents de ceux des modèles de données des environnements OLTP (On-Line Transactional Processing). Les modèles 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 [Cabibbo et Torlone, 1998]. Dans ces modèles, le sujet analysé, appelé mesure ou fait, est représenté dans l'espace des axes d'analyse nommés dimensions. 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 [Agrawal et al., 1995]. Une dimension est représentée par un schéma qui définit différents niveaux de granularité reliés par des liens hiérarchiques. 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. Les données agrégées sont les faits calculés en utilisant des fonctions d'agrégation (les plus courantes étant count, sum, min, max, avg) suivant des granularités différentes. Les agrégations de données sont calculées à partir de la table de fait et des liens entre les membres.

Les modèles multidimensionnels [Cabibbo et Torlone, 1998] [Kimball, 1996] [Lehner, 1998] considèrent habituellement les faits comme la partie dynamique de l'entrepôt de données, et les dimensions comme entités statiques [Mendelzon et Vaisman, 2000]. Dans de tels entrepôts de données, les membres des dimensions sont calculés d'une manière unique. Or dans certains cas, cela peut limiter l'analyse des données, en particulier dans le cas de données multimédias. Les données multimédias sont des données particulièrement volumineuses, de différents formats (textes, graphiques, vidéos, sons,…) et généralement stockées en séquences de bits de longueurs différentes. Elles sont souvent décrites par différents descripteurs et stockées de manière à améliorer leur indexation et leur exploitation. Cela nécessite donc un traitement spécifique et des outils de visualisation adaptés ainsi qu'une redéfinition de fonctions d'agrégat portant non plus sur des données alphanumériques mais sur des faits multimédias. Dans les entrepôts de données multimédias, les descripteurs extraits de ces données constituent les dimensions du modèle multidimensionnel. Ils caractérisent la donnée et permettent de l'analyser. 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) [Han et Kamber, 2001] [Zaïane, 1999]. Par exemple, pour des données images, les descripteurs textuels peuvent être des mots-clé, la résolution et les descripteurs de contenu peuvent être la couleur, la texture. Dans l'exemple des vidéos, les descripteurs textuels peuvent être la date, le réalisateur, et les descripteurs de contenu portent sur le son, la qualité.

Ces descripteurs sont très diversifiés et un même descripteur peut être extrait de la donnée multimédia de diverses manières. En effet, 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 de diverses façons. Dans ce cas, les données des dimensions ne devraient pas représenter une information prédéfinie mais doivent intégrer de multiples méthodes de calcul afin de représenter les données selon les diverses versions fonctionnelles de chaque descripteur. Par conséquent, il nous semble que l'intégration des versions fonctionnelles de dimension (correspondant aux descripteurs de données calculés par différentes fonctions) dans le modèle peut améliorer la caractérisation et l'analyse des

données, l'objectif étant de permettre à l'utilisateur de choisir les modes de calcul pour chaque descripteur afin de définir la meilleure représentation des 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.

Le reste du papier est organisé comme suit. La partie 2 fournit un état de l'art dans ce domaine de recherche. La partie 3 présente l'étude de cas. La partie 4 propose le principe général de notre approche et présente le modèle conceptuel proposé. Dans la partie 5, nous présentons le prototype développé en décrivant son architecture, une manière d'adapter notre modèle au niveau logique et physique, et son implémentation sur l'étude de cas. Enfin, nous concluons dans la partie 6.

2. Etat de l'art

Quelques travaux de construction de cubes de données multimédias ont été menés pour faciliter l'analyse multidimensionnelle de larges bases de données multimédias. Dans [You et al., 2001], 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. Un autre exemple est celui du système d'analyse de données multimédias MultiMediaMiner qui utilise un cube de données multimédias [Zaïane et al., 1998] permettant de stocker les données multidimensionnelles et de les agréger à des niveaux de granularité différents. Dans le domaine médical, les problèmes d'exploitation et d'analyse de données volumineuses sont omniprésents car 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 diagnostic [Zhang et al., 2001]. Une autre étude [Tikekar et Fotouhi, 1995] 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. 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, seuls 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.

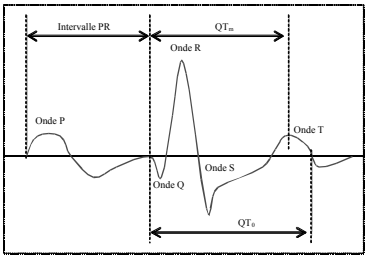
Les aspects dynamiques dans les modèles multidimensionnels ont été abordés dans les études traitant des évolutions des structures d'analyse. Dans ces modèles multidimensionnels, les dimensions et leurs attributs évoluent à travers le temps en même temps que la structure multidimensionnelle. Afin de prendre en compte les évolutions des structures d'analyse, deux types d'approches existent : la première consiste à mettre à jour les modèles [Blaschka et al., 1999] [Blaschka, 1999] [Hurtado et al., 1999a] [Hurtado et al., 1999b], ce qui revient à

transformer l'entrepôt de données de manière à répercuter les évolutions de la structure multidimensionnelle en produisant un modèle unique correspondant souvent au schéma le plus récent; la seconde est basée sur des modèles prenant en compte l'historique des évolutions [Eder et Koncilia, 2001] [Mendelzon et Vaisman, 2000] [Pedersen et al., 2001], ce qui permet de garder une trace des évolutions du système. Cette dernière approche est particulièrement intéressante 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. Les modèles de Mendelzon et Vaisman [Mendelzon et Vaisman, 2000], de Pedersen et al. [Pedersen et al., 2001] et de Eder et Koncilia [Eder et Koncilia, 2001] considèrent les besoins de l'utilisateur qui sont de garder l'historique des évolutions et de pouvoir comparer les données. Ces modèles permettent de transformer les données en une "structure invariante" choisie par l'utilisateur. [Eder et Koncilia, 2001] propose de conserver les liens lors des évolutions et leurs répercutions sur les faits étudiés en introduisant la notion de fonctions de "mapping", [Pedersen et al., 2001] propose un modèle conceptuel qui porte sur l'imprécision et les structures de dimension complexes et [Mendelzon et Vaisman, 2000] définit un modèle multidimensionnel temporel dans lequel tout élément est caractérisé par un temps valide et qui est enrichi par le langage de requête TOLAP. [Body et al., 2002] [Body et al., 2003] intègre la notion de version temporelle dans les dimensions et rassemblent les données dans une table de fait appelée table de fait multiversion, selon différents modes temporels de présentation (MTP). Ce modèle permet donc de prendre en compte les évolutions temporelles dans les entrepôts. Cependant ce modèle autorise une navigation assez limitée puisque l'utilisateur choisit uniquement la version temporelle. Dans le cas des données multimédias et des versions fonctionnelles, cette solution est trop restrictive puisque l'utilisateur peut être amené à choisir, pour chaque descripteur (c'est-à-dire chaque dimension), la version fonctionnelle qu'il souhaite.

3. L'étude de cas

Le travail présenté 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. Dans le domaine cardiologique, l'électrocardiogramme (ECG) est une donnée essentielle pour le suivi des patients et le diagnostic. Un ECG est un signal enregistré sur trois voies (dimensions de l'espace X,Y,Z). Nous proposons d'intégrer ces données multimédias et les informations patients associées au sein d'un entrepôt de données. Cet exemple permet de souligner les difficultés et les besoins liés à ce type de données.

L'équipe ERM107 a notamment travaillé sur les données d'un essai thérapeutique nommé étude EMIAT (European Myocardial Infarct Amiodarone Trial). Cette 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. Elle fournit comme résultat un nombre important de données à exploiter et à analyser dont des données multimédias assez volumineuses. Ces données multimédias sont les 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. Parmi les différentes mesures qui sont effectuées sur un ECG, les plus étudiés sont l'intervalle QT (temps nécessaire pour que le ventricule soit à nouveau repolarisé) et le niveau de bruit (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.... Ainsi, deux types de descripteurs caractérisent les ECGs : les descripteurs textuels (pathologie principale, âge, sexe, date et tranche horaire d'acquisition de l'ECG, technologie avec laquelle l'ECG est obtenu) et les descripteurs de contenu (la durée du QT, le niveau de bruit de l'ECG). Les faits sont des ECGs caractérisés par des descripteurs (correspondant aux dimensions) organisés en hiérarchies complexes. Par exemple, les tranches horaires peuvent être classées en heures puis en périodes (nuit, réveil, jour). Certaines heures, (par exemple 6h) peuvent appartenir à plusieurs périodes (réveil et nuit). La dimension Temps correspondant à ce descripteur est donc organisée en hiérarchie non-stricte. Un modèle multidimensionnel simplifié de l'entrepôt de données montre la table de fait (signal ECG) et les dimensions que nous utilisons (Temps, Durée du QT, Niveau de bruit, Sexe, Age, Pathologie principale, Technologie, Date) (Figure 1). Cet entrepôt de données sera utilisé afin d'analyser les ECGs d'une population ayant certaines caractéristiques sur l'age, le sexe, la pathologie…. Souvent, ces descripteurs peuvent être calculés par divers modes de calcul. Par exemple, la durée du QT peut être obtenue grâce à plusieurs algorithmes. Il est donc nécessaire pour l'utilisateur de choisir le mode de calcul approprié (ou la version fonctionnelle) de chaque descripteur afin d'avoir la représentation ou la vue des données la plus adaptée à ses besoins.

Nous proposons un modèle intégrant les versions de dimension que nous définissons comme des dimensions dont les membres sont calculés selon les différentes versions fonctionnelles de descripteurs. Ce modèle devra supporter les hiérarchies explicites et complexes.

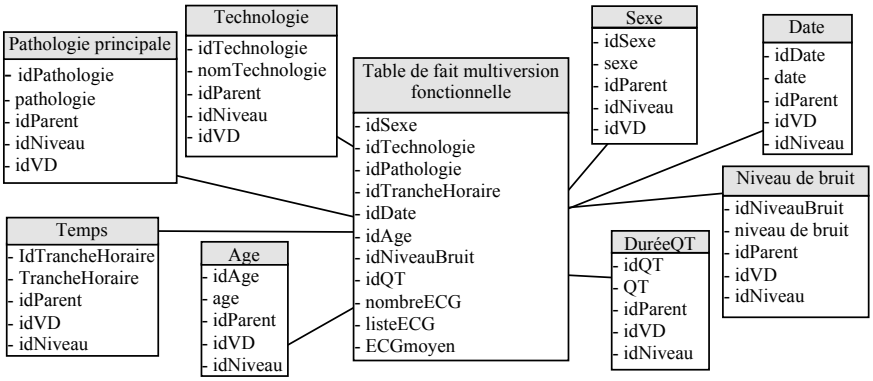


FIG. 1 - Schéma de l'entrepôt de données

4. Approche conceptuelle

4.1 Principe général de notre approche M2F

Notre modèle présente une table de fait regroupant l'ensemble des mesures qui représentent les données à analyser (les données multimédias ou des pointeurs vers celles-ci) et des dimensions qui constituent les axes d'analyse (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, de dimension multiversion, de table de fait multiversion fonctionnelle et de 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 de ces dimensions en décrivant l'ensemble des membres et des filiations. Notre approche permet donc d'avoir des dimensions explicites, les schémas de dimension sont définis explicitement et notre modèle supporte également les hiérarchies complexes (hiérarchie multiple, non-onto, non-stricte, non-couvrante) puisque les instances des dimensions sont construites à partir des membres et des liens hiérarchiques.

4.2 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 $<\boldsymbol{idVD,\mathcal{H},<_{idVD}}>$ 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 $<$***idNiveau$_j$, nomNiveau$_j$, [$\mathcal{A}_j$], [description$_j$]***$>$ où :
  - ***idNiveau$_j$*** est l'identifiant du niveau de version de dimension
  - ***nomNiveau$_j$*** est le nom du niveau de version de dimension
  - $\mathcal{A}_j$ est une propriété optionnelle qui représente l'ensemble des attributs descriptifs de ce niveau
  - ***description$_j$*** est une propriété optionnelle qui permet d'introduire des informations textuelles sur le niveau n_j
- $<_{idVD}$ 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 $<_{idVD}$ est défini tel que : $\forall (n_1,n_2) \in \mathcal{N}\times\mathcal{N}$, si $n_1 <_{idVD} n_2$ 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 les relations selon $<_{idVD}$ 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 ECGs 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.
Soit le schéma $S_{âgeParTranche}$ de la version de dimension "*âgeParTranche*" dont *idVD = 1*. Le schéma de cette version de dimension est défini par :
$S_{âgeParTranche} = <1,\{n_1, n_2, n_3\}, <_1>$ avec
$n_1 = <1,$ "*TrancheDe5*"$>$, $n_2 = <2,$ "*TrancheDe10*"$>$, $n_3 = <3,$ "*TrancheDe50*"$>$
et l'ordonnancement suivant : $n_1 <_1 n_2$, $n_2 <_1 n_3$ *et* $n_3 <_1$ *ALL*
Le schéma peut être représenté par le graphe orienté de la figure 2.

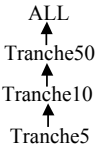


FIG. 2 - Schéma de la version de dimension "âgeParTranche"

On peut également regrouper ces âges en classes d'âge (jeune enfant, enfant, adolescent, jeune adulte, adulte, senior) puis en catégories (mineur, majeur). On peut alors définir le schéma $S_{âgeParClasse}$ de la version de dimension "*âgeParClasse*" dont *idVD = 2*. Le schéma de cette version de dimension est défini par :
$S_{âgeParClasse} = <2,\{n_4, n_5\}, <_2>$ avec
$n_4 = <4,$ "*ClassesAge*"$>$, $n_5 = <5,$ "*Catégories*"$>$
et l'ordonnancement suivant : $n_4 <_2 n_5$ *et* $n_5 <_2$ *ALL*.

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.
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 :
$S_{QTalgo1} = <3,\{n'_1, n'_2, n'_3\},<_3>$ avec
$n'_1 = <1,$ " ValeurQTalgo1">, $n'_2 = <2,$ " TrancheDe100Algo1">,
$n'_3 = <3,$ "TrancheDe400Algo1">
et l'ordonnancement suivant : $n'_1 <_3 n'_2$, $n'_2 <_3 n'_3$ et $n'_3 <_3 ALL$.
De la même manière, on définit le schéma $S_{Qtalgo2}$ 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 $S_{VD} = <idVD, \mathcal{N}, <_{idVD}>$ est définie par le tuple $<$***idVD,nomVD,*** $\mathcal{M}$ $<_{VD}$ ***,[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 $<$***id_j, val_j, [σ_j], $idNiveau_j$***$>$ 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
  - σ_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.
- $<_{VD}$ 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 $n_1 <_{idVD} n_2$, 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 $m_1 <_{VD} m_2$. 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 les relations selon $<_{VD}$ 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 : $\mathcal{MF}_{VD} = \{m_j / m_j \in \mathcal{M} \text{ et } \neg\exists\, m_i \in \mathcal{M} \text{ tel que } (i \neq j \text{ et } m_i <_{VD} m_j)\}$

Exemple 2.
La version de dimension "*âgeParTranche*" dont le schéma $S_{âgeParTranche}$ est présenté dans l'exemple précédent est définie par :
âgeParTranche = <1, "*âgeParTranche*",{m_1,.....,m_7}, $<_{âgeParTranche}$ > avec
m_1 = <1,"0-5" ,1>, m_2 = <2,"6-10",1>, m_3 = <3,"11-15",1>, m_4 = <4,"16-20",1>,
m_5 = <5,"0-10",2>, m_6 = <6,"11-20",2>, m_7 = <7,"0-50",3>
et l'ordonnancement suivant : $m_1 <_{âgeParTranche} m_5$, $m_2 <_{âgeParTranche} m_5$, $m_3 <_{âgeParTranche} m_6$, $m_4 <_{âgeParTranche} m_6$, $m_5 <_{âgeParTranche} m_7$, $m_6 <_{âgeParTranche} m_7$ *et* $m_7 <_{âgeParTranche}$ *all.*
Les membres du niveau n_1 ("*TrancheDe5*") sont donc {m_1, m_2, m_3, m_4}, ceux du niveau n_2 ("*TrancheDe10*") sont {m_5, 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 obtient le graphe orienté de la figure 3.

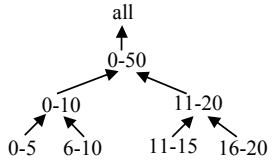


FIG. 3 - Version de dimension "âgeParTranche"

De la même manière, la version de dimension "*âgeParClasse*" dont le schéma $S_{âgeParClasse}$ est présenté dans l'exemple précédent est définie par :
âgeParClasse = < 2, "*âgeParClasse*" , {m_8,.....,m_{15} }, $<_{âgeParClasse}$ > *avec*
m_8 = < 8,"jeune enfant",4 >, m_9 = < 9,"enfant",4 >, m_{10} = < 10,"adolescent",4 >,
m_{11} = < 11,"jeune adulte",4 > ,m_{12} = < 12,"adulte",4 > , m_{13} = < 13,"senior",4 >,
m_{14} = < 14,"mineur",5 > , m_{15} = < 15,"majeur",5 >
et l'ordonnancement suivant : $m_8 <_{âgeParClasse} m_{14}$, $m_9 <_{âgeParClasse} m_{14}$, $m_{10} <_{âgeParClasse} m_{14}$, $m_{11} <_{âgeParClasse} m_{15}$, $m_{12} <_{âgeParClasse} m_{15}$, $m_{13} <_{âgeParClasse} m_{15}$, $m_{14} <_{âgeParClasse}$ *all et* $m_{15} <_{âgeParClasse}$ *all*
Les membres du niveau n_4 ("*ClassesAge*") sont donc { m_8, m_9, m_{10}, m_{11}, m_{12} , m_{13}} et ceux du niveau n_5 ("*Catégories*") sont { m_{14}, m_{15} }.
L'ensemble $\mathcal{MF}_{âgeParClasse}$, est défini par: $\mathcal{MF}_{âgeParClasse}$, = { m_8, m_9, m_{10}, m_{11}, m_{12} , m_{13} }
On peut également définir les versions de dimension "*QTalgo1*" et "*QTalgo2*" dont les schémas sont $S_{QTalgo1}$, $S_{Qtalgo2}$ ainsi que les ensembles $\mathcal{MF}_{QTalgo1}$ et $\mathcal{MF}_{Qtalgo2}$ définis par :
$\mathcal{MF}_{QTalgo1}$ = {209,258,403,510,522,559,709} et $\mathcal{MF}_{Qtalgo2}$ = {205,230,395,475,512,685,750}.
Pour la version de dimension "*QTalgo1*" par exemple, on obtient le graphe orienté de la figure 4.

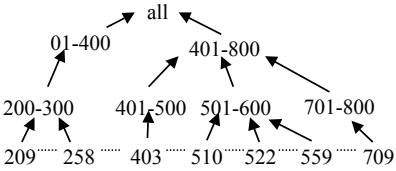


FIG. 4 - Version de dimension " QTalgo1"

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,VD,[descriptionDMV]***> où :

- ***idDMV*** est l'identifiant unique pour la dimension multiversion
- ***nomDMV*** est le nom de la dimension multiversion
- ***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", {" âgeParTranche ", " âgeParClasse "}>
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,"DuréeQT", {"QTalgo1","QTalgo2"}>
On définit de la même manière l'ensemble $\mathcal{MF}_{DuréeQT}$ par :
$\mathcal{MF}_{DuréeQT}$ = {205,209,258,230,395,403,475,510,512,522,559,685,709,750}

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 :

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

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.

Exemple 4 :
Supposons que les faits soient les "*ECGs*" des patients et que l'on ait une mesure du fait (μ_1) qui représente le nombre d'ECG ("*nombreECG*"). Les différentes dimensions multiversions sont "*Age"* et "*DuréeQT* " et leur versions de dimension respectives sont "*âgeParTranche*", "*âgeParClasse" pour "Age" et "QTalgo1*", "*QTalgo2" pour "DuréeQT* ". La table de fait multiversion fonctionnelle "*tf*" correspondante est définie par une fonction telle que :

$$tf : Age \times DuréeQT \rightarrow dom(nombreECG)$$
$$m_{Age}, m_{DuréeQT} \mapsto v_{nombreECG}$$

avec $m_{Age} \in \mathcal{MF}_{Age}$, $m_{DuréeQT} \in \mathcal{MF}_{DuréeQT}$, $v_{nombreECG}$ la valeur correspondante de la mesure μ_1 "*nombreECG*".
Ainsi pour le membre $m_{Age} = m_4$ = "*16-20*" de la version de dimension "*âgeParTranche"* de la dimension multiversion "*Age"* et le membre $m_{DuréeQT}$ ="*209*" de la version de dimension "*QTalgo1*" de la dimension multiversion "*DuréeQT* ", on aura $v_{nombreECG}$ = "*4*" qui correspond au nombre d'ECG dont la valeur du QT est de "*209"* et pour lesquels les patients ont entre *16* et *20* ans.
De la meme manière, pour le membre $m_{Age} = m_{11}$ = "*jeune adulte*" de la version de dimension " *âgeParClasse*" de la dimension multiversion "*Age"* et le membre $m_{DuréeQT}$ ="*209*" de la version de dimension "*QTalgo1*" de la dimension multiversion "*DuréeQT* ", on aura $v_{nombreECG}$ = "*6*" qui correspond au nombre d'ECG dont la valeur du QT est de "*209"* et pour lesquels les patients sont dans la catégorie "*jeune adulte*". (Ce nombre d'ECG est différent de celui obtenu précédemment car la catégorie "*jeune adulte*" ne couvre pas le même domaine de valeur que la classe "*16-20").*

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 f_{VD} est définie par le tuple <$\boldsymbol{idFonction_{VD}}$,***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
- $\boldsymbol{nomFonction_{VD}}$ 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 :

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

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 5 :
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"* et $f_{âgeParTranche}$ *(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) = 205* et $f_{Qtalgo2}$ *(ECG5) = 209*

Définition 6 (structure multidimensionnelle multiversion fonctionnelle). Une structure multidimensionnelle multiversion fonctionnelle ***M2F*** est définie par le tuple **<*DMV*, *tf*, *F*>** où :

- $DMV = \bigcup_{i=1}^{s} DMV_i$ est l'ensemble des dimensions multiversions
- ***tf*** est la table de fait multiversion fonctionnelle
- $F = \bigcup_{j=1}^{r} f_{VD_j}$ 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* :

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

La fonction d'agrégation peut être une fonction classique de la technologie OLAP (sum, count, min, max, avg) pour des mesures numériques ou une fonction plus spécifique (moyenne statistique, enveloppe…) pour des mesures de type image ou signal.

5. Prototype

5.1 Architecture

Pour implémenter notre modèle, nous avons choisi 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 et l'outil Analysis Services de SQL Server pour construire les hypercubes. Nous adoptons une architecture 3-tiers composée des trois parties suivantes :

- un entrepôt de données multimédias multiversion fonctionnelle dans lequel les dimensions multiversions et la table de fait multiversion fonctionnelle ont été intégrées,
- un cube OLAP construit à partir de l'entrepôt de données multimédias multiversion fonctionnelle en utilisant des agrégations et qui permet de faire des requêtes intégrant les versions fonctionnelles des dimensions,
- des outils client OLAP pour visualiser les données.

A partir de la base de données de production, les données sont transférées dans l'entrepôt de données multimédias multiversion fonctionnelle en utilisant les fonctions de versions de dimension.

5.2 Implémentation

Dans un premier temps, nous définissons ce que sont les dimensions multiversion et les metadonnées au niveau du modèle logique (Figure 5). 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. Ensuite, les métadonnées regroupant toutes les informations à propos du cube multidimensionnel ne sont pas intégrées au cube directement mais placées dans des tables relationnelles implémentées séparément. Les informations fournies par ces métadonnées permettent de visualiser le modèle au niveau logique et facilitent le chargement de l'entrepôt.

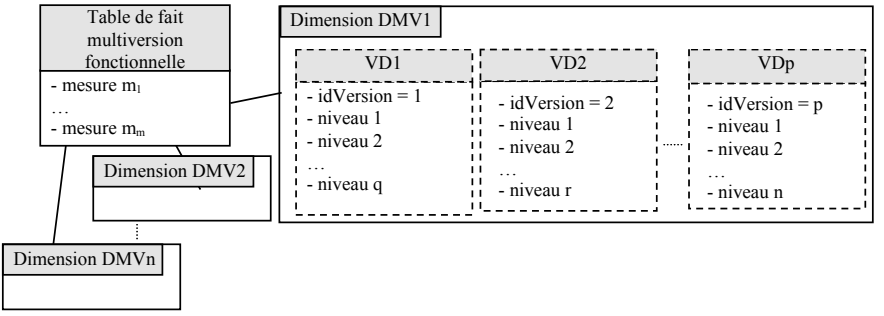


FIG. 5 - Schéma du modèle logique

Nous présentons ensuite, au niveau du modèle physique, la structure multidimensionnelle multiversion fonctionnelle de l'entrepôt en décrivant la table de fait multiversion fonctionnelle et les dimensions multiversions, ainsi que les métadonnées.

La table de fait multiversion fonctionnelle est constituée des valeurs du fait (seuls les liens sur les faits multimédias sont stockés dans la table de fait multiversion fonctionnelle) 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.

Une dimension multiversion regroupe dans une même table les membres de toutes les versions de dimension (pouvant avoir une hiérarchie complexe) 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. Parmi les représentations existantes, nous avons choisi d'utiliser la représentation "parent-enfant" qui s'adapte très bien à notre approche. 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.

Nous introduisons, 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, ce qui permet au modèle de supporter toutes les hiérarchies complexes [Kimball, 2000] [Mendelzon et Vaisman, 2000] [Pedersen et al., 2001]. Pour la prise en compte des hiérarchies multiples, nous transformons la version de dimension en deux versions de dimension de la même dimension multiversion, chacune ayant un des niveaux qui a entraîné la hiérarchie multiple (les niveaux communs et leurs membres sont dupliqués dans chaque version de dimension). Une table des métadonnées permet de conserver les liens pour recréer la réelle hiérarchie multiple. En raisonnant sur le même principe que les hiérarchies multiples, notre approche permet de traiter les hiérarchies non-strictes (le membre qui a plusieurs parents ainsi que tous ses enfants sont dupliqués en établissant un lien de filiation entre chacun des membres obtenus et l'un ou l'autre des membres du niveau supérieur). Une table des métadonnées permet également de conserver les liens permettant ainsi de recréer la hiérarchie non-stricte. De plus, 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. La notion de "parent-enfant" est également utilisée pour les schémas des versions de dimensions construits à partir des niveaux et des liens (représentés par des relations parent-enfant) qui existent entre ces niveaux.

Enfin, les métadonnées sont stockées dans six tables relationnelles. Elles permettent d'avoir des informations sur les dimensions multiversions, les versions de dimension, les niveaux de hiérarchies des versions de dimension qui sont stockés en tant qu'attributs dans les tables des dimensions du cube lui-même, les fonctions de calcul des membres des versions de dimension et les hiérarchies complexes.

5.3 Mise en œuvre sur l'étude EMIAT

En utilisant les données de l'étude EMIAT présentée en début de la partie 2 , nous avons développé un prototype basé sur 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.

L'entrepôt de données est composé d'une table de fait multiversion fonctionnelle et de huit dimensions multiversions. Les faits sont les signaux 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 (pathologie principale, âge, sexe du patient), trois autres sur l'acquisition de l'ECG (date d'acquisition de l'ECG, temps (horaire d'acquisition de l'ECG), technologie utilisée pour effectuer l'ECG) et deux autres sur le contenu de l'ECG (durée du QT, niveau du bruit de l'ECG). 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. Cet 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 implémentées en Visual Basic.

L'application développée permet également 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 (Figure 6) et d'explorer les données agrégées dans un tableau à deux entrées. L'utilisateur peut naviguer en choisissant l'agrégation de données (nombreEcg, listeECG ou ECGmoyen) à analyser et les dimensions multiversions selon lesquelles il veut explorer les données tout en fixant les niveaux des autres dimensions multiversions. Il peut également sélectionner plusieurs dimensions pour une même entrée afin d'explorer les données selon plusieurs critères.

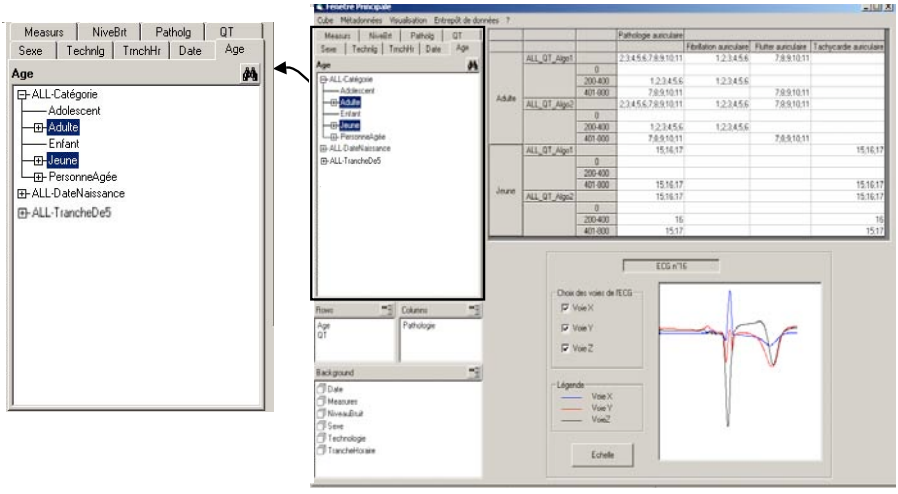


FIG. 6 - Interface de navigation dans un hypercube multimedia

Il est également possible de visualiser plusieurs versions de dimension d'une même dimension multiversion afin d'établir des comparaisons. Puis, les données multimédias peuvent être visualisées en sélectionnant un ECGmoyen, en fournissant un identifiant d'ECG ou en choisissant un ECG dans les données agrégées "listeEcg" obtenues.

De plus, les metadonnées associées peuvent être visualisées pour avoir une vue globale des différentes versions de dimension afin de naviguer plus facilement dans le cube (ex, les schémas et les instances de chaque version de dimension sont représentés pour chaque dimension multiversion). Il est possible de sélectionner la dimension multiversion et la version de dimension que nous souhaitons visualiser. Il est également 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). Enfin, l'utilisateur peut représenter les fonctions de versions de dimension afin d'améliorer l'analyse des résultats obtenus.

6. Discussion et conclusion

Nous avons présenté un nouveau modèle multidimensionnel qui prend en compte les versions fonctionnelles. Il 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. 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. Par ailleurs, à partir de ce modèle conceptuel, nous avons dégagé un modèle logique et physique permettant l'implémentation de notre approche à l'aide des outils existants. Ce modèle a été utilisé pour réaliser une application OLAP permettant de naviguer au sein de données de type signaux. Bien que cette approche multiversion puisse s'appliquer pour tout type de données, notamment alphanumériques, l'exemple d'application montre qu'elle s'adapte particulièrement bien aux données multimédias car celles-ci nécessitent de pouvoir extraire des indicateurs selon différents modes de calcul. 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.

Cependant, le stockage de données de notre modèle pourrait être amélioré. Il est possible d'avoir une certaine redondance dans les schémas de versions de dimension, le stockage de la table de fait multiversion fonctionnel n'est pas optimisé et le traitement des hiérarchies non-strictes et multiples implique des duplications. Notre modèle pourrait être étendu en associant la notion de la version fonctionnelle aux faits, de la même manière que notre modèle associe les versions fonctionnelles aux dimensions. Cela pourrait être possible en ajoutant une dimension "version de fait" permettant ainsi à l'utilisateur de naviguer selon la version du fait.

Références

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

[Blaschka et al., 1999] M. Blaschka, C. Sapia et G. Höfling. On Schema Evolution in Multidimensional Databases. *Proceedings of DaWak'99 Conference*, Florence, Italy, 1999.

[Blaschka, 1999] M. Blaschka. FIESTA: A Framework for Schema Evolution in Multidimensional Information Systems. *Proceedings of 6th Doctoral Consortium*, Germany, 1999.

[Body et al., 2002] M. Body, M. Miquel, Y. Bédard et 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.

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

[Cabibbo et Torlone, 1998] L. Cabibbo et R. Torlone. A Logical Approach to Multidimensional Databases. *Proocedings of the 6th International Conference on Extending Database Technology (EDBT'98)*, Valencia, Spain, 1998.

[Chaudhuri et Dayal, 1997] S. Chaudhuri et U. Dayal. An Overview of Data Warehousing and Olap Technolog. *SIGMOD Record 26(1)*, 1997

[Eder et Koncilia, 2001] J. Eder et C. Koncilia. Evolution of Dimension Data in Temporal Data Warehouses. *Proceedings of the DaWaK'01 Conference*, Munich, Germany, 2001.

[Han et Kamber, 2001] J. Han et M. Kamber. Data mining, concepts and techniques. Morgan Kaufmann Publishers, 2001.

[Hurtado et al., 1999a] C. Hurtado, A.O. Mendelzon et A. Vaisman. Maintaining Data Cubes Under Dimension Updates. *Proceedings of the IEEE/ICDE'99 Conference*, 1999.

[Hurtado et al., 1999b] C. Hurtado, A.O. Mendelzon et A. Vaisman. Updating OLAP Dimensions. *Proceedings of the ACM Second Int. Workshop on Data Warehousing and OLAP*, USA, 1999.

[Inmon, 1996] W.H. Inmon. Building the Data Warehouse, 3rd Edition. Eds.Wiley and Sons, 1996

[Kimball, 1996] R. Kimball. The Data Warehouse Toolkit. J.Wiley and Sons, Inc, 1996.

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

[Lehner, 1998] W. Lehner. Modeling large OLAP scenarios. *Proceedings of the 1998 International Conference on Extending Database Technology*, Valencia, Spain, 1998.

[Mendelzon et Vaisman, 2000] A.O. Mendelzon et A. Vaisman. Temporal Queries in OLAP. *Proceedings of the 26th VLDB'00 Conference*, Cairo, Egypt, 2000.

[Pedersen et al., 2001] T.B. Pedersen, C.S. Jensen et C.E. Dyreson. A foundation for capturing and querying complex multidimensional data. *Information Systems Special Issue:Data Warehousing*, Vol 26, No 5, 2001.

[Tikekar et Fotouhi, 1995] R. V. Tikekar et F. Fotouhi . Storage and retrieval of medical images from data warehouses. *Digital Image Storage and Archiving Systems*, 1995.

[Vassiliadis et Sellis, 1999] P. Vassiliadis et T. Sellis. A Survey of Logical Models for OLAP Databases. *SIGMOD Record* Volume 28, Number 1, March, 1999

[You et al., 2001] J. You, T. Dillon, J.Liu et E. Pissaloux. On hierarchical multimedia information retrieval. *IEEE International Conference on Image Processing (ICIP)*, 2001.

[Zaïane et al., 1998] O. R. Zaïane, J. Han, Z.N. Li et J. Hou. Mining Multimédia Data. *CASCON'98: Meeting of Minds*, pp 83-96, Toronto, Canada, November 1998.

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

[Zhang et al., 2001] H. Zhang, X.H. Cao, S.T.C. Wong, S.L. Lou et E.A. Sickles. Developing a digital mammography data warehouse. *Medical Imaging*, 2001.

Summary

The traditional multidimensional models have a static structure where members of dimensions are computed in a unique way. However, data (particularly multimedia data) is often characterized by descriptors that can be obtained by various computation modes. We define these computation modes as "functional versions" of the descriptors. We propose a Functional Multiversion Multidimensional Model ("F2M model") by integrating the concept of "version of dimension". This concept defines dimensions with members computed according to various functional versions. This new approach integrates a choice of computation modes of these members into the model, in order to allow the user to choose the best representation of data. We implement a multimedia data warehouse in the medical field by integrating the multimedia data of a therapeutic study into a multidimensional model. We formally define a conceptual model and we present a prototype for this study.