Analyse flexible dans les entrepôts de données : quand les contextes s'en mêlent

Yoann PITARCH*, Cécile FAVRE**
Anne LAURENT*, Pascal PONCELET*

*LIRMM - Université Montpellier 2, Montpellier, France
{pitarch,laurent,poncelet}@lirmm.fr
**ERIC, Université Lyon 2, Lyon, France
cecile.favre@univ-lyon2.fr

Résumé. En autorisant l'observation des données à plusieurs niveaux de précision, les hiérarchies occupent une place importante dans les analyses d'entrepôts de données. Malheureusement, les modèles d'entrepôts existants ne considèrent qu'un sous-ensemble restreint des types possibles de hiérarchie. Par exemple, il n'est pas possible de modéliser le fait que le caractère *"faible", "normal" ou "élevé"* de la tension artérielle d'un patient (qui constitue une hiérarchisation de la mesure) dépend de son âge (élément lié à la dimension). Ces hiérarchies, dites *contextuelles*, ont récemment été introduites dans des travaux précédents. Dans cet article, nous proposons la première approche pour les modéliser. Une base experte représentant la connaissance du domaine est créée. Ensuite, un algorithme de réécriture de requêtes est proposé pour permettre une analyse flexible, efficace et adéquate d'un entrepôt possédant de telles hiérarchies. Par exemple, il est désormais possible de répondre à la requête "*Quels patients ont eu une tension faible au cours de la nuit?*" en prenant en compte de manière adéquate les contextes associés au caractère "*faible*" de la mesure tension.

1 Introduction

Les entrepôts de données (Inmon (1996)) permettent de consolider, stocker et organiser ces données à des fins d'analyse. Des faits peuvent alors être analysés à travers des indicateurs (les mesures) selon différents axes d'analyse (les dimensions). En s'appuyant sur des mécanismes d'agrégation, les outils OLAP (On Line Analytical Process)(Agrawal et al. (1997); Chen et al. (1996); Han (1997)) permettent de naviguer aisément le long des hiérarchies des dimensions. La puissance de ces outils place les entrepôts au centre des systèmes d'information décisionnels (Mallach (2000)). Ces considérations justifient l'émergence d'entrepôts dans des domaines aussi variés que l'analyse de ventes, la surveillance de matériel, le suivi de données médicales (Einbinder et al. (2001))... Dans cet article, nous considérons cette dernière application afin d'exhiber un manque d'expressivité dans les solutions actuelles et ainsi illustrer l'intérêt de notre proposition.

Considérons un entrepôt de données médicales enregistrant les paramètres vitaux (*e.g.,* le pouls, la tension artérielle...) des patients d'un service de réanimation. Afin de réaliser un suivi efficace des patients, un médecin souhaiterait par exemple connaître ceux qui ont eu une tension artérielle basse au cours de la nuit. Pouvoir formuler ce type de requête suppose l'existence d'une hiérarchie sur la tension artérielle dont le premier lien d'agrégation serait une catégorisation de la tension artérielle (*e.g.,* basse, normale, élevée). Toutefois, cette catégorisation est délicate car elle dépend à la fois de la tension artérielle mesurée mais aussi de certaines caractéristiques physiologiques (âge du patient, fumeur ou non, ...). Dès lors, une même tension peut être généralisée différemment selon le contexte d'analyse considéré. Par exemple, 13 est une tension *élevée* chez un *nourrisson* alors qu'il s'agit d'une tension *normale* chez un *adulte*. Introduites formellement par Pitarch et al. (2009), ces hiérarchies dites *contextuelles* ne sont implantées dans aucun modèle d'entrepôt de données actuel.

En effet, la plupart des solutions logicielles existantes pour construire un entrepôt de données souffre de deux faiblesses majeures quant à leur gestion des hiérarchies :

- Une hiérarchie ne peut être définie que sur un attribut associé à une dimension d'analyse. Si l'on considère l'exemple précédent, il n'est dont pas possible de définir une hiérarchie sur la tension artérielle car cet attribut est une mesure ;
- Les hiérarchies sont considérées implicitement indépendantes (orthogonales) (Eder et Koncilia (2001)). Dès lors, l'agrégation d'une valeur n'est fonction que d'elle-même. Il n'est donc pas possible de modéliser le fait qu'une caractéristique externe puisse influencer le lien d'agrégation d'une valeur. Cette orthogonalité des dimensions par rapport à la dimension temporelle interdit également toute historisation des dimensions dans la plupart des cas et condamne ces hiérarchies à rester statiques.

Dans cet article, nous abordons la problématique de l'implémentation de ces hiérarchies *contextuelles* dans un entrepôt de données en nous posant les deux questions suivantes : (1) comment stocker efficacement cette connaissance et (2) comment l'exploiter en vue d'accroître les possibilités d'analyse offertes au décideur. En effet, dans le domaine du décisionnel, on parle généralement de contexte d'analyse pour désigner le cadre multidimensionnel qui constitue le cadre d'analyse des faits. Autrement dit, par ce terme de contexte, il est sous-entendu que les faits (à travers les valeurs que prennent les mesures) sont fonction des valeurs prises par les dimensions. Or, dans la réalité des données, il s'avère que finalement les mesures elle-mêmes peuvent être hiérarchisées et que les valeurs des attributs caractérisant cette hiérarchisation de mesure peuvent elles-mêmes dépendre d'un certain contexte.

Afin de répondre à cet objectif, nous proposons de modéliser la connaissance experte du domaine d'application afin de représenter ces hiérarchies contextuelles. Notons que nous parlons de hiérarchie mais nous nous intéresserons ici à la création en particulier du premier niveau de cette hiérarchie (généralisation de la mesure), ceci pouvant s'étendre à plusieurs niveaux. La création d'une base de connaissances externe permet la définition des différents liens d'agrégation en fonction des différents contextes. Cette base de connaissances est composée de deux tables. Une méta-table des connaissances permet de modéliser la structure des différents contextes existants dans l'entrepôt. Par exemple, nous y représentons le fait que la normalité de la tension artérielle d'un patient est fonction de la tension mesurée, de la catégorie d'âge du patient et du fait que le patient soit ou ne soit pas fumeur. Ensuite, la table des connaissances

permet de modéliser les différentes instances des contextes. Par exemple, c'est dans cette table que sera exprimé le fait qu'une tension à 17 chez un adulte fumeur est élevée. Nous décrivons ensuite une méthode pour exploiter cette base externe en permettant la création d'un cube pour l'analyse des mesures généralisées.

La suite de l'article est organisée de la façon suivante. La section 2 expose un cas d'étude sur des données médicales qui permettra d'illustrer le problème posé et le modèle proposé. Nous discutons dans la section 3 de l'inadéquation des différentes solutions existantes par rapport à la problématique de cet article. La section 4 présente justement notre solution pour répondre à cette problématique à travers une formalisation. Dans la section 5, nous montrons comment mettre en œuvre cette approche en nous appuyant sur le cas d'étude précédemment présenté. Enfin, nous concluons et indiquons les perspectives à ce travail dans la section 6.

2 Cas d'étude

Pour illustrer la problématique liée à notre approche, nous considérons le cas d'un entrepôt de données enregistrant, pour chaque patient d'un service de réanimation, sa tension ainsi que les médicaments prescrits au fil du temps. Ces valeurs sont mesurées par des capteurs et alimentent directement l'entrepôt. Dès lors, le volume des données stockées au sein de l'entrepôt est potentiellement immense. Cette considération justifie de modéliser l'entrepôt sans normaliser les dimensions (principe du modèle en étoile mais avec plusieurs table de faits) afin de minimiser le nombre de jointures à effectuer lors de son interrogation. Le schéma décrivant cet entrepôt est présenté dans la figure 1. Notons qu'une hiérarchie implicite sur l'âge existe dans la table `PATIENT`.

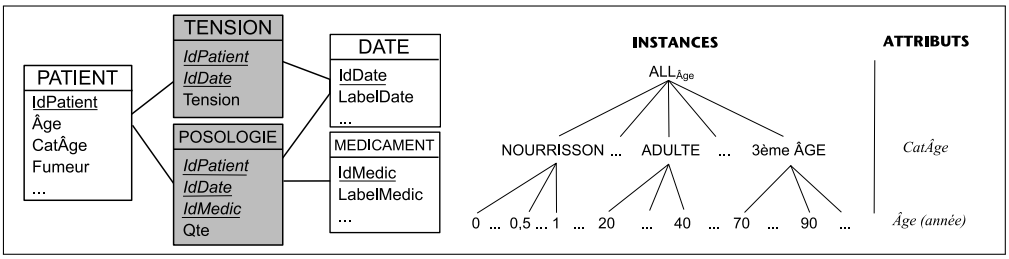


FIG. 1 – *Schéma simplifié de l'entrepôt de données pour l'analyse de la tension et de la posologie ; extrait de la hiérarchie associée à l'attribut* Âge *des patients.*

En pratique, un médecin qui consulte un tel entrepôt peut trouver que les informations qui y sont stockées sont insuffisantes pour assurer un suivi efficace des patients du service. En effet, offrir la possibilité de formuler des requêtes telles que "*Quels sont les patients dont la tension artérielle a été élevée pendant la nuit ?*" ou "*Quels sont les patients qui se sont vus prescrire une quantité trop importante de médicament X ?*" faciliterait le travail des médecins en leur évitant une analyse manuelle des tables de faits.

Malheureusement, le modèle présenté sur la figure 1 ne permet pas de répondre à de telles requêtes pour deux raisons. Premièrement, la notion de tension (resp. posologie) élevée peut

être considérée comme une généralisation de la tension mesurée (resp. de la quantité prescrite). Dans la mesure où les modèles classiques ne permettent pas d'établir une hiérarchie sur les mesures, ces requêtes ne peuvent être formulées. De plus, même si l'on suppose que de telles requêtes sont formulables, les notions de *tension élevée* ou de *posologie élevée* sont directement liées à certaines caractéristiques des patients et/ou des médicaments prescrits. Par exemple, un bébé ne doit pas recevoir la même quantité d'un médicament qu'un adulte. Ainsi, une même posologie pourra être considérée comme *faible*, *normale* ou *élevée* selon l'âge du patient considéré. Une connaissance experte est alors nécessaire pour (1) définir quels sont les attributs qui impactent sur la généralisation d'une mesure et (2) décrire cette généralisation en fonction des valeurs prises par ces attributs. Dans la suite de cet article, nous nous focalisons sur la catégorisation d'une tension pour illustrer l'approche proposée. Le tableau 1 présente quelques exemples de connaissances expertes sur la catégorisation d'une tension en fonction des attributs *CatÂge* et *Fumeur* d'un patient. Par exemple, une tension à 13 est normale chez un adulte fumeur mais est élevée chez un nourrisson.

CatÂge	Fumeur	Tension	CatTension
Nourrisson	Oui ou Non	>12	Élevée
Adulte	Oui	>14	Élevée
$3^{ème}$ âge	Oui ou Non	> 16	Élevée
Nourrisson	Oui ou Non	Entre 10 (inclus) et 12 (inclus)	Normale
Adulte	Oui	Entre 12 (inclus) et 14 (inclus)	Normale
...	...	...	...

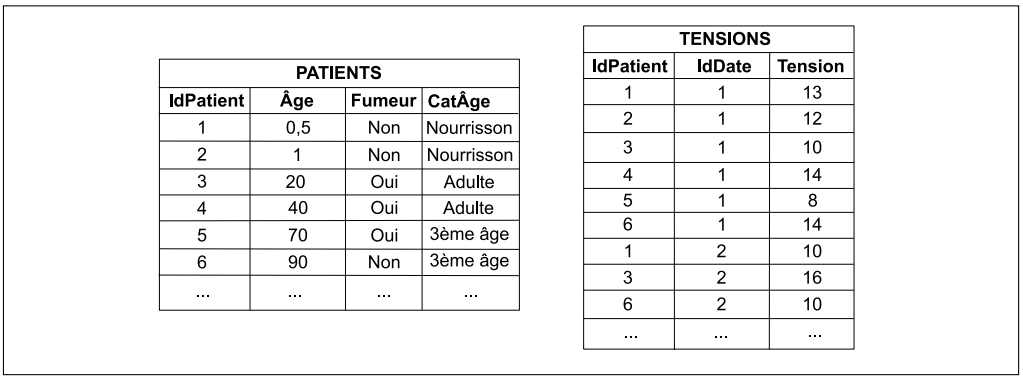
TAB. 1 – *Exemple de règles expertes décrivant la catégorie d'une tension (CatTension) en fonction de la tension mesurée, de la classe d'âge d'un patient (CatÂge) et de l'attribut Fumeur.*

Considérons les extraits de la table de dimension `PATIENTS` et de la table de faits `TENSIONS` (figure 2). En s'appuyant sur les règles expertes du tableau 1, la réponse à la requête "*Quels sont les patients (idPatient) qui ont eu une tension élevée à la date 1 ?*" sera "*Les patients 1 et 4*". Dans la suite de cet article, nous décrivons le modèle utilisé pour représenter cette base experte ainsi que l'algorithme mis au point pour répondre efficacement à ce type de requêtes.

3 État de l'art

De nos jours une des grandes puissances des outils d'analyse en ligne est donc la navigation au travers de données plus ou moins détaillées. Comme évoqué en introduction, ceci s'appuit sur des opérateurs d'agrégation qui fonctionnent à partir de la hiérarchisation des dimensions. Ainsi un certain nombre de travaux se sont d'ores et déjà intéressés aux hiérarchies de dimension, et aux problèmes d'additivité des mesures selon les dimensions qui peuvent en découler. Cependant, à notre connaissance, aucune proposition sur la hiérarchisation de mesure n'a été apportée. On retrouve «simplement» la notion de mesure dérivée (ou calculée).

Compte-tenu de notre problématique de hiérarchisation de mesure contextuelle (non figée vis-à-vis du chemin d'agrégation), nous nous sommes donc intéressés aux différents travaux



PATIENTS			
IdPatient	**Âge**	**Fumeur**	**CatÂge**
1	0,5	Non	Nourrisson
2	1	Non	Nourrisson
3	20	Oui	Adulte
4	40	Oui	Adulte
5	70	Oui	3ème âge
6	90	Non	3ème âge
...	...	...	...

TENSIONS		
IdPatient	**IdDate**	**Tension**
1	1	13
2	1	12
3	1	10
4	1	14
5	1	8
6	1	14
1	2	10
3	2	16
6	2	10
...	...	...

FIG. 2 – *Extraits des tables PATIENTS et TENSIONS pour l'analyse de la catégorie d'une tension.*

introduisant une certaine forme de flexibilité ou de capacité d'expression au niveau des dimensions et de leurs hiérarchies afin de voir dans quelle mesure une adaptation pour la hiérarchisation de mesure était envisageable.

Notons tout d'abord qu'un travail important de formalisation conceptuelle des différentes hiérarchies a été proposée par Malinowski et Zimányi (2004), tentant d'établir une liste des hiérarchies possibles, plus ou moins complexes, posant plus au moins de problème quant à leur exploitation lors du processus d'agrégation. Mentionnons également le travail fait dans le domaine des hiérarchies floues présenté par Laurent (2003). Mais ces travaux se situent dans le contexte de dimensions indépendantes. Et finalement, dans le problème que nous avons posé, l'enjeu est de pouvoir prendre en compte le fait que les dimensions ne sont pas forcément indépendantes, et surtout, que les mesures peuvent elles-mêmes être hiérarchisées en prenant en compte le contexte.

Afin de pouvoir rendre l'analyse plus flexible, un langage à base de règles a été développé dans Espil et Vaisman (2001) pour la gestion des exceptions dans le processus d'agrégation. Le langage IRAH (Intensional Redefinition of Aggregation Hierarchies) permet de redéfinir des chemins d'agrégation pour exprimer des exceptions dans les hiérarchies de dimensions prévues par le modèle. Ce travail permet aux utilisateurs d'exprimer eux-mêmes les exceptions dans le processus d'agrégation. En effet, afin de prendre en compte ces exceptions, les utilisateurs définissent et exécutent un programme IRAH, produisant ainsi une révision des chemins d'agrégation. L'exemple considéré est l'étude des prêts d'une compagnie de crédit en fonction de la dimension emprunteur qui est hiérarchisée. La catégorie de l'emprunteur est définie en fonction de son revenu. Mais les auteurs expliquent qu'il est possible que l'analyste veuille réaffecter un emprunteur dans une autre catégorie en voulant tenir compte d'autres paramètres que le revenu. Dans ce cas, le processus d'agrégation doit tenir compte de cette «exception». Dans ces travaux les auteurs proposent alors un langage à base de règles qui permet de définir des analyses révisées qui tiennent compte de ce type d'exception. Si ce langage constitue une alternative à la rigidité du schéma multidimensionnel dans le processus d'agrégation pour les utilisateurs, il ne fait qu'en modifier les chemins en fonction d'exceptions dans les hiérarchies de dimension. Or dans notre cas, il ne s'agit pas de prendre en compte des exceptions, mais bel et bien de prendre en compte des chemins d'agrégation qui dépendent d'un contexte, au niveau

des mesures, en se basant sur des connaissances d'experts.

Dans Favre et al. (2007), un précédent travail s'est penché sur la proposition d'un modèle d'entrepôt à base de règles qui visait à représenter la création de nouveaux niveaux d'analyse pour répondre à un besoin de personnalisation des analyses en fonction de la connaissance individuelle d'utilisateurs. Les nouveaux niveaux étaient créés le long des hiérarchies de dimension (insertion d'un niveau ou ajout en fin de hiérarchie), forcément au-dessus du premier niveau (dimension). Le lien d'agrégation exprimé ne peut être directement lié à la table des faits, empêchant l'expression d'une hiérarchisation de mesure comme nous avons besoin de le faire.

Une alternative pouvant présenter un certain intérêt vis-à-vis du problème posé par la nécessité de prendre en compte le fait que les chemins d'agrégation le long d'une hiérarchie peuvent changer est le versionnement. Il s'agit de pouvoir exprimer un changement (plutôt temporel) au niveau des instances de dimension comme proposé par Bliujute et al. (1998), au niveau des liens d'agrégation comme envisagé par Mendelzon et Vaisman (2000) ou de la structure comme proposé par Morzy et Wrembel (2003). Ceci répond au problème évoqué dans l'introduction sur l'indépendance des dimensions, par rapport à la dimension temporelle entre autres. Le versionnement est mis en avant pour la possibilité de stocker le fait que les dimensions puissent évoluer. Le versionnement permet donc non seulement d'historiser les données de la table des faits à travers une dimension temporelle, mais également les modifications au sein même des hiérarchies de dimension. Le problème majeur de ces approches et que ces versions sont développées par rapport à une évolution dans le temps et ne sont pas faites pour prendre en compte une modification non pas dans le temps mais par rapport à ce que nous avons appelé «contexte». Par ailleurs, cela ne résoud pas le problème du point de vue de la hiérarchisation de mesure que nous avons besoin de modéliser.

Si l'ensemble des travaux présentés ici apportent une flexibilité dans l'analyse des données en se focalisant sur ce qui définit en l'occurrence le contexte d'analyse, à savoir les dimensions, ils ne prennent pas en compte le besoin de flexibilité au niveau de la mesure. Et il ne s'agit pas de « simples » hiérarchies dépendant seulement de la valeur de la mesure. A notre connaissance, il n'y a pas de travaux proposant ou permettant de solutionner ce problème. Ainsi, par la suite, nous proposons une modélisation, un stockage, une exploitation permettant de prendre en compte cette hiérarchisation contextuelle de mesure.

4 Formalisation pour une prise en compte des connaissances

Notre approche vise à permettre la prise en compte de contextes dans le processus d'agrégation vis-à-vis des mesures lorsque cela est nécessaire. Ainsi, ce que nous cherchons à pouvoir représenter et à exploiter est le fait que la détermination de la valeur d'un attribut pour généraliser une mesure dépend de connaisances experts que nous allons représenter et qui devront être ainsi prises en compte dans le processus d'analyse.

Si nous nous focalisons sur le besoin de prise en compte de contexte, il s'agit plus particulièrement d'attributs dont la valeur va dépendre des valeurs prises par d'autres attributs de dimension, ainsi que de la table des faits : de la mesure que l'on cherche à généraliser en l'occurrence. De ce fait, pour garder une cohérence du modèle, cette donnée ne peut figurer ni dans la table de dimension, ni dans la table de faits. Ceci ne peut donc être traité lors de la phase classique de chargement de l'entrepôt mais doit donc être pris en compte dans le cadre de la phase

d'analyse. Il s'agit de connaissances dont il faut tenir compte lors du processus d'agrégation, connaissances qui pourraient faire d'ailleurs l'objet d'un enrichissement au cours du temps par les experts. Ainsi, l'objectif est de pouvoir disposer d'une méthode de prise en compte de ces connaissances durant le processus d'analyse.

Pour supporter le processus qui vise à la prise en compte de contextes par rapport à la détermination de la valeur de certains attributs généralisant les mesures, il est alors crucial de disposer d'un modèle d'entrepôt qui retrace cette contextualisation, par conséquent un modèle plus flexible.

Dans cette formalisation, nous restreignons donc le cadre de cet article au cas d'un schéma en étoile présentant donc une seule table des faits avec des dimensions dénormalisées (mais dont les attributs peuvent être porteurs d'informations hiérarchisées implicitement).

Définition 1 (Dimensions)**.** *Soit $\{D_s, s = 1..t\}$ l'ensemble des t tables de dimension. Soit $A_s = \{a_{sg}/s = 1..t, g = 1..h_s\}$ l'ensemble des h_s attributs de la dimension D_s pour $s = 1..t$. Notons id_{D_s} l'attribut de A_s identifiant la dimension D_s.*

Exemple 1. *Dans notre étude sur la tension, $t = 2$: $D_1 \equiv PATIENT$, $D_2 \equiv DATE$.*
$A_1 = \{IdPatient, Age, CatAge, Fumeur\}$, $A_2 = \{IdDate, LabelDate\}$.
$id_{D_1} \equiv IdPatient$, $id_{D_2} \equiv IdDate$

Définition 2 (Faits)**.** *La table des faits est déterminée structurellement par un ensemble de dimensions et de mesures.*
$F = (\mathcal{D}, \mathcal{M})$ avec $\mathcal{D} = \{id_{D_s}, s = 1..t\}$ les identifiants des t dimensions décrivant F et $\mathcal{M} = \{M_u, u = 1..v\}$ dénote l'ensemble des v mesures de F.

Exemple 2. *Pour l'étude de la tension, on a :*
$F = (\{IdPatient, IdDate\}, \{Tension\})$

Définition 3 (Attributs contextualisés et contextualisant)**.** *Un attribut L est dit contextualisé si sa valeur dépend des valeurs prises par un ensemble d'autres attributs de l'entrepôt (qualifiés alors de contextualisant). Ainsi, cet attribut contextualisé est la généralisation de la mesure que l'on calcule et les attributs contextualisant sont les attributs de dimensions et la mesure elle-même.*

Exemple 3. *Dans l'étude de la tension, $CatTension$ est un attribut contextualisé puisque sa valeur dépend des attributs contextualisant $CatAge$, $Fumeur$ et $Tension$.*

Définition 4 (Univers de l'entrepôt)**.** *L'univers $\mathcal{U}$ de l'entrepôt est un ensemble d'attributs, tel que :*

$$\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$$

où $\mathcal{U}_1 = \{K_\alpha, \alpha = 1..z\} = \{A_s \bigcup M_u, /s = 1..t, u = 1..v\}$ est l'ensemble des z attributs appartenant aussi bien à la table de faits qu'aux tables de dimension et $\mathcal{U}_2 = \{L_\beta, \beta \geq 1\}$ est l'ensemble des attributs contextualisés, c'est à dire dont les valeurs dépendent du contexte et qui sont donc déterminées par des connaissances.

Exemple 4. $\mathcal{U}_1 = \{IdPatient, Age, CatAge, Fumeur, IdDate, LabelDate, Tension\}$
$\mathcal{U}_2 = \{CatTension\}$

Définition 5 (Contexte : structure)**.** *Soit $\{c_i, i = 1..n\}$ l'ensemble des n structures de contexte.*

La structure du contexte c_i est définie par un sous-ensemble d'attributs de $\mathcal{U}_1$ et un sous-ensemble de $\mathcal{U}_2$:

$$c_i = (\{K_\Omega\}, \{L_\Psi\})$$

avec $\{K_\Omega\} \subset \mathcal{U}_1$ et $\{L_\Psi\} \subset \mathcal{U}_2$ tels que $1 \leq \Omega \leq z'$ avec $z' \geq 2$, $\Psi \geq 1$.

Notons ainsi que $\{L_\Psi\}$ peut se ramener à un singleton (un seul attribut contextualisé défini par un contexte) et, qu'en général, $\{K_\Omega\}$ contient au minimum deux attributs (dans le cas contraire, la valeur de l'attribut contextualisé ne dépendrait plus que d'une seule valeur, et on se ramènerait à une conception classique de hiérarchie).

Exemple 5. $c_1 = (\{CatAge, Fumeur, Tension\}, \{CatTension\})$

Définition 6 (Contexte : instances)**.** *Chaque structure de contexte est ensuite instanciée. On a alors $\mathcal{C} = \{c_i^j, i = 1..n, j = 1..m_i\}$ qui constitue l'ensemble de toutes les instances de contextes.*

Notons c_i^j l'instance j de la structure du contexte i. Elle est définie par l'instanciation de chacun des attributs.

L'instanciation des attributs correspond à l'affectation d'une expression. En terme d'implémentation en base de données qui sera évoquée par la suite, cette expression devra respecter une syntaxe SQL valide.

L'instanciation $\mathcal{I}^j$ des attributs contextualisés de $\{L_\Psi\}$ correspond à une expression caractérisant l'affectation d'une et une seule valeur :

$\mathcal{I}^j(L_\Psi) =$ "$= val_\Psi^j$"

Exemple 6. *Instanciation d'attribut contextualisé :*

$\mathcal{I}^1(L_1)=$ "$='Elevée'$"

Par contre, l'instanciation des autres attributs du contexte ne correspond pas forcément à l'expression d'une affectation de valeur. Elle correspond à une expression pouvant représenter plusieurs valeurs.

Ainsi ce terme peut être noté comme suit :

Soit $T_i^{\omega j}$ le terme correspondant à l'instance j de l'attribut ω du contexte i. $T_i^{\omega j}$ est de la forme " op $\{ens|val\}$" où op est un opérateur ($=$, $<$, $\leq$, $\geq$, $\neq$, $\in$, ...) ; ens est un ensemble de valeurs et val une valeur.

Exemple 7. *Instanciation d'attributs contextualisant :*

$T_1^{11} =$ "$='Nourrisson'$"
$T_1^{21} =$ "$IN('Oui','Non')$"
$T_1^{31} =$ "> 12"

Exemple 8. *Contexte :*

$c_1^1 = (\{$"$='Nourrisson'$","$IN('Oui','Non')$","> 12"$\}, \{$"$='Elevée'$"$\})$

Dans la suite, l'appelation contexte représente en fait une instance de contexte (à la fois sa structure et ses valeurs) lorsque nous ne précisons pas spécifiquement qu'il s'agit de la structure.

5 Mise en œuvre de notre approche : stockage et interrogation

Maintenant que les hiérarchies contextuelles ont été formellement introduites, nous nous focalisons sur leur implantation et leur utilisation au sein d'un entrepôt de données.

5.1 Stockage des connaissances : méta-table et table

Cette section aborde la problématique de la représentation et du stockage de ces connaissances expertes dans l'hypothèse d'une mise en œuvre dans un système relationnel (ROLAP) pour le stockage de l'entrepôt de données. Pour cela, nous proposons la création d'une base de données externe dont nous détaillons ici le contenu.

Dans la mesure où plusieurs contextes peuvent cohabiter au sein d'un même entrepôt (*e.g.,* la catégorisation d'une tension artérielle et celle d'une posologie), il est nécessaire de stocker quels sont les attributs qui interviennent dans chaque contexte (*i.e.,* quels sont les attributs contextualisants) car ceux-ci peuvent différer pour chaque contexte. Nous proposons alors la création d'une table `MTC` (Méta-Table des Connaissances) afin de stocker la structure associée à chaque contexte présent dans l'entrepôt.

Définition 7 (`MTC`)**.** *Soit* `MTC` $= (Contexte, Attribut, Table, Type)$ *une table relationnelle où :*

- ***Contexte*** *désigne l'identifiant du contexte (c_i) ;*
- ***Attribut*** *désigne un attribut intervenant dans le contexte c_i (*i.e., $Attribut \in K_\Omega \cup L_\Psi$*) ;*
- ***Table*** *correspond à la table relationnelle de l'entrepôt où* Attribut *est instancié ;*
- ***Type*** *définit le rôle joué par* Attribut *dans le contexte c_i. Cet attribut vaut "*Contexte*" si* Attribut *est un attribut* contextualisant *de c_i et vaut "*Résultat*" si* Attribut *est l'attribut* contextualisé *de c_i.*

Exemple 9. *Considérons le cas d'étude présenté dans la section 2 et analysons comment est stockée la structure du contexte associé à la généralisation d'une tension artérielle. Nous rappelons que la catégorisation d'une tension est fonction de plusieurs paramètres : la tension mesurée, la catégorie d'âge du patient et du fait qu'il soit fumeur ou non. Par convention, le nom donné à chaque contexte est celui de la mesure à généraliser. Ici, l'attribut* Contexte *vaut donc "*Tension*". Ensuite, les attributs dont la modalité est* Contexte *(*i.e, *les attributs* contextualisant*) sont* C*atÂge,* F*umeur et* T*ension. Ceux-ci sont définis dans les tables* `PATIENTS` *et* `TENSIONS`*. Enfin, l'attribut contextualisé est* C*atTension. Le tableau de la figure 3 présente un extrait de la table* `MTC` *associé au contexte* T*ension.*

Une fois la structure de chaque contexte définie dans `MTC`, nous nous intéressons à la matérialisation des instances de chaque contexte. Pour cela, nous définissons au sein de la base de données externe une nouvelle table relationnelle `TC` (Table des Connaissances). Notons que sur l'exemple précédent, la table associée à l'attribut contextualisé (*catTension*) est cette nouvelle table `TC`. Dans la mesure où les valeurs prises par cet attribut sont dépendante des valeurs prises par les attributs *contextualisants*, ce choix de modélisation semble le plus adapté.

MTC			
Contexte	**Attribut**	**Table**	**Type**
Tension	CatÂge	Patient	Contexte
Tension	Fumeur	Patient	Contexte
Tension	Tension	Tension	Contexte
Tension	CatTension	Tension	Résultat
Posologie	...	...	...

FIG. 3 – *Extraits de la table MTC décrivant le contexte associé à la généralisation d'une tension artérielle.*

Définition 8 (TC)**.** *Soit* TC $= (Contexte, Instance\_contexte, Attribut, Valeur)$ *une table relationnelle de la base de données externe telle que :*

- ***Contexte*** *désigne l'identifiant du contexte (*c_i*) ;*
- ***Instance_Contexte*** *identifie l'instance du contexte concernée (correspond à l'indice* j *pour* c_i^j*) ;*
- ***Attribut*** *désigne un attribut intervenant dans le contexte* c_i (i.e., $attribut \in K_\Omega \cup L_\Psi$*) ;*
- ***Valeur*** *correspond à la valeur affectée à Attribut dans* c_i^j *.*

Pour diminuer la taille de la table TC, le nombre d'instances de contexte stockées peut être réduit en autorisant l'attribut Valeur à contenir une expression SQL syntaxiquement correcte. Par exemple, "*IN (Oui,Non)*" évite la création de deux instances de contexte (une pour chaque valeur de l'opérateur *IN*).

Considérons le contexte *Tension* présenté lors de l'exemple précédent. Comme illustré dans le tableau 1, de nombreuses règles expertes existent pour déterminer la généralisation d'une tension artérielle. Alors que la structure de ces règles (*i.e.,* quels sont les attributs qui impactent sur la généralisation d'une mesure) est définie au niveau de la méta-table des connaissances MTC, les instances de ces contextes (*i.e.,* les règles expertes) sont définies au niveau de la table des connaissances TC. Le tableau 4 présente un extrait du contenu stocké dans TC. Chaque ligne du tableau 1 représente une instance du contexte *T*ension.

Exemple 10. *Illustrons le stockage des instances de contexte en considérant la connaissance "*chez un nourrisson, une tension supérieure à 12 est élevée*". Cette instance est représentée dans* TC *par les 4 n-uplets dont la modalité de l'attribut* Contexte *est* Tension *et celle de l'attribut* Instance_contexte *est 1. Parmi ces 4 n-uplets, les 3 premiers permettent de décrire les conditions à réunir pour que la généralisation (stockée dans le quatrième n-uplet) soit valide.*

5.2 Prise en compte des hiérarchies contextuelles lors de l'interrogation

Maintenant que nous avons décrit comment représenter et stocker les hiérarchies contextuelles, nous nous focalisons sur leur exploitation et leur interrogation afin de fournir à l'utilisateur une plus grande flexibilité dans l'analyse des données de l'entrepôt. La solution adoptée doit satisfaire deux conditions : ne pas diminuer la puissance d'analyse des outils OLAP et

TC			
Contexte	**Instance_contexte**	**Attribut**	**Valeur**
Tension	1	CatÂge	= Nourisson
Tension	1	Fumeur	IN (Oui,Non)
Tension	1	Tension	>12
Tension	1	CatTension	= Élevée
Tension	2	CatÂge	= Adulte
Tension	2	Fumeur	= Oui
Tension	2	Tension	>14
Tension	2	CatTension	= Élevée
Tension	3	CatÂge	= 3ème âge
Tension	3	Fumeur	IN (Oui,Non)
Tension	3	Tension	>16
Tension	3	CatTension	= Élevée
Tension	4	...	...

FIG. 4 – *Extraits de la table TC présentant quelques instances de contexte associées au contexte Tension.*

autoriser l'interrogation des hiérarchies contextuelles. Pour cela, nous proposons la prise en compte de ces hiérarchies lors de la création du cube de données pour l'analyse.

Ainsi, en assimilant la construction du cube à la création d'une vue, notre problème d'interrogation est résolu par la création de la vue adéquate. Pour illustrer cette solution, nous partons sur l'hypothèse de la création d'une vue `FG` dont les attributs sont (1) ceux de la table de faits et (2) un attribut contenant la généralisation de la mesure considérée. Les analyses OLAP *classiques* sont alors possibles en considérant au choix la table des faits ou la nouvelle vue `FG`. L'interrogation des hiérarchies contextuelles peut quant à elle être réalisée en interrogeant `FG`. Notons en outre que, par définition, la création de cette vue n'entraîne pas d'augmentation du volume de données stockées (si cette vue n'est pas matérialisée).

La génération de la requête associée à cette vue est présentée dans l'algorithme 1. Dans un premier temps, nous recherchons quels sont les attributs ayant un impact sur la mesure que nous désirons généraliser. Pour cela, il est nécessaire de recherche dans `MTC` quels sont les attributs *contextualisant* appartenant au contexte recherché. Afin de faciliter les jointures futures, les tables dans lesquelles sont stockés ces attributs sont également récupérées. L'algorithme se poursuit par la recherche de l'attribut *contextualisé*. L'étape suivante (lignes 4 à 12) permet d'écrire la requête générant la vue souhaitée. Pour cela, une sous-requête calculant l'instance du contexte associé à chaque n-uplet de la table de faits est mise en place et se base sur le principe suivant. Pour chaque attribut *contextualisant*, il faut d'abord récupérer la valeur associée puis trouver dans `TC` quelles sont les instances de contexte qui concordent avec cette valeur. Enfin, l'intersection de ces ensembles de contextes permet de récupérer le contexte qui valide toutes les conditions associées à chaque n-uplet de la table de faits. En vue de la mise en œuvre de ce modèle, nous envisageons l'utilisation du modèle relationnel-objet. Finalement, la requête permettant de créer la vue est générée.

Algorithme 1: Construction de la vue étendant une table de faits

Data : $F = (k_1, \ldots, k_n, m_1, \ldots, m_i, \ldots, m_l)$ une table de faits où m_i est la mesure à généraliser, $\mathcal{T} = \{T_1, \ldots, T_n\}$ (avec $T_i = (k_i, a_i^1, ...)$) les dimensions d'analyse associées avec $k_1, \ldots, k_n$les clés primaires de ces tables, MTC la Méta Table des connaissances et TC la table des connaissances

Result : ReqVue,une chaîne de caractères contenant la requête permettant la création de la vue généralisée

`/* 1) Recherche des couples <Table,Attribut> associés au contexte étudié */`
PairAttrib = SELECT Table, Attribut FROM MTC WHERE Contexte='m_i' and Type='Contexte' ;
`/* 2) Recherche de l'attribut contextuaisé */`
attrGen = SELECT Table, Attribut FROM MTC WHERE Contexte='m_i' AND Type='Resultat' ;
`/* 3) Construction du nom de la vue qui sera créée */`
nameView = m_i+'_View' ;
ReqVue = "CREATE OR REPLACE VIEW *nameView* AS
SELECT' F.k_1,. . .,F.k_n,F.$m_1, \ldots, F.m_i, \ldots, F.m_l$, (SELECT Valeur FROM TC WHERE Attribut = '*Resultat*' AND Contexte = 'm_i' ;
pour chaque $\langle T_i, A_i \rangle \in$ *PairAttrib* **faire**
  ReqVue = ReqVue + " AND instanceCtxt IN (select instance_match FROM ";
  si $T_i = F$ **alors**
    ReqVue = ReqVue + " instance_match('m_i',m_i,'m_i',F) ";
  sinon
    ReqVue = ReqVue + " instance_match('m_i',F,k_i,'k_i',A_i,T_i) ";
ReqVue = ReqVue + ") as *AttrGen* FROM F ";

retourner *ReqVue* ;

Illustrons cet algorithme en utilisant le cas d'étude présenté au cours de la section 2. Ici, nous cherchons à généraliser les tensions des patients. Dès lors, les attributs contextualisants sont *C*atÂge, *F*umeur et *T*ension. L'attribut contextualisé est *C*atTension. Étudions maintenant la requête générée grâce à la boucle qui permet de déterminer l'instance du contexte associé à chaque n-uplet de la table des faits. Pour cela, nous considérons le premier n-uplet de la table des faits. Si l'on considère l'extrait de la table `TC` présentée dans la figure **??**, l'instance du contexte *T*ension associée à un nourrisson dont la tension est 13 est l'instance 1. En effet, l'instance 1 est la seule dont les modalités de l'attribut *Valeur* associées aux attributs *CatÂge* et *Tension* coïncident avec celles du n-uplet de la table de faits étudié ici. Nous remarquons par contre que les contextes 1 et 3 sont tous les deux valides pour l'attribut *F*umeur. L'intersection de ces ensembles d'instances permet de déduire que l'instance 1 du contexte *Tension* doit être associée au premier n-uplet de la table de faits. Par conséquent, lors de la création de la vue `FG`, la modalité de l'attribut contenant la généralisation sera "=*Élevée*" (*i.e.,* celle stockée dans l'attribut *V*aleur dont le contexte est *Tension*, l'instance du contexte est *1* et la modalité de

l'attribut *A*ttribut est *CatTension*).

A travers cet exemple, nous avons illustré comment est stockée et exploitée la connaissance experte du domaine. Dès lors, la puissance d'analyse d'un entrepôt utilisant des hiérarchies de mesure contextuelles est accrue. De plus, cette apport de flexibilité n'entraîne pas une complexification du processus d'analyse dans la mesure où la création de la vue est transparente pour l'utilisateur. En outre, notons que l'ajout ou la modification de connaissances au sein des méta-table et table de connaissances peut être rendue très aisée grâce à une interface de saisie.

5.3 Discussion

Notre approche permet ainsi de répondre à la problématique de représentation et de stockage de contextes. Elle offre ainsi un moyen de prendre en compte une certaine forme de hiérarchisation de mesure, définie par un contexte.

Cette approche présente différents avantages. Ce mode de représentation permet de représenter différents contextes composés de structures différentes. Elle constitue ainsi une approche générique et permet l'ajout facile de contextes, autrement dit de nouvelles connaissances. Les contextes sont stockés dans une même table évitant que chaque contexte soit représenté par différentes tables, facilitant à terme le processus de réécriture de requêtes.

Un des intérêts de cette proposition est également de pouvoir regrouper des instances ensemble pour définir la fonction d'agrégat plutôt que de stocker le chemin d'agrégation de chaque instance, ce qui constitue un avantage en terme de complexité. Par exemple, chaque valeur de tension ne va pas faire l'objet de l'expression de son propre chemin d'agrégation mais un chemin pourra être défini pour un ensemble de valeurs (ainsi > 12 pour l'attribut *Tension*, reprend un ensemble de tensions).

Le choix d'une représentation relationnelle permet alors d'exploiter la puissance d'interrogation relationnelle. Et dans un contexte d'implémentation tel que celui-ci, le recours à la construction d'une vue semble constituer une approche pertinente. L'algorithme de construction de vue constitue une première proposition pour l'exploitation de ce type de connaissances. Il méritera d'être affiné au niveau de la construction des cubes, prenant donc en entrée d'autres paramètres. Mais dans un premier temps, cela permet une exploitation réelle de ces connaissances introduites.

6 Conclusion et perspectives

Dans cet article, nous proposons la première approche pour prendre en compte les hiérarchies contextuelles au sein d'un entrepôt de données du point de vue des mesures. Afin de représenter les différents contextes de généralisation, nous proposons la construction de deux tables externes pour stocker la connaissance experte du domaine. La méta-table des connaissances stocke la structure des différents contextes de l'entrepôt (e.g., la normalité d'une tension artérielle dépend de la tension mesurée, de la catégorie d'âge du patient et du fait qu'il fume ou non). La table des connaissances permet quant à elle d'exprimer les différentes instances d'un contexte donné (e.g., une tension supérieure à 14 chez un adulte fumeur est une tension élevée). En nous appuyant sur ces tables, nous proposons la création d'une vue étendant la table

de faits et garantissant à l'utilisateur une analyse flexible, efficace et adéquate de l'entrepôt de données. Cette création peut s'apparenter à la construction d'un cube.

De nombreuses perspectives s'ouvrent à la suite de ce travail. A court terme, l'implémentation de cette méthode sur un jeu de données proche du cas d'étude présenté dans cet article permettra de valider expérimentalement ce modèle. La mise en place d'une interface centrée utilisateur pour la saisie, la définition et la modification des contextes sera alors une étape fondamentale dans le processus de validation du système par l'utilisateur. Il faudra bien évidemment porter une attention particulière à l'aspect performances. En outre, la proposition d'une formalisation générique adaptée aux différents modèles d'entrepôts présents dans la littérature (modèle en flocon, constellation, ...) améliorerait l'applicabilité de la solution proposée dans cet article. Une perspective connexe est la mise en place d'un système de découverte semi-automatique de contextes grâce à des approches de fouille de données. Enfin, l'étude de l'adéquation de notre approche aux différents modèles d'entrepôts existants dans la littérature est envisagée. Plus particulièrement, dans la mesure où notre modèle permet la mise en place de hiérarchies sur des mesures, il serait intéressant d'étudier l'intégration des hiérarchies contextuelles à un modèle dit « en galaxie » proposé par Ravat et al. (2007) ne comportant pas de faits (tout y est dimension initialement).

Références

Agrawal, R., A. Gupta, et S. Sarawagi (1997). Modeling multidimensional databases. In *Data Engineering, 1997. Proceedings. 13th International Conference on*, pp. 232–243.

Bliujute, R., S. Saltenis, G. Slivinskas, et C. Jensen (1998). Systematic Change Management in Dimensional Data Warehousing. In *IIIrd International Baltic Workshop on Databases and Information Systems, Riga, Latvia*, pp. 27–41.

Chen, M., J. Han, et P. S. Yu (1996). Data mining : An overview from a database perspective. *IEEE Trans. on Knowl. and Data Eng. 8*(6), 866–883.

Eder, J. et C. Koncilia (2001). Changes of dimension data in temporal data warehouses. In *IIIrd International Conference on Data Warehousing and Knowledge Discovery (DaWaK 01), Munich, Germany*, pp. 284–293. Springer-Verlag.

Einbinder, J. S., K. W. Scully, R. D. Pates, J. R. Schubart, et R. E. Reynolds (2001). Case study : a data warehouse for an academic medical center. *Journal of Healthcare Information Management : JHIM 15*(2), 165–175. PMID : 11452578.

Espil, M. M. et A. A. Vaisman (2001). Efficient Intensional Redefinition of Aggregation Hierarchies in Multidimensional Databases. In *IVth ACM International Workshop on Data Warehousing and OLAP (DOLAP 01), Atlanta, Georgia, USA*, pp. 1–8. ACM Press.

Favre, C., F. Bentayeb, et O. Boussaid (2007). Evolution et personnalisation des analyses dans les entrepôts de données : une approche orientée utilisateur. In *XXVème congrès INFormatique des ORganisations et Systèmes d'Information et de Décision (INFORSID 07), Perros-Guirec, France*, pp. 308 – 323.

Han, J. (1997). OLAP mining : An integration of OLAP with data mining. *IN PROCEEDINGS OF THE 7TH IFIP 2.6 WORKING CONFERENCE ON DATABASE SEMANTICS (DS-7*, 1—9.

Inmon, W. H. (1996). *Building the Data Warehouse, 2nd Edition* (2 ed.). Wiley.

Laurent, A. (2003). Querying fuzzy multidimensional databases : unary operators and their properties. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 11*(Supplement), 31–45.

Malinowski, E. et E. Zimányi (2004). OLAP Hierarchies : A Conceptual Perspective. In *XVIth International Conference on Advanced Information Systems Engineering (CAiSE 04), Riga, Latvia*, Volume 3084 of *LNCS*, pp. 477–491. Springer.

Mallach, E. G. (2000). *Decision Support and Data Warehouse Systems*. McGraw-Hill Higher Education.

Mendelzon, A. O. et A. A. Vaisman (2000). Temporal Queries in OLAP. In *XXVIth International Conference on Very Large Data Bases (VLDB 00), Cairo, Egypt*, pp. 242–253. Morgan Kaufmann.

Morzy, T. et R. Wrembel (2003). Modeling a Multiversion Data Warehouse : A Formal Approach. In *Vth International Conference on Enterprise Information Systems (ICEIS 03), Angers, France*, pp. 120–127.

Pitarch, Y., A. Laurent, et P. Poncelet (2009). A conceptual model for handling personalized hierarchies in multidimensional databases. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, France, pp. 107–111. ACM.

Ravat, F., O. Teste, R. Tournier, et G. Zurfluh (2007). A Conceptual Model for Multidimensional Analysis of Documents. In *International Conference on Conceptual Modeling (ER), Auckland, New Zealand*, Number 4801 in LNCS, pp. 550–565. Springer.

Summary

Allowing multilevel analysis, hierarchies are considered as a central concept in datawarehouses. Unfortunately, the current datawarehouse models do not consider all the categories of hierarchy which are described in the litterature. For instance, there is no way to model the fact that a given arterial pressure is either low, normal or high depending on the age of the patient. In a previous work, we present these contextual hierarchies. In this paper, we propose the first approach to implement these hierarchies in a datawarehouse system. An external database stores the domain knowledge. Then, in order to provide a flexible, efficient and appropriate analysis, we propose a query rewriting algorithm to take the various contexts into account. Thus, it is now possible to correctly answer to the query "*Who has had a low arterial pressure this night?*".