Agrégations multiples différenciées dans les bases de données multidimensionnelles

Ali Hassan, Franck Ravat, Olivier Teste, Gilles Zurfluh

Agrégations multiples différenciées dans les bases de données multidimensionnelles

Ali Hassan[1], Franck Ravat[1], Olivier Teste[2], Ronan Tournier[1], Gilles Zurfluh[1]

1. Université Toulouse 1 Capitole – IRIT (UMR 5505)
118, Route de Narbonne – F-31062 Toulouse cedex 9

2. Université Toulouse 2 IUT Blagnac – IRIT (UMR 5505)
1, Place Georges Brassens BP 60073, F-31703 Blagnac cedex

{hassan, ravat, teste, tournier, zurfluh}@irit.fr

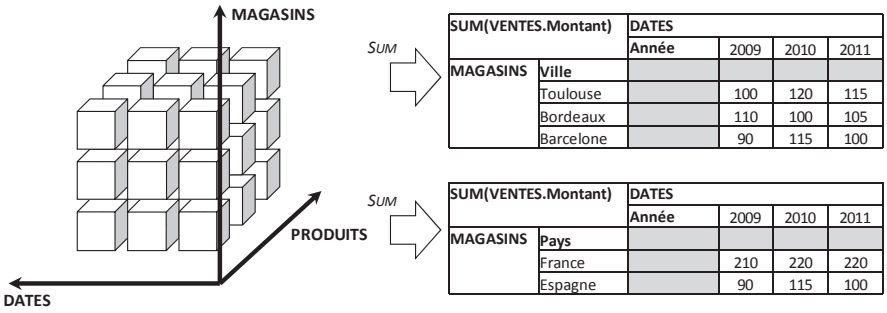
RÉSUMÉ. *De nombreux modèles ont été proposés pour la modélisation de données multidimensionnelles dans les entrepôts. Ces propositions considèrent une même fonction d'agrégation pour déterminer les valeurs d'une mesure aux différents niveaux de granularité de l'espace multidimensionnel. Nous proposons un nouveau modèle conceptuel plus expressif supportant des agrégations multiples différenciées. L'agrégation multiple permet d'associer à une même mesure, des fonctions d'agrégation différentes pour chaque axe d'analyse ou pour chaque hiérarchie. L'agrégation différenciée autorise des agrégations spécifiques à chaque niveau de granularité. Le modèle proposé repose sur des formalismes graphiques suffisamment expressifs pour contrôler la validité des fonctions d'agrégation qui peuvent être distributives, algébriques ou holistiques. Nous montrons également comment la modélisation conceptuelle peut être exploitée au niveau logique R-OLAP pour construire efficacement des treillis de pré-agrégats.*

ABSTRACT. *Many models have been proposed for multidimensional data warehouses modeling. These approaches consider the same aggregate function to determine the values of a measure with different levels of granularity into the multidimensional space. We define a new conceptual model for multidimensional representation of data supporting multiple differentiated aggregations. Multiple aggregations consist in associating different aggregation functions to each dimension or hierarchy. Differentiated aggregations allow specific aggregations at each level of granularity. The defined model is based on graphical formalisms, which are expressive enough to control the validity of aggregate functions that can be distributive, algebraic and holistic. We also show how conceptual modeling can be exploited in R-OLAP context to build valid and efficient lattices of pre-aggregates.*

MOTS-CLÉS : *systèmes décisionnels, bases de données multidimensionnelles, modélisation conceptuelle d'entrepôts de données, mécanismes d'agrégations multiples, treillis multidimensionnels de pré-agrégats.*

1. Introduction

Les systèmes d'information d'aide à la prise de décision ont montré leur capacité à intégrer de larges volumes de données tout en supportant efficacement des analyses sur les données entreposées. Ces systèmes décisionnels sont élaborés à partir de sources de données, provenant généralement du système opérationnel d'une organisation ; les données identifiées pertinentes dans les sources sont extraites, transformées, puis chargées (Vassiliadis *et al.*, 2002) dans un espace de stockage appelé entrepôt de données (« data warehouse »). Afin de rendre efficace l'interrogation et l'analyse de ces données entreposées, des techniques d'organisation des données spécifiques ont été développées (Kimball, 1996) reposant sur des bases de données multidimensionnelles (BDM). Ce type de modélisation considère la donnée à analyser comme un point dans un espace à plusieurs dimensions, formant ainsi un cube de données (Gray *et al.*, 1996). Les décideurs visualisent un extrait des cubes de données, généralement une tranche à deux dimensions (Gyssens et Lakshmanan, 1997). A partir de cette structure, appelée table multidimensionnelle (TM), le décideur peut interagir par des opérations de manipulation (Ravat *et al.*, 2007). Les opérations les plus emblématiques sont les forages qui consistent à modifier le niveau de granularité des données observées et les opérations de rotation qui consistent à changer de tranche du cube manipulé. On parle d'analyse en ligne ou encore de processus OLAP « On-Line Analytic Processing ».



SUM(VENTES.Montant)		DATES			
		Année	2009	2010	2011
MAGASINS	Ville				
	Toulouse		100	120	115
	Bordeaux		110	100	105
	Barcelone		90	115	100

SUM(VENTES.Montant)		DATES			
		Année	2009	2010	2011
MAGASINS	Pays				
	France		210	220	220
	Espagne		90	115	100

Figure 1. Agrégation uniforme appliquée aux tranches d'un cube

Cet environnement offre un cadre adéquat aux analyses des décideurs, cependant les structures de données imposées peuvent s'avérer imparfaites. En particulier, lors d'une analyse, une BDM classique supporte des calculs d'agrégation uniforme réalisés à partir de la même fonction d'agrégation dans les différentes tranches du

cube. Par exemple, si l'on considère des montants de ventes, ces derniers peuvent être calculés en effectuant la somme des produits vendus en fonction des villes et des années. Le calcul de ces mêmes montants de ventes en fonction des pays est généralement réalisé avec la même fonction d'agrégation (SUM) comme l'illustre la figure 1. Dès lors que l'utilisateur souhaite changer les fonctions d'agrégation entre deux tranches de cube manipulées, les BDM classiques ne garantissent plus la validité des données calculées, voire ne supportent pas ce type de manipulation.

Les travaux présentés dans cet article visent à rendre possible des agrégations non uniformes lors de la manipulation tout en garantissant leur validité. Nous proposons un modèle multidimensionnel permettant de supporter des *agrégations multiples différenciées*. Notre proposition vise à développer un modèle multidimensionnel suffisamment expressif pour autoriser la conception de cube intégrant différentes fonctions d'agrégation aux différents niveaux d'agrégation.

1.1. Cas d'étude

Pour illustrer nos propos, nous utilisons le cas d'un jury de délibération des diplômes. Dans cet exemple, les décideurs (enseignants membres du jury) délivrent les diplômes en analysant les notes (moyennes maximales, minimales) des étudiants et en regardant le taux d'absentéisme aux contrôles.

Nous considérons que l'année universitaire se compose de deux semestres. Chaque semestre comprend des unités d'enseignement (UE). Chaque UE se compose de matières. Les matières peuvent comprendre plusieurs contrôles. La note de chaque contrôle représente une partie de la note totale de la matière considérée : chaque contrôle a un coefficient qui représente le pourcentage de ce contrôle dans la matière. De manière analogue, chaque matière est elle-même associée à un coefficient qui représente l'importance de la matière dans l'UE. Il faut prendre en compte cet autre coefficient pour calculer la note de l'UE, qui elle-même est liée à une valeur de crédit (ECTS) utilisée pour calculer la note par semestre qui est donc une moyenne pondérée. Chaque semestre accumule le même total d'ECTS.

L'absence est enregistrée par contrôle. Les analystes peuvent souhaiter surveiller le taux d'absentéisme selon deux manières différentes :

– la première, **simple**, consiste à calculer le pourcentage des contrôles où l'étudiant est absent sans distinction entre les matières ou les différentes UE ;

– la deuxième, **pondérée**, utilise les mêmes coefficients pour calculer les notes des UE et des semestres et pour calculer les taux d'absentéisme.

Les étudiants sont classés selon leurs années de naissance et selon leurs statuts (FI étudiant en formation initiale, FC étudiant en formation continue, FA étudiant en formation en alternance). Dans notre exemple, en plus des contrôles et des étudiants, les enseignants peuvent analyser les notes et les taux d'absentéisme en fonction des dates (années scolaires) et des formations. Les formations sont organisées selon les diplômes et selon les cycles d'étude (licence, master, doctorat).

Une BDM est mise en place et alimentée par des processus d'extraction, de transformation et de chargement des données issues du système opérationnel que nous ne détaillons pas dans cet article. La figure 2 décrit conceptuellement le schéma en étoile de cette BDM (Golfarelli *et al.,* 1998 ; Ravat *et al.*, 2007 ; 2008). Cette dernière vise à analyser les mesures (moyennes des notes 'Avg_Note', notes maximales 'Max_Note', notes minimales 'Min_Note' et taux d'absentéisme 'Taux_Abs') en fonction de chaque contrôle, de chaque étudiant, de chaque formation et des dates (dimensions).

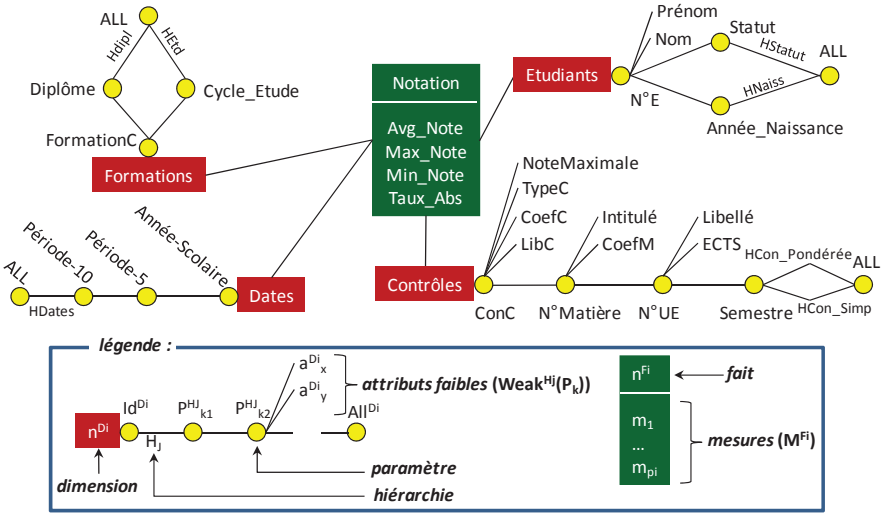


Figure 2. La BDM de l'exemple du jury des diplômes

La dimension 'Contrôles' a deux hiérarchies 'HCon_Simp' et 'HCon_Pondérée'. Chaque hiérarchie correspond à une manière d'analyser les taux d'absentéisme (simple et pondérée). Un contrôle est caractérisé par un code 'ConC', par un numéro de matière 'N°Matière', un numéro d'unité d'enseignement 'N°UE' et un Semestre. Chaque contrôle a un attribut 'NoteMaximale' qui représente la note totale du contrôle. Il a aussi un coefficient 'CoefC' qui représente le taux de la partie de la note totale que le contrôle couvre.

La dimension 'Formations' a deux hiérarchies 'HDipl' et 'HEtd' qui correspondent à l'organisation des formations selon les diplômes et selon les cycles d'étude. Chaque formation est caractérisée par un code 'FormationC', un nom 'Diplôme' et un cycle d'étude 'Cycle_Etude'. La dimension 'Etudiants' a deux hiérarchies 'HStatut' et 'HNaiss' qui correspondent à l'organisation des étudiants selon leurs statuts et leurs années de naissance. Chaque étudiant est caractérisé par un numéro 'N°E', un nom 'Nom', un prénom 'Prénom', une année de naissance 'Année_Naissance' et un statut 'Statut'. Les années scolaires 'Année-Scolaire' de la

dimension ‘Dates’ sont agrégées selon des périodes de cinq ans ‘Période-5’ et des périodes de dix ans ‘Période-10’.

1.2. Illustration du problème

Ce schéma de BDM permet par exemple d’obtenir la moyenne des notes ‘Avg_Note’ d’un étudiant par contrôle (figure 3).

AVG(Avg_Note)		CONTROLES							
		Semetsre	S1						
		N°UE	U1					U2	
		N°Matière	M1		M2			M3	
		ConC/NoteMaximale	M1C1/20	M1C2/10	M2C1/20	M2C2/10	M2C3/10	M3C1/20	M3C2/20
ETUDIANTS	N°E (Nom)								
	E1 (Martin)		14	8	13	7.5	5	13	12
	E2 (Duval)		8	6	15	6.8	6	12	14

Figure 3. TM visualisant la moyenne des étudiants par contrôle

Pour obtenir la moyenne des notes par matière ou par UE dans cet environnement multidimensionnel classique, il suffit d’agréger les moyennes des notes par contrôles conformément à la fonction d’agrégation AVG associée à la mesure ‘Avg_Note’. Or une telle opération donne un résultat incorrect compte tenu des modalités d’examens. En effet, la moyenne des notes par matière est calculée en additionnant les notes des contrôles en tenant compte du coefficient (normalisé) de chaque contrôle ‘CoefC’ (1). La moyenne par UE est calculée en faisant la moyenne des notes des matières en tenant compte des coefficients (non normalisés) des matières (2). De manière analogue, pour obtenir la moyenne par semestre, l’application de la fonction d’agrégation prévue est inappropriée puisque cette moyenne est calculée en prenant en compte les crédits (ECTS) de chaque UE (3).

$$Moyenne\_Matière = \sum Avg\_Note * CoefC \tag{1}$$

$$Moyenne\_UE = \frac{\sum Moyenne\_Matière * CoefM}{\sum CoefM} \tag{2}$$

$$Moyenne\_Semestre = \frac{\sum Moyenne\_UE * ECTS}{\sum ECTS} = \frac{\sum \left(\frac{\sum Moyenne\_Matière * CoefM}{\sum CoefM} \right) * ECTS}{\sum ECTS} = \frac{\sum \left(\frac{\sum (\sum Avg\_Note * CoefC) * CoefM}{\sum CoefM} \right) * ECTS}{\sum ECTS} \tag{3}$$

Les approches classiques qui considèrent une fonction d’agrégation unique pour tous les niveaux d’agrégation modélisés dans le schéma en étoile souffrent de plusieurs limites :

– *la variabilité de la fonction d'agrégation.* Le modèle traditionnel ne donne pas la possibilité d'utiliser des fonctions d'agrégation évoluant avec les niveaux des hiérarchies ou avec les dimensions. Dans l'exemple du jury de diplôme, la fonction d'agrégation des moyennes change avec les niveaux de hiérarchie entre 'ConC', 'N°Matière', 'N°UE' et 'Semestre' ;

– *les lacunes des fonctions de base.* Nous remarquons que, pour agréger les données entre les niveaux de hiérarchie, nous utilisons des fonctions d'agrégation non standard qui utilisent des données autres que les valeurs de la mesure (coefficients 'CoefC', 'CoefM', 'ECTS') ;

– *les contraintes des agrégations.* Dans la littérature (Gray *et al.*, 1996), les fonctions d'agrégation appartiennent à trois catégories différentes. La première correspond aux fonctions *distributives* qui calculent les valeurs agrégées à un niveau de granularité à partir des valeurs déjà agrégées au niveau de granularité directement inférieur (par exemple, la somme - SUM - d'un montant par année peut se calculer à partir de la somme des montants par semestre). La deuxième correspond aux fonctions *algébriques* qui calculent les valeurs agrégées à partir de résultats intermédiaires stockés (par exemple, la moyenne - AVG - d'un montant par année peut se calculer à partir de la somme - SUM - des montants et du nombre - COUNT - des occurrences). Enfin, la troisième correspond aux fonctions *holistiques* qui ne peuvent pas être calculées à partir de résultats intermédiaires. Dans ce cas, il faut calculer les valeurs agrégées à partir des valeurs élémentaires correspondant au niveau de granularité le plus bas (par exemple, RANK). Outre ces catégories de fonctions, des contraintes sur la manière d'opérer le calcul de l'agrégation peuvent exister. Dans notre exemple, comme l'illustre la formule (2), le calcul de la moyenne des notes par UE ne peut être obtenu directement à partir des notes des contrôles ; pour obtenir la moyenne d'une UE, il est nécessaire de calculer les moyennes des notes par matières pour ensuite faire la moyenne pondérée de ces moyennes. De la même façon, formule (3), la moyenne par semestre est nécessairement calculée à partir du calcul de la moyenne par UE.

Notre objectif est donc de proposer un modèle multidimensionnel suffisamment expressif pour supporter ces types d'agrégations. Nous étudions ensuite les conséquences au niveau logique sur les treillis de pré-agrégats (Gray *et al.,* 1996).

L'article est organisé comme suit. La section 3 présente les extensions de notre modèle conceptuel multidimensionnel aux agrégations multiples différenciées. Nous présentons le formalisme graphique de ces extensions. La section 4 décrit le modèle logique en R-OLAP avec ses relations d'optimisation et les impacts de nos extensions sur ce modèle. Nous expérimentons notre proposition dans la section 5 en détaillant notre prototype.

2. Positionnement et contributions des travaux

Il existe classiquement deux approches pour la modélisation des BDM : une approche reposant sur la métaphore du cube de données suivant laquelle la BDM est représentée par des cubes, et une approche dite de modélisation multidimensionnelle où la BDM est décrite par un schéma en étoile ou en constellation (Kimball, 1996).

Nos travaux s'inscrivent dans cette seconde approche. La métaphore du cube souffre d'une modélisation peu expressive (Torlone, 2003) : difficultés à représenter l'organisation hiérarchique des données, à représenter des hypercubes où l'espace multidimensionnel doit être constitué de plus de trois axes d'analyse, à représenter des constellations de faits et de dimensions partagées.

Plusieurs synthèses du domaine (Chaudhuri et Dayal, 1997 ; Vassiliadis *et al.,* 1999 ; Mazón *et al.,* 2009) et d'études comparatives (Gyssens et Lakshmanan, 1997 ; Vassiliadis et Skiadopoulos, 2000 ; Pedersen *et al.,* 2001 ; Abelló *et al*., 2006 ; Lujàn-Mora *et al.,* 2006 ; Ravat *et al.*, 2008 ; Prat et Akoka, 2010 ; Oliveira *et al*., 2011 ; Boulil *et al.,* 2011) sont disponibles dans la littérature scientifique. La plupart des propositions existantes considèrent qu'une mesure est associée à une fonction d'agrégation qui sera utilisée à tous les niveaux d'agrégation modélisés. Cette fonction calcule la même agrégation pour toutes les combinaisons de tous les paramètres modélisés.

Le traitement de l'agrégation des mesures dans l'espace multidimensionnel a évolué (tableau 1). Les travaux de (Gyssens et Lakshmanan, 1997) et (Vassiliadis et Skiadopoulos, 2000) ne précisent pas des fonctions d'agrégation pour les mesures, mais ils laissent la possibilité d'utiliser pour chaque mesure plusieurs fonctions d'agrégation au cours du processus OLAP. Cela donne une grande flexibilité, mais laisse la possibilité de commettre des erreurs en utilisant des fonctions inappropriées. Des travaux (Hurtado, *et al.,* 2002 ; Ghozzi, *et al*., 2003) ont proposé d'intégrer des contraintes dans la modélisation multidimensionnelle afin de palier certaines manipulations invalides, mais ces approches se focalisent sur les dimensions et hiérarchies sans tenir compte des problématiques liées à l'additivité (Gray *et al.*, 1996) des fonctions d'agrégation. Les travaux de Pedersen *et al.* (2001) proposent de lier à chaque mesure un ensemble de fonctions qui ne comprend que les fonctions valides. Néanmoins, chaque fonction est utilisée uniformément pour toutes les dimensions et tous les niveaux des hiérarchies. Des travaux plus récents (Abelló *et al*., 2006) permettent d'utiliser une fonction d'agrégation différente pour chaque dimension, sans donner la possibilité de faire évoluer la fonction avec les niveaux de hiérarchies. Cette limite a été levée par les modèles d'agrégation des travaux les plus récents (Prat et Akoka, 2010) et (Boulil *et al.*, 2011). Ces travaux nous permettent d'associer à chaque mesure, une fonction d'agrégation pour chaque dimension, pour chaque hiérarchie ou pour chaque niveau d'agrégation. Le modèle de (Prat et Akoka, 2010) ne traite que le cas où il y a des fonctions standard (SUM, AVG, MIN, MAX, COUNT). Boulil *et al.* (2011) étendent la proposition à des fonctions non standard, mais ne traitent pas le cas des fonctions d'agrégation non commutatives.

En ce qui concerne les outils commerciaux, « Business Objects » utilise une seule fonction d'agrégation pour chaque mesure. En revanche, l'outil « Analysis Services de Microsoft » offre la possibilité d'appliquer un « Rollup personnalisé » à une hiérarchie de plusieurs façons (Harinath *et al.,* 2009) :

– par l'utilisation des opérateurs unaires qui sont utilisés pour résoudre le problème de l'agrégation sur un type particulier de hiérarchie (hiérarchie d'attributs parent-enfant). Une hiérarchie parent-enfant est construite à partir d'un seul attribut

parent. Un attribut parent décrit une relation de jointure réflexive dans une table de dimension principale ;

– par l'utilisation de scripts MDX, soit directement, soit par l'utilisation de la propriété « CustomRollupColumn » qui indique à une colonne où sont stockés les scripts MDX.

Les deux approches représentent des fonctions d'agrégation mais elles ne sont liées ni à une dimension, ni à une hiérarchie, ni à un niveau d'agrégation. Elles sont liées à un membre (une instance) d'un niveau d'agrégation d'une hiérarchie, c'est-à-dire, à une ligne dans la table de la dimension. Donc, pour appliquer ce « Rollup personnalisé » à un seul niveau d'agrégation il faut le répéter pour toutes les instances de ce niveau. Cela pose un problème de stockage et diminue la performance (Harinath *et al.,* 2009). D'un autre côté, la liaison de « Rollup personnalisé » avec une instance spécifique peut entraîner des difficultés en ce qui concerne la mise à jour des données.

Le tableau 1 montre dans la colonne 'Générale' comment les propositions existantes intègrent les fonctions d'agrégation au cours du processus OLAP durant l'interrogation ou dans le modèle. Il montre également si ces propositions offrent la possibilité de changer la fonction d'agrégation avec les dimensions, les hiérarchies et les niveaux de granularité (colonnes 'Dimension', 'Hiérarchie' et 'Niveau de granularité'). De plus, il présente si les travaux traitent le cas des fonctions non commutatives (colonne 'Non-commutativité') ou le cas des agrégations contraintes, c'est-à-dire lorsque la mesure doit être calculée à partir d'un niveau différent du niveau de base (colonne 'Agrégation contrainte').

Tableau 1. Synthèse des travaux sur les agrégations multidimensionnelles

	Générale	**Agrégation multiple**		**Agrégation différenciée**	Non-commutativité	Agrégation contrainte
		Dimension	Hiérarchie	Niveau de granularité		
Gyssens, 1997	OLAP	-	-	-	-	-
Vassiliadis, 2000	OLAP	-	-	-	-	-
Pedersen, 2001	Modèle	-	-	-	-	-
Abelló, 2006	Modèle	✓	-	-	✓	-
Prat, 2010	Modèle	✓	✓	✓	-	-
Boulil, 2011	Modèle	✓	✓	✓	-	-
Business Objects	Modèle	-	-	-	-	-
Analysis Services de Microsoft	Modèle	-	-	✓ (Instance)	✓	-
Notre modèle	Modèle	✓	✓	✓	✓	✓

Grâce à ce tableau, nous constatons que la possibilité de changer la fonction d'agrégation avec les dimensions, les hiérarchies et les niveaux d'agrégation avait

été traitée (Prat et Akoka, 2010) et (Boulil *et al.*, 2011), mais partiellement car ces propositions ne prennent pas en compte les fonctions non commutatives. Par ailleurs, les travaux prenant en compte les fonctions non commutatives, supportent seulement les fonctions d'agrégation au niveau dimensions (Abelló *et al*., 2006).

Notre objectif est de lever ces limites en développant un modèle conceptuel de représentation des agrégations multidimensionnelles multiples différenciées. Par *multiples* nous signifions qu'une même mesure peut être agrégée par plusieurs fonctions selon les dimensions ou les hiérarchies et par *différenciées* nous indiquons que ces agrégations peuvent varier en fonction du niveau d'agrégation.

A notre connaissance, toutes les propositions existantes supposent qu'il est possible de calculer l'agrégation d'une mesure à partir des niveaux de base. Nous proposons d'ajouter le moyen de traiter le cas où la mesure ne peut pas être calculée à partir du niveau de base (colonne 'Agrégation contrainte'), en utilisant des *contraintes d'agrégation.*

3. Modèle conceptuel de données

3.1. Concepts classiques

Soit $\mathcal{N}$= {n_1, n_2,...} un ensemble fini de noms non redondants, F = {F_1,..., F_n} un ensemble fini de faits, $n \geq 1$, D = {D_1,..., D_m} un ensemble fini de dimensions, $m \geq 2$.

DÉFINITION 1. — Un *fait*, noté F_i, $\forall\, i \in [1..n]$, est défini par (n^{Fi}, M^i).

– $n^{Fi} \in \mathcal{N}$ est le nom identifiant le fait,

– M^i = {m_1,..., m_{pi}} est un ensemble de *mesures*.

On pose $M = \bigcup_{i=1}^{n} M^i$

DÉFINITION 2. — Une *dimension*, notée D_i, $\forall i \in [1..m]$,est définie par (n^{Di},A^i,H^i).

– $n^{Di} \in \mathcal{N}$ est le nom identifiant la dimension,

– A^i = {a^i_1,..., a^i_{ri}}$\cup${Id^i, All^i} est l'ensemble des *attributs de dimension*,

– H^i = {H^i_1,..., H^i_{Si}} est un ensemble de *hiérarchies*.

Les hiérarchies organisent les attributs d'une dimension, appelés paramètres, de la graduation la plus fine (paramètre racine noté Id^i) jusqu'à la graduation la plus générale (paramètre extrémité noté All^i). Ainsi une hiérarchie définit les chemins de navigation valides sur un axe d'analyse.

On pose $A = \bigcup_{i=1}^{m} A^i$ et $H = \bigcup_{i=1}^{m} H^i$

DÉFINITION 3. — Une *hiérarchie*, notée H_j (notation abusive de H^i_j, $\forall i \in [1..m]$, $\forall j \in [1..s_i]$), est définie par (n^{Hj}, P^j, $\prec^{Hj}$, $Weak^{Hj}$).

– $n^{Hj} \in \mathcal{N}$ est le nom identifiant la hiérarchie ;

– $P^j = \{p^j_1,..., p^j_{qj}\}$ est un ensemble d'attributs de la dimension appelés *paramètres*, $P^j \subseteq A^i$;

– $\prec^{Hj} = \{(p^j_x, p^j_y) \mid p^j_x \in P^j \wedge p^j_y \in P^j\}$ est une relation binaire antisymétrique et transitive. Rappelons que l'antisymétrie signifie que $(p^j_{k1} \prec^{Hj} p^j_{k2}) \wedge (p^j_{k2} \prec^{Hj} p^j_{k1}) \Rightarrow p^j_{k1} = p^j_{k2}$ tandis que la transitivité signifie que $(p^j_{k1} \prec^{Hj} p^j_{k2}) \wedge (p^j_{k2} \prec^{Hj} p^j_{k3}) \Rightarrow p^j_{k1} \prec^{Hj} p^j_{k3}$;

– $Weak^{Hj} : P^j \rightarrow 2^{A^i \setminus P^j}$ est une application qui associe à chaque paramètre un ensemble d'attributs de dimension, appelés *attributs faibles*.

On pose $P^i = \bigcup_{j=1}^{Si} P^j$, $P = \bigcup_{i=1}^{m} P^i$ et $W^i = \bigcup_{\forall j \in [1..s_i], \forall k \in [1..q_j]} Weak^{H_j}(p^j_k)$

LEMME 1. — Pour chaque dimension D_i, ses attributs de dimension sont de manière exclusive soit des paramètres, soit des attributs faibles, $P^i \cap W^i = \varnothing$ et $P^i \cup W^i = A^i$.

3.2. Extensions pour les agrégations multiples différenciées

Afin que le modèle multidimensionnel réponde à notre problématique, nous l'enrichissons par les extensions suivantes :

– La première extension concerne les mécanismes d'agrégation.

- **Agrégation générale :** c'est la fonction que l'on utilise pour agréger les valeurs d'une mesure entre n'importe quel paramètre. Cette fonction n'est associée qu'à la mesure sans prendre en compte ni les paramètres, ni les hiérarchies, ni les dimensions. Cette fonction représente la fonction d'agrégation dans le modèle classique.

- **Agrégation multiple dimensionnelle :** c'est la fonction que l'on utilise pour agréger les valeurs d'une mesure entre les paramètres d'une dimension. Cette fonction est associée à une mesure et à une dimension. Il est possible d'associer à une même mesure, plusieurs fonctions d'agrégation différentes selon les dimensions.

- **Agrégation multiple hiérarchique :** c'est la fonction que l'on utilise pour agréger les valeurs d'une mesure entre les paramètres sur une hiérarchie. Cette fonction est associée à une mesure et à une hiérarchie. Il est possible d'associer à une même mesure, plusieurs fonctions d'agrégation, une pour chaque hiérarchie.

- **Agrégation différenciée :** c'est la fonction que l'on utilise pour agréger les valeurs d'une mesure entre deux paramètres (niveaux d'agrégation) d'une hiérarchie. Elle est associée à une mesure et à un paramètre. Cette fonction donne la possibilité d'appliquer une agrégation de manière spécifique à chaque niveau de granularité.

– La deuxième extension concerne **l'ordre d'exécution** des fonctions d'agrégation impliquées dans l'analyse. Il est possible d'avoir plusieurs fonctions d'agrégation différentes sur les dimensions considérées durant une analyse. Ces

fonctions sont généralement non commutatives. Il faut donc pour contrôler la validité des résultats imposer un ordre d'exécution.

– La troisième extension concerne **les contraintes d'agrégation**. Les agrégations ne s'effectuent pas toutes nécessairement de manière uniforme à partir de tous les niveaux inférieurs (contrairement au mécanisme d'agrégation prévu dans les modèles multidimensionnels classiques). Par conséquent, nous introduisons un mécanisme de contrainte sur l'agrégation pour fixer le niveau d'agrégation valide permettant d'obtenir une agrégation supérieure.

Soit $\mathcal{F} = \{f_1, f_2, ...\}$ un ensemble fini de fonctions d'agrégation.

DÉFINITION 4. — Un *schéma multidimensionnel*, noté S, est défini par (F, D, Star, Aggregate).

– $F = \{F_1, ..., F_n\}$ est l'ensemble des faits, si $|F|=1$ alors le schéma multidimensionnel est appelé schéma en étoile alors que si $|F|>1$ alors le schéma est appelé schéma en constellation ;

– $D = \{D_1, ..., D_m\}$ est l'ensemble des dimensions ;

– Star : $F \rightarrow 2^D$ est une fonction qui associe à chaque fait un ensemble de dimensions en fonction desquelles il peut être analysé ;

– Aggregate : $M \rightarrow 2^{N^* \times F \times 2^D \times 2^H \times 2^P \times N^-}$ associe à chaque mesure un ensemble de fonctions d'agrégation. Cette fonction permet de prendre en compte les différents types d'agrégations supportés par notre modèle (générale, multiple dimensionnelle, multiple hiérarchique, différenciée) :

- si $2^D = \emptyset$, $2^H = \emptyset$ et $2^P = \emptyset$, alors la fonction est une fonction d'agrégation générale ;

- si $2^H = \emptyset$ et $2^P = \emptyset$, alors la fonction est une fonction d'agrégation multiple dimensionnelle utilisée pour agréger la mesure sur toute la dimension considérée ;

- si $2^P = \emptyset$, alors la fonction est une fonction d'agrégation multiple hiérarchique utilisée pour agréger la mesure sur toute la hiérarchie considérée ;

- si $2^D \neq \emptyset$, $2^H \neq \emptyset$ et $2^P \neq \emptyset$, alors la fonction est une fonction d'agrégation différenciée utilisée pour agréger la mesure entre le paramètre considéré et celui directement supérieur.

$\mathbb{N}^*$ associe à chaque fonction d'agrégation un numéro d'ordre qui représente la priorité dans l'exécution. La fonction d'agrégation avec l'ordre le plus petit a la priorité la plus élevée. Si les fonctions d'agrégation sont commutatives, alors elles sont du même ordre. Le choix d'un ordre valide dépend des besoins de l'utilisateur. Il peut différer d'un cas à l'autre, même si les fonctions sont les mêmes dans les deux cas. Le modèle permet de fixer l'ordre qui donne un résultat valide pour l'utilisateur.

$\mathbb{N}^-$ sert à contraindre une agrégation en indiquant un niveau d'agrégation spécifique à partir duquel l'agrégation considérée doit se calculer. Une agrégation non contrainte sera associée à 0 tandis qu'une agrégation contrainte sera associée à une valeur négative pour forcer le calcul à partir d'un niveau inférieur choisi par rapport au niveau considéré. En utilisant ces contraintes, nous pouvons surmonter le

troisième problème présenté (les contraintes des agrégations). Ainsi, nous pouvons forcer les calculs des moyennes des notes par 'UE' et 'Semestre' à partir des niveaux 'Matière' et 'UE' respectivement.

LEMME 2. — Les fonctions d'agrégation assurent la *couverture* du schéma multidimensionnel, c'est-à-dire qu'il ne doit pas exister de paramètre (niveaux d'agrégation) auquel il est impossible d'appliquer une fonction d'agrégation.

$\forall\, i \in [1..n], \forall\, m_k \in M^{Fi}, \exists\, f \in \mathcal{F}, \exists\, x_1 \in \mathbb{N}^*, \exists\, x_2 \in \mathbb{N}^-,$

$$\begin{cases} (x_1, f, \{\}, \{\}, \{\}, x_2) \in Aggregate(m_k) \\ \forall\, D_j \in Star(F_i) \,|\, (x_1, f, \{D_j\}, \{\}, \{\}, x_2) \in Aggregate(m_k) \\ \forall\, H_s \in H^j \,|\, (x_1, f, \{D_j\}, \{H_s\}, \{\}, x_2) \in Aggregate(m_k) \\ \forall\, P_q \in P^s \setminus \{All^j\} \,|\, (x_1, f, \{D_j\}, \{H_s\}, \{P_q\}, x_2) \in Aggregate(m_k) \end{cases}$$

La couverture du schéma est réalisée de plusieurs façons par l'utilisation :

– d'une fonction d'agrégation générale,

– d'une fonction d'agrégation multiple dimensionnelle pour chaque dimension, ou une fonction d'agrégation multiple hiérarchique pour chaque hiérarchie,

– d'une fonction d'agrégation différenciée pour chaque niveau d'agrégation,

– par combinaison des fonctions d'agrégation multiples et différenciées où la dimension ou la hiérarchie, qui n'a pas de fonction multiple doit avoir une fonction différenciée pour chaque niveau d'agrégation.

3.3. Formalismes graphiques

L'exemple du jury de délibération des diplômes présenté dans le cas d'étude se définit par (F, D, Star, Aggregate) où

– F = $\{F_{Notation}\}$, le fait est défini par $F_{Notation}$ = ('Notation', {Avg_Note, Max_Note, Min_Note, Taux_Abs}) ;

– D = $\{D_{Contrôles}, D_{Etudiants}, D_{Formations}, D_{Dates}\}$, pour simplifier, nous présentons les définitions de deux dimensions $D_{Contrôles}$ et $D_{Etudiants}$:

1- $D_{Contrôles}$ = ('Contrôles', {ConC, LibC, CoefC, TypeC, NoteMaximale, N°Matière, CoefM, Intitulé, N°UE, ECTS, Libellé, Semestre, $ALL^{Contrôles}$}, $\{H_{HCon\_Pondérée}, H_{HCon\_Simp}\}$) avec :

- $H_{HCon\_Pondérée}$ = ('HCon_Pondérée', {ConC, N°Matière, N°UE, Semestre, $ALL^{Contrôles}$}, {(ConC, N°Matière), (N°Matière, N°UE), (N°UE, Semestre), (Semestre, $ALL^{Contrôles}$)}, {(ConC, {LibC, CoefC, TypeC, NoteMaximale}), (N°Matière, {CoefM, Intitulé}), (N°UE, {ECTS, Libellé})}) ; et

- H_{HCon_Simp} = ('HCon_Simp', {ConC, N°Matière, N°UE, Semestre, $ALL^{Contrôles}$}, {(ConC, N°Matière), (N°Matière, N°UE), (N°UE, Semestre), (Semestre, $ALL^{Contrôles}$)}, {(ConC, {LibC, CoefC, TypeC, NoteMaximale}), (N°Matière, {CoefM, Intitulé}), (N°UE, {ECTS, Libellé})}).

2- $D_{Etudiants}$ = (‘Etudiants’, {N°E, Nom, Prénom, Statut, Année_Naissance, $ALL^{Etudiants}$}, {$H_{HStatut}$, H_{HNaiss}}) avec

- $H_{HStatut}$ = (‘HStatut’, {N°E, Statut, $ALL^{Etudiants}$}, {(N°E, Statut), (Statut, $ALL^{Etudiants}$)}, {(N°E, {Nom, Prénom})}) ; et

- H_{HNaiss} = (‘HNaiss’, {N°E, Année_Naissance, $ALL^{Etudiants}$}, {(N°E, Année_Naissance), (Année_Naissance, $ALL^{Etudiants}$)}, {(N°E, {Nom, Prénom})}).

– Star : $F \rightarrow 2^D$ |

Star($F_{Notation}$) = {$D_{Contrôles}$, $D_{Etudiants}$, $D_{Formations}$, D_{Dates}},

– Aggregate : $M \rightarrow 2^{N^* \times F \times 2^D \times 2^H \times 2^P \times N^-}$

Aggregate(Avg_Note) = {(2, AVG(Avg_Note), {}, {}, {}, 0)[1],
(1, SUM_W(Avg_Note, CoefC), {Contrôles}, {HCon_Pondérée}, {ConC}, 0),
(1, AVG_W(Avg_Note, CoefM), {Contrôles}, {HCon_Pondérée},
{N°Matière}, -1)[2],
(1, AVG_W(Avg_Note, ECTS), {Contrôles}, {HCon_Pondérée}, {N°UE},-1),
(1, AVG(Avg_Note), {Contrôles}, {HCon_Pondérée}, {Semestre}, -1)}
(1, SUM_W(Avg_Note, CoefC), {Contrôles}, {HCon_Simp},{ConC}, 0),
(1, AVG_W(Avg_Note, CoefM), {Contrôles},{HCon_Simp},{N°Matière},-1),
(1, AVG_W(Avg_Note, ECTS), {Contrôles}, {HCon_Simp}, {N°UE}, -1),
(1, AVG(Avg_Note), {Contrôles}, {HCon_Simp}, {Semestre}, -1)}

Aggregate(Max_Note) = {(2, MAX(Max_Note), {}, {}, {}, 0),
(1, SUM_W(Avg_Note, CoefC), {Contrôles}, {HCon_Pondérée}, {ConC}, 0),
(1, AVG_W(Avg_Note, CoefM), {Contrôles}, {HCon_Pondérée},
{N°Matière}, -1),
(1, AVG_W(Avg_Note, ECTS), {Contrôles}, {HCon_Pondérée}, {N°UE},-1),
(1, AVG(Avg_Note), {Contrôles}, {HCon_Pondérée}, {Semestre}, -1)}
(1, SUM_W(Avg_Note, CoefC), {Contrôles}, {HCon_Simp},{ConC}, 0),
(1, AVG_W(Avg_Note, CoefM), {Contrôles},{HCon_Simp},{N°Matière},-1),
(1, AVG_W(Avg_Note, ECTS), {Contrôles}, {HCon_Simp}, {N°UE}, -1),
(1, AVG(Avg_Note), {Contrôles}, {HCon_Simp}, {Semestre}, -1)}

Aggregate(Taux_Abs) = {(2, RATE(Taux_Abs), {}, {}, {}, 0),
(1, RATE(Taux_Abs), {Contrôles}, {HCon_Pondérée}, {ConC}, 0),
(1, AVG_W(Taux_Abs, CoefM), {Contrôles}, {HCon_Pondérée},
{N°Matière}, -1),
(1, AVG_W(Taux_Abs, ECTS), {Contrôles}, {HCon_Pondérée}, {N°UE},-1),

1. Il n’y a pas de contrainte sur l’agrégation.
2. Les valeurs sont agrégées à partir des valeurs agrégées au niveau directement inférieur.

(1, AVG(Taux_Abs), {Contrôles}, {HCon_Pondérée}, {Semestre}, -1)}

Aggregate(Min_Note) est comparable à Aggregate(Max_Note) sauf qu'il utilise la fonction MIN au lieu de MAX.

La fonction SUM_W(X,Y) prend deux entrées numériques. Elle retourne la somme des valeurs X pondérées par Y. Autrement dit, la somme pondérée SUM_W(X,Y) = $\sum(X \times Y)$. Par exemple, si les contrôles dans figure 3 : M2C1, M2C2 et M2C3 de la matière M2 ont les coefficients (CoefC) 0.5, 0.6 et 0.4 consécutivement, alors la note de cette matière de l'étudiant E1(Martin) est SUM_W(Avg_Note, CoefC) = 13×0.5 + 7.5×0.6 + 5×0.4 = 13. La fonction AVG_W(X, Y) prend deux entrées numériques. Elle retourne la moyenne des valeurs X pondérées par Y. Autrement dit, la moyenne pondérée

$$\text{Avg\_W(X, Y)} = \frac{\sum(X \times Y)}{\sum Y}$$

La fonction RATE(X) prend une entrée numérique. Elle retourne le pourcentage des valeurs X qui ne sont pas zéro

$$\text{RATE(X)} = \frac{\text{COUNT}(X \neq 0)}{\text{COUNT}(X)} \times 100$$

On utilise cette fonction parce que dans notre exemple si l'étudiant n'était pas absent au contrôle, la valeur de la mesure 'Taux_Abs' serait zéro.

En ce qui concerne les mesures 'Avg_Note', 'Max_Note' et 'Min_Note', elles sont agrégées d'une manière identique sur les deux hiérarchies de la dimension 'Contrôles'. En outre, l'agrégation des mesures 'Max_Note' et 'Min_Note' sur la dimension 'Contrôles' s'appuie sur l'agrégation de 'Avg_Note'. Cela apparaît clairement à travers l'utilisation de la mesure 'Avg_Note' dans les fonctions d'agrégation de 'Max_Note' et 'Min_Note'. Pour connaître la note maximale 'Max_Note' d'une matière ou d'une UE pour un groupe d'étudiants, on doit d'abord calculer la note 'Avg_Note' de cette matière ou de cette UE pour chaque étudiant, ensuite on détermine parmi les notes obtenues, la note maximale.

Associé aux définitions formelles, nous introduisons un formalisme graphique facilitant la compréhension du schéma de la BDM. Ces représentations graphiques sont de deux niveaux.

3.3.1. Schéma structurel

Le schéma structurel permet de visualiser globalement les éléments structurels (faits, dimensions et hiérarchies) du schéma multidimensionnel en masquant les mécanismes d'agrégation. Cette vue globale est obtenue à partir de la fonction Star. La BDM décrite dans l'exemple précédent, se représente graphiquement conformément à la figure 2.

3.3.2. Schéma d'agrégation

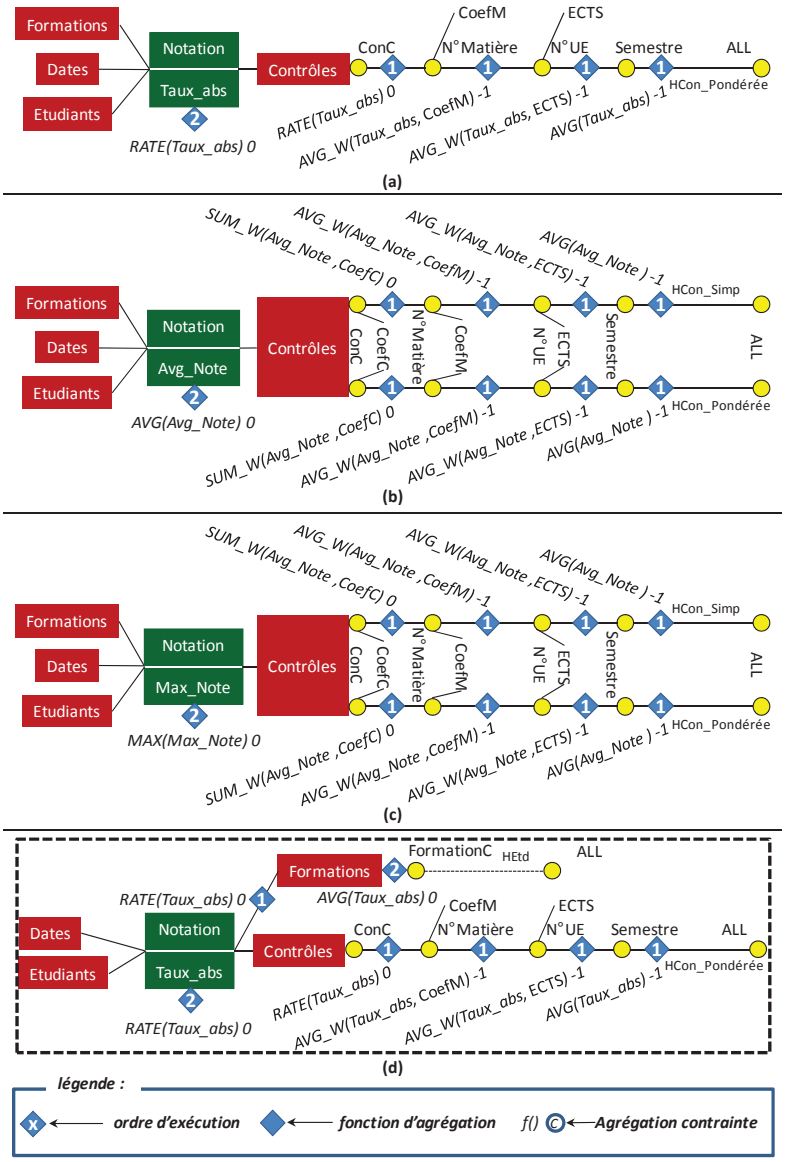


Figure 4. Schémas d'agrégation

Pour chaque mesure $m_k \in F_i$, un schéma d'agrégation peut être obtenu grâce à la fonction Aggregate. Cette vision détaille les mécanismes d'agrégation impliqués durant une analyse portant sur la mesure considérée en simplifiant les éléments

structurels autant que possible. Ce schéma est une extension de nos précédents travaux (Hassan *et al.*, 2012a ; 2012b).

La figure 4 décrit les trois schémas d'agrégation (a, b, c) correspondant aux mesures 'Taux_Abs', 'Avg_Note' et 'Max_Note' (nous ne présentons pas celui de la mesure 'Min_Note'). Comme le montre la figure 4, les hiérarchies sont présentées en version divisée contrairement au schéma structurel (figure 2) où elles sont présentées en version compacte ; par exemple, les hiérarchies 'HCon_Simp', 'HCon_Pondérée' dans la figure 4 (b et c).

Les fonctions d'agrégation sont modélisées par des losanges. Nous utilisons le même symbole (losange) pour toutes les fonctions pour ne pas surcharger le schéma. Chaque losange indique également l'ordre d'exécution et la contrainte d'agrégation possible. Les positions des losanges dépendent du type de fonction :

– la fonction générale est représentée par un losange sur le bord du fait ;

– la fonction d'agrégation multiple dimensionnelle est localisée sur l'arc reliant le fait à la dimension (ce cas n'est pas utilisé dans notre exemple) ;

– la fonction d'agrégation multiple hiérarchique est localisée en bas de la hiérarchie (ce cas n'est pas utilisé dans notre exemple) ;

– la fonction d'agrégation différenciée étiquette l'arc reliant deux paramètres.

La figure 4 (d) présente les fonctions d'agrégation multiple (dimensionnelle et hiérarchique) et la commutativité dans l'ordre d'exécution. Nous supposons qu'il y a une fonction multiple dimensionnelle Rate(Taux_abs) sur la dimension 'Formations'. Cette fonction est commutative avec les fonctions de la dimension 'Contrôles'. Nous supposons également qu'il y a une fonction multiple hiérarchique Avg(Taux_abs) sur la hiérarchie 'HEtd'. Cette fonction est commutative par rapport à la fonction générale.

Les agrégations avec des contraintes fixées à -1 sont calculées à partir du niveau directement inférieur ; par exemple, la moyenne de notes 'Avg_Note' par semestre est calculée à partir des moyennes des notes par UE. Dans l'hypothèse où nous aurions choisi de calculer cette moyenne par semestre à partir des notes par contrôle, la contrainte aurait été fixée à -3.

4. Modèle logique R-OLAP

4.1. Approche classique

4.1.1. Etoile R-OLAP

L'implantation courante repose sur l'approche dite R-OLAP. Elle consiste à utiliser l'approche relationnelle pour implanter les schémas multidimensionnels (Kimball, 1996). Cette approche procure de nombreux avantages : la réutilisation des mécanismes de gestion des données éprouvés et la capacité à gérer des volumes de données importants. Dans le contexte relationnel, les structures multidimensionnelles conceptuelles sont donc traduites au niveau logique sous la

forme de relations (Kimball, 1996). Appliqué à notre exemple, le schéma R-OLAP en étoile est le suivant :

CONTRÔLES (**ConC**, LibC, CoefC, TypeC, NoteMaximale, N°Matière, CoefM, Intitulé, N°UE, ECTS, Libellé, Semestre)

ETUDIANTS (**N°E**, Nom, Prénom, Statut, Année_Naissance)

DATES (**Année-Scolaire**, Période-5, Période-10)

FORMATIONS (**FormationC**, Diplôme, Cycle_Etude)

NOTATION (**ConC#, N°E#, Année-Scolaire#, FormationC#**, Avg_Note, Max_Note, Min_Note, Taux_Abs)

4.1.2. Etoile optimisée

La modélisation conceptuelle permet de structurer hiérarchiquement les graduations (paramètres) des axes d'analyses. Ces hiérarchies sont exploitées pour optimiser la BDM. Cette optimisation consiste à compléter le schéma par un ensemble de relations pré-calculant les agrégations nécessaires aux décideurs lors de leurs interrogations et analyses OLAP. Classiquement, les pré-agrégations sont modélisées par un treillis de pré-agrégats (Gray *et al.*, 1996 ; Chaudhuri et Dayal, 1997) où chaque nœud représente un pré-agrégat et chaque arc représente le chemin des calculs d'agrégation. Lorsque la fonction d'agrégation utilisée est distributive ou algébrique (Gray *et al.*, 1996), un agrégat est calculable directement à partir de l'agrégat inférieur direct, tandis que dans le cas d'une agrégation holistique (Gray *et al.*, 1996), l'agrégat se calcule en cheminant jusqu'aux relations de base.

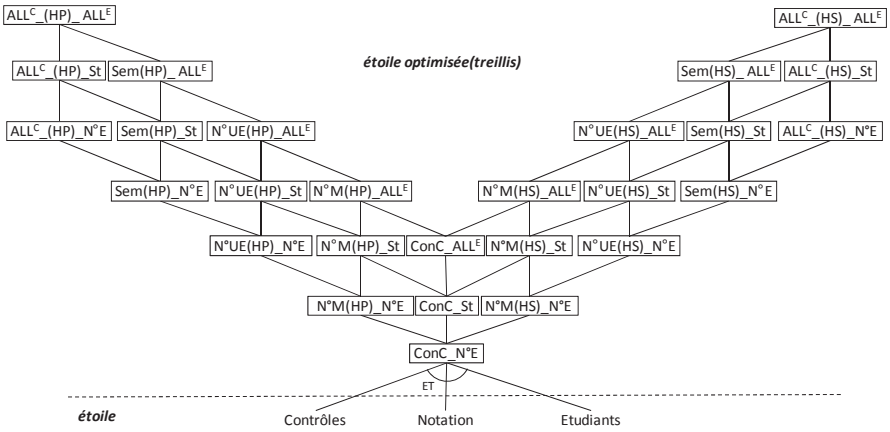


Figure 5. Treillis classique[3]

3. Nous utilisons les abréviations (N°M pour N°Matière, Sem pour Semestre, HS pour HCon_Simp, HP pour HCon_Pondérée, ALL^{E} pour $ALL^{Etudiants}$, ALL^{C} pour $ALL^{Contrôles}$).

Pour éviter que le treillis soit trop complexe, nous simplifions l'exemple du jury. Nous ne prenons en compte que deux dimensions :

– 'Contrôles' avec ses deux hiérarchies 'HCon_Simp' et 'HCon_Pondérée',

– 'Etudiants' avec une seule hiérarchie 'HStatut'.

La figure 5 représente le treillis de pré-agrégats de la mesure 'Taux_Abs'. Chaque nœud représente une relation. Dans ces relations, l'attribut 'Taux_Abs' représente le taux d'absentéisme calculé par la fonction d'agrégation RATE. Il s'agit, ici, d'un cas de fonction algébrique. Dans cette approche classique, contrairement à notre proposition, une fonction d'agrégation unique est utilisée dans l'ensemble du treillis pour la mesure 'Taux_Abs'.

4.2. Extensions par les agrégations multiples et différenciées

L'expressivité que nous avons introduite dans le modèle conceptuel peut être exploitée au niveau du treillis.

4.2.1. Typage des arcs

Comme l'illustre la figure 6, les fonctions d'agrégation multiples et différenciées impliquent l'utilisation d'agrégations différentes sur chaque arc du treillis. La possibilité pour une même mesure d'utiliser différentes fonctions d'agrégation selon les paramètres nécessite de typer les arcs du treillis. Ce typage permet d'indiquer entre deux nœuds la fonction d'agrégation correspondante. Lorsque plusieurs chemins sont possibles, le chemin le moins coûteux est préféré. La fonction de coût, que nous ne détaillons pas, privilégie les temps de calcul les plus efficaces (Kotidis *et al.,* 1999). Nous pouvons néanmoins remarquer que l'utilisation de fonctions d'agrégation différentes sur chaque arc rend l'estimation du coût plus complexe que dans les treillis habituels.

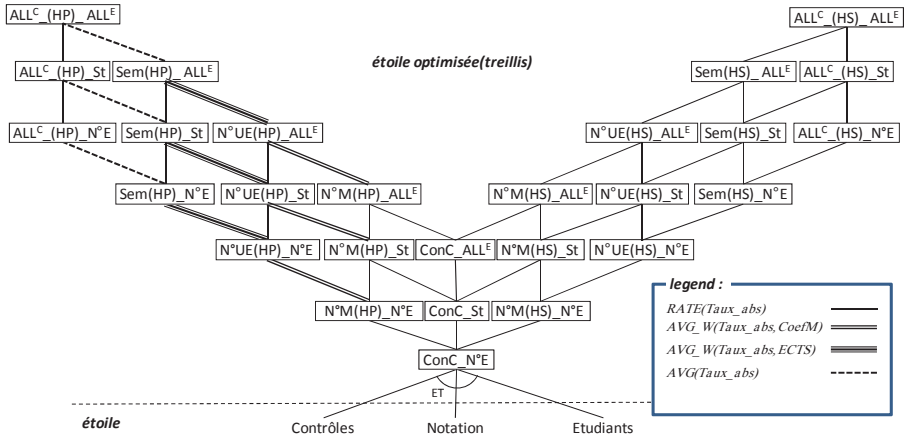


Figure 6. Treillis avec des arcs typés

4.2.2. Blocage de la transitivité

Les contraintes associées aux fonctions d'agrégation ont une autre répercussion sur le treillis : les arcs obtenus à partir de ces contraintes imposent de calculer un nœud à partir d'un nœud précis. Il est alors interdit de calculer un nœud supérieur par transitivité des nœuds inférieurs comme cela est classiquement possible. Ainsi les chemins de calcul sont bloqués dès qu'un arc contraint intervient. Par exemple, le nœud 'Sem(HS)_N°E' est calculable à partir du nœud inférieur direct 'N°UE(HS)_N°E', par transitivité, il est également calculable à partir du nœud inférieur 'N°M(HS)_N°E'. Par contre, l'arc contraint issu de la contrainte de la fonction 'AVG_W(Taux_Abs, ECTS)' qui opère sur l'arc ('N°UE(HP)_N°E', 'Sem(HP)_N°E') bloque la transitivité des calculs. Donc, le nœud 'Sem(HP)_N°E' est calculable à partir du nœud inférieur direct 'N°UE(HP)_N°E' mais il ne l'est pas par transitivité à partir du nœud inférieur 'N°M(HP)_N°E'.

De la même manière, le changement des ordres d'exécution ou des fonctions entre les arcs provoque un blocage de la transitivité. Autrement dit, si tous les arcs précédents pour un arc spécifique correspondent à des fonctions ou des ordres d'exécution différents, alors cet arc est non transitif. Par exemple, l'arc ('N°UE(HP)_N°E', 'N°UE(HP)_St') correspond à la fonction 'RATE(Taux_abs)' avec un ordre d'exécution de valeur 2. Cet arc a un seul arc précédent ('N°M(HP)_N°E', 'N°UE(HP)_N°E') qui correspond à la fonction 'AVG_W(Taux_abs, CoefM)' avec un ordre d'exécution de valeur 1. A cause de la différence entre les ordres d'exécution, l'arc ('N°UE(HP)_N°E', 'N°UE(HP)_St') est non transitif. Donc, le nœud 'N°UE(HP)_ALL^{E}' est calculable par transitivité à partir du nœud 'N°UE(HP)_N°E' mais il n'est pas calculable par transitivité à partir du nœud 'N°M(HP)_N°E', parce que le schéma d'agrégation figure 4 impose à l'ordre d'exécution de calculer d'abord les taux d'absentéisme en fonction de la dimension 'Contrôles' (nœud 'N°UE(HP)_N°E') pour pouvoir calculer ensuite les taux d'absentéisme en fonction de la dimension 'Etudiants' ('N°UE(HP)_ALL^{E}').

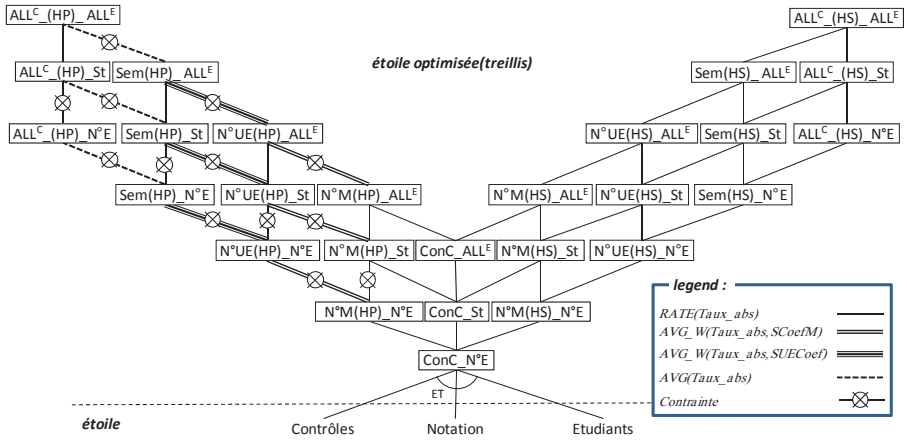


Figure 7. Treillis avec des arcs contraints

La figure 7 décrit le treillis dans lequel les arcs étiquetés par un cercle barré sont obtenus soit à partir des contraintes d'agrégation, soit à partir du changement d'ordre d'exécution ou de fonction d'agrégation entre les arcs.

4.2.3. Elagage du treillis

Certains chemins ou arcs sont invalides et peuvent ainsi être éliminés pour réduire le treillis (figure 8). Cet élagage est rendu possible par l'utilisation de l'ordre d'exécution. Dans notre exemple, on ne peut pas appliquer la fonction 'AVG_W(Taux_Abs, CoefM)' après la fonction 'RATE(Taux_abs)' sur la dimension 'Etudiants' car cela donnerait un résultat invalide. Ainsi, pour obtenir le nœud 'N°UE(HP)_St' (le taux d'absentéisme par Statut d'étudiant et par UE sur la hiérarchie 'HCon_Pondérée'), on ne peut pas le calculer à partir du nœud 'N°M(HP)_St' (le taux d'absentéisme par Statut d'étudiant et par matière sur la hiérarchie 'HCon_Pondérée'). L'arc entre 'N°M(HP)_St' et 'N°UE(HP)_St' peut donc être supprimé.

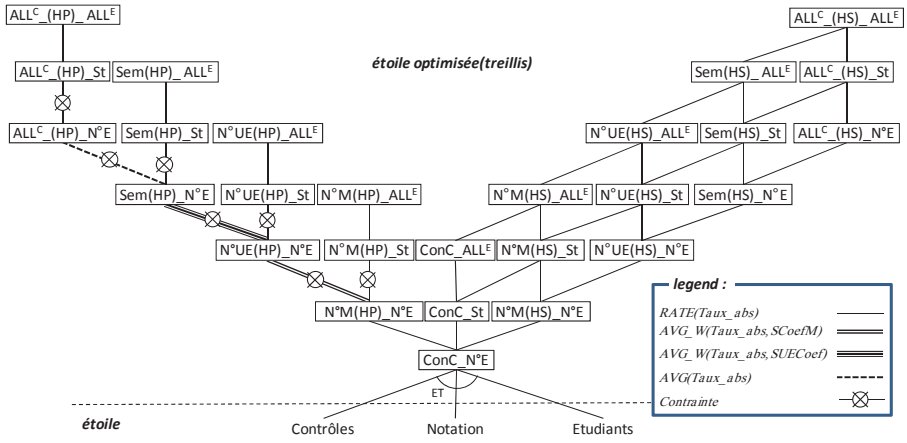


Figure 8. Treillis de pré-agrégats contrôlés

4.2.4. Modification d'arcs

Dans notre modèle, nous avons proposé un mécanisme de contrainte sur l'agrégation pour fixer le niveau d'agrégation valide à partir duquel se calcule une agrégation supérieure. Ce niveau valide n'est pas forcément le niveau directement inférieur. Nous exprimons ce cas lorsque nous utilisons une valeur de la contrainte différente de 0 (l'agrégation est calculable à partir de n'importe quel niveau inférieur) ou -1 (l'agrégation est calculable uniquement à partir du niveau directement inférieur). Les contraintes différentes de 0 et -1 induisent des changements de chemins dans le treillis. Dans notre exemple, le taux d'absentéisme par semestre sur la hiérarchie 'HCon_Pondérée' est calculée à partir des taux d'absentéisme par UE (valeur de contrainte = -1). Dans l'hypothèse où nous aurions

choisi de calculer ce taux par semestre à partir des taux par contrôles, la contrainte aurait été fixée à -3 et le treillis correspondrait à la figure 9.

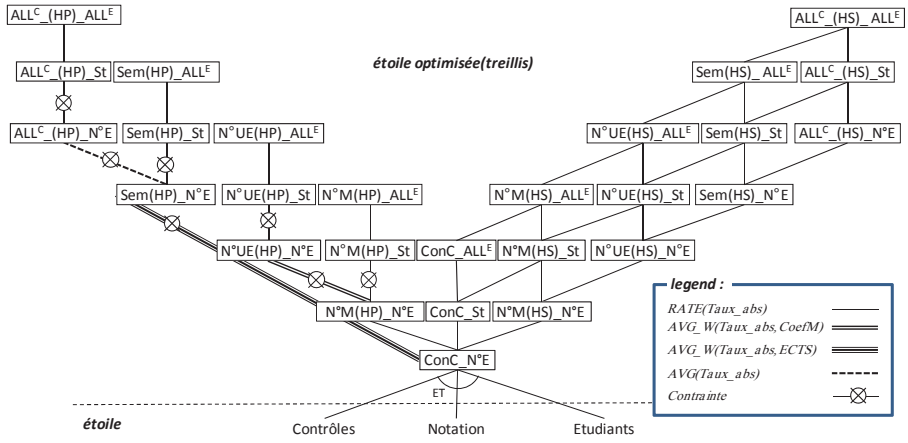


Figure 9. Treillis avec contrainte = -3

Si le système ne s'appuie pas sur la stratégie de treillis de pré-agrégats, d'autres stratégies sont possibles. Il peut notamment recalculer systématiquement le résultat d'une requête à partir des données de base (données stockées dans le fiat et les dimensions). Dans ce cas, le système doit utiliser des requêtes imbriquées dont la complexité dépend du nombre de fonctions nécessaires pour effectuer l'analyse.

5. Validation

Nous développons un prototype, appelé *OLAP-Multi-Functions*, permettant de concevoir une BDM à agrégations multiples et différenciées, ainsi que de superviser les manipulations OLAP effectuées par un analyste. Dans cette section, nous décrivons l'architecture modulaire du prototype, puis nous détaillons le méta-schéma et le générateur des requêtes sur lequel est basé notre prototype. Ce dernier nous sert de plateforme expérimentale dont nous détaillons quelques expérimentations menées.

5.1. Prototype

Notre proposition est mise en œuvre dans le prototype *OLAP-Multi-Functions.* Nous utilisons Java 7 au-dessus du SGBD Oracle 11g. Il permet la définition d'une constellation à agrégations multiples et différenciées, ainsi que la visualisation et l'interrogation des données multidimensionnelles.

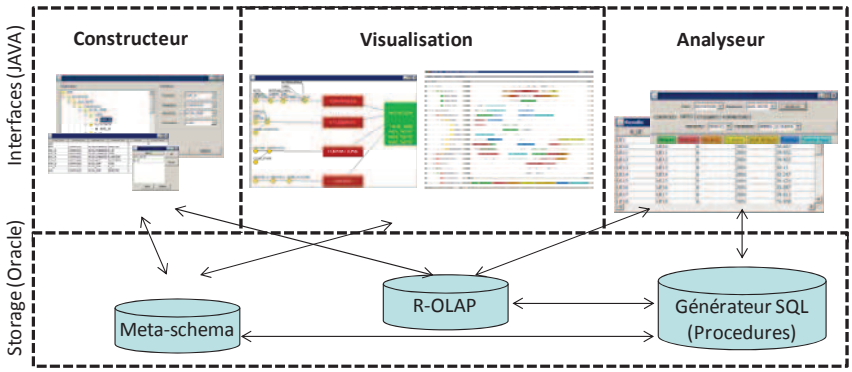


Figure 10. Architecture du prototype

La fonctionnalité principale de l'*OLAP-Multi-Functions* (figure 10) est de visualiser et de faciliter l'intégration des fonctions d'agrégation dans le modèle multidimensionnel. Il repose sur un ensemble d'interfaces graphiques (Constructeur) permettant de définir les quatre types de fonctions d'agrégation (générale, multiple dimensionnelle, multiple hiérarchique, différenciée), leur ordre d'exécution et les éventuelles contraintes d'agrégation. Le schéma structurel est visualisé sous la forme d'un graphe en constellation basé sur des formalismes graphiques des faits, des dimensions et des hiérarchies introduits dans (Ravat, *et al*., 2007, 2008). Les différents schémas d'agrégation sont quant à eux visualisés sous la forme d'un graphe hyperbolique. Pour l'interrogation, l'analyste sélectionne dans les graphes la mesure et les niveaux d'agrégation souhaités. Après validation, *OLAP-Multi-Functions* calcule automatiquement le résultat qui est présenté sous forme d'une table R-OLAP.

5.1.1. Méta-schéma

Les fonctions d'agrégation sont décrites dans un méta-schéma. Ce méta-schéma décrit également les structures du schéma multidimensionnel (faits, dimensions et hiérarchies). La figure 11 montre ce méta-schéma en diagramme de classes UML. Les classes blanches/claires décrivent le modèle classique (F, D, Star) et les classes bleues/grisées décrivent les mécanismes d'agrégations multiples et différenciées *(Aggregate)*.

Selon ce méta-schéma, un fait se compose de mesures. Il peut être analysé selon plusieurs dimensions. Chaque dimension comprend des hiérarchies, des paramètres et des attributs faibles. Les paramètres peuvent appartenir à plusieurs hiérarchies et ils sont ordonnés en *niveaux d'agrégation*. Chaque *niveau* peut être associé à un ensemble d'attributs faibles. Une mesure peut être agrégée par plusieurs fonctions d'agrégation (classe *Fonction*). La classe *Fonction* possède un attribut *contrainte* qui est utilisé pour forcer l'agrégation en indiquant le *niveau* inférieur à partir duquel l'agrégation considérée doit être calculée. L'attribut *ordre d'exécution* est utilisé

pour ordonner l'exécution des fonctions d'agrégation non commutatives. Une fonction prend au moins une entrée. Cette entrée est soit une mesure soit un paramètre soit un attribut faible.

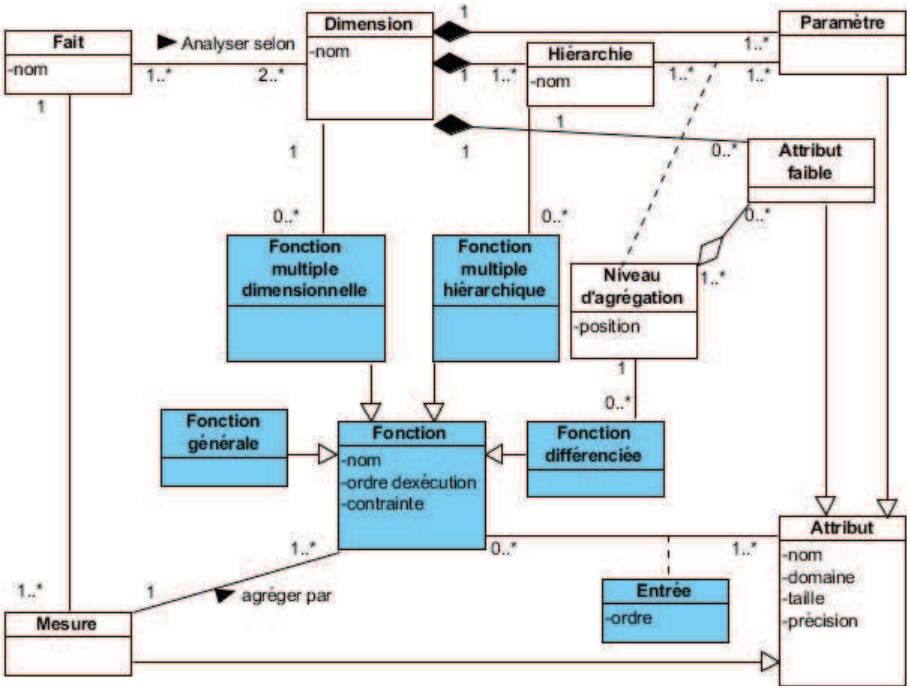


Figure 11. Méta-schéma

Quatre types de fonctions d'agrégation héritent de la classe *Fonction* :

– la fonction générale est liée uniquement à une mesure (une mesure ayant au maximum une fonction d'agrégation générale) ;

– la fonction multiple dimensionnelle est liée à une mesure et une dimension ;

– la fonction multiple hiérarchique est liée à une mesure et une hiérarchie ;

– la fonction différenciée est liée à un niveau d'agrégation.

Toutefois, chaque dimension, chaque hiérarchie et chaque niveau d'agrégation peut avoir plusieurs fonctions d'agrégation ; une pour chaque mesure différente.

5.1.2. Générateur des requêtes SQL

Pour superviser les analyses, le prototype dispose d'un générateur de requêtes SQL. L'analyste paramètre le calcul d'agrégation qu'il souhaite réaliser : l'utilisateur doit préciser la mesure et les niveaux d'agrégation souhaités.

Le générateur traduit les interactions en générant un script SQL exécutable dans le contexte d'implantation R-OLAP. Le processus de génération comprend quatre étapes, comme l'illustre la figure 12 en diagramme BPMN *(Business Process Modeling Notation)*.

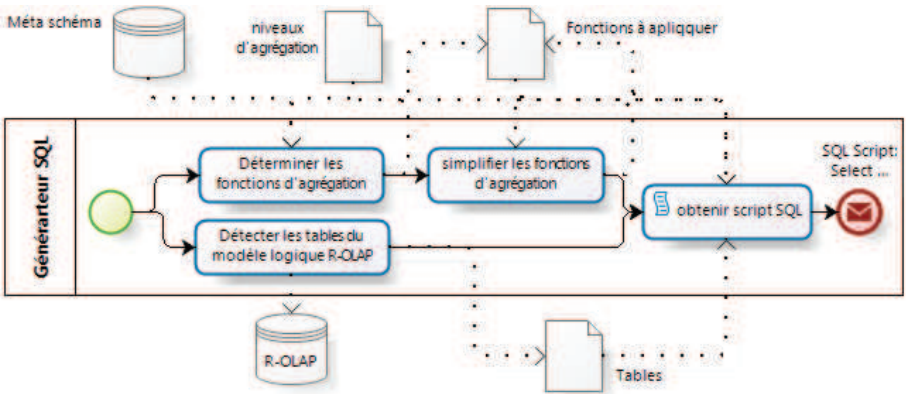


Figure 12. Générateur des requêtes SQL représenté en BPMN

[1] Détecter les tables du modèle logique R-OLAP : cette étape identifie les tables utilisées pour stocker les données pour l'analyse ;

[2] Déterminer les fonctions d'agrégation : à partir du méta-schéma et des niveaux d'agrégation demandés, cette étape identifie les fonctions d'agrégation à appliquer pour calculer les données de l'analyse ;

[3] Simplifier les fonctions d'agrégation : cette étape consiste à détecter d'éventuels calculs redondants, c'est-à-dire une répétition inutile ou invalide d'une fonction d'agrégation ;

[4] Obtenir script SQL : à partir du méta-schéma et des étapes précédentes, cette étape génère la requête SQL finale. Elle l'envoie au SGBD qui calcule la requête et restitue les données au prototype.

5.2. Expérimentations

Les premières expérimentations que nous avons réalisées visent à étudier les conséquences de nos propositions sur le temps d'exécution des requêtes d'interrogations et d'analyses OLAP.

La figure 13 (b) montre les temps d'exécution (seconde) de trois requêtes.

– la première agrège les moyennes des notes au niveau matière. Elle repose sur une seule fonction d'agrégation (comme dans le modèle classique) 'SUM_W' ;

– la deuxième requête agrège les moyennes des notes au niveau UE. Elle repose sur deux fonctions d'agrégation 'SUM_W' et 'AVG_W' ;

– la troisième requête agrège les moyennes des notes au niveau semestre. Elle nécessite trois fonctions d'agrégation 'SUM_W' et deux fois 'AVG_W'.

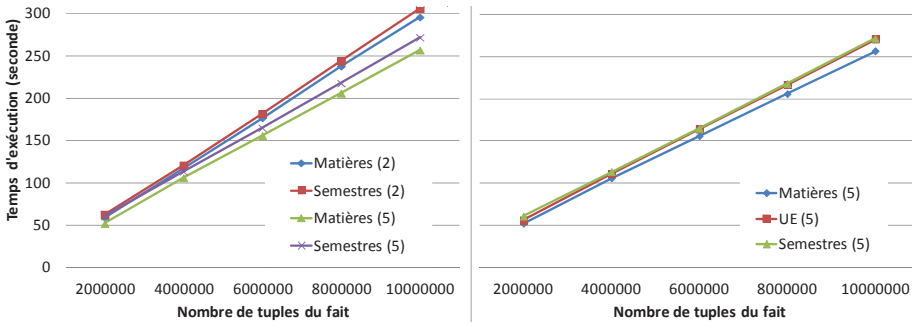
La taille de groupement pour les trois niveaux est de valeur 5 : chaque instance d'un niveau supérieur correspond à cinq instances de niveau inférieur (par exemple, chaque matière a cinq contrôles). Le temps d'exécution des requêtes augmente régulièrement en fonction du nombre de tuples. La distance entre les courbes de première et de deuxième requête représente le temps supplémentaire demandé pour appliquer la deuxième fonction. Ce temps est d'environ 5 % du temps total pour exécuter la requête. Le temps supplémentaire pour appliquer la troisième fonction semble être non remarquable. En réalité, ce phénomène est lié au volume des données qui décroît avec les fonctions précédemment appliquées. Ainsi lors du calcul de la troisième fonction, le volume des données est proportionnellement fortement réduit par rapport au volume initialement impliqué.

La figure 13 (a) étudie la relation entre le temps d'exécution et le nombre de tuples en fonction de la taille des regroupements. Nous entendons par taille de regroupements le nombre de valeurs d'un paramètre inférieur qui sont regroupées en une valeur d'un paramètre supérieur. Nous présentons le temps d'exécution de quatre requêtes :

– deux au niveau matière (une avec des tailles de groupement à 2 et l'autre à 5) qui utilisent une fonction d'agrégation ; et

– deux autres au niveau semestre (une avec des tailles de regroupement à 2 et l'autre à 5) qui utilisent trois fonctions d'agrégation.

Nous remarquons que le temps d'exécution des requêtes avec une taille de groupement à 5 est moindre que celui des requêtes avec une taille de groupement à 2. Nous remarquons également que le temps d'exécution des requêtes semble principalement influencé par la taille des regroupements. Ainsi, la requête avec une taille de groupement à 2 et une seule fonction d'agrégation (Matière(2)) est plus coûteuse en temps de calcul que la requête avec une taille de groupement à 5 malgré trois fonctions d'agrégation (Semestre(5)). La taille des groupements apparaît comme avoir un impact primordial sur le temps d'exécution.



(a) (b)

Figure 13. Expérimentations

6. Conclusion

Cet article définit un nouveau modèle conceptuel pour représenter les données multidimensionnelles. Ce modèle est suffisamment expressif pour permettre au concepteur de spécifier des agrégations multiples et différenciées. Ce modèle permet la combinaison d'une mesure avec différentes fonctions d'agrégation en fonction des dimensions, des hiérarchies et des paramètres utilisés. En outre, le modèle permet de contrôler la validité des calculs des fonctions. Les contraintes d'agrégation définissent le niveau à partir duquel l'agrégation doit être calculée. L'ordre d'exécution définit l'ordre nécessaire entre les fonctions d'agrégation non commutatives. Ce modèle s'appuie sur des formalismes graphiques à deux niveaux : un schéma structurel décrivant les structures multidimensionnelles en masquant la complexité des agrégations et des schémas d'agrégation détaillant les mécanismes d'agrégation liés à chaque mesure. En outre, au niveau relationnel, le schéma R-OLAP peut être optimisé par un treillis de pré-agrégats contrôlé. Par contre, notre modèle ne permet pas de spécifier une fonction d'agrégation pour une instance d'un paramètre comme « Analysis Services de Microsoft ». En effet, l'utilisation de plusieurs fonctions pour effectuer une analyse, peut augmenter significativement le temps nécessaire pour l'exécuter.

Nous envisageons de poursuivre nos travaux en revisitant les algorithmes de calcul des pré-agrégats en les adaptant à notre modélisation et en étudiant les effets des changements dans le treillis contrôlé lors de la sélection de nœuds pour améliorer les performances. Nous envisageons également de poursuivre ces travaux par l'étude des opérateurs OLAP appliqués à notre modèle. Nous souhaitons notamment étudier de manière expérimentale la relation entre la complexité de l'analyse et la taille des regroupements de données.

Bibliographie

Abelló A., Samos J., Saltor F. (2006). YAM2: A multidimensional conceptual model extending UML. *Information Systems*, vol. 31, n° 6, p. 541-567.

Chaudhuri S., Dayal U. (1997). An Overview of Data Warehousing and OLAP Technology. *SIGMOD Record*, vol. 26, n° 1, p. 65-74.

Ghozzi F., Ravat F., Teste O., Zurfluh G. (2003). Constraints and Multidimensional Databases. *5th International Conference on Enterprise Information Systems, ICEIS'03*, p. 104-111, Angers, France.

Golfarelli M., Maio D., Rizzi S. (1998). Conceptual Design of Data Warehouses from E/R Schemes. *HICSS*, vol. 7, p. 334-343.

Gray J., Bosworth A., Layman A., Pirahesh H. (1996). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. *ICDE*, p. 152-159.

Gyssens M., Lakshmanan L. V. S. (1997). A Foundation for Multi-Dimensional Databases. *VLDB*, p. 106-115.

Harinath S., Zare R., Meenakshisundaram S., Carroll M., Guang-Yeu Lee D. (2009). Professional Microsoft SQL Server Analysis Services 2008 with MDX.

Hassan A., Ravat F., Teste O., Tournier R., Zurfluh G. (2012). Agrégations multiples différenciées dans les bases de données multidimensionnelles. *30e congrès INFormatique des ORganisations et Systèmes d'Information et de Décision (INFORSID'12),* p. 447-462, Montpellier, France.

Hassan A., Ravat F., Teste O., Tournier R., Zurfluh G. (2012). Differentiated Multiple Aggregations in Multidimensional Databases. *14th International Conference on Data Warehousing and Knowledge Discovery (DAWAK'12),* p. 93-104, Vienna (Austria). doi:10.1007/978-3-642-32584-7_8.

Hurtado C., Mendelzon A. (2002). OLAP Dimension Constraints. *21st ACM Symposium on Principles of Database Systems, PODS'02,* Madison, USA.

Jaecksch B., Lehner W. (2011). The Planning OLAP Model - A Multidimensional Model with Planning Support. *DaWaK*, p. 14-25.

Kimball R. (1996). *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses.* John Wiley & Sons, USA.

Kotidis Y., Roussopoulos N. (1999). DynaMat: A Dynamic View Management System for Data Warehouses. *SIGMOD,* p. 371-382.

Luján-Mora S., Trujillo J., Song I. Y. (2006). A UML profile for multidimensional modeling in data warehouses. *Data & knowledge Engineering 59,* p. 725-769.

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

Oliveira R., Rodrigues F., Martins P., Moura J. P. (2011). Extending the Dimensional Templates Approach to Integrate Complex Multidimensional Design Concepts. *DaWaK*, LNCS, vol. 6862, p. 26-38.

Pedersen T., Jensen C., Dyreson C. (2001). A foundation for capturing and querying complex multidimensional data. *Information Systems,* vol. 26*,* n° 5, p. 383-423.

Ravat F., Teste O., Tournier, R., Zurfluh G. (2008). Algebraic and graphic languages for OLAP manipulations. *International Journal of Data Warehousing and Mining, IGI Publishing, D. Taniar,* vol. 4, n° 1, p.17-46. doi: 10.4018/jdwm.2008010102.

Ravat F., Teste O., Tournier R., Zurfluh G. (2007). Graphical Querying of Multidimensional Databases. *11th East-European Conference on Advances in Databases and Information Systems, ADBIS'07*, p.298-313, Bulgarie. doi: 10.1007/978-3-540-75185-4_22.

Torlone R. (2003). Conceptual Multidimensional Models. *Multidimensional Databases: Problems and Solutions.* IGI Publishing Group, p. 69-90.

Vassiliadis P., Simitsis A., Skiadopoulos S. (2002). Modeling ETL activities as graphs*. DMDW*, p. 52-61.

Vassiliadis P., Sellis T. K. (1999). A Survey of Logical Models for OLAP Databases. *SIGMOD Record*, vol. 28, n° 4, p. 64-69.