Problèmes d'additivité dus à la présence de hiérarchies complexes dans les modèles multidimensionnels : définitions, solutions et travaux futurs

Marouane Hachicha, Jérôme Darmont

Problèmes d'additivité dus à la présence de hiérarchies complexes dans les modèles multidimensionnels : définitions, solutions et travaux futurs

Marouane Hachicha, Jérôme Darmont

Université de Lyon (ERIC Lyon 2)
5 Avenue Pierre Mendès-France
69676 Bron Cedex
France
Courriel:prénom.nom@univ-lyon2.fr

Résumé. De nos jours, les entrepôts de données et les outils d'analyse OLAP sont très utilisés dans les entreprises qui ont besoin de systèmes décisionnels qui s'adaptent à toutes les situations particulières du monde réel, pour éviter les erreurs d'analyse (plus connues dans la littérature sous le nom de problèmes d'additivité ou *summarizability issues* en Anglais). Dans cet article, nous présentons un état de l'art des travaux sur les problèmes d'additivité dus à la présence de hiérarchies complexes dans les modèles multidimensionnels et discutons des travaux restant à mener dans ce domaine.
Mots clés : Entrepôts de données, OLAP, Problèmes d'additivité, Hiérarchies non-strictes, Hiérarchies non-couvrantes, Hiérarchies complexes, Normalisation, Transformation, Solutions en temps réel.

1 Introduction

De nos jours, les systèmes d'information décisionnels, basés sur les entrepôts de données et les analyses OLAP (*On-Line Analytical Processing*), sont très utilisés en entreprise dans un but d'aide à la décision. Afin de les préparer au stockage et à l'analyse, les données décisionnelles sont modélisées, au niveau conceptuel, d'une façon dite multidimensionnelle. La *modélisation multidimensionnelle* consiste à représenter les données décisionnelles selon leurs niveaux d'agrégation (faits et dimensions qui peuvent être hiérarchisées). Par exemple, le modèle conceptuel UML de la figure 1 représente des faits *ventes* décrits selon deux dimensions : (1) la dimension *géographique* organisée en hiérarchie : *ville-région-pays* (du plus détaillé au plus général) et (2) la dimension *date* organisée en hiérarchie : *jour-semaine-mois-année* (du plus détaillé au plus général). D'après Mazón et al. (2009), dans un modèle multidimensionnel, le concepteur doit garantir (1) la représentation adéquate des associations fait-dimension et (2) la représentation adéquate des différents niveaux d'agrégation d'une dimension exprimée en hiérarchie. Généralement, les situations du monde réel sont modélisées selon des hiérarchies simples. Les associations entre les différents niveaux d'une *hiérarchie simple* sont de type "un à plusieurs". Mais, certaines situations du monde réel sont modélisées selon des hiérar-

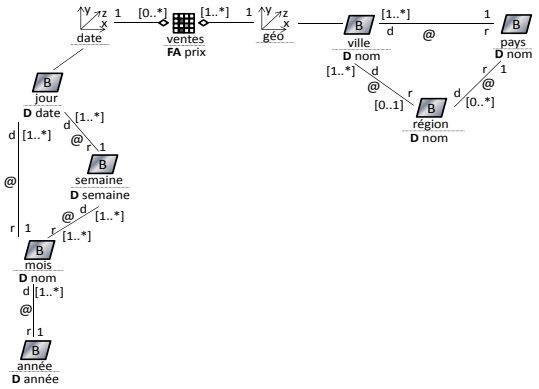

FIG. 1 – *Exemple de modélisation multidimensionnelle de ventes (voir Mazón et al., 2009, figure 1).*

chies dites complexes. Par définition, une *hiérarchie complexe* est une hiérarchie qui est à la fois non-stricte et non-couvrante (Hachicha et al., 2012).

Une hiérarchie est dite *non-stricte* lorsque l'association entre deux niveaux hiérarchiques est de type "plusieurs à plusieurs" et non de type "un à plusieurs" (Malinowski et Zimányi, 2006). Par exemple, une semaine peut appartenir à plusieurs mois et non un seul. Par ailleurs, un mois contient plusieurs semaines (figure 1). Une hiérarchie est dite *non-couvrante* lorsque deux niveaux hiérarchiques sont reliés l'un à l'autre en "sautant" un ou plusieurs niveaux intermédiaires (Malinowski et Zimányi, 2006). L'association entre *pays* et *région* dans la figure 1 (zéro à plusieurs) reflète le cas réel où certains pays ne contiennent pas de région (comme Monaco, qui est à la fois une ville et un état).

Les hiérarchies complexes sont un sujet très sensible dans les entrepôts de données et l'OLAP. Le fait d'ignorer la nature de ces hiérarchies en concevant l'entrepôt de données (et/ou les opérateurs OLAP) peut être à l'origine d'erreurs d'analyse et, par la suite, la cause principale de l'échec de tout un projet décisionnel. Les problèmes dus à la présence des hiérarchies non-couvrantes, non-strictes ou complexes sont connus dans la littérature sous le nom de *problèmes d'additivité* ou *summarizability issues* en Anglais (Mazón et al., 2009).

L'objectif de cet article est de présenter un état des lieux des problèmes d'additivité dus aux hiérarchies complexes et de leur traitement, inédit en Français. Notons qu'il n'est pas une traduction de l'état de l'art de Mazón et al. (2009). Nous classons en effet et discutons les travaux sur les problèmes d'additivité d'une façon différente, et nous élargissons de plus notre étude aux solutions en temps réel, qui ne sont pas étudiées par Mazón et al. (2009) et nous semblent très prometteuses. Notons enfin que nous n'avons pas dans cet article l'espace suffisant pour étudier d'autres travaux sur l'additivité comme les travaux sur les contraintes de compatibilité de type (Lenz et Shoshani, 1997; Prat et al., 2011).

Le reste de cet article est organisé comme suit. Dans la section 2, nous définissons l'additivité et étudions les contraintes d'additivité liées aux hiérarchies complexes. Dans la section 3, nous étudions les solutions présentées dans la littérature pour les problèmes d'additivité dus aux hiérarchies complexes et nous les discutons. Nous présentons les travaux que nous anticipons pour le futur dans la section 4. Nous concluons cet article dans la section 5.

2 Additivité

2.1 Définition de l'additivité

La notion d'additivité (*summarizability*) a été introduite par Rafanelli et Shoshani (1990) dans le contexte des bases de données statistiques et analysée par Lenz et Shoshani (1997) dans le contexte des bases de données multidimensionnelles. Soit une dimension, appartenant à l'ensemble des dimensions d'un modèle multidimensionnel et exprimée en hiérarchie. Par définition, l'additivité désigne le *calcul correct* des valeurs agrégées à un niveau d'agrégation donné à partir des valeurs agrégées à un niveau d'agrégation inférieur (Mazón et al., 2009). Ce calcul n'est autre que l'opération OLAP de forage vers le haut (*roll-up*) (Teste, 2009).

Notons que le terme additivité que nous utilisons dans cet article est la traduction imparfaite du terme anglo-saxon *summarizability*. Dans la littérature, le terme anglo-saxon *additivity* désigne un cas particulier de *summarizability* lorsque la fonction d'agrégation utilisée est la somme (Horner et al., 2004). Dans cet article, nous étudions les problèmes d'additivité (*summarizability issues*) qui concernent toutes les fonctions d'agrégation.

2.2 Contraintes d'additivité liées aux hiérarchies complexes

Rafanelli et Shoshani (1990) et Lenz et Shoshani (1997) introduisent les contraintes suivantes pour de garantir l'additivité. (1) Les cardinalités de type "plusieurs à plusieurs" ne doivent pas être utilisées, c'est-à-dire que la cardinalité maximale associée au niveau supérieur (dans une association entre un niveau inférieur et un niveau supérieur d'une hiérarchie) ne peut être que 1. (2) Toutes les associations entre les niveaux hiérarchiques doivent être de type "un à plusieurs". (3) Dans une hiérarchie de dimension, les relations de type "zéro à plusieurs" ne doivent pas être utilisées : toute instance du niveau inférieur doit être liée à au moins une instance du niveau supérieur.

Nous concluons donc que, pour garantir l'additivité dans un modèle multidimensionnel, deux contraintes majeures doivent être vérifiées : les associations entre les niveaux hiérarchiques d'une dimension doivent être à la fois strictes et couvrantes. Il est alors évident que la présence de hiérarchies complexes dans un modèle multidimensionnel provoque des problèmes d'additivité.

2.2.1 Hiérarchies de dimension non-strictes et problèmes d'additivité

Soit la dimension *date* de la figure 1. Dans la vie de tous les jours, nous disons couramment qu'un mois se compose de quatre semaines. Or, en réalité, quatre semaines correspondent à vingt-huit jours, ce qui est strictement inférieur au nombre de jours d'un mois (généralement trente ou trente-et-un). Donc, un mois est composé de quatre semaines et d'une partie de la cinquième semaine, elle-même première semaine du mois suivant. En conclusion, une semaine peut appartenir à plusieurs mois et un mois contient plusieurs semaines. L'association entre *semaine* et *mois* est de type "plusieurs à plusieurs", donc elle est non-stricte.

Supposons maintenant que nous voulions calculer le total des ventes d'un produit donné pour les mois de juin et de juillet 2011 (tableau 1(b)) à partir des ventes par semaine (tableau 1(a)). Le tableau 1(b) montre que si nous comptons la cinquième semaine deux fois (une fois pour chaque mois), nous obtenons un total incorrect (différent du celui du tableau 1(a)).

Cet exemple, inspiré d'un cas réel, illustre comment les hiérarchies non-strictes provoquent des problèmes d'additivité.

Semaine	Ventes
S1-06-2011	10
S2-06-2011	10
S3-06-2011	10
S4-06-2011	10
S5-06-2011	10
S2-07-2011	10
S3-07-2011	10
S4-07-2011	10
S5-07-2011	10
Total	90

(a) Ventes par semaine

Mois	Ventes
06-2011	50
07-2011	50
Total	100

(b) Ventes par mois

TAB. 1 – *Hiérarchies non-strictes et problèmes d'additivité.*

2.2.2 Hiérarchies de dimension non-couvrantes et problèmes d'additivité

Soit la la dimension *géographique* de la figure 1. Les cardinalités entre *région* et *pays* (zéro à plusieurs) reflètent le cas réel où certains pays du monde ne contiennent pas de région (comme Monaco, qui est à la fois une ville et un état). Cette relation est donc non-couvrante.

Les tableaux 2(a) et 2(b) représentent les sommes des ventes d'un produit donné par région et par pays, respectivement. La somme des ventes par pays est différente de la somme des ventes par région car il n'existe pas de région à Monaco. Cet exemple, inspiré d'un cas réel, démontre que les hiérarchies non-couvrantes provoquent des problèmes d'additivité.

Région	Ventes
Île-de-France	200
Rhône-Alpes	200
Bavière	200
Total	600

(a) Ventes par région

Pays	Ventes
France	400
Allemagne	200
Monaco	200
Total	800

(b) Ventes par pays

TAB. 2 – *Hiérarchies non-couvrantes et problèmes d'additivité.*

Remarque. Toutes les contraintes d'additivité que nous avons étudiées jusqu'à maintenant concernent les différents niveaux d'une dimension organisée en hiérarchie. Ces contraintes sont appelées *contraintes intradimensionnelles* par Lehner et al. (1998). Les auteurs introduisent également un autre type de contraintes d'additivité : les contraintes sur les relations entre faits et dimensions dans un modèle multidimensionnel (*contraintes interdimensionnelles*). Même si certains auteurs identifient les relations inter et intradimensionnelles comme des problèmes

séparés (Mazón et al., 2008), ces relations sont de même nature et les solutions pour régler les problèmes d'additivité sont exactement les mêmes (Pedersen et al., 1999).

3 Résolution des problèmes d'additivité

Dans cette section, nous étudions les solutions présentées dans la littérature pour les problèmes d'additivité dus aux hiérarchies complexes : la normalisation des modèles multidimensionnels (section 3.1), la transformation de données multidimensionnelles (section 3.2) et les solutions en temps réel (section 3.3). Nous concluons et discutons l'ensemble de ces travaux dans la section 3.4.

3.1 Normalisation des modèles multidimensionnels

La normalisation d'un modèle multidimensionnel est définie comme la transformation des hiérarchies de dimension (ou des associations fait-dimension) qui sont à l'origine des problèmes d'additivité en hiérarchies simples (strictes et couvrantes). Nous classons les stratégies de normalisation en deux familles. Dans la première, le but est de définir des contraintes et des règles de transformation du modèle multidimensionnel. Hurtado et al. (2005) présentent deux solutions pour les problèmes d'additivité dans les hiérarchies de dimension non-couvrantes (appelées hétérogènes par les auteurs). Par définition, une hiérarchie hétérogène est décrite de plusieurs façons dans le même schéma.

Pour faire face à ce problème, Hurtado et al. (2005) définissent un graphe direct G enrichi par deux types de contraintes : *les contraintes de chemin* (*Path atoms*) et *les contraintes d'égalité* (*Equality atoms*). Les contraintes de chemin sont de la forme [c_0, c_1,..., c_n] tel que [c_0, c_1,..., c_n] est un chemin dans la hiérarchie hétérogène. Les contraintes d'égalité sont de la forme [c_1 = k] tel que $k \in$ M (M est un ensemble d'instances d'un membre d'une dimension). Plusieurs signes sont autorisés dans la définition des contraintes comme $\neg$ (négation), $\vee$ (différent de), etc. Ensuite, Hurtado et al. (2005) introduisent la notion de dimensions couvrantes minimales (*frozen dimensions*). Ces dimensions sont représentées dans le même graphe avec les dimensions non-couvrantes. Le but de cette représentation est de rendre plus facile la compréhension des hiérarchies hétérogènes.

La deuxième stratégie de normalisation ajoute de nouvelles structures au modèle logique dans le but de garantir l'additivité, appelées les tables de jointure ou *bridge tables* en Anglais (Malinowski et Zimányi, 2006). De cette manière, les autres tables contiennent les données qui ne posent pas de problèmes d'additivité. Dans ce sens, Mazón et al. (2008) proposent un modèle conceptuel UML pour les différentes associations fait-dimension, non-strictes et non-couvrantes, afin de faciliter la représentation des situations du monde réel. Ils dérivent ensuite, à partir de ce modèle, un modèle conceptuel "normalisé" qui évite les problèmes d'additivité. Pour normaliser les associations fait-dimension non-strictes, deux solutions sont proposées par Mazón et al. (2008). Dans la première solution, si nous connaissons la contribution de chaque instance de dimension pour chaque instance de fait, les valeurs des mesures sont partagées entre les différentes instances des dimensions en question. Dans ce cas, la multiplicité maximale côté dimension est transformée de plusieurs à un. Cette méthode ne nécessite pas l'ajout de nouvelle classe au modèle conceptuel, mais il suffit de regrouper les différentes instances de dimensions qui posent des problèmes d'additivité dans un nouvel attribut ajouté aux faits.

Dans la deuxième solution, si nous ne connaissons pas la contribution de quelques instances de dimension pour chaque instance de fait, les auteurs procèdent comme suit : (1) les instances connues sont traitées comme dans la première solution et (2) les instances inconnues sont séparées dans une nouvelle classe de dimension.

3.2 Transformation de données multidimensionnelles

Les approches de transformation de données multidimensionnelles consistent à modifier les valeurs des instances de dimensions afin d'assurer l'additivité. Dans un premier travail, Pedersen et al. (1999) proposent une solution pour résoudre les problèmes d'additivité dans les hiérarchies de dimension complexes. Ils présentent trois algorithmes : (1) *MakeCovering* et (2) *MakeOnto* pour remplacer les valeurs manquantes dans les hiérarchies non-couvrantes et non-ontos [1], respectivement ; et (3) *MakeStrict* pour fusionner les valeurs des instances multiples dans les hiérarchies non-strictes.

- *MakeCovering :* Soit la hiérarchie *adresse-ville-région-pays*. L'algorithme commence par donner des valeurs aux valeurs manquantes à partir du niveau le plus détaillé de la hiérarchie (*adresse* dans cet exemple) jusqu'au niveau supérieur (*pays* dans cet exemple) en passant par les niveaux intermédiaires (*ville* puis *région* dans cet exemple). *MakeCovering* recherche les valeurs manquantes dans les métadonnées associées aux données multidimensionnelles et/ou fait appel à un expert. Par exemple, supposons que nous voulions compléter l'adresse d'une ruelle en Australie ou aux États-Unis, un avis d'expert peut être décisif en l'absence des métadonnées nécessaires.
- Le principe de *MakeOnto* est le même que celui de *MakeCovering*.
- *MakeStrict :* Une fois les hiérarchies non-couvrantes et non-ontos transformées en hiérarchies couvrantes et ontos, respectivement, l'algorithme *MakeStrict* est appelé afin de fusionner les valeurs des instances d'un même niveau hiérarchique ayant le même élément parent en une seule valeur. Les valeurs fusionnées sont ensuite insérées entre les deux niveaux hiérarchiques parent et enfant.

Dans une extension de ce travail, Mansmann et Scholl (2007) regroupent toutes les instances d'un niveau hiérarchique donné pour lesquelles nous ne connaissons pas les valeurs des instances de l'élément parent en dessous d'un niveau hiérarchique artificiel nommé "Others", dans les hiérarchies non-couvrantes et non-ontos. Les auteurs transforment les hiérarchies non-strictes en utilisant *MakeStrict*.

3.3 Solutions en temps réel

Le principe des solutions en temps réel est de détecter et de résoudre les problèmes d'additivité au moment du calcul des agrégations et non *a priori* (normalisation et transformation). L'opérateur *projection généralisée* de Pedersen et al. (2002) permet de calculer des agrégations. Au cas où il détecte des hiérarchies non-strictes ou non-couvrantes, cet opérateur retourne un résultat vide. Dans un deuxième travail, Horner et Song (2005) ouvrent des perspectives pour des nouvelles solutions en temps réel. Pour résoudre les problèmes d'additivité dans les hiérarchies non-strictes, les auteurs proposent d'exécuter des scripts pour (1) déterminer combien

1. Les hiérarchies non-ontos représentent un cas particulier des hiérarchies non-couvrantes. Une hiérarchie est dite non-onto lorsque la distance entre le plus haut et le plus bas niveaux de granularité de la hiérarchie varie d'un cas à un autre (Pedersen et al., 1999).

de fois les mesures ont été agrégées dans ces hiérarchies et (2) identifier le nombre total de valeurs multiples. Ensuite, Horner et Song (2005) distinguent trois types de valeurs manquantes : *orphelines*, *omises* et *non-applicables*, et proposent un traitement spécial pour chaque type.

Dans une approche plus récente, nous présentons deux opérateurs XOLAP d'*agrégation et regroupement* et de *forage vers le haut* (Hachicha et al., 2012). Dans cette approche, nous modélisons les données multidimensionnelles à l'aide d'arbres de données, que nous appelons les arbres de données multidimensionnels. Pour résoudre les problèmes d'additivité, nos deux opérateurs XOLAP sont basés sur l'algorithme QBS (*Query-Based Summarizability*) qui est composé de deux étapes : (1) détection des problèmes d'additivité et (2) création/mise à jour du résultat. QBS parcourt les faits de l'arbre de données un par un. Dans l'étape (1), si la hiérarchie en cours est non-couvrante, il regroupe toutes les instances correspondantes à chaque critère de regroupement dans une seule valeur artificielle "Other". En cas de hiérarchie non-stricte, il regroupe toutes les instances correspondantes à chaque critère de regroupement dans un seul groupe, comme dans *MakeStrict* (Pedersen et al., 1999). En cas de hiérarchie complexe, QBS associe ces deux solutions. Dans l'étape (2), QBS vérifie si l'ensemble de groupes créés à l'étape (1) existe dans un des arbres de données du résultat. Si c'est le cas, il met à jour le résultat d'agrégation avec la mesure en cours. Sinon, il crée un nouvel arbre dans le résultat final. Une fois tous les faits parcourus, tous les arbres résultats sont réunis sous une seule racine pour former le résultat final de la requête.

3.4 Discussion

3.4.1 Transformation ou normalisation ?

Toutes les approches de transformation se basent sur les algorithmes *MakeStrict* et *MakeCovering* de Pedersen et al. (1999). En fait, les hiérarchies non-strictes existent dans plusieurs situations du monde réel (Pedersen et al., 1999) et leur transformation à l'aide du processus de fusion nécessite le respect de la sémantique des données exprimées dans ces hiérarchies (Mansmann et Scholl, 2007).

Li et al. (2005) démontrent toutefois que l'algorithme *MakeCovering* n'est pas adapté à certains cas réels. Soit l'exemple de la *dimension géographique* liée à la Chine : *district-ville-province-pays*, où certaines provinces sont elles même des villes, comme Pékin. *MakeCovering* ne permet pas de résoudre ce problème. Li et al. (2005) étendent alors *MakeCovering* afin de pouvoir traiter les problèmes d'additivité liés à la hiérarchie géographique en Chine. L'inconvénient de ce travail est qu'il ne traite que ce cas particulier.

La transformation souffre de quelques problèmes de performance. En effet, à chaque fois que l'entrepôt est mis à jour, les contraintes d'additivité doivent être vérifiées. C'est pourquoi le choix de la transformation comme solution pour les problèmes d'additivité a été abandonné par les auteurs des approches de normalisation (Hurtado et al., 2005; Mazón et al., 2008). Nous pensons d'ailleurs que les dimensions non-couvrantes minimales de Hurtado et al. (2005) peuvent être associées à l'algorithme *MakeCovering* pour résoudre les problèmes présentés dans l'article de Li et al. (2005). Si nous associons la hiérarchie non-couvrante minimale *district-province-pays* à toutes les provinces-villes, *MakeCovering* pourra traiter les hiérarchies non-couvrantes minimales sans qu'il soit nécessaire de le modifier. Enfin, cette proposition (théorique) nécessite l'intervention d'un expert, ce qui ne contredit pas le principe de *MakeCovering*.

La deuxième technique de normalisation, les tables de jointure (*bridge tables*), est utilisée au niveau logique (Malinowski et Zimányi, 2006). Seuls Mazón et al. (2008) emploient cette technique au niveau conceptuel. Leur motivation est que l'utilisation de ces techniques au niveau logique nécessite une grande expertise pour modéliser les situations du monde réel les plus complexes. Les avantages des tables de jointure se résument dans leur très faible taille et leur utilité lorsque les informations reliées aux mesures ne changent pas dans le temps (Malinowski et Zimányi, 2006). Par contre, côté performance, des efforts supplémentaires sont nécessaires pour agréger toutes les mesures, distribuées entre les faits originaux et les tables de jointure. Enfin, dans la liste suivante, nous récapitulons les limitations de la normalisation et de la transformation :

(1) Modification des données et des situations réelles (normalisation, transformation).

(2) Besoin d'un expert de la modélisation multidimensionnelle et de la technique des tables de jointure (normalisation).

(3) Besoin de métadonnées et/ou d'un expert du domaine et des données entreposées, non prise en compte de toutes les situations complexes du monde réel et nécessité de mettre à jour toutes les données transformées à chaque fois que l'entrepôt est mis à jour (transformation).

3.4.2 Vers des solutions en temps réel ?

Même si elles sont différentes, la transformation et la normalisation partagent le même objectif : s'assurer de ne pas avoir des problèmes d'additivité au moment de l'exécution des requêtes OLAP (solutions *a priori*). Nous avons présenté dans la section 3.3 les seuls travaux qui proposent des solutions en temps réel pour les problèmes d'additivité. Ces travaux partagent le même objectif : la prise en compte des problèmes d'additivité au moment de l'analyse. L'opérateur *projection généralisée* de Pedersen et al. (2002), proposé dans le cadre d'une approche XOLAP, permet de signaler la présence des hiérarchies non-strictes ou non-couvrantes en retournant un résultat vide. Nous pensons que la détection de ces hiérarchies ne représente pas une solution à la hauteur des solutions de transformation ou de normalisation. Les directives de Horner et Song (2005) présentent plus de choix en terme de solutions pour les problèmes d'additivité que Pedersen et al. (2002). Toutefois, les auteurs ne présentent pas d'algorithme ni de programme pour implémenter ces directives. (Hachicha et al., 2012) est la première solution en temps réel "complète", dans le sens où elle détecte et résout les problèmes d'additivité en sortie de la requête utilisateur.

4 Perspectives de recherche

Mazón et al. (2009) attirent l'attention sur le fait que les entrepôts de données sont de plus en plus utilisés dans de nouveaux domaines (biologie, multimédia ou spatio-temporel). Ils concluent que l'additivité ne doit pas être assurée sans la prise en compte des propriétés sémantiques de chaque domaine et proposent d'intégrer ces propriétés dans l'entrepôt de données. Mazón et al. (2009) proposent aussi de traiter l'additivité en définissant de nouveaux opérateurs OLAP qui utilisent les techniques de fouille de données nécessaires à la gestion de ces données.

Comme Mazón et al. (2009), nous encourageons la proposition de nouvelles solutions de normalisation et de transformation pour faire face aux problèmes d'additivité dans tous les

outils basés sur les entrepôts de données et les analyses OLAP. Mais, pour qu'elles soient efficaces, ces nouvelles propositions doivent prendre en compte les inconvénients que nous citons à la fin de la section 3.4.1, ainsi que toutes les propositions et directives que nous trouvons dans la littérature comme les trois types de valeurs manquantes de Horner et Song (2005) (orphelines, omises et non-applicables).

Par ailleurs, nous préférons l'utilisation des solutions en temps réel pour faire face aux problèmes d'additivité. Même si ces solutions sont actuellement immatures, nous pensons que plusieurs travaux futurs sont possibles. Par exemple, il serait intéressant que les propositions de Horner et Song (2005) soient implémentés, afin de tester leur efficacité dans le cadre d'une solution en temps réel. Dans nos travaux futurs, nous comptons en premier lieu améliorer notre approche en temps réel (Hachicha et al., 2012) par la définition de nouveaux opérateurs XOLAP qui prennent en compte les problèmes d'additivité. Comme le temps d'exécution de l'étape (2) de l'algorithme QBS augmente de façon considérable en fonction du nombre des dimensions (critères de regroupement) dans la requête et de la taille des données, nous comptons élargir notre solution XOLAP à une approche holiste. Le but d'une approche holiste est de limiter l'accès à la mémoire en utilisant les étiquettes des nœuds pour le parcours et le traitement des arbres de données (voir Hachicha, 2012, chapitre 6, section 6.2).

5 Conclusion

Dans cet article, nous avons présenté un état de l'art des problèmes d'additivité dus aux hiérarchies complexes. En premier lieu, nous avons défini l'additivité et nous avons étudié les contraintes qui permettent de la garantir. Ensuite, nous avons étudié et discuté les trois familles de solutions pour les problèmes d'additivité : normalisation, transformation et en temps réel. Pour chaque famille de solution, nous étudions son principe, ses avantages et ses inconvénients. Nous concluons cette étude par des propositions de travaux futurs. En tout cas, quelle que soit la nature des données, nous pensons que l'additivité doit être considérée dans tout projet décisionnel.

Références

Hachicha, M. (2012). *Modélisation de hiérarchies complexes dans les entrepôts de données XML et traitement des problèmes d'additivité dans l'analyse en ligne XOLAP*. Thèse de Doctorat, Université Lumière Lyon 2, France.

Hachicha, M., C. Kit, et J. Darmont (2012). A Novel Query-Based Approach for Addressing Summarizability Issues in XOLAP. In *COMAD'12, Pune, India*, pp. 56–67. CSI.

Horner, J. et I.-Y. Song (2005). A Taxonomy of Inaccurate Summaries and Their Management in OLAP Systems. In *ER'05, Klagenfurt, Austria*, Volume 3716 of *LNCS*, pp. 433–448. Springer.

Horner, J., I.-Y. Song, et P. P. Chen (2004). An Analysis of Additivity in OLAP Systems. In *DOLAP'04, Washington, DC, USA*, pp. 83–91. ACM.

Hurtado, C. A., C. Gutiérrez, et A. O. Mendelzon (2005). Capturing Summarizability with Integrity Constraints in OLAP. *ACM Trans. Database Syst. 30*(3), 854–886.

Lehner, W., J. Albrecht, et H. Wedekind (1998). Normal Forms for Multidimensional Databases. In *SSDBM'98, Capri, Italy*, pp. 63–72. IEEE Computer Society.

Lenz, H.-J. et A. Shoshani (1997). Summarizability in OLAP and Statistical Data Bases. In *SSDBM'97, Olympia, Washington, USA*, pp. 132–143. IEEE Computer Society.

Li, Z., J. Sun, J. Zhao, et H. Yu (2005). Transforming Non-covering Dimensions in OLAP. In *APWeb'05, Shanghai, China*, Volume 3399 of *LNCS*, pp. 381–393. Springer.

Malinowski, E. et E. Zimányi (2006). Hierarchies in a multidimensional model : From conceptual modeling to logical representation. *Data & Knowledge Engineering 59*(2), 348–377.

Mansmann, S. et M. H. Scholl (2007). Empowering the OLAP Technology to Support Complex Dimension Hierarchies. *Int. Journal of Data Warehousing and Mining 3*(4), 31–50.

Mazón, J.-N., J. Lechtenbörger, et J. Trujillo (2008). Solving Summarizability Problems in Fact-Dimension Relationships for Multidimensional Models. In *DOLAP'08, Napa Valley, USA*, pp. 57–64. ACM.

Mazón, J.-N., J. Lechtenbörger, et J. Trujillo (2009). A survey on summarizability issues in multidimensional modeling. *Data & Knowledge Engineering 68*(12), 1452–1469.

Pedersen, D., K. Riis, et T. B. Pedersen (2002). A Powerful and SQL-Compatible Data Model and Query Language for OLAP. In *ADC'02, Melbourne, Australia*, Volume 5 of *CRPIT*. Australian Computer Society.

Pedersen, T. B., C. S. Jensen, et C. E. Dyreson (1999). Extending Practical Pre-Aggregation in On-Line Analytical Processing. In *VLDB'99, Edinburgh, UK*, pp. 663–674. Morgan Kaufmann.

Prat, N., I. Comyn-Wattiau, et J. Akoka (2011). Combining objects with rules to represent aggregation knowledge in data warehouse and OLAP systems. *Data Knowl. Eng. 70*(8), 732–752.

Rafanelli, M. et A. Shoshani (1990). STORM : A Statistical Object Representation Model. In *SSDBM'90, Charlotte, USA*, Volume 420 of *LNCS*. Springer.

Teste, O. (2009). *Modélisation et manipulation des systèmes OLAP : de l'intégration des documents à l'usager*. Mémoire d'HDR, Université Paul Sabatier - Toulouse III, France.

Summary

Nowadays, data warehousing and OLAP tools play a decisive role in companies all around the world. These companies have to use data analysis tools adapted to all real-world situations in order to avoid summarizability issues, which in turn may lead to erroneous analysis results. In this paper, we present a survey on summarizability issues in complex hierarchies: definitions, existing solutions and suggestions for future works.

Keywords: Data warehouses, OLAP, Summarizability issues, Non-strict hierarchies, Incomplete hierarchies, Complex hierarchies, Normalization approaches, Transformation approaches, Query-time approaches.