Vers une intégration de la connaissance dans l'analyse exploratoire des cubes OLAP

Towards an Integration of Knowledge in OLAP Cubes Exploratory Analysis

Sami Naouali

LARIM, Université du Québec en Outaouais,

sami.naouali@uqo.ca

Résumé

Cet article propose une approche pour l'intégration de la connaissance dans les entrepôts de données et en particulier dans le processus d'analyse et d'exploration des cubes de données. Cette proposition permet d'enrichir l'exploration des cubes OLAP en faisant intervenir la connaissance extraite à partir de ces cubes et intégrée au même niveau que les données multidimensionnelles. Un modèle physique supportant les données et les connaissances est par la suite proposé ainsi qu'un ensemble d'opérateurs pour l'analyse multidimensionnelle et l'extraction de connaissances. Une interface utilisateur est implémentée et permet d'interagir avec l'utilisateur ainsi que de visualiser le cube intégrant les données aussi bien que la connaissance. Pour illustrer l'approche proposée et montrer l'intérêt d'un tel enrichissement des systèmes décisionnels, une application réelle traitant de l'analyse du comportement des utilisateurs naviguant sur un serveur Web est présentée.

Abstract

This paper presents a general framework allowing a tight coupling between data warehousing and data mining capabilities in a uniform and integrated way. The proposed approach consists in integrating knowledge into data warehouses and especially in OLAP cubes exploration. This integration aims to make the OLAP process guided both by multidimensional data and knowledge. A multidimensional structure is then proposed to support data and knowledge. We propose also a set of analytical operators and multidimensional data mining operators. To illustrate the proposed approach, a real OLAP cube built to understand the behaviour of users navigating on the Web is proposed and used to show by concrete examples the interest of such extension of data warehouses.

1.Introduction

La mise en place d'entrepôts de données a permis la construction de bases de données multidimensionnelles supportant l'analyse et la compréhension de phénomènes cachés dans les bases de données opérationnelles. Le processus OLAP *(On Line Analytical Processing)* d'analyse en ligne des données multidimensionnelles de l'entrepôt s'appuie sur une structure particulière de l'entrepôt, un langage d'interrogation et une visualisation 3D de l'hypercube de données OLAP (Codd, Codd et Salley, 1993, Kimbal, 1996).

Par ailleurs, l'extraction de connaissances est devenue populaire grâce aux développements scientifiques et techniques dans plusieurs champs de recherche tels que l'algorithmique, l'apprentissage automatique, les statistiques, les bases de données, etc. À cela s'ajoute un développement technologique permettant le stockage et les échanges de gros volumes de données à des coûts raisonnables. Mais, c'est principalement l'engouement des industriels pour l'extraction de connaissances qui a permis le développement de solutions diverses pour des domaines d'application variés tels que la médecine, le marketing, la finance, etc.

Cependant, l'analyse exploratoire et interactive des cubes OLAP reste focalisée sur les données multidimensionnelles. Ainsi, seules les données sont visualisées dans les cubes, et même les opérations de manipulation des cubes ne traitent que les données. De plus, l'approche OLAP n'intègre pas les avancées réalisées par la communauté l'extraction de connaissances. La connaissance n'est pas représentée explicitement dans l'entrepôt et aucun opérateur classique ne permet de la manipuler lors de la visualisation et de l'exploration des cubes. A cela s'ajoute le fait que l'analyse exploratoire des cubes OLAP est principalement basée sur les intuitions et les connaissances acquises par l'utilisateur qui, par conséquent, doit être expert du domaine. Cette dépendance entre la connaissance cachée dans les cubes OLAP et les hypothèses et les intuitions des utilisateurs de ces cubes peut mener deux experts à interpréter différemment le même cube OLAP.

D'autre part, l'extraction de connaissances à partir de données connaît des limites dues au volume et à la qualité des données considérées, au temps de calcul ainsi qu'au volume et à la qualité de la connaissance extraite. Ainsi, les travaux de la communauté l'extraction de connaissances continuent, en grande partie, à se focaliser sur le développement d'algorithmes efficaces pour faire face au grand volume et à la mauvaise qualité des données. Aussi, bien que la visualisation prend de plus en plus d'importance dans les recherches actuelles, elle est en général perçue comme un processus sans un réel retour sur les données ou une remise en cause des méthodes et du processus d'extraction de connaissances utilisé. A cela s'ajoute le fait que rares sont les approches capables d'extraire de la connaissance à partir d'une véritable base de données en utilisant son schéma, son catalogue, les procédures stockées disponibles, etc.

Nous pensons que les travaux réalisés dans le cadre de l'entreposage de données (*i.e., data warehousing*), de l'analyse OLAP et de l'extraction de connaissances peuvent se renforcer mutuellement pour permettre une exploration des cubes basée sur les données et les connaissances multitype (statistiques, règles, arbres de décisions, etc.).

En réponse à ces constats, nous présentons dans cet article un système OLAP intégrant, en plus des données multidimensionnelles, des métadonnées ainsi que des connaissances extraites à partir de ces données multidimensionnelles. Notre contribution se situe au niveau de la modélisation et de la manipulation :

Modélisation : enrichir l'entrepôt de données avec des métadonnées (décrivant ces données) et connaissances de différentes natures.

Manipulation : définir un ensemble d'opérateurs structurés selon trois niveaux :

- **Niveau 1 :** Définition d'un modèle couvrant la totalité des opérateurs classiques OLAP.
- **Niveau 2 :** Enrichissement avec de nouveaux opérateurs de manipulation multidimensionnelle des données, des opérateurs d'extraction d'informations et d'extraction de connaissances.
- **Niveau 3 :** Ajout d'opérateurs de visualisation, d'interaction et d'exploration conjointe des données et connaissances. Par conséquent, l'approche proposée conduit à une redéfinition de la notion de l'hypercube de données OLAP qui inclut non seulement les données, mais aussi la connaissance (règles, mesures statistiques, etc.).

Cet article procède comme suit. Un aperçu sur les travaux en rapport avec cette problématique est fourni dans la section 2 suivi dans les sections 3, 4 et 5 respectivement par la proposition d'une structure multidimensionnelle pour supporter le modèle d'entrepôt enrichi par la connaissance, une brève description des approches proposées pour l'extraction et l'intégration de la connaissance dans les cubes de données, et la proposition d'un ensemble d'opérateurs pour la manipulation des cubes OLAP. Avant de conclure la proposition avec des perspectives d'extension, la section 6 présente une application du système proposé sur des données réelles extraites à partir d'un serveur Web pour l'analyse et la compréhension du comportement des utilisateurs ayant navigué sur ce serveur.

2. Travaux connexes et motivations

De nombreux travaux ont été réalisés dans la thématique de l'extraction de connaissances à partir des entrepôts de données (Han et Kamber, 2001). Nous citons parmi les approches proposées la génération d'arbres de décision et de règles d'association à partir des données multidimensionnelles (Han, Fu, Wang et al., 1996), la découverte d'exceptions (écarts significatifs entre la valeur observée et la valeur estimée, i.e., *outliers*) (Barnett et Lewis, 1994) (Knorr, Ng et Tucakov, 2000), la génération des *cubegrades* (Imielinski, Khachiyan et Abdulghani, 2002) (généralisation des règles d'association pour l'étude de l'impact de la manipulation des cubes de données sur les mesures), l'analyse des tendances dans les cubes OLAP (Dong, Han, Lam et al., 2001) et l'exploration des cubes guidée par la découverte (Sarawagi, Agrawal et Megiddo, 1998).

D'autres travaux ont été proposés non pas pour l'extraction de connaissances à partir des entrepôts de données mais pour l'optimisation du calcul d'agrégats, c'est à dire *iceberg*. Ainsi, Xin *et al.* (Xin, Han et Wah, 2003) se sont focalisés sur cet aspect et ont proposé une approche pour le calcul d'agrégats selon plusieurs dimensions simultanément. Dans ce même objectif, Ross *et al.* (Ross et Srivastava, 1997) ont proposé de partitionner le cube de données en plusieurs partitions pouvant tenir en mémoire centrale. Dans ce même esprit, Li *et al.* (Li, Han et Gonzalez, 2004) ont proposé de réduire la dimensionnalité d'un cube OLAP en le partitionnant en plusieurs sous-cubes (fragments) disjoints. Les identifiants des tuples (faits) du cube initial servent par la suite à maintenir la relation entre le cube initial et les sous-cubes dérivés, et ce dans le but de pouvoir régénérer à tout moment le cube initial ou pour répondre à des requêtes complexes à partir de plusieurs sous-cubes.

Toujours à des fins d'optimisation, des travaux ont été proposés pour l'approximation des cubes OLAP. Les auteurs dans (Babcock, Chauhuri et Das, 2003) se basent pour cela sur des techniques d'échantillonnage. Chakrabarti *et al.* (Chakrabarti, Garofalakis, Rastogi et al., 2001) proposent une approche d'approximation des réponses aux requêtes par ondelettes (*wavelets*). Shanmugasundaram *et al.* (Shanmugasundaram, Fayyad et Bradley, 1999) se basent sur une estimation de la densité de probabilité des données pour construire une représentation compacte du cube de données supportant des requêtes d'agrégation. Dans ce même esprit, Vitter *et al.* (Vitter et Wang, 1999) utilisent le principe des ondelettes pour la compression des cubes de données éparses ainsi que pour fournir des réponses approximatives à des requêtes de calcul d'agrégats. Lakshmanan *et al.* (Lakshmanan, Pei et Han, 2002) proposent une approche, *Quotient Cube*, pour générer un résumé du contenu sémantique d'un cube OLAP en s'appuyant sur le principe de partitionnement du cube en classes d'équivalence regroupant les cellules avec les mêmes valeurs d'agrégats. Ces classes peuvent par la suite remplacer les cellules pour aboutir à un cube plus compact. Toutefois, les structures obtenues sont peu compactes et non adaptées aux mises à jour des données. En réponse à cette défaillance, Lakshmanan *et al.* (Lakshmanan, Pai et Zhao, 2003) ont proposé l'approche *QC-Tree* permettant l'extraction de cubes intéressants résumant les données du cube initial tout en réservant sa sémantique. Dans ce même ordre d'idées, Feng *et al.* (Feng, Agrawal, Abbadi et al., 2003) proposent la technique *Range Cube* permettant la compression d'un cube en se basant sur la corrélation qui existe entre ses cellules et qui permet d'aboutir à un cube plus compact et moins coûteux en terme d'espace de stockage et de temps de réponse aux requêtes qui lui sont destinées. D'autres travaux ont été destinés à la matérialisation partielle des cubes de données. Ainsi Barbara et al. (Barbara et Sullinvan, 1997) se basent sur une description (incomplète mais suffisante) d'un cube initial pour en produire des vues matérialisées correspondant à l'essentiel de ses données.

Par la suite, l'objectif de cet article est d'intégrer la connaissance qu'on peut extraire du cube OLAP au même niveau que les données multidimensionnelles d'une façon uniforme et homogène. Le but est qu'à tout moment de l'exploration des cubes OLAP, les données multidimensionnelles, les métadonnées et les connaissances extraites sont manipulées, explorées et visualisées conjointement pour une exploration des cubes OLAP guidée non seulement par les données, mais également par la connaissance et surtout par les interactions entre les données et les connaissances.

3. Structure multidimensionnelle des données

Nous proposons dans cette section une structure multidimensionnelle permettant l'enrichissement de l'entrepôt de données par de la connaissance. Cette structure est un cadre général permettant une réelle intégration des données multidimensionnelles et connaissances extraites à partir de ces données. Elle est présentée plus en détail dans (Naouali, 2004)

Une dimension de l'entrepôt est un axe sémantique permettant la représentation des données selon plusieurs perspectives dans le but de faciliter et d'enrichir leur analyse.

Soit DIM = $\{dim_1,\ldots, dim_n\}$ un ensemble de *n* dimensions directement connectées à la table des faits. À chacune de ces dimensions correspond un ensemble d'attributs qu'on note $A_{dim\,i}$ chacun de ces attributs admet un ensemble de valeurs possibles. Considérons alors A = $\{a_1,a_2,\ldots\}$ l'ensemble global des attributs de toutes les dimensions et dom_{ai} le domaine d'un

attribut quelconque . Par la suite *f* est une fonction informative attribuant à chaque dimension l'ensemble des attributs qui lui correspondent:

$$f : DIM \rightarrow P(A)$$
$$\dim_i \mapsto \vec{a}_{\dim i}$$

avec *P*(*A*) désigne la fonction "partie de".

Chaque dimension est établie selon une certaine hiérarchie permettant d'organiser ses attributs selon plusieurs niveaux d'abstraction ce qui lui offre un aspect granulaire. C'est la raison pour laquelle à chaque attribut d'une dimension quelconque correspond une table accessible à partir de la table dimension en question, et c'est cette séquence de tables liées qui décrit la hiérarchie de cette même dimension[1] . Une dimension peut avoir plus d'une hiérarchie, chacune d'entre elles organise différemment ses attributs.

On définit une expression de chemin comme étant le moyen d'accéder à un certain attribut (soit *a*) d'une dimension quelconque (soit dim_i) en parcourant une ou plusieurs tables reliées à cette même dimension (soient $t_j,\ldots,t_k$) et décrivant sa hiérarchie. Une expression de chemin permet par la suite de changer le niveau d'abstraction courant de la dimension dim_i et est notée comme suit : $dim_i \rightarrow t_j \rightarrow \ldots \rightarrow t_k \rightarrow a$ avec $a \in \vec{a}_i$. Toutes les expressions de chemin commençant par dim_i appartiennent au chemin générique

$\xrightarrow{dim\,i}$.

Une mesure représente généralement une valeur agrégée selon un ensemble de dimensions, chacune de ces dimensions est donnée selon un niveau quelconque d'abstraction (ce qui explique l'utilisation d'une expression de chemin pour désigner une dimension quelconque) :

$$\text{mesure} = \text{agrégation} \left(\xrightarrow{dim\,1},\ldots, \xrightarrow{dim\,n} \right) \quad (1)$$

avec $dim_1,\ldots,dim_n$ des tables dimensions directement reliées à la table des faits en question. Cette fonction d'agrégation permettra de recalculer les agrégats à la suite du changement du niveau hiérarchique d'une ou de plusieurs dimensions du cube de données (section 5.1.1).

La construction d'un cube fait appel à un ensemble de tables:

- Une table des faits (TF). C'est le lieu de connexion de *n* dimensions, et contenant *m* mesures. Elle admet des coordonnées correspondant aux différentes valeurs (membres ou encore modalités) des dimensions du cube et un contenu correspondant aux valeurs prises par ses mesures. Une table des faits peut être la table initiale (table des faits principale contenant les données correspondant au plus bas niveau hiérarchique de chacune des dimensions du cube) ou une table générée à la suite de l'application d'une (ou plusieurs) opération(s) de manipulation du cube et/ou d'extraction de connaissances.
- Les tables dimensions sont associées à chaque dimension référencée dans la table des faits initiale à laquelle elles sont directement connectées.
- Les tables reliées aux tables dimensions sont utilisées pour représenter la décomposition de la hiérarchie de chacune de ces dimensions en sous niveaux de granularité.

La structure multidimensionnelle proposée suit ainsi un schéma en flocons de neige. Une cellule de ce cube correspond par la suite à un enregistrement de la table des faits (i.e., *fait*) et est représenté comme suit :

fait (TF) : [coordonnées : [$v_1, \ldots, v_n$*], contenu : [g*($v_1, \ldots, v_n$*)]]*

avec:

- $vi = (\xrightarrow{dimi} : di)$ où $di \in dom_{di} \in \delta_{dim_i}$
- *g* : fonction informative retournant le contenu du fait dont les coordonnées sont passées en paramètre. Cette fonction retourne une des valeurs suivantes:
- " !∃" si la cellule correspondante n'existe pas.
- "? " si la cellule correspondante est manquante.
- La (ou les) valeur(s) correspondante(s) sinon[2] .

Nous tenons à mentionner ici que les cellules manquantes ou inexistantes sont prises en compte dans la structure proposée afin de pouvoir introduire des approches pour leur prédiction ou leur approximation (modèles log-linéaires, théorie des ensembles approximatifs, etc). Pour ces mêmes raisons, et toujours dans le but d'enrichir l'entrepôt de données par de nouvelles approches d'extraction de connaissances et d'analyse de données, nous représentons un groupement de faits (sous-cube) en délimitant leurs coordonnées entre crochets comme suit:

$$G^{j}_{FT} = [w1, \ldots, wn]$$

avec *j* l'identifiant du groupe et $w_{i:1,\ldots,n}$ correspond à un ou plusieurs membres de la $i^{ème}$ dimension (soit dim_i). w_i prend une des formes suivantes:

- v_k: une valeur unique correspondant au $k^{ième}$ membre de dim_i
- $v_{k,m}$: le $k^{ième}$ et le $m^{ième}$ membre de dim_i
- $v_{k\text{-}m}$: du $k^{ième}$ au $m^{ième}$ membre de dim_i
- $v_?$: tous les membres de dim_i.

4. Intégration de la connaissance dans les entrepôts de données

Nous présentons dans cette section deux approches pour l'extraction et l'intégration de la connaissance dans les entrepôts en tenant compte de la structure multidimensionnelle proposée ci-dessus. La première approche est une extension des capacités d'analyse OLAP vers l'approximation de concepts dans les cubes de données selon la théorie des ensembles approximatifs, *i.e., Rough Set Theory* (RST). La seconde approche consiste en la découverte de similarités entre les cellules d'un cube de données à des fins de segmentation, *i.e., clustering*. Cette approche est basée sur le calcul des ensembles fréquents à partir des cubes de données. Le présent document ne comporte pas des détails techniques (notamment d'évaluation de performances) de ces deux approches. On se contente juste de montrer leurs intérêts dans le cube de données ainsi que dans le processus d'analyse OLAP.

4.1 Approximation des cubes de données

Nous présentons dans cette section une approche permettant l'approximation des réponses aux requêtes OLAP soumises à un entrepôt de données. Cette approche est basée sur une adaptation de la théorie des ensembles approximatifs aux données multidimensionnelles et offre de nouvelles possibilités d'exploration et d'extraction de connaissances à partir des cubes OLAP. Puisque les entrepôts proviennent généralement de multiples sources de données hétérogènes et peu fiables, les utilisateurs peuvent accepter une réponse approximative à une requête OLAP et donc une perte d'information.

Cette approche permet également l'intégration d'outils d'approximation dans les entrepôts de données dans le but de produire de nouvelles vues que l'on peut par la suite analyser et explorer en faisant appel à des opérateurs OLAP et/ou des algorithmes d'extraction de connaissances. Cette intégration permet par la suite à l'utilisateur de travailler soit en mode restreint en utilisant une approximation basse[3] du cube OLAP, ou en mode relâché en utilisant une approximation haute de ce dernier[4]. Le premier mode est utile dans le cas où la réponse à la requête est volumineuse, permettant ainsi à l'utilisateur de focaliser son attention sur un ensemble réduit de cellules (faits) fortement similaires. Le second est utile dans le cas d'une requête retournant un ensemble vide ou réduit de cellules, permettant ainsi de relâcher les contraintes de la requête afin d'élargir le volume du résultat.

Dans cet objectif, nous avons tout d'abord intégré les principes des ensembles approximatifs dans le contexte multidimensionnel des données dans le but de fournir des réponses approximatives aux requêtes et définir des concepts (partitions du cube en sous-ensemble de faits) en réponse aux requêtes des utilisateurs, ensuite nous avons proposé un enrichissement des techniques OLAP avec de nouveaux opérateurs apportant plus de flexibilité lors des interactions entre l'utilisateur et les entrepôts de données, et enfin, nous avons défini des vues matérialisées de données pour encapsuler et exploiter les réponses aux opérateurs d'approximation à des fins d'extraction de connaissances (segmentation des cubes et calcul des règles d'association). En conséquence, l'approximation concerne non seulement la réponse aux requêtes mais également le résultat d'extraction de connaissances. Pour plus de détails sur cette approche, le lecteur peut se référer à (Naouali, 2004,Naouali et Missaoui, 2005).

4.2 Découverte de relations entre les cellules dans les cubes OLAP

Cette approche propose l'enrichissement de l'entrepôt de données en permettant la découverte et la mise en évidence de relations de similarité et de partage de propriétés communes entre les cellules du cube OLAP. Le but étant de visualiser et par la suite de pouvoir explorer ces liens simultanément avec les données multidimensionnelles elles-mêmes. Cette approche est basée sur la découverte des ensembles fréquents à partir des cubes de données qui a été communément utilisée comme une première étape dans un processus de calcul de règles d'association. Notre objectif est par la suite, en plus du calcul des règles d'association, d'explorer cette connaissance en la visualisant dans le cube de données et en permettant la génération de nouvelles vues matérialisées très particulières dont l'exploration peut conduire à des aspects d'analyse plus avancés que ce que permet les systèmes OLAP classiques. Notons également que notre but n'est pas de calculer la liste exhaustive des ensembles fréquents, seulement une approximation de cet ensemble complet est calculée, puis intégrée dans les cubes de données. Quant aux règles d'association, elles sont calculées et sauvegardées indépendamment du schéma de l'entrepôt et selon la demande de l'utilisateur.

L'approche proposée considère en entrée la table des faits et comprend essentiellement les étapes suivantes: En premier la sélection et la transformation de données à partir de l'entrepôt et en second lieu la génération des ensembles fréquents à partir de ces données. Comme on s'intéresse dans cette proposition à la connaissance exprimée par le partage d'ensembles fréquents entre les cellules d'un cube, nous ajoutons à ce processus la troisième et dernière étape qui consiste en la découverte de liens sémantiques entre les cellules du cube de données. Ces liens sont alors basés sur les ensembles fréquents que se partagent les cellules du cube.

L'objectif de la première étape du processus est la restructuration de la table des faits. Elle consiste à attribuer à chaque modalité de chacune des dimensions un item dans la nouvelle table des faits en ne gardant que ceux dont la fréquence est supérieure à un certain seuil minimum prédéfini dans le but d'aboutir à une représentation qui soit dense. Un fait de la table ainsi prétraitée est représenté par N bits (N étant la somme des cardinalités des dimensions du cube initial moins le nombre d'items supprimés lors du prétraitement) avec la valeur 1 si l'item en question est présent et 0 est absent. Un tel fait est par la suite considéré comme un *chromosome* ce qui nous permet l'application du principe des algorithmes génétiques dans le but d'aboutir à une liste non exhaustive des ensembles fréquents du moment que le calcul de la liste exhaustive se révèle NP-complet (Ganter et Wille, 1999). L'approche adoptée pour la découverte des ensembles fréquents est basée sur les notions de frontière et de pseudo-frontière introduites par Zaki *et al.* (Zaki, Parthasarathy, Ogihara et al., 1997).

Frontière : La frontière est l'ensemble des itemsets fréquents maximaux. On dit qu'un itemset fréquent est maximal s'il n'existe pas un autre itemset fréquent qui le contient. Tout itemset fréquent maximal devient rare (non fréquent) si on lui rajoute un nouvel item, c'est d'ailleurs ce qui explique l'utilisation du terme frontière car c'est la limite pour qu'un itemset (appartenant à cette frontière) reste fréquent.

Pseudo-frontière : Une pseudo-frontière est un ensemble d'itemsets fréquents potentiellement maximaux. Un itemset fréquent est potentiellement maximal si, à un moment précis de la découverte, aucun autre itemset fréquent l'incluant n'a été découvert. Les pseudo-frontières sont donc des approximations de la frontière réelle. Ce sont, en d'autres termes, ce qu'on a pu trouver comme itemsets maximaux à un moment donné du processus de recherche de la frontière.

Le but du processus génétique est par la suite de calculer une pseudo-frontière qui s'approche le maximum possible de la frontière réelle. A chaque itération du processus génétique, une nouvelle pseudo-frontière est calculée et évaluée. Le processus est arrêté systématiquement dès que les performances de la découverte deviennent obsolètes, et l'ensemble total des itemsets fréquents est par la suite obtenu en générant tous les sous-itemsets possibles à partir des itemsets de la pseudo-frontière finale[5] . La performance de la découverte est calculée par un paramètre η selon :

$$\eta = \frac{FD}{FC + RC} \qquad (2)$$

où FD est l'ensemble des itemsets fréquents découverts (inclus les itemsets maximaux calculés ainsi que les sous-itemsets générés à partir de ces itemsets maximaux), FC est l'ensemble des itemsets fréquents calculés[6] et RC est l'ensemble des itemsets non fréquents

(rares) calculés. Un algorithme classique, comme APRIORI (Agrawal et Srikant, 1994), donne $\eta<1$, voire $\eta<<1$, ce qui signifie que sur le nombre total d'itemsets calculés[7] , beaucoup s'avèrent non fréquents. Notre approche vise à réduire cet inconvénient et faire tendre η vers 1.

La survie d'un chromosome quelconque d'une génération à une autre est déterminée moyennant une fonction de qualité, *i.e., fitness,* calculée selon :

$$fitness(C) = \frac{\text{support}(C) \times \alpha^{longueur(C)}}{age(C)}, \alpha = \frac{1}{\sqrt[n]{p(X_1) \times ... \times p(X_n)}} \qquad (3)$$

où *support(C), age(C)* et *longueur(C)* sont des fonctions déterminant le support, l'age et le nombre d'items de *C*. *N* représente le nombre total d'items de la table des faits, $p(X_i)$ la fréquence d'apparition de l'item X_i dans la table des faits et α est un facteur constant égal à l'inverse de la moyenne géométrique des fréquences de tous les items de la table des faits.

Les expérimentations effectuées ont montré que la découverte se stabilise au delà de 80 à 90% du nombre total des ensembles fréquents ce qui est suffisant pour avoir une approximation de cet ensemble en temps raisonnable.

Une fois les ensembles fréquents calculés, ils sont utilisés pour la découverte de liens entre les cellules du cube de données. Deux cellules c_i et c_j quelconques sont connectées si et seulement si les ensembles fréquents que vérifie chacune de ces cellules vérifient la relation φ définie par :

$$c_i \xi c_j \Leftrightarrow \varphi(F(c_i), F(c_j)) \qquad (4)$$

où ξ exprime la liaison sémantique entre c_i et c_j tandis que *F* retourne l'ensemble des itemsets fréquents que vérifie la cellule passée en paramètre. Pour des raisons de simplicité, φ vérifie par défaut que l'intersection des deux ensembles passés en paramètre est non vide. Par conséquent, deux cellules sont connectées si elles vérifient au moins un même ensemble fréquent. Plus ce nombre d'ensembles fréquents est élevé, plus les cellules en question sont fortement connectées. Ceci est graphiquement illustré par la couleur des liens entre les cellules du cube.

L'approche proposée s'étend également pour permettre la génération de nouvelles vues de données contenant seulement les groupements de cellules inter-connectées, des groupements de cellules faiblement ou fortement liées (en se fixant un seuil minimum pour le degré de liaison entre les cellules), ou de nouvelles vues avec seulement les faits n'appartenant à aucun groupement et illustrant ainsi des exceptions dans le cube de données. Pour plus de détails sur cette approche, le lecteur peut se référer à (Naouali, Quafafou et Nachouki, 2004) et (Naouali, 2004).

5. Manipulation des cubes de données

5.1 Niveau 1: Intégration des outils d'analyse OLAP classiques

Cette section présente une intégration des outils d'analyse OLAP tels que connus et référencés dans la littérature. Nous nous sommes appuyés pour leur étude et leur collecte sur des travaux de recherche sur les besoins utilisateurs (Han et Kamber, 2001,Inmon, 1994,Kimbal, 1996,Codd, Codd et Salley, 1993). Ces outils consistent en des définitions informelles de ces besoins utilisateurs. Ces besoins peuvent être regroupés en deux catégories d'opérations : D'une part observer les données multidimensionnelles selon plusieurs niveaux d'abstraction (opérateurs d'agrégation), et d'autre part pouvoir changer la structure selon laquelle les données sont représentées[8] (opérateurs de restructuration). Le but de ces manipulations est de pouvoir découvrir des aspects insoupçonnables dans la grande masse de données disponibles dans un entrepôt permettant ainsi l'enrichissement de leur analyse exploratoire.

5.1.1 Opérateurs d'agrégation

Nous présentons en ce qui suit une version simple et plus spécifique[9] des opérations d'agrégation Roll_up (connue aussi sous le terme Drill_up) et Drill_down que nous appellerons SRoll_up and SDrill_down. Ces opérateurs permettent de représenter les données d'un cube à un niveau de granularité immédiatement supérieur (pour le SRoll_up) ou inférieur (pour le SDrill_down) selon une dimension:

TF'=SRoll_up(TF, $\xrightarrow{dim\,i}$) TF'=SDrill_down(TF, $\xrightarrow{dim\,i}$)

5.1.2 Opérateurs de restructuration

Rotate: opère sur un cube une rotation autour d'un des axes correspondant à ses dimensions, de manière à rendre visible de nouvelles facettes de ce cube. Cet opérateur prend en entrée la table *TF* et la dimension correspondant à l'axe de rotation. Il fournit en sortie une nouvelle table *TF*':

TF'=Rotate(TF, $\xrightarrow{dim\,i}$)

Switch: permet d'échanger les positions de deux membres d'une même dimension. Il prend en entrée une table *TF*, une expression de chemin sur les membres dont on veut échanger les positions ainsi que deux valeurs désignant ces deux membres et fournit en sortie une nouvelle table *TF*' :

TF'=Switch(TF, $\xrightarrow{dim\,i}$,val_1($\xrightarrow{dim\,i}$),val_2($\xrightarrow{dim\,i}$))

avec val_1($\xrightarrow{dim\,i}$) et val_2($\xrightarrow{dim\,i}$) prises dans le domaine de $\xrightarrow{dim\,i}$.

Pull: permet de changer le statut de certaines mesures d'un cube en membres, constituant ainsi une nouvelle dimension. Il prend en entrée une table *TF*, la mesure dont on veut changer le statut et un nom pour la nouvelle dimension:

TF= Pull (TF, mes_j, dim)

Push: consiste à faire passer des membres d'une dimension quelconque passée en paramètre comme contenu de cellules, et donc transformer une dimension en mesure:

TF= Push (TF, dim_j, mes)

Nest: permet de regrouper sur une même représentation bi-dimensionnelle toutes les informations d'un cube ou d'un hypercube en imbriquant des membres de différentes dimensions. Cette opération consiste alors à transformer la table *TF* à *n* dimensions et *k* mesures en une table *TF*' à seulement 2 dimensions et (*k*+*n*-2) mesures. Elle prend en entrée la table *TF* ainsi que deux expressions de chemin sur les deux dimensions à garder:

TF'=Nest(TF, $\xrightarrow{dim\,i}$, $\xrightarrow{dim\,j}$)

Split: permet de réduire le nombre de dimensions d'un cube de données. Le nombre de tables ou de cubes résultats dépend des informations de la représentation initiale. Cet opérateur transforme alors une table *TF* à *n* dimensions et *k* mesures en un ensemble de tables à *n*-1 dimensions et toujours *k* mesures. Le nombre de tables générées est donc égal au nombre de modalités dans la dimension éclatée:

Split (TF, dim_j, /TF_1, . . . , /TF_{card})

5.2 Niveau 2: Extension avec de nouveaux opérateurs d'analyse et d'extraction de connaissances

5.2.1 Nouveaux opérateurs d'analyse en ligne des données

Cette classe d'opérateurs comprend une extension des opérateurs d'agrégation, respectivement le Roll_up et le Drill_down, pour permettre respectivement d'agréger ou de détailler les données d'un cube à plus d'un niveau selon une dimension quelconque. Ces opérateurs sont par la suite appelés MRoll_up et le MDrill_down[10] . Ces opérateurs prennent en entrée une table des faits ainsi que deux expressions de chemins sur la même dimension tout en décrivant deux niveaux hiérarchiques différent, le MDrill_down prend en plus en entrée la table des faits initiale qui est la seule à contenir les données au plus bas niveau de détails:

TF'=MRoll_up(TF, $\xrightarrow{dim\,i}$, $\xrightarrow{dim\,i'}$) TF'=MDrill_down(TF,TV, $\xrightarrow{dim\,i}$, $\xrightarrow{dim\,i'}$)

Cube: cet opérateur restructure le cube de données en rajoutant le résultat d'agrégation de tous les membres de chaque dimension comme un nouveau membre pour la dimension en question.

TF'=Cube(TF, $\xrightarrow{dim\,i}$, $\xrightarrow{dim\,j}$, $\xrightarrow{dim\,k}$)

Roll_up2: cet opérateur permet la restructuration d'une dimension quelconque du cube en agrégeant quelques uns de ses membres tout en gardant le même niveau hiérarchique de cette même dimension[11] . Cet opérateur prend en entrée une table des faits, la dimension à restructurer ainsi que l'ensemble des membres à agréger et un libellé pour ce nouveau membre:

$$TF'=Roll\_up2(TF, \xrightarrow{dim\,i},\{val_1(\xrightarrow{dim\,i}),\ldots,val_n(\xrightarrow{dim\,i})\},lib)$$

Il est important de noter que cet opérateur doit être utilisé avec beaucoup de précaution car il permet de restructurer les hiérarchies préétablies.

Drop: réduit le nombre de dimensions d'un cube de données en supprimant celle passée en paramètre et en recalculant les agrégations nécessaires dans le cube en question (passé également en paramètre):

$TF'=Drop(TF, dim_j)$

Add-measure: rajoute une nouvelle mesure, dont le nom est passé en paramètre, à un cube de données. Cette nouvelle mesure est calculée grâce à une requête SQL passée également en paramètre:

TF'=Add-measure(TF, lib, requêteSQL)

Select_Links: opérateur d'ajout de contraintes dans le cube de données en générant des liens entre les cellules partageant les mêmes valeurs correspondant à la dimension ou la mesure passée en paramètre:

Select_links from <TF> on <dimension|mesure>

5.2.2 Intégration des opérateurs issus du modèle relationnel

Les opérateurs issus du modèle relationnel sont adaptés à la structure multidimensionnelle proposée. Ces opérateurs sont la sélection, la jointure, l'union, l'intersection, la différence, le renommage et la suppression. Cette intégration a pour objectif d'enrichir la manipulation des cubes de données en permettant de sélectionner des sous-cubes, calculer l'union, l'intersection ou la différence de deux cubes, etc. L'application de l'un de ces opérateurs produit par la suite une nouvelle vue matérialisée avec le résultat de la requête utilisateur. Ce dernier peut alors continuer l'exploration du cube répondant à sa requête avec des opérateurs d'analyse et/ou d'extraction de connaissances.

5.2.3 Opérateurs d'approximation des cubes de données

Nous décrivons dans cette section les opérateurs d'approximation des cubes OLAP que nous avons brièvement introduits dans la section 4.1. Ces approximations ainsi que les règles de classification et de description sont accessibles via les opérateurs suivants (dont l'appel génère non pas un ensemble de tuples mais plutôt une nouvelle vue matérialisée avec les faits répondant à la requête en question[12]):

select_lower: génère un nouveau cube avec les faits répondant sans ambiguïté à la requête utilisateur.

select_ upper: le cube généré peut contenir des faits ne répondant pas parfaitement à la requête.

select_ boundary_region: génère un cube avec seulement les faits douteux (répondant de façon incertaine à la requête).

select_ classification_rules: génère des règles basées sur la description des faits répondant sans ambiguïté à la requête utilisateur permettant ainsi de déterminer l'appartenance de nouveaux objets (faits) à la requête utilisateur.

select_ characteristic_rules: les règles générés sont basées sur la description de faits pouvant ne pas répondre parfaitement à la requête utilisateur permettant ainsi de donner des descriptions possibles aux faits répondant à la requête utilisateur.

Ces opérateurs permettent ainsi de filtrer les cubes OLAP pour ne garder que les cellules répondant à une requête donnée. L'utilisateur peut poursuivre par la suite son processus exploratoire en focalisant son attention sur des données plus précises et peut y appliquer des opérateurs d'analyse et/ou d'extraction de connaissances.

5.2.4 Opérateurs de découverte de liens entre les cellules d'un cube de données

Les opérateurs suivants permettent le calcul et la visualisation des relations inter-cellules telles que décrites dans la section 4.2:

Show_Links_All : calculer et visualiser l'ensemble total des liens découverts entre les cellules du cube.

Show_Links_One(cell_id) : ne générer que les liens sortant d'une seule cellule à la fois.

Show_Links_Off : désactiver la visualisation des liens entre les cellules du cube de données.

Associer ces opérateurs avec d'autres opérateurs de sélection et de visualisation permet à l'utilisateur de générer de nouveaux cubes de données avec seulement des cellules fortement ou faiblement liées ou encore des cellules exceptionnelles, etc. L'exploration de ces nouveaux cubes avec des opérateurs d'analyse et/ou d'extraction de connaissances reste toujours possible.

5.3 Niveau 3: opérateurs de visualisation et d'interaction

Pour ce dernier niveau d'opérateurs, nous mettons à la disposition de l'utilisateur une interface graphique permettant d'accéder à toutes les opérations ci-dessus décrites. Cette interface permet également la visualisation en 2*D* et 3*D* du cube de données. L'impact des opérateurs d'analyse et/ou d'extraction de connaissances est visualisé directement sur le cube ainsi manipulé.

En plus des opérateurs issus des deux premiers niveaux, nous décrivons ci-dessous de nouveaux opérateurs dédiés à la visualisation des données et des connaissances ainsi que des opérateurs d'enrichissement des interactions avec l'utilisateur.

Convert: permet de convertir la table des faits *TF*, que l'utilisateur veut visualiser, sous un format (fichier texte) propre à la visualisation: *text_file f=convert(TF)*. L'intérêt est d'éviter la visualisation de toute une succession de vues que l'utilisateur génère dans le seul but d'atteindre un objectif bien précis.

Project: permet de montrer une des facettes du cube.

Zoom_in, Zoom_out: élargir ou rétrécir le cube de données.

Enlarge(dim_i**), Reduce(**dim_i**):** élargir ou rétrécir une dimension donnée du cube.

GetInfo(cell_id): retourne toutes les informations relatives à la cellule dont l'identifiant est fourni en paramètre.

Plane_Clipping(x,y,z): permet différentes possibilités de plans de coupe selon une ou plusieurs dimensions permettant ainsi d'explorer le cube de données de l'intérieur. Cet opérateur prend en entrée trois coefficients appartenant à l'intervalle [0,1) et qui correspondent aux trois axes du cube.

6. Application: analyse du comportement des utilisateurs via le Web

Un fichier journal, *i.e., log file*, est un fichier en code ASCII dans lequel un serveur Web enregistre toutes les requêtes qui lui sont adressées par les navigateurs. Ces requêtes décrivent par la suite les transactions entre le serveur Web et les clients navigateurs. Une ligne du fichier journal correspond à une requête. La taille du fichier journal est par la suite proportionnelle à la fréquentation du serveur Web. Ce fichier contient des informations telles que l'adresse IP de l'utilisateur, la date de la requête utilisateur, l'URL demandé, etc. Analyser les fichiers journaux permet de répondre à des interrogations comme qui vient visiter notre site? d'où viennent ces internautes? pour quelle raison viennent-ils sur notre site? etc.

6.1 Données expérimentales et résultats de prétraitement

Les données expérimentales utilisées dans cet article proviennent de fichiers journaux générés par le serveur LOGIN qui est une association d'étudiants-chercheurs de l'Université de Nantes en France. Ces fichiers couvrent la période du 25 Mars 2002 au 28 Avril 2002. Cet article ne traite pas de l'étape de prétraitement des fichiers journaux, on se contente de mentionner qu'on ne considère que les requêtes demandant des pages HTML ou PHP et qu'on ignore les requêtes générées par les robots Web. Les données sont également transformées en sessions plutôt qu'en requêtes. Une session est un ensemble de pages visitées par un même utilisateur tel que le laps de temps écoulé entre deux pages successives ne dépasse pas un certain seuil prédéfini. Le prétraitement des données expérimentales ici considérées permet de générer 663 sessions. Cette étape de prétraitement a également montré que la page la plus visitée traite, étrangement, d'une activité de baseball[13] , ce qui n'a rien à voir avec le thème du

serveur qui est la recherche scientifique, la vie des étudiants chercheurs et leur insertion professionnelle. Cette page est accessible via la page personnelle d'un membre de l'association.

Par conséquent, le but de notre étude est l'identification et la compréhension du comportement des utilisateurs accédant à cette page de sport via le serveur de l'association LOGIN.

6.2 Modélisation multidimensionnelle des données provenant des fichiers journaux

Pour répondre à notre objectif, et comme l'illustre la figure 1 nous proposons le modèle d'entrepôt en étoile permettant de relier la table des faits *Trafic* aux dimensions suivantes:

- Domaine_Internet : information extraite de l'adresse IP de l'utilisateur ;
- Date : correspond à la date à laquelle l'URL a été demandé ;
- Commande : Get, Post ou HEAD ;
- URL : l'URL demandé, restructuré en URL–adresse_hôte–domaine ;
- Protocole : http ou autre ;
- Code : la réponse du serveur, restructurée en code–état–réponse ;
- Taille : la taille du fichier objet de la requête, restructuré en taille–intervalle;
- Référent : L'URL d'origine, restructuré en référent–adresse_hôte–domaine ;
- Navigateur : le navigateur utilisé par l'utilisateur, extrait à partir du champ user_agent des fichiers journaux, mentionne "autre" si le navigateur est inconnu par le système ;
- SE : le système d'exploitation utilisé par l'utilisateur, extrait à partir du champ user_agent des fichiers journaux, mentionne "autre" si le SE est inconnu par le système.

En plus des identifiants de dimensions, la table des faits contient comme mesure *NbHits* illustrant le nombre de requêtes (clicks) utilisateur, *i.e., hits*, dans la session. Il est important de mentionner ici que l'entrepôt ci-dessus décrit peut être utilisé pour tout serveur Web et son utilisation n'est dans aucun cas restreinte au serveur ici utilisé. Ainsi, les différentes catégories de fichiers journaux, leur prétraitement et fusion, etc, sont tous des aspects pris en compte dans notre proposition. Plus de détails sur ces aspects sont disponibles dans (Naouali, 2004).

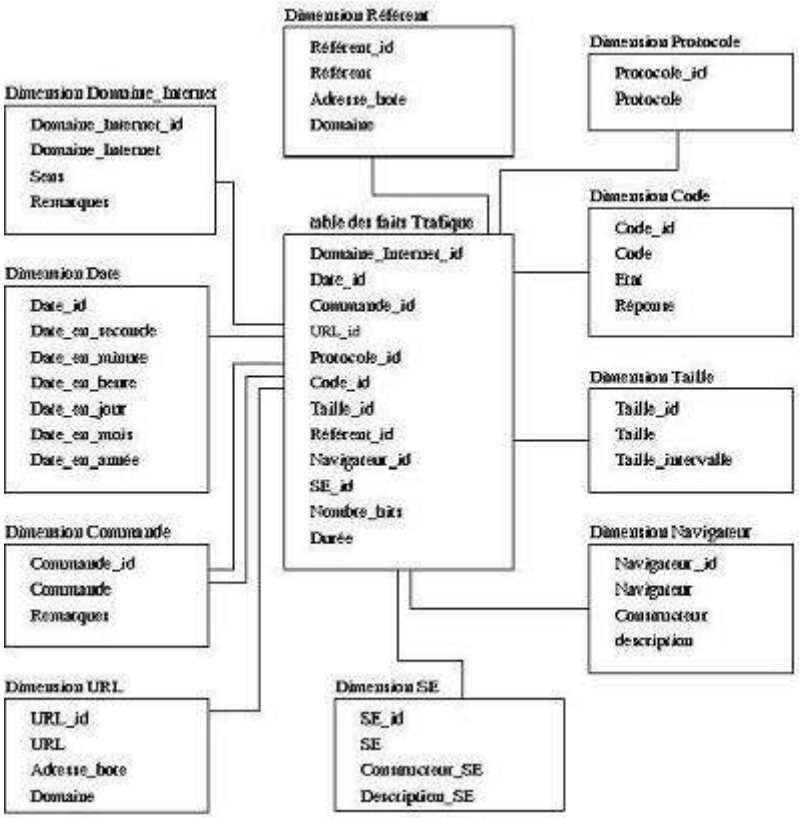

Figure 1. Schéma en étoile de l'entrepôt construit pour les fichiers journaux

6.3 Analyse et exploration de l'entrepôt de données expérimental

Dans le but de mieux cibler les utilisateurs demandant les pages de sport via le serveur LOGIN, nous utilisons le modèle d'entrepôt enrichi proposé précedemment. Notre but étant de poursuivre et comprendre le parcours de ces utilisateurs sur le serveur de LOGIN avant de demander les pages de sport. En effet, l'identification de leur parcours permettra d'identifier l'objectif pour lequel ces utilisateurs visitent le serveur LOGIN rien qu'en interprétant la relation sémantique pouvant exister entre les pages sources et celles de LOGIN (e.g., un utilisateur venant d'une page de sport cherche bien évidemment des informations sportives tandis qu'un utilisateur provenant du serveur de l'université de Nantes, à titre d'exemple, cherche fort probablement des informations sur LOGIN).

Pour ceci, nous produisons en premier lieu un cube de données avec les dimensions Référent, URL et Date et comme mesure (illustrée par la couleur des cellules) NbHits. Le scénario nécessaire pour générer ce cube est illustré dans la figure 2 où on fait appel principalement à l'opérateur *Drop* pour ne garder dans le cube final que les dimensions désirées. Le résultat de ce scénario est représenté dans le cube de la figure 3 qui illustre l'interface utilisateur proposée mettant à la disposition de l'utilisateur les trois niveaux d'opérateurs comme expliqué plus haut dans ce papier. Ce cube contient 6479 cellules dont la couleur illustre de faibles valeurs pour la mesure NbHits. Pour résoudre ce problème, nous procédons à l'agrégation de la dimension Date au moyen de l'opérateur MRoll_up pour l'exprimer en jour plutôt qu'en seconde. Nous appelons la table des faits résultat Trafic_réduit[14] .

Considérons à présent le concept (sous-cube) GTrafic_réduit[1] défini à partir de la table des faits Trafic_réduit comme étant l'ensemble des faits demandant la page "http:

//www.irin.univ-nantes.fr/mariners" (page de sport) ainsi que toute page qui découle de ce répertoire. On calcule par la suite l'approximation haute du concept cible GTrafic_réduit[1] pour prendre en compte tous les faits demandant certainement une page de sport ainsi que les faits qui leurs sont indiscernables. L'objectif est de focaliser notre attention sur les parcours utilisateurs qui peuvent conduire à une page de sport et ignorer le reste. Le cube de données résultant de cette approximation est illustré dans la figure 4 mettant en évidence (par le cadre blanc) des cellules correspondant à des parcours réguliers et fréquents dont l'exploration (avec les outils intégrés au système) montre qu'ils mènent au concept cible. Nous supprimons par la suite les connexions venant de l'intérieur du serveur LOGIN pour focaliser notre attention sur les connexions depuis l'extérieur. Le cube résultat est illustré dans la figure 5. Par la suite nous colorons en blanc toutes les cellules avec NbHits inférieur à 4 pour ne garder que les cellules les plus intéressantes (cube de la figure 6). L'exploration du cube ainsi obtenu montre que les cellules du côté droit du cube appartiennent au concept cible (demandent des pages de sport) alors que celles encerclées en jaune ne le sont pas, elles sont juste indiscernables des autres cellules (ce qui explique leur appartenance à l'approximation haute) tout en provenant du serveur *google*. Nous concluons alors que les utilisateurs ayant visité des pages de sport proviennent essentiellement de "http: //fantasybaseball.free.fr" ou de "http: //mariners.free.fr". Ces utilisateurs adoptent un parcours indiscernable de celui effectué par des utilisateurs provenant du serveur *google*.

Pour résumer, nous avons pu identifier des cellules indiscernables illustrant deux types d'utilisateurs ; des utilisateurs en quête d'informations sportives et ceux en quête d'informations sur LOGIN. Néanmoins, on peut se poser la question sur l'homogeneité du comportement de chacun des deux groupes.

Pour répondre à cela, nous introduisons à ce même cube une nouvelle mesure associant à chaque cellule du cube le temps moyen mis par l'utilisateur à visualiser la page en question et ce en utilisant l'opérateur Add_mesure. Le cube ainsi obtenu contient dès lors trois dimensions et deux mesures. Pour visualiser la deuxième mesure, nous dressons un segment de droite entre deux cellules si et seulement si elles ont quasiment la même valeur pour cette deuxième mesure. Le résultat de ceci est représenté dans la figure 7 montrant que les cellules demandant des pages de sport bien qu'elles soient reliées entre elles par la première mesure NbHits (dans la mesure où elles correspondent à des nombres élevés de requêtes utilisateurs), ne le sont pas par rapport à la deuxième mesure (les utilisateurs ne mettent pas tous le même temps pour visualiser les pages correspondantes). Ces cellules sont par la suite homogènes par rapport à la première mesure mais hétérogènes par rapport à la seconde.

En conclusion, nous avons pu grâce au modèle d'entrepôt enrichi ici proposé, identifier les utilisateurs ayant effectué les parcours les plus fréquents via le serveur de LOGIN. Ce groupe d'utilisateurs comprend en réalité deux sous-groupes d'utilisateurs ; ceux en quête d'informations sportives et ceux en quête d'informations sur LOGIN. Et même si les parcours des utilisateurs de chacun de ces deux sous-groupes semblent homogènes, les utilisateurs en quête d'informations sportives ne passent pas le même temps à visualiser les pages correspondantes même s'ils ont les mêmes intérêts. En plus, les deux sites dont proviennent les trafics les plus importants sont "http: //fantasybaseball.free.fr" et "http: //mariners.free.fr". Ces pages correspondent à des associations sportives dont le Web Master est un membre de LOGIN. Les utilisateurs provenant de ces deux pages ne sont pas là pour s'informer sur LOGIN mais seulement pour du sport, c'est pourquoi ils quittent rapidement le serveur.

```
1: BEGIN{Scenario 1}
2: Trafic1=Drop (Trafic, commande)
3: Trafic2=Drop (Trafic1, protocole)
4: Trafic3=Drop (Trafic2, code)
5: Trafic4=Drop (Trafic3, taille)
6: Trafic5=Drop (Trafic4, adresse_ip)
7: Trafic6=Drop (Trafic5, navigateur)
8: Trafic7=Drop (Trafic6, SE)
9: Trafic7_converted=Convert(Trafic7)
10: Show(Trafic7_converted)
11: END{Scenario 1}
```

Figure 2. Scénario 1

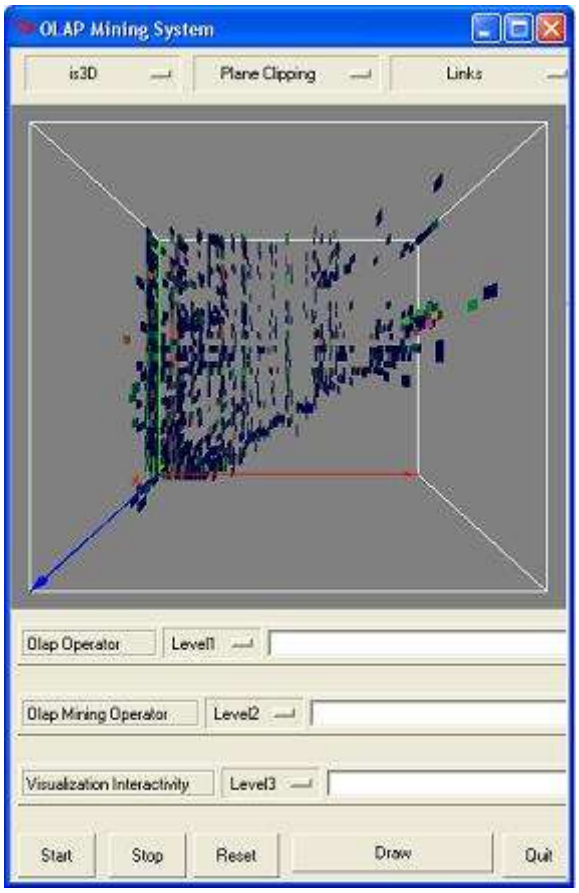

Figure 3. Cube de données initial (Url, Référent, Date ; NbHits)

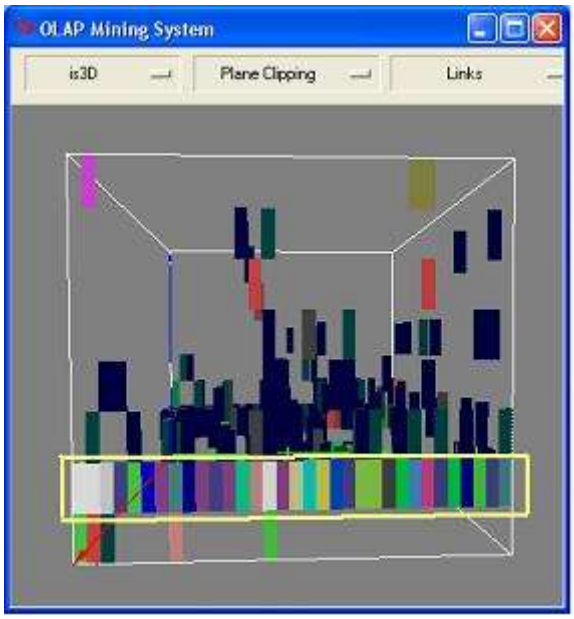

Figure 4. Cube de données illustrant l'approximation haute du concept $G_{Reduced\_Traffic}$ [1].

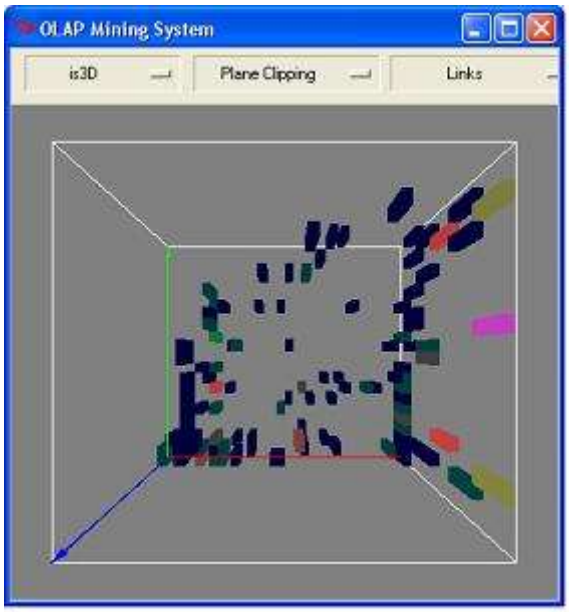

Figure 5. Suppression des requêtes internes.

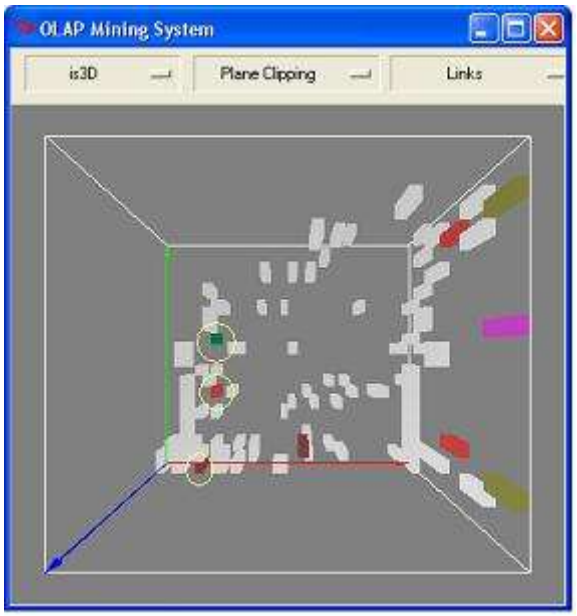

Figure 6. Suppression des cellules avec NbHits<4.

Figure 7. Visualisation de la nouvelle mesure.

7. Conclusion

Cet article présente un cadre général pour l'intégration d'outils d'analyse OLAP et d'extraction de connaissances. Nous avons présenté un entrepôt de données enrichi avec de la connaissance. Le but de cette proposition est de mieux guider l'utilisateur dans son processus d'analyse et d'exploration du cube de données. Pour illustrer cette proposition, nous avons construit et utilisé un entrepôt de données réel dont l'objectif est la compréhension du comportement des utilisateurs en naviguant sur un serveur donné. Nous avons alors proposé un modèle en étoile pour cet entrepôt que nous avons exploré en utilisant des opérateurs d'analyse OLAP, des opérateurs d'extraction de connaissances ainsi que des opérateurs de visualisation et d'interaction avec l'utilisateur. Nous avons montré avec un exemple réel et concret que l'enrichissement des entrepôts de données avec de tels outils amène certainement à des possibilités d'analyse et d'exploration des cubes de données avancées et non supportées par les systèmes OLAP classiques. Nos travaux actuels traitent, en partie, de la redéfinition des opérateurs d'analyse OLAP pour agir non seulement sur les données multidimensionnelles mais également sur la connaissance extraite à partir de ces données. Le but de cette extension est qu'à tout moment du processus exploratoire du cube de données, les données multidimensionnelles ainsi que les connaissances sont conjointement manipulées, explorées et visualisées.

1Un exemple serait une table dimension *Lieu* liée à la table *Ville* qui à son tour est liée à la table *Région* qui à son tour est liée à la table *Pays*.
2Plusieurs valeurs si plusieurs mesures dans le cube.
3Sous-ensemble de la réponse exacte contenant seulement les faits répondant sans ambiguïté à la requête
4Sur-ensemble de la réponse exacte pouvant contenir des faits ne répondant pas à la requête OLAP mais qui sont indiscernables avec d'autres faits qui eux répondent sans ambiguïté à la requête
5Les sous-itemsets d'itemsets fréquents sont forcément fréquents
6On appelle itemset calculé tout itemset candidat qui a été considéré et traité dans le processus de découverte des itemsets fréquents. Cet ensemble est par la suite inclut dans FD qui contient en plus les sous-itemsets générés à partir des itemsets de FC.
7Ce nombre n'est rien d'autre que *FC+RC*.
8changer les dimensions observées, ajouter de nouvelles mesures, interchanger les membres d'une ou de plusieurs dimensions, etc.
9Car tel que connu dans la littérature, l'opérateur Roll_up, à titre d'exemple, peut agréger les données d'un cube de plus d'un niveau selon une dimension donnée. Nous avons choisi d'implanter une telle action séparément car elle implique que l'utilisateur connaît à priori, et exactement, le niveau hiérarchique auquel il désire agréger les données ce qui n'est pas toujours le cas.
10Multi-niveau_Roll_up et Multi-niveau_Drill_down.
11Il est par exemple possible d'agréger les mois de décembre 2000 et janvier 2001, ce qui n'est pas supporté par la hiérarchie préétablie de la dimension *temps*.
12Sauf pour les opérateurs de génération de règles.
13Cette page est: "http: //login.irin.sciences.univ-nantes.fr/mariners/fantasybaseball/index.php3".
14Trafic_réduit=MRoll_up(Trafic7, date, date_en_jour).

Bibliographie

(Agrawal et Srikant, 1994) Agrawal, R., Srikant, R., (1994). Fast algorithms for mining association rules. In Proceeding of the VLDB'1994 Conference, pages 487–499.

(Babcock, Chauhuri et Das, 2003) Babcock, B., Chaudhuri, S., Das, G., (2003). Dynamic sample selection for approximate query processing. In Proceeding of the SIGMOD'03 Conference, pages 539–550.

(Barbara et Sullinvan, 1997) Barbara, D., Sullivan, M., (1997). Quasi-cubes: exploiting approximations in multidimensional databases. Proceeding of the SIGMOD'97 Conference, 26(3): 12–17.

(Barnett et Lewis, 1994) Barnett, V., Lewis, T., (1994). Outliers in Statistical Data. free paper.

(Chakrabarti, Garofalakis, Rastogi et al., 2001) Chakrabarti, K., Garofalakis, M., Rastogi, R., Shim, K., (2001). Approximate query processing using wavelets. The VLDB Journal, 10(2-3): 199–223.

(Codd, Codd et Salley, 1993) Codd, E.F., Codd, S.B., Salley, C.T., (1993). Providing olap (on line analytical processing) to user analysts : an it mandate [on line].

(Dong, Han, Lam et al., 2001) Dong, G., Han, J., Lam, J., Pei, J., Wang, K., (2001). Mining multi-dimensional constrained gradients in data cubes. In Proceeding of the VLDB'01 Conference.

(Feng, Agrawal, Abbadi et al., 2003) Feng, V., Agrawal, D., Abbadi, A., Metwally, A., (2003). Range cube: Efficient cube computation by exploiting data correlation. In Proceeding of the ICDE'2004 Conference, pages 658–670.

(Ganter et Wille, 1999) Ganter, B., Wille, R., (1999). Formal Concept Analysis - Mathematical Foundations. Springer.

(Han, Fu, Wang et al., 1996) Han, J., Fu, Y., Wang, W., Chiang, J., Gong, W., Koperski, K., et al., (1996). Dbminer: A system for mining knowledge in large relational databases. In Proceeding of the KDD'1996 Conference, Portland, Oregon, pages 250–255.

(Han et Kamber, 2001) Han, J., Kamber, V., (2000). Data Mining: Concepts and Techniques. Morgan Kaufmann.

(Imielinski, Khachiyan et Abdulghani, 2002) Imielinski, T., Khachiyan, L., Abdulghani, A., (2002). Cubegrades: Generalizing association rules. Data Mining and Knowledge Discovery, 6(3): 219–257.

(Inmon, 1994) Inmon, W., (1994). Building the Dada Warehouse Toolkit. Wiley Computer Publishing.

(Kimbal, 1996) Kimbal, (1996). The Data Warehouse Toolkit. Wiley Computer Publishing.

(Knorr, Ng et Tucakov, 2000) Knorr, E.M., Ng, R.T., Tucakov, V., (2000). Distance-based outliers: algorithms and applications. The VLDB Journal, 8(3-4): 237–253.

(Lakshmanan, Pei et Han, 2002) Lakshmanan, L., Pei, J., Han, J., (2002). Quotient cube: How to summarize the semantics of a data cube. In Proceeding of the VLDB'2002 Conference, pages 778–789.

(Lakshmanan, Pai et Zhao, 2003) Lakshmanan, L.V.S., Pei, J., Zhao, Y., (2003). Qc-trees: an efficient summary structure for semantic olap. In Proceeding of the SIGMOD'2003 Conference, New York, USA, pages 64–75.

(Li, Han et Gonzalez, 2004) Li, X., Han, J., Gonzalez, H., (2004). High-dimensional olap: A minimal cubing approach.

(Naouali, 2004) Naouali, S., (2004). Enrichissement d'Entrepôts de Données par la Connaissance : Application au Web. PhD thesis, Université de Nantes, France, December 2004.

(Naouali et Missaoui, 2005) Naouali, S., Missaoui, R., (2005). Flexible query answering in data cubes. In Proceeding of the DAWAK'2005 Conference, Copenhagen, Denmark, Springer Verlag, pages 221–232.

(Naouali, Quafafou et Nachouki, 2004) Naouali, S., Quafafou, M., Nachouki, G., (2004). Mining olap cubes: Semantic links based on frequent itemsets. In Proceedings of the IEEE Int. Conf. on Information and Communication Technologies: from Theory to Applications, Damascus, Syria.

(Ross et Srivastava, 1997) Ross, K.A., Srivastava, D., (1997). Fast computation of sparse datacubes. In Proceeding of the VLDB'1997 Conference, San Francisco, CA, USA, pages 116–125.

(Sarawagi, Agrawal et Megiddo, 1998) Sarawagi, S., Agrawal, R., Megiddo, N., (1998). Discovery-driven exploration of olap data cubes. In Proceeding of the EDBT'1998, London, UK, pages 168–182.

(Shanmugasundaram, Fayyad et Bradley, 1999) Shanmugasundaram, J., Fayyad, U., Bradley, P.S., (1999). Compressed data cubes for olap aggregate query approximation on continuous dimensions. In Proceeding of the KDD'1999 Conference, pages 223–232.

(Vitter et Wang, 1999) Vitter, J.S., Wang, M., (1996). Approximate computation of multidimensional aggregates of sparse data using wavelets. In Proceeding of the SIGMOD'99 Conference, pages 193–204.

(Xin, Han et Wah, 2003) Xin, D., Han, J., Wah, B.W., (2003). Starcubing: Computing iceberg cubes by top-down and bottom-up integration. In Proceeding of the VLDB'03 Conference.

(Zaki, Parthasarathy, Ogihara et al., 1997) Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W., (1997). Parallel algorithms for discovery of association rules. Data Mining and Knowledge Discovery, 1(4): 343–373.