THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Université Toulouse 1 Capitole (UT1 Capitole)

Discipline ou spécialité :

Informatique

Présentée et soutenue par :
Houssem Jerbi

le : vendredi 20 janvier 2012

Titre :

Personnalisation d'analyses décisionnelles sur des données multidimensionnelles

---

Ecole doctorale :
Mathématiques Informatique Télécommunications (MITT)

Unité de recherche :
Institut de Recherche en Informatique de Toulouse – UMR 5505

Directeur(s) de Thèse :

Gilles Zurfluh

Rapporteurs :
O.Boussaid, Pr. à l'Université Lyon 2
C. Cauvet, Pr. à l'Université Aix-Marseille 3

Autre(s) membre(s) du jury

C. Chrisment, Pr. à l'Université Toulouse 3 - J. Feki, MC HDR à l'Université de Sfax
A. Laurent, Pr. à l'Université Montpellier 2 - F. Ravat, Pr. à l'Université Toulouse 1 Capitole
O. Teste, MC HDR à l'Université Toulouse 3 - G. Zurfluh, Pr. à l'Université Toulouse 1 Capitole

A mes parents

Résumé

Les systèmes OLAP (On-Line Analytical Processing) ont été proposés pour améliorer les processus de prise de décision par l'analyse de grandes masses de données. Ces systèmes organisent les données sous forme d'une constellation de faits (sujets d'analyse) et de dimensions (axes d'analyse). De nos jours, les besoins des décideurs ne se limitent plus à la simple navigation des données mais ils sollicitent des « analyses à la carte » qui facilitent leurs tâches quotidiennes de prise de décision. Les systèmes OLAP actuels s'avèrent inadaptés à ces exigences d'adaptation. Dans cette thèse, nous nous intéressons à la personnalisation des analyses OLAP.

Nous proposons un cadre générique pour la prise en compte de l'usager dans l'analyse OLAP. Cette analyse est considérée comme une exploration interactive des données basée sur une succession de requêtes. Chaque étape de l'analyse constitue un contexte d'analyse qui est représenté par un arbre spécifique. Nous modélisons une analyse OLAP par un graphe de contextes d'analyse. Afin de personnaliser ces analyses, nous proposons une modélisation des préférences de l'usager portant sur le schéma ainsi que les valeurs d'une constellation. Chaque préférence est associée avec un contexte d'analyse qui précise son cadre d'application.

En nous basant sur le modèle de graphe d'analyse, nous définissons un mécanisme de personnalisation globale des analyses OLAP en fonction des préférences de l'usager. Ce mécanisme permet de personnaliser le contexte d'analyse courant de l'usager (personnalisation de requête) et de recommander les prochains contextes d'analyse à visiter (recommandation de requête).

La personnalisation de requête permet d'enrichir la requête de l'usager en fonction de préférences dans le but de fournir un résultat personnalisé. Cette personnalisation est effectuée suivant une perspective utilisateur afin d'augmenter l'intérêt du résultat ou selon une perspective système où seules les K meilleures préférences qui satisfont une contrainte de personnalisation sont considérées.

La recommandation de requête est effectuée suivant trois scénarios. L'usager est guidé dans la formulation de ses requêtes. Après l'exécution de la requête, le système recommande à l'usager la requête suivante en anticipant ses besoins d'analyse et des requêtes alternatives qu'il n'est pas susceptible de demander. Les recommandations sont construites progressivement à partir des préférences de l'usager par l'appariement du contexte d'analyse courant avec les contextes des préférences.

Afin de valider nos propositions, nous présentons un prototype de manipulations OLAP personnalisées qui est écrit en Java. Les préférences et les contextes d'analyse sont stockés en extension de la constellation dans un environnement R-OLAP. Nous montrons l'efficacité de nos approches par une série d'expérimentations.

Mots-clés: OLAP, analyse décisionnelle, personnalisation de requête, recommandation de requête, préférence utilisateur, contexte d'analyse.

Remerciements

Mes remerciements s'adressent d'abord à Claude Chrisment et Gilles Zurfluh, responsables de l'équipe Systèmes d'Informations Généralisées (SIG) pour m'avoir accueilli au sein de leur équipe afin que je puisse mener à bien cette thèse.

Je remercie très sincèrement Omar Boussaid, Professeur à l'Université Lyon 2 et Corine Cauvet, Professeure à l'Université Aix-Marseille 3 pour avoir accepté d'être rapporteurs de ce mémoire, pour leurs remarques pertinentes et pour leur participation à mon jury de thèse.

Je tiens à remercier également Jamel Feki, Maître de Conférences HDR à l'Université de Sfax et Anne Laurent, Professeure à l'Université Montpellier 2 pour tout l'intérêt qu'ils ont manifesté envers mon travail et pour l'honneur qu'ils m'accordent en participant au jury.

Ma reconnaissance va envers Gilles Zurfluh, professeur à l'Université Toulouse 1 Capitole pour avoir dirigé et encadré mes recherches, pour sa rigueur scientifique, ses conseils et ses critiques constructives. Je le remercie pour m'avoir laissé une grande liberté dans mes travaux.

Je remercie Franck Ravat, Professeur à l'Université Toulouse I Capitole et Olivier Teste, Maître de Conférences HDR à l'Université Toulouse 3 pour avoir encadré et suivi cette thèse. Leurs remarques, leurs conseils et nos différents échanges m'ont été d'un grand intérêt.

Je tiens aussi à remercier tous les membres de l'équipe SIG pour leur accueil chaleureux et leur gentillesse. Je remercie mes amis de l'équipe pour leur présence, leur aide et leur collaboration. Plus particulièrement, je tiens à remercier Anass, Hamdi, Nacim et Ronan.

Mes remerciements vont aussi à l'ensemble du personnel de l'IRIT, pour leur disponibilité, leur aide généreuse et leur gentillesse.

Je remercie mes parents à qui je dédie cette thèse. Je leur suis très reconnaissant de m'avoir assuré de leurs encouragements et de leur soutien sans borne au cours de ce si long cursus universitaire. J'espère rester un sujet de fierté à leurs yeux. J'ai également une pensée affectueuse à mes frères et à mes nièces Eya, Salma et Myriam.

Enfin, je remercie mon épouse qui dans l'ombre m'a apporté durant les deux dernières années de thèse tout le soutien. Elle a su m'aider dans les moments difficiles et a fait preuve de patience.

Sommaire général

Liste des Figures

Liste des Tableaux

Chapitre 1

Introduction Générale

Sommaire

1 Introduction

L'information représente un capital immatériel dont la bonne gestion est un facteur primordial pour la réussite de toute organisation. Les systèmes d'information ont pour objectif de supporter la réalisation des activités d'une organisation. Ils sont construits à partir des exigences des métiers et des processus définis par l'entreprise afin de stocker, traiter et communiquer les informations.

Les systèmes d'information des entreprises peuvent accumuler au fil du temps un volume important de données stockées sur plusieurs sites internes à l'entreprise ou provenant de son environnement externe (partenaires, Web, …). Le problème des entreprises est d'exploiter efficacement ces données afin de permettre aux décideurs d'optimiser leurs choix et de leur faciliter le pilotage à moyen terme via une meilleure anticipation. Ainsi, le besoin d'une exploitation efficace des données dans une perspective décisionnelle a donné lieu à l'élaboration de nouveaux systèmes, dits systèmes d'aide à la décision, facilitant le stockage et le traitement synthétique de grands volumes de données (Inmon, 1996).

2 Les systèmes d'aide à la décision

Un système d'information permet de faciliter la mise en œuvre des stratégies de l'entreprise. Le rôle d'un système d'aide à la décision est plutôt d'aider à déterminer les bonnes stratégies. Plus précisément, un système d'aide à la décision permet de développer la capacité de réflexion et d'action de l'entreprise en aidant à l'apprentissage (analyse et suivi des activités précédentes) et au pilotage des plans d'actions (prévision et planification des activités futures).

Les systèmes d'aide à la décision sont employés dans tous les domaines où la prise de décision est nécessaire, à savoir, les domaines du commerce (marketing, ventes), de la logistique, de la santé (aide à la décision médicale), de la science (par exemple en bioinformatique), des télécommunications, des transports (trafic autoroutier), des banques, ...

Définition. Un système d'aide à la décision est l'ensemble des outils matériels et logiciels qui permettent de collecter, de stocker et d'analyser des données issues du système d'information des entreprises dans le but de faciliter la prise de décision par les décideurs.

Les décideurs, utilisateurs de ces systèmes, sont des experts d'un métier chargés d'analyser les données décisionnelles pour le pilotage de l'organisation. Ils sont généralement non informaticiens. Dans la suite, nous les désignerons simplement par le terme *usagers*.

2.1 Architecture de systèmes d'aide à la décision

Afin d'offrir une vision transversale de l'activité de l'entreprise, les systèmes d'aide à la décision collectent et stockent des données en provenance des bases de données des différents métiers de l'entreprise et de sources externes (sites web, emails,…). A notre sens, la conception d'un système d'aide à la décision doit être basée sur la séparation entre deux espaces de stockage : l'entrepôt qui regroupe toute l'information décisionnelle et les magasins qui contiennent une partie de cette information, dédiée à un thème, un métier, ou une analyse.

En s'appuyant sur cette dichotomie, nous définissons l'architecture des systèmes d'aide à la décision décrite à la Figure 1. Cette architecture est composée de deux niveaux de stockage (entrepôt et magasin de données) et de deux niveaux fonctionnels (outils ETL et outils de restitution et d'analyse).

- L'entrepôt est le lieu de stockage centralisé d'un extrait des sources. Son organisation doit faciliter l'intégration des données et la conservation de leurs évolutions.
- Le magasin est un extrait de l'entrepôt. L'organisation des données d'un magasin suit un modèle spécifique qui facilite les analyses décisionnelles.
- Les outils d'extraction, de transformation et de chargement ou ETL (Extract, Transform, Load) permettent d'alimenter et de rafraîchir les données contenues dans les entrepôts et les magasins de données à partir des sources.
- Les outils de restitution et d'analyse permettent d'interroger et d'analyser les données sous une forme adaptée aux décideurs.

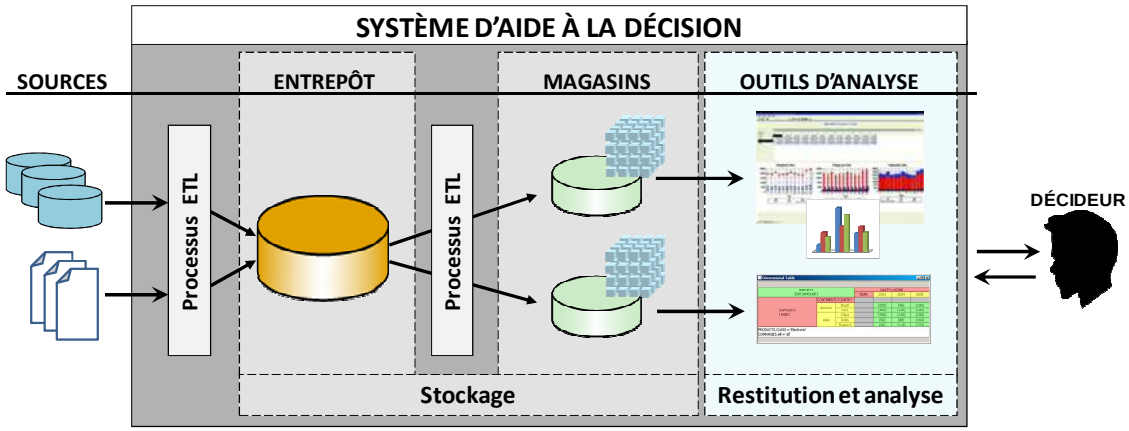


Figure 1. Architecture d'un système d'aide à la décision

2.2 Stockage de données

2.2.1 Entrepôts

Bill Inmon définit l'entrepôt de données comme « une collection de données orientées sujet, intégrées, non volatiles et historisées, organisées pour le support d'un processus d'aide à la décision » (Inmon, 1996).

Définition. *L'entrepôt de données* est un espace de stockage centralisé qui permet de stocker et d'historiser des données hétérogènes qui sont pertinentes pour la prise de décision.

L'organisation des données au sein de l'entrepôt de données suit un modèle assurant la gestion efficace des données.

2.2.2 Magasins de données

Définition. *Un magasin de données* constitue un extrait de l'entrepôt adapté à une classe de décideurs ou à un usage particulier et organisé suivant un modèle adapté aux traitements décisionnels.

Les magasins de données reposent sur une modélisation multidimensionnelle des données afin de supporter efficacement le processus de prise de décision reposant sur des analyses OLAP (« On-Line Analytical Processing » (Codd, et al., 1993)).

Dans la suite, nous désignons par le terme *Base de Données Multidimensionnelles* (BDM) un magasin de données suivant une organisation multidimensionnelle.

Modélisation conceptuelle

Au niveau conceptuel, les données d'un magasin sont vues sous la forme de points dans un espace à plusieurs dimensions avec la métaphore de cube ou d'hypercube de données (Gray, et al., 1996). Chaque donnée représente une cellule du cube. Les arêtes du cube représentent les axes d'analyse des données et comportent plusieurs graduations afin de permettre l'observation des données selon différents niveaux de détail.

Exemple. La figure suivante présente un exemple de cube de données qui permet l'analyse des publications scientifiques. Le cube est formé de nombres de publications en cellules et de trois arêtes graduées respectivement par des équipes d'auteurs, des niveaux de manifestations et des trimestres.

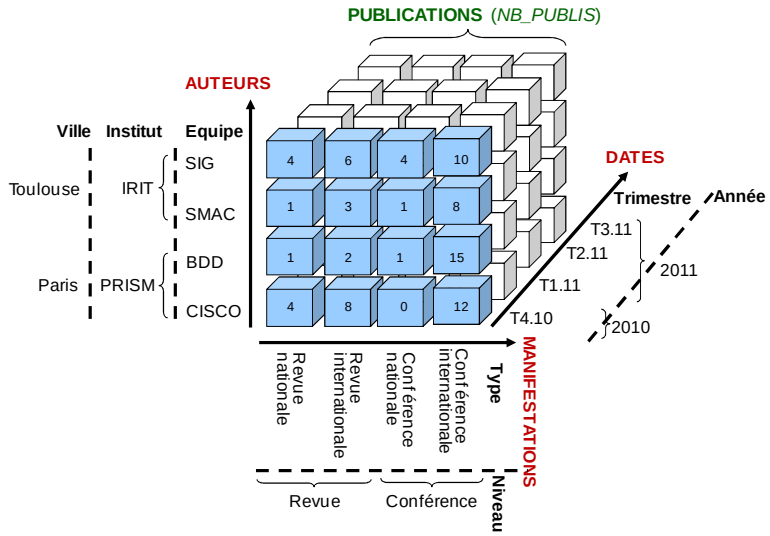


Figure 2. Exemple de cube de données

La modélisation en cube est très limitée en termes de représentation des structures hiérarchiques des axes d'analyse (Torlone, 2003) et de séparation entre structures et contenu. Des structures avancées ont été définies permettant la modélisation de sujets d'analyse appelés faits, et d'axes d'analyse appelés dimensions (Kimball, 1996). Chaque fait est composé d'indicateurs d'analyse appelés mesures. Les dimensions sont composées d'attributs, appelés *paramètres*, agencés de manière hiérarchique, qui modélisent les différents niveaux de détails des axes d'analyse. Les magasins sont modélisés par des schémas en étoile qui organisent les données en fait et dimensions (Kimball, 1996 ; Abelló et al., 2001). Une généralisation possible du schéma en étoile est le schéma en constellation qui est constitué de plusieurs faits et de plusieurs dimensions éventuellement partagées.

Exemple. Le schéma en étoile associé à l'exemple précédent est présenté en Figure 3. Le sujet d'analyse est modélisé par le fait *publications* qui est composé de la mesure *nb_publis*.

Les axes de l'analyse sont représentés par les dimensions *dates*, *auteurs* et *manifestations*. La dimension *auteurs* est caractérisée par les trois paramètres *équipe*, *institut* et *ville*.

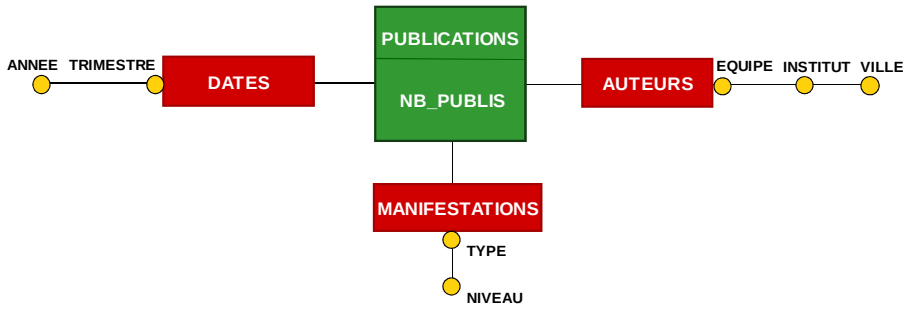


Figure 3. Exemple de schéma en étoile

Modélisation logique

Au niveau logique, les BDM sont modélisées suivant une approche relationnelle, multidimensionnelle ou objet (Chaudhuri et Dayal, 1997).

L'approche la plus répandue est l'approche relationnelle « Relational OLAP » (R-OLAP) où les bases de données multidimensionnelles sont traduites par des relations (Kimball, 1996 ; Mangisengi et Tjoa, 1998). Cette approche admet de nombreux avantages tels que la réutilisation des mécanismes de gestion des données éprouvés depuis des décennies et la capacité à gérer des volumes de données très importants.

L'approche « Object OLAP » (O-OLAP) s'appuie sur le paradigme objet (Cauvet et Semmak, 1994). Les magasins de données sont modélisés par des classes de fait et de dimensions. L'avantage de l'approche O-OLAP est lié au niveau d'abstraction fort de l'objet qui prend en compte des concepts plus riches.

Une autre approche, dite « Multidimensional OLAP » (M-OLAP), consiste à stocker les données nativement sous une forme multidimensionnelle (Dinter, et al., 1998). L'intérêt de cette approche réside dans l'optimisation du temps d'accès. Mais, cette approche nécessite de redéfinir tous les mécanismes des systèmes de gestion de base de données pour manipuler les structures multidimensionnelles.

2.3 Analyse et restitution de données

Les données d'une BDM sont interrogées via la technologie OLAP à l'aide d'outils graphiques ou suivant un langage textuel (Ravat et al., 2007b, 2008).

Une requête OLAP est une requête multidimensionnelle permettant d'agréger les données d'une ou de plusieurs mesures d'un fait suivant les attributs d'une ou plusieurs dimensions.

2.3.1 Opérations de manipulation OLAP

De nombreuses propositions concernent la définition d'opérations de manipulation OLAP. Les structures de données manipulées sont des cubes de données ou des tranches du cube. Il n'existe pas de consensus sur la définition d'un ensemble minimum d'opérateurs assurant l'intégralité des opérations de manipulation OLAP, mais la plupart des propositions offrent un support partiel des différentes catégories d'opérations suivantes :

- Forage vers le bas (drill-down), qui consiste à descendre dans une hiérarchie de dimension vers un niveau plus détaillé,
- Forage vers le haut (roll-up), qui consiste à remonter d'un niveau dans une hiérarchie de dimension vers un niveau plus agrégé,
- Rotation, qui réoriente une analyse en changeant l'axe d'analyse en cours (rotation de dimension),
- Sélection de tranches : slice, qui sélectionne un sous-ensemble de l'hypercube réduit à un ensemble de membres sur une ou plusieurs dimensions, et dice qui réduit l'hypercube d'une ou plusieurs dimensions.

2.3.2 Structure de visualisation

L'affichage des données d'une BDM peut être effectué selon diverses structures de visualisation telles que les courbes, les histogrammes, … Même si la modélisation multidimensionnelle est basée sur la métaphore du cube ou de l'hypercube, la structure de visualisation la plus utilisée dans le contexte OLAP est la table multidimensionnelle (TM) qui affiche les données selon deux axes (Gyssens et Lakshmanan, 1997 ; Lehner 1998). Une TM permet de visualiser une tranche du cube.

2.4 Système OLAP

Les systèmes d'aide à la décision que nous considérons reposent sur l'analyse en ligne (OLAP) des données décisionnelles afin de prendre des décisions. Nous les désignons par les systèmes OLAP.

Par la suite, le terme *système OLAP* désignera un système d'aide à la décision dont les données sont stockées selon une dichotomie entrepôt et magasins de données, et les outils d'analyse et de restitution de données sont basés sur la technologie OLAP.

Les systèmes OLAP stockant les données selon l'approche relationnelle rajoutent une couche logicielle (moteur de requête OLAP) qui analyse et traduit les requêtes OLAP en requêtes de bases de données, puis organise le résultat du moteur de requête d'un SGBDR selon un format multidimensionnel.

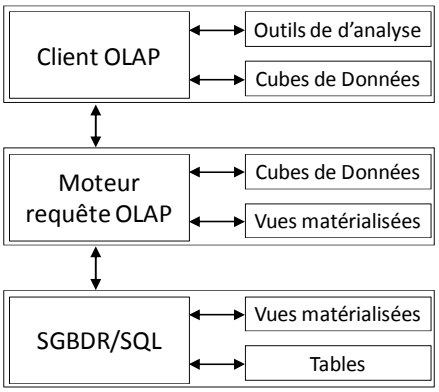


Figure 4. Implantation d'un système OLAP au dessus d'un SGBDR

3 Personnalisation de l'information : au-delà de l'analyse des données

La personnalisation constitue un enjeu capital pour l'industrie informatique. En effet, le problème de personnalisation se pose dès lors qu'un service est offert en réponse à des besoins de l'usager. Par exemple, dans le domaine de l'interaction Homme-Machine, l'adaptation de l'interface aux caractéristiques de l'usager (novice ou expert, personne handicapée ou valide, jeune ou âgé, …) constitue un facteur clé du succès ou du rejet d'un système. Plus particulièrement, dans le domaine des systèmes d'informations, la personnalisation a été abordée par différentes communautés scientifiques telles qu'en bases de données (Bouzeghoub et Kostadinov, 2005; Koutrika et Ioannidis, 2004, 2005a, 2005b) et en recherche d'information (Chirita et al., 2007; Liu et al., 2004). Elle a été définie comme solution à deux problèmes mutuels. D'une part, la prolifération de l'information disponible suite à l'essor des technologies de l'information et l'avènement d'Internet engendre des résultats massifs comportant des données sans intérêt. D'autre part, le manque d'expressivité des langages d'interrogation des données, tels que SQL en bases de données et les requêtes sous forme de mots clés en recherche d'information, rend la tâche de description des informations recherchées difficile. Ainsi, la personnalisation a pour objectif, d'une part, de faciliter l'expression des besoins de l'usager et, d'autre part, de sélectionner des informations pertinentes.

3.1 Définitions et terminologie

La notion de personnalisation de l'information est souvent vague dans la littérature et peut se rapporter à des concepts différents.

Définition. Nous définissons la personnalisation de l'information comme l'action d'adapter l'accès à l'information en fonction d'informations sur un usager ou sur un groupe d'usagers.

Modèle et profil utilisateur

De manière générale, un service personnalisé nécessite la connaissance des utilisateurs, ce qui peut être exprimé en tant que modèle utilisateur. Aucun consensus n'existe sur la définition du modèle de l'utilisateur. On peut relever dans la littérature diverses informations qui composent ce modèle :

- Données démographiques : âge, sexe, statut marital, …
- Habitudes comportementales décrivant les interactions fréquentes avec le système
- Préférences et centres d'intérêt
- Données relatives au métier et aux compétences de l'usager

Certains auteurs (Ioannidis et Koutrika, 2005) considèrent les données d'accessibilité (privilèges,…) et le contexte d'utilisation d'un système (connectivité, dispositif utilisé, …) comme une partie du modèle de l'utilisateur. Par ailleurs, Bouzeghoub et Kostadinov (2005) définissent un modèle générique multidimensionnel afin de couvrir une majorité d'informations caractérisant un utilisateur.

Définition. Le modèle utilisateur est un ensemble de caractéristiques, à court et à long terme, permettant de configurer ou d'adapter le fonctionnement d'un système à l'utilisateur afin de lui fournir un accès personnalisé à l'information.

Ainsi, certains éléments du modèle de l'utilisateur sont des informations de durée de vie limitée. L'exemple le plus connu est celui des préférences exprimées dans les requêtes de bases de données à l'aide d'opérateurs spécifiques (Chomicki, 2003; Kießling, 2002).

Dans la littérature, le « modèle utilisateur » est parfois confondu au « profil utilisateur ». A notre sens, il s'agit de deux concepts différents. D'une part, un profil utilisateur comporte des informations à long terme, donc représente une partie du modèle de l'utilisateur. D'autre part, il s'agit de concepts de niveaux d'abstraction différents. Le modèle utilisateur est une vision conceptuelle de connaissances sur l'utilisateur, tandis que le profil représente une implémentation de ce modèle qui peut varier en fonction du domaine d'application et des choix techniques.

Définition. Un profil utilisateur est une instance du modèle de l'usager qui est stockée d'une façon permanente.

La représentation du profil d'un utilisateur varie selon le domaine d'application. Par exemple, en recherche d'information, les profils sont généralement représentés sous forme de mots clés pondérés (Ferreira et Silva, 2001; Soltysiak et Crabtree, 1998) ou d'un ensemble de fonctions d'utilité sur un domaine d'intérêt (Cherniack et al., 2003), tandis qu'en bases de données, les profils peuvent contenir des conditions de sélection ou de jointure des requêtes SQL (Koutrika et Ioannidis, 2004, 2005a).

Les données du profil peuvent être entrées manuellement par l'utilisateur, ou inférées automatiquement à partir de ses interactions précédentes avec le système.

Services de la personnalisation

Les systèmes de personnalisation exploitent des profils utilisateur afin d'offrir différents services.

- Filtrage des résultats. Ce service permet d'affiner les résultats de requêtes en écartant l'information non pertinente. Deux méthodes sont possibles : affiner la requête avant son exécution afin de réduire le résultat (Koutrika et Ioannidis, 2004, 2005a), ou exécuter la requête puis appliquer un post traitement sur le résultat afin d'éliminer les résultats non pertinents (Bradley et al., 2000).
- Tri des résultats. Le principe du tri est de présenter en premier niveau les informations les plus pertinentes afin de rendre les données sélectionnées intelligibles à l'utilisateur (Sun et al., 2008).
- Recommandation. C'est un service de personnalisation visant à proposer les éléments d'information qui sont susceptibles d'intéresser l'utilisateur.

3.2 Recommandation

Robin Burke (Burke, 2002) définit d'une manière générale un système de recommandation comme « tout système capable de générer des recommandations ou permettant de guider l'utilisateur vers des objets utiles au sein d'un espace de données important ».

Les systèmes de recommandation sont souvent utiles dans la prise de décisions. Dotés de compétences cognitives, les moteurs de recherche de données sont capables de prédire l'intérêt d'un usager à des informations, d'exprimer des opinions et d'influer les choix de l'usager. Par exemple, en commerce électronique, ces systèmes proposent à l'internaute des objets de types variés tels que des films (Miller et al., 2003), des livres (Mooney et Roy, 2000), des articles scientifiques (Pavlov et al., 2004), des pages Web (Pitkow et Pirolli, 1999), etc. En base de données, le système recommande des tuples (Stefanidis et al., 2009).

Les approches de recommandation sont généralement classées en fonction de leurs algorithmes de calcul en des approches basées sur le contenu et des approches de filtrage collaboratif (Adomavicius et Tuzhilin, 2005).

La recommandation basée sur le contenu consiste à proposer à l'usager des objets qui sont similaires à ceux qu'il a appréciés dans le passé (Maes, 1994 ; Pazzani et Billsus, 2007 ; Zhang et al., 2002). Une mesure de similarité entre les objets est souvent utilisée afin d'identifier ceux qui sont susceptibles d'être utiles pour l'usager.

Le filtrage collaboratif, exploite les appréciations d'une communauté d'utilisateurs sur les objets afin de découvrir des corrélations entre les utilisateurs. L'utilisateur courant se verra recommander des objets que des utilisateurs similaires ont appréciés (Konstan et al., 1997 ; Resnick et al., 1994). Ainsi une mesure de similarité entre utilisateurs est généralement établie (Satzger et al., 2006).

Des approches qualifiées d'hybrides combinent entre le calcul basé sur le contenu et le filtrage collaboratif (Balabanovic et Shoham, 1997). D'autres approches de recommandation basées sur les données démographiques ou sur les connaissances de l'usager ont été néanmoins proposées (Burke, 2002).

3.3 Personnalisation et recommandation dans cette thèse

Certains travaux limitent la définition de la personnalisation de l'accès à l'information à l'adaptation du résultat d'une requête au profil utilisateur. Dans cette thèse, nous considérons une définition plus générale de la personnalisation dans laquelle le service de recommandation est inclus. Une recommandation ne peut sortir du cadre de la personnalisation. Même si la génération d'une recommandation n'est pas totalement basée sur le profil de l'usager, elle serait calculée selon des heuristiques ou par des fonctions d'utilité qui considèrent indirectement des propriétés du profil.

Ainsi, la recommandation représente un service de la personnalisation. Cependant, une recommandation peut être personnalisée après sa génération en triant ou en filtrant la liste de ses éléments.

4 Problématique : personnalisation de l'interrogation des BDM

Les systèmes OLAP sont élaborés pour un sujet d'analyse (« subject-oriented » (Inmon, 1996)) ou pour un groupe de décideurs (métier, département, …) pour lesquels sont présumés des besoins parfaitement identiques (Rizzi et al., 2006). Cette simplification rend les systèmes OLAP parfois mal adaptés à un usage particulier.

Le décideur qui essaie de trouver des données expliquant un phénomène spécifique ne sait pas à priori ce qu'il cherche exactement. Confronté à un espace multidimensionnel, souvent très vaste, le décideur est obligé de poser plusieurs requêtes afin d'obtenir un résultat le plus proche possible de son besoin. Ainsi, la recherche d'information dans une BDM se déroule en plusieurs étapes et exige souvent un effort non négligeable. De plus, il est parfois difficile aux décideurs de traduire leurs besoins d'analyse par des requêtes textuelles structurées ou graphiques. Ceci nécessite une maitrise d'un langage d'interrogation et une compréhension approfondie du schéma multidimensionnel. Or, le décideur, qui est généralement un utilisateur non informaticien, n'est pas souvent en mesure de maitriser parfaitement le schéma de la BDM qui intègre plusieurs structures multidimensionnelles complexes (Garrigós et al., 2009). Ces problèmes gênent clairement l'exploration des BDM et réduisent les avantages d'employer un système OLAP. Ainsi, les systèmes OLAP doivent faciliter la tâche du décideur en l'assistant durant le processus d'analyse décisionnelle.

Au-delà des problèmes d'exploration des données, bien qu'un système OLAP permette de définir des requêtes sur de gros volumes de données, l'adaptation du résultat de ces requêtes aux besoins spécifiques de chaque décideur et la restriction du résultat massif aux données les plus pertinentes reste un grand défit. En effet, les BDM stockent généralement de gros volumes de données dans le but d'être analysées par différents décideurs. Or, ces décideurs ont souvent différentes perceptions des données en fonction de leurs centres d'intérêts et de leurs objectifs d'analyse (Rizzi, 2007). Par exemple, deux décideurs différents peuvent s'attendre à des résultats différents pour la même requête. Or, il est difficile d'envisager plusieurs BDM qui soient conformes aux attentes typiques de chaque décideur. En effet, la conception d'une BDM est une tâche complexe qui à l'heure actuelle ne repose pas sur des formalismes et une démarche standards (Rizzi et al., 2006). De plus, l'implantation d'un système OLAP composé de nombreuses BDM nécessite la mise en place de processus ETL d'alimentation et de rafraîchissement des données lourds (Vassiliadis et al., 2002) ainsi que des efforts de maintenance importants. Les systèmes OLAP doivent évoluer pour permettre un accès personnalisé à l'information afin d'aider les usagers partageant la même BDM à trouver rapidement les données pertinentes.

Ainsi, grâce à leur modèle multidimensionnel et à leur langage d'analyse en ligne, les systèmes OLAP actuels sont capables de répondre aux questions du type « quoi ? » (données modélisées en sujets et axes d'analyse) et « par quel moyen ? » (manipulation des données à l'aide d'opérations spécifiques), mais ils ne sont pas en mesure de prendre en compte d'autres aspects, notamment « pour qui ? » (mieux connaitre les usagers) et « comment ? » (les assister à trouver les données pertinentes).

Les systèmes OLAP doivent non seulement pouvoir intégrer et organiser des données multidimensionnelles selon des modèles appropriés, mais doivent aussi offrir des analyses « à la carte » de ces données. Nous souhaitons adapter les systèmes OLAP aux besoins spécifiques des décideurs afin de faciliter les analyses OLAP. L'évolution des systèmes OLAP vers des systèmes centrés sur l'usager requiert une connaissance préalable de l'usager. L'ensemble de la problématique de ce mémoire de thèse est résumé ainsi :

- Comment intégrer des informations sur l'usager dans un système OLAP ?
- Comment réduire le résultat d'une requête aux données pertinentes ?
- Comment assister l'usager au cours de son processus d'analyse décisionnelle ?

5 Plan de la thèse

Ce mémoire de thèse est structuré de la manière suivante. Le chapitre 2 est consacré à l'état de l'art. Nous y présentons les notions inhérentes à l'intégration de la personnalisation dans les systèmes OLAP. Ce chapitre commence par présenter un aperçu des travaux sur la modélisation et les langages de manipulation des données OLAP, avec une étude des approches de modélisation des analyses OLAP. Il se poursuit sur les approches de personnalisation des systèmes OLAP puis en particulier sur les travaux de personnalisation de l'interrogation des données OLAP. Le chapitre se termine sur une synthèse des travaux de personnalisation des analyses OLAP et discute les insuffisances et les limites identifiées lors de l'examen de ces approches qui constituent une base pour fonder nos propositions.

Le chapitre 3 présente les modèles de base de nos contributions. D'abord le modèle de données multidimensionnelles adopté dans notre contexte est défini. Ensuite, un modèle générique d'analyse OLAP est proposé. Ce modèle générique met en relief le concept de contexte d'analyse sur lequel seront basés nos différents algorithmes. Une modélisation de préférences contextuelles est ensuite introduite.

Les chapitres 4 et 5 décrivent nos mécanismes de personnalisation basés sur le modèle d'analyse et le modèle de préférences du chapitre 3. Le chapitre 4 présente une approche de personnalisation de requêtes OLAP. Le chapitre 5 introduit un mécanisme de recommandation comprenant trois scénarios couvrant la démarche golbale d'une analyse OLAP.

Le chapitre 6 valide nos travaux à travers la réalisation d'un prototype. Ce prototype étend un outil d'analyse multidimensionnelle afin d'intégrer les fonctionnalités de personnalisation que nous proposons.

Le Chapitre 7 conclue ce mémoire en rappelant nos contributions et les originalités de notre travail. Nous terminons en indiquant les perspectives de recherche envisageables.

Références

Abelló, A., Samos, J., Saltor, F. (2001). Understanding Facts in a Multidimensional Object-Oriented Model. Intl. Workshop on Data Warehousing and OLAP (DOLAP), ACM Press, pages 32–39.

Adomavicius, G., Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering (TKDE), Vol. 17, No. 6, pages 734–749.

Balabanovic, M., Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. Communications of the ACM, Vol. 40, No. 3, pages 66–72.

Bouzeghoub, M., Kostadinov, D. (2005). Personnalisation de l'information : aperçu de l'état de l'art et définition d'un modèle flexible de profils. Conférence en recherche d'infomations et applications (CORIA), pages 201–218.

Bradley, K., Rafter, R., Smyth, B. (2000). Case-Based User Profiling for Content Personalisation. Intl. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems (AH), pages 62–72.

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction, Vol. 12, No. 4, pages 331–370.

Cauvet, C., Semmak, F. (1994). Abstraction Forms in Object-Oriented Conceptual Modeling: Localization, Aggregation and Generalization Extensions. Intl. Conf on Advanced Information Systems Engineering (CAiSE), pages 149–171.

Chaudhuri, S., Dayal, U. (1997). An Overview of Data Warehousing and OLAP Technology. ACM SIGMOD Record, Vol. 26, No. 1, ACM Press, pages 65–74.

Cherniack, M., Galvez, E.F., Franklin, M.J., Zdonik, S.B. (2003). Profile-Driven Cache Management. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 645–656.

Chirita, P.-A., Firan, C.S., Nejdl, W. (2007). Personalized query expansion for the Web. Intl. Conf. on Research and Development in Information Retrieval, pages 7–14.

Chomicki, J. (2003). Preference formulas in relational queries. ACM Trans. Database Syst. 28, 4, pages 427–466.

Codd, E.F., Codd, S.B., Salley, C.T. (1993). Providing OLAP (On Line Analytical Processing) to user analyst: an IT mandate. Rapport technique, E.F. Codd and associates.

Dinter, B., Sapia, C., Höfling, G., Blaschka, M. (1998). The OLAP Market: State of the Art and Research Issues. Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 22–27.

Ferreira J., Silva A. (2001). MySDI: A Generic Architecture to Develop SDI Personalised Services. Intl. Conf. on Enterprise Information Systems (ICEIS), pages 262–270.

Garrigós, I., Pardillo, J., Mazón, J., Trujillo, J. (2009). A Conceptual Modeling Approach for OLAP Personalization. Intl. Conf. on Conceptual Modeling (ER), pages 401–414.

Gray, J., Bosworth, A., Layman, A., Pirahesh, H. (1996). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 152–159.

Gyssens, M., Lakshmanan, L.V.S. (1997). A foundation for multi-dimensional databases. Intl. Conf. on Very Large Data Bases (VLDB), pages 106–115.

Inmon, W.H. (1996). Building the Data Warehouse, John Wiley and Sons, New York, NY, ISBN : 0764599445, 1996 (2ème ed.), 4ème ed. 2005.

Ioannidis, Y., Koutrika, G. (2005). Personalized systems: models and methods from an IR and DB perspective. Intl. Conf. on Very Large Data Bases (VLDB), pages 1365–1365.

Kießling, W. (2002). Foundations of preferences in database systems. Intl. Conf. on Very Large Data Bases (VLDB), pages 311–322.

Kimball, R. (1996). The data warehouse toolkit: Practical Techniques for Building Dimensional Data Warehouses, John Wiley and Sons, ISBN : 0-471-15337-0, 1996, 2ème ed. : Ralph Kimball, Margaery Ross, The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd Edition, John Wiley & Sons, 2002.

Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J. (1997). GroupLens: Applying Collaborative Filtering to Usenet News. Communications of the ACM, Vol. 40, No. 3, pages 77–87.

Koutrika, G. Ioannidis, Y. E. (2004). Personalization of queries in database systems. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 597–608.

Koutrika, G., Ioannidis, Y. E. (2005a). Personalized queries under a generalized preference model. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 841–852.

Koutrika, G., Ioannidis, Y. (2005b). Constrained Optimalities in Query Personalization. ACM SIGMOD Intl. Conf.on Management of Data (SIGMOD), ACM Press, pages 73–64.

Lehner, W. (1998). Modelling Large Scale OLAP Scenarios. Intl. Conf. on Extending Database Technology (EDBT), pages 153–167.

Liu, F.,Yu, C., Andmeng,W. (2004). Personalized Web search for improving retrieval effectiveness. IEEE Transactions on Knowledge and Data Engineering (TKDE), Vol. 16, No. 1, pages 28–40.

Maes, P. (1994). Agents that reduce work and information overload. Communications of the ACM, Vol. 37, No. 7, pages 31–40.

Mangisengi, O., Tjoa, A.M. (1998). A multidimensional modeling approach for OLAP within the framework of the relational model based on quotient relations. Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 40–46.

Miller, B., Albert, I., Lam, S., Konstan, J., Riedl, J. (2003). MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System. Intl. Conf. on Intelligent user interfaces (IUI), pages 263–266.

Mooney, R., Roy, L. (2000). Content-based Book Recommending using Learning for Text Categorization. Intl. Conf. on Digital Libraries (DL), ACM, pages 195–204.

Pavlov, D., Manavoglu, E., Pennock, D., Giles, C. (2004). Collaborative Filtering with Maximum Entropy. IEEE Intelligent Systems, Vol. 19, No. 6, pages 40–48.

Pazzani, M., Billsus, M. (2007). Content-Based Recommendation Systems. The Adaptive Web, pages 325–341.

Pitkow, J., Pirolli, P. (1999). Mining Longest Repeating Subsequences to Predict World Wide Web Surfing. Intl. Conf. on USENIX Symposium on Internet Technologies and Systems (USITS), pages 139–150.

Ravat F., Teste O., Tournier R., Zurfluh G. (2007b). Querying Multidimensional Databases. Conf. on Advances in Databases and Information Systems (ADBIS), Springer-Verlag, pages 298–313.

Ravat F., Teste O., Tournier R., Zurfluh G. (2008). Algebraic and graphic languages for OLAP manipulations. Intl. Journal of Data Warehousing and Mining (IJDWM), Vol. 4, No. 1, pages 17–46.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. ACM Conf. on Cumputer-Supported Cooperative Work, pages 175–186.

Rizzi S. (2007). OLAP preferences: a research agenda. Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 99–100.

Rizzi, S., Abelló, A., Lechtenbörger, J., Trujillo, J. (2006). Research in data warehouse modeling and design: dead or alive? Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 3–10.

Satzger, B., Endres, M., Kießling, W. (2006). A Preference-Based Recommender System. Intl. Conf. on Electronic Commerce and Web Technologies (EC-Web), Springer, Heidelberg, pages 31–40.

Soltysiak S., Crabtree B. (1998). Automatic learning of user profiles - Towards the personalisation of agent services. BT Technology Journal, Vol 16, No 3, pages 110–117.

Stefanidis, K., Drosou, M., Pitoura, E. (2009). "You May Also Like" Results in Relational Databases. Intl. Workshop on Personalized Access, Profile Management and Context Awareness in Databases (PersDB), VLDB Workshops.

Sun, Y., Li, H., Councill, I.G, Huang, J., Lee, W-C., Giles, C. L. (2008). Personalized ranking for digital libraries based on log analysis. Intl. Workshop on Web Information and Data Management (WIDM), ACM, pages 133–140.

Torlone, R. (2003). Conceptual Multidimensional Models. Chapitre III, Multidimensional Databases: Problems and Solutions, Maurizio Rafanelli (Ed.), Idea Publishing Group (IGP), ISBN 1-59140-053-8, pages 69–90.

Vassiliadis, P., Simitsis, A., Skiadopoulos, S. (2002). Modeling ETL activities as graphs. Intl. Workshop on Design and Management of Data Warehouses (DMDW), CAISE Workshops, pages 52–61.

Zhang, Y., Callan, J., Minka, T. (2002). Novelty and Redundancy Detection in Adaptive Filtering. Intl. Conf. on Research and Development in Information Retrieval (SIGIR), ACM, pages 81–88.

Chapitre 2

État de l'art

Sommaire

1 Introduction

Ce chapitre présente l'état de l'art sur l'intégration du concept de personnalisation dans les systèmes OLAP. Il est articulé autour de trois axes principaux :

- Que personnaliser ?
- Par quel biais ? et
- Comment ?

Concernant le premier axe, les travaux de personnalisation en OLAP ont porté aussi bien sur les BDM que sur leurs outils d'interrogation. Afin de bien cerner le cadre d'intégration de la personnalisation, nous présentons dans la section 2 les modèles de données multidimensionnelles, puis les opérations de manipulation et le concept d'analyse relatifs à l'exploration des ces données.

Un modèle utilisateur est généralement employé pour fournir un service personnalisé (*cf.* chapitre 1, section 3.1). A l'exception d'un travail embryonnaire de Garrigós et al. (2009), les travaux sur la modélisation des usagers en OLAP se sont facalisés sur les préférences. La section 3 dresse un panorama de ces travaux.

En ce qui concerne le troisième axe, la section 4 présente un état de l'art général des approches de personnalisation des systèmes OLAP. Puis, la section 5 présente une étude bibliographique détaillée des travaux centrés sur la personnalisation de l'interrogation des BDM. Nous exposons dans la section 6 les constatations de cet état de l'art qui vont conduire à nos propositions.

2 Modélisation et interrogation des bases de données multidimensionnelles

2.1 Modélisation des données OLAP

Issue originellement du monde industriel, la modélisation multidimensionnelle vise à organiser les données de telle sorte que les applications OLAP soient performantes et efficaces (Kimball, 1996). Deux approches existent.

Modélisation en cube

Les premiers modèles proposés reposent directement sur la métaphore de cube (Agrawal et al., 1997 ; Datta et Thomas, 1999 ; Gray et al., 1996 ; Gyssens et Lakshmanan, 1997; Li et Wang, 1996). Cette approche supporte une séparation entre les éléments de structure et les valeurs (Torlone, 2003) : modélisation des axes de l'analyse peu expressive (difficulté à représenter l'organisation hiérarchique des données). Elle se heurte aussi à la représentation des espaces multidimensionnels constitués de plus de trois axes d'analyse. Elle s'avère enfin limitée lorsqu'il s'agit de représenter des constellations de faits et de dimensions potentiellement partagées.

Modélisation multidimensionnelle

Face à ces limites, d'autres approches sont apparues par développement de modèles multidimensionnels (Cabibbo et Torlone, 1998, 2000 ; Golfarelli, et al., 1998 ; Ravat et al., 2008 ; Schneider, 2003) qui distinguent les éléments de structuration des valeurs tout en maintenant un nombre limité de concepts : fait, dimension et hiérarchie. Ces modèles restent spécialisés dans la représentation de données multidimensionnelles et ne reposent pas sur des notations standards (Torlone, 2003). Les concepts et les formalismes associés à la modélisation multidimensionnelle souffrent de l'absence d'un consensus standardisé (Rizzi, et al., 2006).

2.2 Manipulations des données OLAP

La communauté des bases de données multidimensionnelles s'est intéressée à la définition d'un langage d'interrogation des données OLAP. De nombreux opérateurs et langages ont été proposés. Ces propositions visent à répondre aux besoins d'analyse OLAP des décideurs en définissant des opérateurs interactifs facilitant la navigation au sein des données multidimensionnelles (Abelló et al., 2003). Différentes études comparatives ont été réalisées dans (Abelló et al., 2006 ; Rafanelli, 2003 ; Ravat et al., 2007b, 2008 ; Torlone, 2003). Les premiers travaux sur les manipulations OLAP ont étendu les opérateurs de l'algèbre relationnelle pour le modèle en cube (Agrawal et al., 1997 ; Datta et Thomas, 1999 ; Gray et al., 1996 ; Gyssen et Lakshmanan, 1997 ; Li et Wang, 1996).

Pour mieux prendre en compte les structures multidimensionnelles, d'autres travaux ont proposé des opérateurs pour spécifier et manipuler un cube (Abelló et al., 2003 ; Cabibbo et Torlone, 1997 ; Franconi et Kamble, 2004 ; Pedersen et al., 2001). La majorité des travaux reposent sur une structure de visualisation simplifiée dans laquelle le concept de hiérarchie n'est pas exploité. Certaines des propositions (Gyssen et Lakshmanan, 1997) ne supportent qu'un niveau de paramètre en entête des lignes et des colonnes de la structure de visualisation.

Certains travaux (Sarawagi, 1999, 2000; Sathe and Sarawagi, 2001) ont défini des opérateurs avancés pour l'exploration d'un cube de données en se basant sur des techniques de la fouille des données (*cf.* section 5.2.3). Ces opérateurs ne sont pas largement utilisés dans les outils commerciaux et au niveau de la recherche à l'image des opérations classiques telles que Drill-down et Roll-up.

En pratique, certains serveurs OLAP (Mondrian[1], Oracle[2]) utilisent des langages qui ne traduisent pas l'aspect navigationnel des analyses OLAP. Nous pouvons citer le langage MDX qui est doté d'une syntaxe de type SQL, ainsi que l'extension de SQL par les opérateurs Roll-up et Cube (Gray et al., 1996) qui a été intégrée à Oracle 11g.

2.3 Analyse des données OLAP

Contrairement à l'interrogation d'une base de données, l'interrogation d'une BDM est une sucession d'opérations d'exploration. Elle est souvent désignée par *analyse*

[1] Mondrian open source OLAP engine, http://mondrian.pentaho.org
[2] Oracle OLAP 11g. http://www.oracle.com/technology/products/bi/olap

multidimensionnelle relativement au type des données manipulées, ou par *analyse OLAP* en faisant référence à la techonlogie utilisée. Bien qu'elle représente le cœur d'un système OLAP, l'analyse OLAP souffre d'un manque de formalisation. Différents travaux la qualifie d'une *navigation* (Dittrich et al., 2005 ; Kumar et al., 2006 ; Sarawagi et al., 1998 ; Thalhammer et al., 2001), un concept emprunté du web. Ainsi, une analyse OLAP se déroule comme suit :

1. Sélection d'une première requête
2. Navigation des données par
   - forage vers le bas
   - forage vers le haut
   - restriction des données
   - etc.

Différentes visions de l'intuition de la navigation sont adoptées :

- Une analyse est basée sur l'objectif (Dittrich et al., 2005 ; Thalhammer et al., 2001). L'usager cherchant à répondre à une question ou à expliquer un phénomène commence par définir une première requête qui traduit son besoin. Puis, il modifie la requête suite à l'observation du résultat. Une requête est généralement la transformation de la requête précédente.
- Une analyse est pilotée par la découverte (Kumar et al., 2006 ; Sarawagi et al., 1998). L'objectif de l'analyse est de détecter les anomalies dans les valeurs de la BDM. L'analyse démarre généralement au niveau hiérarchique le plus haut d'une dimension. Des opérateurs spécifiques sont utilisés en plus des manipulations OLAP classiques afin de découvrir les anomalies cachées.

Une analyse OLAP est considérée comme la transition entre différents états d'analyse. Les approches de représentation des analyses OLAP se distinguent par la représentation d'un état d'analyse et les types de transition possibles.

D'après (Kumar et al., 2006), une analyse OLAP est un chemin au sein du treillis de cuboïdes[3]. La transition d'un nœud à un autre est assurée par un forage vers le bas suivant la dimension du nœud précédent ou par l'ajout d'une nouvelle dimension. Des manipulations OLAP comme la rotation ou les restrictions ne sont pas prises en compte.

D'une manière plus générale, Dittrich et al. (2005) définissent une analyse par un graphe. Chaque nœud traduit un état de l'analyse et est représenté par les deux dimensions affichées et les conditions de restriction courantes. Un arc traduit la requête utilisateur appliquée. Ainsi, la représentation d'un état de l'analyse n'intègre pas les structures multidimensionnelles fait et mesures et se limite à deux dimensions.

Par ailleurs dans (Thalhammer et al., 2001), une analyse est vue comme un graphe de cubes de données, où le nœud racine est le cube le plus général (dont les données sont agrégées selon le niveau le plus général de chaque dimension). L'état d'analyse suivant représente un cube où les données sont agrégées selon un niveau de granularité plus détaillé. Il faut noter que Sathe et Sarawagi (2001) adoptent une vision contraire des transitions d'une analyse OLAP. En effet, le passage d'un état d'analyse (un cube de données) à un autre est effectué par un forage vers le haut.

[3] Un cuboïde est une vue définie par un niveau hiérarchique pour chaque dimension.

Ainsi, les transitions d'un état de l'analyse à un autre traduisent différents sens de navigation. Les données sont explorées du plus général au plus détaillé (Kumar et al., 2006 ; Thalhammer et al., 2001 ; Sarawagi, 2000), du plus détaillé au plus général (Sathe et Sarawagi, 2001), ou sans sens prédéfini (Dittrich et al., 2005).

Dans les travaux étudiés, des restrictions sont imposées sur les transitions d'une analyse. Par exemple, dans (Sarawagi, 2000 ; Thalhammer et al., 2001), la première requête d'une analyse est typiquement définie au niveau le plus haut des hiérarchies des dimensions, tandis que dans (Kumar et al., 2006), la première requête est définie sur une seule dimension. Certains travaux se limitent à l'emploi d'un langage de requêtes particulier. Nous pouvons citer le travail de (Giacometti et al., 2009) où une analyse OLAP est une séquence de requêtes MDX[4].

2.4 Synthèse

Bien qu'il y ait un accord sur les catégories principales des manipulations OLAP, les opérateurs OLAP ne font encore aujourd'hui l'objet d'un consensus au sein de la communauté OLAP (Romero et Abelló, 2007). Par conséquent, divers langages sont proposés à cause de l'absence d'un standard de description des opérateurs OLAP. Ainsi, des problèmes de fiabilité et de réutilisabilité se posent pour toute approche qui est dépendante d'un langage particulier.

Les analyses OLAP sont définies par une sucession d'états d'analyse et de transitions entre ces états. Divers travaux représentent une analyse OLAP de différentes manières sans avoir défini un formalisme. Dans ces travaux, la représentation d'un état d'analyse est restreinte à un sous-ensemble des éléments de la requête (Dittrich et al., 2005 ; Kumar et al., 2006) ou aux données manipulées (Thalhammer et al., 2001). Malgré que les outils OLAP permettent l'analyse des structures de la requête et des valeurs ensemble (par exemple, via une table multidimensionnelle (Gyssen et Lakshmanan, 1997), aucune approche ne permet d'intégrer ces deux niveaux dans le modèle d'un état de l'analyse.

3 Modélisation et exploitation des préférences OLAP

De nombreuses recherches sur la modélisation des préférences ont été menées par les communautés de recherche d'information et de base de données. Toutefois, peu de travaux ont étudié les préférences dans un environnement OLAP. Pourtant, la prise en compte des préférences est nécessaire dans le contexte OLAP afin d'éviter les résultats vides d'une part, et les résultats volumineux d'une autre part (Rizzi, 2007).

Afin de recenser les axes de recherche potentiels sur les préférences, nous faisons référence à certains travaux du domaine des bases de données relationnelles. Une étude détaillée de ces travaux peut être trouvée dans (Stefanidis et al., 2011).

Les rares travaux sur les préférences OLAP (Bellatreche et al., 2005 ; 2006 ; Golfarelli et Rizzi, 2009 ; Ravat et Teste, 2008 ; Xin et Han, 2008) sont classés selon les aspects de modélisation des préférences et les méthodes de leur exploitation.

[4] MultiDimensional eXpressions, http://msdn.microsoft.com/fr-fr/site/aa216767

3.1 Modélisation de préférences

Les modèles des préférences OLAP sont définis à divers niveaux selon deux approches de modélisation différentes.

3.1.1 Niveau des préférences

Au niveau des bases de données, les préférences sont généralement exprimées sur les n-uplets. Ces modèles sont inadaptés aux BDM où les préférences doivent spécifier le chemin des données que l'usager désire analyser (Golfarelli et al., 2011). Ainsi, les modèles des préférences OLAP ont porté sur le schéma ainsi que les valeurs de la BDM.

Les préférences sur le schéma de la BDM sont définies sur deux niveaux :

- les préférences sur les dimensions (Bellatreche et al., 2006) décrivent l'ensemble des dimensions pertinentes pour l'analyse d'un fait.
- les préférences sur les paramètres (Golfarelli et Rizzi, 2009, Ravat et Teste, 2008) spécifient les niveaux de granularité préférés au long d'une dimension.

Les modèles proposées dans (Bellatreche et al., 2006 ; Golfarelli et Rizzi, 2009 ; Xin et Han, 2008) supportent des préférences sur des valeurs de mesures élémentaires (par exemple, *Nb_publis > 2*) ou agrégées (par exemple, *SUM(Nb_publis) > 10*). Il peut s'agir également de préférences sur les valeurs des attributs de dimension (Bellatreche et al., 2006 ; Golfarelli et Rizzi, 2009), par exemple *Année > 2010*.

La sémantique des préférences définies sur le même élément du schéma varie d'une approche à une autre. Considérons par exemple une préférence de l'usager sur l'attribut « Ville ». Selon Ravat et Teste (2008), cette préférence qualifie l'importance de l'attribut ville par rapport aux autres attributs de la dimension géographique. Cette préférence spécifie selon Golfarelli et Rizzi (2009) l'importance des valeurs de mesures agrégées par ville par rapport aux valeurs des mêmes mesures lorsqu'elles sont agrégées selon d'autres attributs.

Il faut noter que dans le modèle de Golfarelli et Rizzi (2009), une préférence sur les valeurs d'un attribut est propagée suivant la hiérarchie. Par exemple, une préférence de la ville de Toulouse siginifie que l'usager préfère, en plus des données de cette ville, les données de la France (pays de cette ville) et des différents départements toulousains.

3.1.2 Formulation de préférences

Les travaux de modélisation des préférences OLAP se sont basés sur des approches initialement définies pour les bases de données. Les préférences sont formulées dans ces approches selon une approche quantitative (Agrawal et Wimmers, 2000; Koutrika et Ioannidis, 2004, 2005a) ou qualitative (Kießling, 2002 ; Chomicki, 2003).

Selon les approches *qualitatives*, les préférences sur un ensemble sont formulées par des relations d'ordre entre ses éléments.

Soit E un ensemble et une relation binaire sur cet ensemble notée « $\leq$ », cette relation est une relation d'ordre si pour tous x, y et z éléments de E :

- $x \leq x$ (réflexivité)
- $(x \leq y \text{ et } y \leq x) \Rightarrow x = y$ (antisymétrie)

- $(x \leq y \text{ et } y \leq z) \Rightarrow x \leq z$ (transitivité)

La plupart des travaux proposent la définition d'un ordre strict partiel sur les attributs de dimensions (Bellatreche et al., 2006) ou sur les n-uplets de la BDM (Golfarelli et Rizzi, 2009). Un ordre partiel strict sur un ensemble E est une relation binaire irréflexive, transitive et asymétrique (Chomicki, 2003) sur les éléments de E.

Les approches *quantitatives* permettent d'exprimer les préférences d'une façon indirecte par l'utilisation de fonctions de score qui associent un nombre réel à chaque n-uplet du résultat de la requête. Dans ce cas, un n-uplet t_1 est préféré à un n-uplet t_2 si son score est supérieur à celui de t_2.

Approches qualitatives Vs. Approches quantitatives

Les approches qualitatives pemettent une formulation relative des préférences à travers une comparaison entre deux éléments (par exemple « je préfère le domaine des entrepôts de données aux réseaux »). Cette formulation est intuitive pour les usagers. Par contre, les approches quantitatives permettent de formuler des préférences d'une manière absolue sur les éléments désirés (par exemple « j'aime beaucoup le domaine des entrepôts de données » et « je préfère le domaine des réseaux avec un degré inférieur »).

En terme d'expressivité, les approches qualitatives sont plus générales puisqu'on ne peut pas traduire toutes les relations d'ordre par des fonctions de score. Cependant, ces approches ne permettent pas de traduire la différence de l'intensité des préférences. Par exemple, elles ne permettent pas de distinguer les préférences « j'aime beaucoup les bases de données » et « j'aime un peu les bases de données ».

Quant à l'acquisition des préférences, il est plus facile de définir des relations de préférences entre des couples d'éléments que de spécifier des scores. Cependant, l'emploi de telles préférences pour ordonner des valeurs agrégées est plus compliqué que l'évaluation d'une fonction de score. Il faut noter qu'une méthode typique pour inférer implicitement les préférences qualitatives est de calculer à partir du log le nombre d'occurrences des éléments sélectionés au passé (Holland et al., 2003). Ces occurrences, qui peuvent être assimilées à des scores de préférences, sont ensuite utilisées pour déduire les relations d'ordre. Récemment Aligon et al. (2011) ont proposé une approche d'extraction de règles d'association à partir de l'historique des requêtes MDX, qui sont ensuite traduites en préférences.

3.1.3 Contextualisation

Une préférence peut être associée à un contexte. Dans ce cas, elle est dite contextuelle (ou conditionnelle). Le contexte d'une préférence définit sa portée, c'est-à-dire l'environnement dans lequel elle doit être prise en compte.

Définition. Une préférence contextuelle est un couple (P, C), où P est une préférence et C est un contexte.

La partie contexte spécifie les conditions sous lesquelles la préférence P sera activée, où P peut être formulée selon une approche quantitative ou qualitative. Dans cette section, nous nous focalisons sur la partie contexte. Comme aucun travail n'a été dédié à la modélisation des contextes en OLAP, nous faisons référence à des travaux dans les domaines connexes afin d'étudier le besoin pour la contextualisation.

La prise en compte du contexte courant de l'usager pour l'adaptation du comportement d'une application a été intensivement étudiée. Plusieurs définitions du contexte ont été proposées (Brown et al. 1997; Schmidt et al. 1999). Une définition générale est la suivante:

« Le contexte est toute information succeptible de caractériser la situation d'une entité. Une entité est une personne, un lieu ou un objet qui est considéré pertinent pour l'interaction entre l'usager et l'application, incluant l'utilisateur et l'application » (Dey, 2001).

Dans cette perspective, Stefanidis et al. (2011) définissent le contexte dans les bases de données par toute information externe à la base qui permet de caractériser la situation de l'usager ou toute information qui peut caractériser les données en soi. Ainsi, nous distinguons entre deux catégories de contextes : les contextes internes et les contextes externes.

Un *contexte interne* est décrit par des attrinbuts ou des conditions sur les valeurs de la base de données ou de la requête. Les contextes internes dans (Agrawal et al. 2006; Chomicki 2003) sont définis par des conditions sur la présence de valeurs spécifiques d'attributs de la base de données.

Un *contexte externe* définit une situation externe à la base de données. Il est typiquement défini par des attributs spécifiques appelés paramètres de contexte (Stefanidis et al., 2007). L'accompagnement de l'usager, son emplacement et la date courante sont des exemples de paramètres de contexte.

Exemple. Considérons une préférence de l'usager pour les publications en entrepôts de données. Un contexte interne associé à cette préférence peut être « l'année de publication est supérieure à 2000 ». Un exemple de contexte externe pour cette préférence est « en réunion de travail ».

La relation entre les préférences et les contextes externes est typiquement représentée par une relation de type M:N selon le formalisme Entité/Association (Holland et Kießling, 2004). L'association de préférences à des contextes internes peut être représentée graphiquement par les réseaux CP-net, Conditional Preference network (Boutilier et al. 2004).

3.2 Exploitation de préférences

Durant un processus de personnalisation basé sur les préférences, différents problèmes se posent par rapport au traitement des préférences : Comment exprimer les préférences ? Comment sélectionner celles qui sont pertinentes ? Comment intégrer ces préférences dans la requête ? Quels impacts auront-elles sur l'exécution de la requête ?

Expression des préférences

Les préférences qui expriment des besoins spécifiques de l'usager à long terme sont stockées. D'autres préférences qui représentent des besoins à court terme sont exprimées explicitement au moment de la requête suivant différentes manières.

Les préférences quantitatives sont formulées dans (Ravat et Teste, 2008) par des règles ECA (Evènement-Condition-Action) qui associent un poids à un attribut de dimension lorsqu'une opération OLAP est invoquée. Cette action est contrainte par la satisfaction d'une condition.

Exemple. Considérons une préférence de l'usager pour le niveau de granularité *Année* de la dimension temporelle. Cette préférence est décrite par la règle suivante.

CREATE RULE R1 ON Dates

WHEN displayed

THEN priority(Dates. Année, 1);

Golfarelli et Rizzi (2009) ont défini une algèbre de formulation de préférences qualitatives simples (POS, NEG, CONTAIN, …). Cette algèbre est enrichie par deux opérateurs de composition de préférences : Pareto (⊗) pour indiquer le même degré d'importance entre deux préférences et Priorisation (⊳) pour définir un ordre de priorité entre préférences. Ces différents opérateurs sont ensuite intégrés dans une requête MDX à l'aide de clause PREFERRING.

Exemple. La préférence de l'exemple précédent est formulée à l'aide de l'opérateur suivant : CONTAIN(Dates,Année). Lors de la définition d'une requête MDX, cette préférence sera intégrée à l'aide de la clause suivante : PREFERRING Dates CONTAIN Année.

Les préférences de Xin et Han (2008) sont exprimées par des fonctions de score. Par exemple, afin de rechercher les meilleurs doctorants qui publient dans des conférences avec un taux d'acceptation proche de 20% et qui sont organisées depuis 1980, la clause suivante est définie au niveau de la requête SQL :

Order By $(\text{Tx\_accep} - 20)^2 + \alpha(\text{Date\_deb} - 1980)^2$; α étant un paramètre de pondération des deux préférences.

D'autres méthodes d'expression explicite de préférences ont été étudiées en bases de données. Nous pouvons citer les modèles de description logique de Bunningen et al. (2006) et de Chomicki (2003).

Les méthodes explicites demandent un effort de formulation de la part de l'usager qui ne sait parfois pas exprimer ses préférences par un langage descriptif. Afin de remédier à ces problèmes, certains travaux proposent d'acquérir implictement les préférences à partir de l'historique des interactions de l'usager avec le système (Holland et al., 2003). Ces préférences sont stockées dans un profil.

Remarque. Il faut noter que la notion de profil est différente de celle du log. Le log est composé de requêtes ou de sessions de requêtes, alors qu'un profil est composé d'éléments de requêtes (Bellatreche et al., 2006) auxquels sont rajoutées éventuellement des caractéristiques de l'usager telles que ses données démographiques (Garrigós et al. 2009). Certaines approches exploitent directement un log sans construire un profil (Giacometti et al., 2009 ; Chatzopoulou et al., 2009).

Sélection des préférences

Une étape de sélection des préférences est nécessaire pour déterminer celles qui seront utilisées dans le processus de personnalisation.

Une première méthode est centrée sur l'applicabilité de la préférence. Une préférence P est applicable à une requête *Q* si l'exécution de *Q* combinée conjonctivement avec P ne renvoie pas un résultat vide (Golfarelli et al., 2011) ou si le score de P est supérieur à un seuil exprimé dans *Q* (Ravat et Teste, 2008). Nous pouvons aussi citer l'approche de (Cuppens et Demolombe, 1991) où une préférence est applicable à une requête *Q* si son prédicat est impliqué logiquement par un prédicat de *Q*.

Une deuxième méthode a été proposée en bases de données où les préférences sont sélectionnées si leurs contextes apparient avec le contexte de la requête.

- Si une préférence P est rattachée à un contexte interne C, la sélection de P dépend d'une confrontation entre le contexte C et les tuples de la base de données (Stefanidis et al., 2009) ou les attributs de la requête (Agrawal et al. 2006).
- Dans le cas de contexte externe, le contexte courant de l'usager CC(U) est d'abord détecté. Une préférence est sélectionnée si son contexte apparie avec CC(U). Selon (Bunningen et al., 2006 ; Stefanidis et al., 2007), l'appriement de contexte revient à déterminer les contextes qui sont égaux ou plus généraux que CC(U). Par exemple, si l'usager est localisé à Toulouse au moment de la requête, les préférences qui sont associées à la localisation France sont sélectionnées.

Impact des préférences

Les préférences OLAP ont des impacts différents sur la restitution des données. Selon une première approche, les préférences sont intégrées dans la requête en tant que contraintes optionnelles. Les n-uplets qui satisfont les préférences autant que possible sont restituées, même si aucun n-uplet ne satisfait toutes les préférences (Golfarelli et Rizzi, 2009 ; Golfarelli et al, 2011). Dans une deuxième approche, les préférences sont intégrées dans la requête comme des contraintes fortes. Par exemple, les préférences sur les valeurs d'un attribut de dimension sont intégrées en tant que conditions de sélection au sein de la requête initiale (Bellatreche et al., 2006). Les préférences sur les attributs de dimension sont intégrées au niveau de la clause Group-By d'une requête SQL (Ravat et Teste, 2008).

3.3 Synthèse sur la modélisation et l'exploitation des préférences OLAP

En résumé, les modèles de préférences OLAP varient selon le niveau de définition (schéma ou valeurs de la BDM) et la méthode de formulation (quantitative ou qualitative). Les préférences exprimées implictement ou explicitement sont utilisées en tant que contraintes fortes ou optionnelles. Le Tableau 1 donne une vision synthétique de ces approches. A partir de ce tableau, nous pouvons constater des insuffisances au niveau de la modélisation et de l'usage des préférences OLAP.

En terme de modélisation, nous avons relevé le manque de modèles de préférences portant sur toutes les structures multidimensionnelles de la BDM. De plus, nous avons constaté l'absence de modèle de préférence OLAP contextuelle. La prise en compte du contexte permettrait de décrire plus précisément les préférences de l'usager (Garrigós et al., 2009). Il faut noter que malgré de nombreuses propositions de modèles de préférences contextueles en bases de données, aucune n'est parfaitement adaptée au domaine OLAP. Par exemple, dans (Agrawal et al. 2006) la modélisation des contextes internes sous forme de vecteur composé des attributs de la requête et de leurs valeurs ne prend pas en compte l'organisation multidimensionnelle et hiérarchique des stuctures et valeurs d'une BDM.

En ce qui concerne l'usage des préférences, la prise en compte du contexte courant permettrait de restreindre l'ensemble des préférences sélectionnées.

				Bellatreche et al., 2006	Golfarelli et al , 2010, ; Aligon et al., 2011	Ravat et Teste, 2008	Xin et Han, 2008
Modèlisation	Niveau	Schéma	Mesures			×	
			Dimensions	×			
			Paramètres		×	×	
		Valeurs		×	×		×
	Forme			Qual.	Qual.	Quant.	Quant.
	Contextualisation			Non	Non	Non	Non
Exploitation	Expression	explicite			×	×	×
		implicte			×		
	Sélection			Ordre	BMO[5]	Score>seuil	
	Intégration			Condition de sélection	Clause Preferring	Clause Group-By	Clause Order By
	Impact			forte	optionnelle	forte	optionnelle

Tableau 1. Récapitulatif des travaux sur la modélisation et l'usage des préférences OLAP

4 Personnalisation des systèmes OLAP

Selon (Garrigós et al., 2009), la personnalisation est un processus d'adaptation du système OLAP à certaines informations relatives à l'usager telles que ses objectifs, ses caractéristiques, son comportement et son contexte. Ainsi, les travaux sur la personnalisation dans la communauté des bases de données multidimensionnelles ont porté sur les différents niveaux de l'architecture des systèmes OLAP. Nous présentons dans la suite les travaux de personnalisation par niveau : la personnalisation du schéma multidimensionnel, la personnalisation de l'interrogation des données (couvrant les trois services de personnalisation de l'accès à l'information définis dans la section 3.1 du chapitre 1), la personnalisation de la visualisation des données et la personnalisation de la prise de décision.

4.1 Personnalisation du schéma OLAP

Après la conception du schéma de la BDM, les besoins de l'usager peuvent évoluer dans le temps (Favre et al., 2007; Hurtado et al., 1999 ; Garrigós et al. 2009). Les travaux sur la personnalisation du schéma multidimensionnel répondent à la problématique suivante : *Comment adapter le schéma d'une base de données multidimensionnelles aux besoins évolutifs de chaque usager ?*

Deux approches ont été proposées : l'évolution du schéma et la gestion de vues du schéma.

[5] « Best Matches Only »

La première approche est centrée sur la mise à jour des hiérarchies des dimensions. Deux niveaux de mise à jour sont distingués : 1) l'ajout ou la suppression d'un niveau de granularité et 2) l'ajout ou la suppression d'une instance de dimension.

Dans (Blaschka et al., 1999 ; Hurtado et al., 1999), des opérateurs d'évolution sont définis pour mettre à jour les hiérarchies des dimensions, tandis que plus récemment, Favre et al. (2007) ont proposé une approche à base de règles. Les travaux d'évolution du schéma engendrent la mise à jour du schéma physique de la base de données multidimensionnelles. Une étape de maintenance est effectuée en conséquence afin de propager les mises à jour au niveau des agrégats. Ceci implique des mises à jour, qui peuvent s'avérer lourdes et complexes, des processus d'alimentation et de rafraîchissement de la BDM.

La deuxième approche de personnalisation du schéma permet de générer une vue personnalisée du schéma en fonction du profil de l'usager courant (Garrigós et al., 2009). Il s'agit de cacher des éléments de la constellation à l'usager. La personnalisation du schéma est répartie en deux étapes. Les profils utilisateurs sont définis durant la phase de conception. Ils sont ensuite combinés à des règles ECA afin de générer des vues personnalisées du schéma au moment de sa consultation ou de son interrogation. Cette approche n'induit pas de modification du schéma physique de la base. Son inconvénient majeur est l'impossibilité d'ajout d'un élément de la constellation pour répondre à un nouveau besoin de l'usager.

4.2 Personnalisation de l'interrogation des données

Les travaux de personnalisation de l'interrogation des données se situent au niveau « restitution et analyse » du système OLAP (*cf.* chapitre 1, Figure 1). Ils répondent à la problématique suivante :

Parmi les éléments du schéma multidimensionnel et le volume important des données stockées, comment déterminer une requête qui répond au mieux aux besoins de l'usager et comment renvoyer ensuite un résultat pertinent ?

Ainsi, ces travaux permettent de personnaliser l'interrogation du schéma et/ou des instances de la base de données multidimensionnelles. Deux catégories de travaux peuvent être distinguées:

- des travaux permettant la personnalisation des requêtes de l'usager
- des travaux visant à assister l'usager dans la définition des requêtes, appelés communément des travaux de recommandation.

4.2.1 Personnalisation de requêtes

La personnalisation de requête est basée sur le constat que des usagers peuvent juger des résultats différents pertinents lors du requêtage des données (Pitkow et al. 2002). L'objectif de cette approche est de restituer les données les plus pertinentes pour chaque usager.

Définition. La personnalisation de requête est un mécanisme effectué avant ou après l'évaluation de la requête afin de changer la requête ou l'ordre du résultat.

Deux approches de personnalisation des requêtes OLAP ont été proposées:

- Expansion de la requête afin de mieux traduire les besoins de l'usager.
- Tri du résultat en fonction de préférences afin de retourner les meilleurs objets. On parle dans ce cas de requêtes de tri ou de préférences.

Expansion de requêtes

Les méthodes d'expansion des requêtes supposent l'existence d'un profil de l'usager. Des éléments du profil sont utilisés pour étendre la requête de l'usager. Ceci se traduit par l'ajout de conditions de sélection (Bellatreche et al., 2005, 2006), d'attributs d'agrégation (Ravat et al., 2007a), … La version étendue de la requête est ensuite exécutée en substitution de la requête initiale afin de générer un résultat adapté à l'usager.

Le processus de personnalisation de requêtes par expansion se déroule en deux étapes:

- Sélection d'un sous-ensemble des éléments du profil ou des préférences définies en ligne qui sera utilisé pour personnaliser la requête. Notons que plusieurs travaux effectués dans la communauté des bases de données (Agrawal et al., 2006 ; Bunningen et al., 2006 ; Holland et Kießling, 2004 ; Stefanidis et al., 2007) considèrent cette étape comme un processus de résolution de contexte. Il s'agit du cas de préférences contextuelles où celles qui sont rattachées au contexte de la requête sont sélectionnées.
- Intégration des éléments sélectionnés dans la requête initiale.

Exemple. Supposons que la requête *Q* à personnaliser concerne les publications de l'équipe SIG, et que le profil de l'usager indique son intérêt aux publications dans des revues internationales durant l'année en cours. Une requête personnalisée *Q** de la requête *Q* sera les publications de l'équipe SIG dans des revues internationales en 2011.

Requêtes de tri

La catégorie des requêtes de tri la plus étudiée est celles des *requêtes Top-K* qui permettent de renvoyer seulement les k meilleurs objets du résultat (Li et al., 2007a; Li et al., 2007b; Loh et al., 2002; Xin et Han, 2008). Les critères de tri sont définis en tant que conditions faibles (soft) afin de rendre la sélection des n-uplets flexible.

Les requêtes top-k permettent de trier le résultat selon des fonctions de score. Ces fonctions portent sur un ou plusieurs attributs (par exemple, les 10 appartements avec le loyer le moins cher et la surface la plus grande), d'une ou plusieurs tables de la base (par exemple, les 10 appartements avec le loyer et le coût de vie de la ville les moins chers).

Typiquement, une fonction de score est utilisée pour associer chaque n-uplet du résultat avec un score, puis les k n-uplets avec les scores les plus élevés sont restitués. Cependant, d'autres critères de tri du résultat ont été étudiés. Selon Ilyas et al. (2008), il existe trois modèles de requêtes de type top-k selon le type des objets auxquels on attribue les scores: le modèle top-k sélections, le modèle top-k jointures et le modèle top-k groupes d'agrégation.

Dans le modèle *top-k sélections*, les scores sont attachés aux n-uplets de la base. Le processus d'exécution d'une requête est effectué comme suit. Les n-uplets sont sélectionnés selon les conditions de base de la requête. Puis les n-uplets sélectionnés sont triés selon la fonction de score. L'ensemble non vide des meilleurs n-uplets est ensuite renvoyé.

Dans le modèle *top-k jointures*, les scores sont attribués aux résultats de jointures entre différentes tables de la base. Les jointures entre les tables sont effectuées, puis les n-uplets qui résultent des jointures sont triés pour en retourner les meilleurs.

Selon le modèle *top-k groupes d'agrégation*, les scores sont calculés pour des groupes de n-uplets. Le traitement d'une requête nécessite d'abord de calculer les groupes de n-tuples puis les trier selon une fonction de score de groupes (Li et al., 2006), telle que la somme. Le regroupement peut être basé sur les attributs d'agrégation de la requête ou sur d'autres attributs (Li et al., 2007a).

Ainsi, le processus de tri peut être effectué avant (modèle top-k jointures) ou après (modèles top-k sélections et top-k groupes d'agrégation) l'évaluation de la requête.

Il faut noter que lorsqu'il est difficile de définir une fonction de score ou lorsque les préférences ne sont pas précises, d'autres types de requêtes de tri sont utilisés. Il s'agit essentiellement de requêtes Skyline où seulement les objets non dominés du résultat sont retournés. Introduit initialement par Börzsönyi et al. (2001), plusieurs travaux ont étudié l'évaluation et l'optimisation de l'opérateur Skyline.

4.2.2 Recommandation de requêtes

Les approches de recommandation ont été largement étudiées dans le domaine du web. Un état de l'art détaillé peut être trouvé dans (Adomavicius et Tuzhilin, 2005 ; Ioannidis et Koutrika, 2005). Certains travaux se basent sur le paradigme OLAP afin de générer des recommandations (Adomavicius et Tuzhilin, 2001) pour les applications de commerce électronique. Mais, peu de travaux (Giacometti et al., 2008, 2009 ; Sarawagi, 1999, 2000) ont considéré la recommandation dans le contexte OLAP.

Contrairement au domaine du web où les objets recommandés sont de différents niveaux de granularité (un article, plusieurs articles, une page web, …), les travaux en OLAP se focalisent sur la recommandation de requêtes (Marcel et Negre, 2011) afin de prendre en compte l'aspect navigationnel de l'interrogation des données OLAP. Ils répondent à la problématique suivante : « *comment guider l'usager dans l'exploration de la BDM afin de l'aider dans son processus de prise de décision?* ».

Définition. La recommandation de requête est l'action de proposer à l'usager une requête ou des parties de requête d'une manière adaptée à ses intérêts et/ou à son analyse en cours afin de l'assister dans l'exploration des données.

La recommandation de requête fournit deux fonctionnalités : 1) l'assistance à la définition de requête par la proposition de parties de requête et 2) la proposition de requêtes complètes afin de faciliter l'exploration de l'espace multidimensionnel. Cependant, seule la deuxième fonctionnalité a été étudiée par la communauté des bases de données multidimensionnelles (Giacometti et al., 2008, 2009). Aucun travail sur l'assistance à la formulation de requête OLAP n'a été proposé à l'image des travaux sur les requêtes SQL en base de données (Fan et al., 2011).

Outre les approches de recommandation des domaines du web et des bases de données, une nouvelle approche a été proposée en OLAP, à savoir la recommandation basée sur les exceptions (Kumar et al., 2006 ; Sarawagi et al., 1998) permettant de retrouver des données qui sont difficilement repérées à l'aide des analyses OLAP classiques.

Il faut noter que malgré la richesse des travaux sur la recommandation des requêtes SQL de bases de données (Akbarnejad et al. 2010 ; Chatzopoulou et al., 2009 ; Khoussainova et al., 2009), les approches proposées ne peuvent pas être directement appliquées à l'OLAP à cause de l'incohérence de leurs modèles de données avec les concepts de base de l'OLAP, à savoir la modélisation multidimensionnelle.

4.3 Personnalisation de la visualisation des données

Le problème du volume souvent important du résultat des requêtes OLAP a fait l'objet d'étude de la personnalisation de la visualisation dans les BDM (Stolte, 2003). Certains travaux considèrent la personnalisation de la visualisation comme une approche de personnalisation de l'interrogation des données (Golfarelli, 2010). A notre sens, il s'agit d'un axe de personnalisation indépendant. D'une part, la personnalisation de la visualisation relève de l'interaction Homme-Machine et concerne un autre niveau de l'architecture des systèmes OLAP (structures de visualisation). D'autre part, il s'agit d'un mécanisme qui est généralement effectué après la personnalisation de l'interrogation de données.

La personnalisation de la visualisation des données est l'action d'adapter l'interface de visualisation des données en fonction d'un modèle de l'usager ou de critères externes tels que le type et la taille du dispositif utilisé.

Les travaux dans cet axe ont porté sur :

- La personnalisation de la structure d'affichage du résultat d'une requête, en définissant par exemple la disposition des données sur les axes d'une table multidimensionnelle (Bellatreche et al., 2005).
- La mise en valeur de certains indicateurs décisionnels, par exemple les chemins de navigation les plus visités (Garrigós et al., 2009).

4.4 Personnalisation de la prise de décision

Les travaux de personnalisation dans cet axe se situent au niveau de la dernière étape d'un processus de prise de décision.

Une première approche permet d'aider l'usager à mieux interpréter les données multidimensionnelles afin de prendre une décision pertinente. Cabanac et al. (2007) proposent d'intégrer des annotations sur le schéma et sur les valeurs d'une BDM afin de conserver les commentaires, les réflexions et les explications formulés lors des analyses. Ces annotations sont restituées par le système OLAP conjointement aux éléments auxquels elles sont rattachées (Jerbi, 2007). Elles sont utilisées à des fins personnelles (réutilisation des réflexions précédentes) ou collectives (confrontation de différentes interprétations).

Une deuxième approche permet d'aller au-delà de l'assistance à la prise de décision par l'automatisation de la prise de décision (Thalhammer et al., 2001). Des règles d'analyse sont définies par l'usager afin de traduire le processus de prise de décision manuel. Une action, correspondant à une décision (par exemple changer le prix d'un article) est effectuée suite à

un évènement au niveau des bases de données sources (par exemple la diminution de la quantité en stock).

4.5 Synthèse

Les travaux sur la personnalisation des systèmes OLAP ont couvert, au sens général, les différents services de personnalisation que nous avons identifiés dans les domaines de la recherche d'information et des bases de données (*cf.* chapitre 1, section 3.1).

D'autres services ont été étudiés afin de répondre à des problématiques crutiales en OLAP, telles que la personnalisation du schéma multidimensionnel et la personnalisation de la prise de décision.

En ce qui concerne la personnalisation de l'interrogation des données, plusieurs travaux ont été menés sur les requêtes de tri. Cependant, peu d'effort a été consacré à la personnalisation des requêtes par expansion ainsi qu'à la recommandation des requêtes. Pourtant, ces deux techniques représenteraient des solutions potentielles au problème du volume important des données décisionnelles stockées, ainsi que celui des résultats renvoyées par les requêtes multidimensionnelles.

5 Personnalisation de l'interrogation des bases de données multidimensionnelles

Après avoir présenté en section 4 un état de l'art général sur la personnalisation des systèmes OLAP, nous présentons dans cette section une étude plus détaillée qui est centrée sur le niveau de l'interrogation des données.

Nous avons considéré les travaux les plus significatifs qui répondent au mieux à notre problématique de recherche. Nous examinons les travaux retenus selon les axes *approche globale*, *algorithme de personnalisation* et *implantation du système* et relativement à des critères d'évaluation jugés importants pour ces axes.

5.1 Critères d'étude des travaux de personnalisation de l'interrogation des BDM

Soit O l'objet renvoyé par la fonctionnalité de personnalisation, O peut être une requête, un ensemble de requêtes, ou un ensemble de tuples.

Les travaux considérés sont évalués suivant trois axes :

- l'approche de personnalisation, concernant la nature et les caractéristiques de la fonctionnalité de personnalisation qui est offerte à l'usager
- l'algorithme appliqué, précisant comment $\mathcal{O}$ est généré
- l'implantation du système, indiquant comment la fonctionnalité de personnalisation est mise en œuvre

5.1.1 Critères liés à l'approche

Objectif

Rappelons qu'il existe deux objectifs majeurs des approches de personnalisation de l'interrogation des BDM (*cf.* section 4.2) : la personnalisation et la recommandation de requête. Afin de donner une classification détaillée des objectifs, nous distinguerons dans le cas de recommandation entre l'assistance à la définition de requête et la recommandation de requête complète.

Catégorie

Selon le principe de génération du service personnalisé, une approche peut être:

- centrée sur l'utilisateur : un modèle de l'usager est principalement utilisé pour la production du service personnalisé.
- centrée sur les données internes à la BDM (schéma, tuples, métadonnées).
- basée sur l'historique des manipulations des usagers (sans en déduire un modèle de l'utilisateur).

Notons que dans le cas des approches centrées sur les données ou sur l'historique, un service est généré puis est personnalisé en exploitant un modèle utilisateur (personnalisation d'un service généré). Cependant, une approche centrée sur l'utilisateur utilise un modèle utilisateur pour produire un service (génération d'un service personnalisé).

Modèle d'analyse OLAP

Ce critère définit le modèle d'analyse OLAP adopté (*cf.* section 2.3). Nous considérons en particulier le sens de navigation adpoté, le modèle d'un état d'analyse et la présence d'une modélisation générique de l'analyse OLAP.

Proactivité

Ce critère évalue la capacité de l'approche à intervenir activement sans dépendre d'éléments qui ne sont disponibles qu'en temps réel, tels que les résultats de requêtes intermédiaires.

5.1.2 Critères liés à l'algorithme

L'algorithme de personnalisation se déroule en deux étapes : la génération d'un ensemble d'objets $\mathcal{L}$, puis le tri de cet ensemble afin de renvoyer le(s) meilleur(s) objet(s) $\mathcal{O}$.

Entrée et sortie

Nous avons recensé des travaux qui renvoient une requête, un ensemble de requêtes ou un ensemble de tuples.

Timing

L'algorithme de personnalisation peut se dérouler au moment de la conception de la BDM, durant la définition d'une requête, ou après l'application d'une requête.

Méthode de génération

La génération de l'ensemble $\mathcal{O}$ peut être effectuée selon une méthode *descendante* par sélection d'un sous-ensemble à partir d'un ensemble existant ou selon une méthode *ascendante* qui calcule un ensemble complet à partir d'éléments et/ou de sous-ensembles.

Appariement

- Matrice d'appariement

Soit q_i une requête ou une partie de requête. Le problème de personnalisation est représenté par une matrice $M = \{q_i\} \times E$, où $E = \{o_i\}$ est un ensemble d'objets $\mathcal{O}_i$ candidats ou des éléments servant au calcul de $\mathcal{O}_i$. La valeur de $M(q_i,o_i)$ est un booléen. La valeur 0 indique que o_i est exclu du processus de personnalisation.

- Technique

Il s'agit de la technique utilisée pour le calcul d'un ensemble d'objets candidats $\mathcal{O}_1, \ldots, \mathcal{O}_n$ à partir des éléments retenus lors de l'étape d'appariement.

Matrice d'utilité

Ce critère spécifie comment les objets candidats $\mathcal{O}_1, \ldots, \mathcal{O}_n$ sont triés afin de renvoyer le(s) meilleur(s) objet(s).

5.1.3 Critères liés au système

Cet axe concerne la mise en œuvre de la fonctionnalité de personnalisation dans les systèmes OLAP.

Implication de l'usager

Ce critère juge le niveau d'automatisation du système de personnalisation. Nous distinguons des systèmes où les paramètres nécessaires à la personnalisation sont spécifiés par l'usager et des systèmes inférant ces paramètres à partir du profil utilisateur ou du contexte.

Implantation du système

- Généricité

Ce critère décrit l'indépendance de l'approche par rapport aux langages de requête et aux structures de visualisation.

- Performance

Ce critère juge la performance de l'implantation de la fonctionnalité de personnalisation au sein d'un système OLAP. Il ne s'agit pas de mesurer la performance de l'approche mais de recenser les problèmes importants que peut causer l'implantation du service.

- Extesion du moteur de requête

Certaines approches sont implantées au niveau applicatif d'un système OLAP. D'autres approches suggèrent des modifications au niveau du moteur de requête OLAP afin de mettre en œuvre le comportement personnalisé.

5.2 Etude comparative des travaux de personnalisation de l'interrogation des BDM

Dans cette section nous examinons en détail, selon les critères d'étude retenus en section 5.1, des travaux théoriques traitant la conception de systèmes OLAP personnalisés (Garrigós et al. 2009) et des travaux proposant l'intégration de fonctionnalités de personnalisation dans des systèmes OLAP (Bellatreche et al., 2005, 2006 ; Giacometti et al., 2009 ; Golfarelli et al, 2011 ; Ravat et al., 2007a ; Sarawagi, 1998, 1999, 2000; Sathe and Sarawagi, 2001).

Ces travaux traitent la personnalisation dans le domaine OLAP. Il est à noter que dans (Sapia 1999, 2000), une approche de prédiction du prototype de la prochaine requête est proposée afin d'améliorer les performances du système OLAP. Cette approche n'est pas retenue car elle s'inscrit dans une perspective de personnalisation qui est différente de celle étudiée.

Exemple de référence. Afin de mener une étude comparative entre les différentes approches existantes, nous considérons le scénario suivant d'interrogation de la BDM dont le schéma est présenté dans le chapitre 1 (*cf.* chapitre 1, Figure 3) :

Q_E^1 : L'utilisateur demande d'afficher le nombre de publications par année et par ville d'auteur.

5.2.1 Proposition de Golfarelli et *al.*

Les travaux de Golfarelli et al. (Golfarelli et Rizzi, 2009 ; Golfarelli et al, 2011) traitent la problématique des résultats volumineux ou vides des requêtes OLAP. Inspirés des travaux de (Kießling, 2002) en bases de données, ils se basent sur le modèle BMO (« Best Matches Only ») d'exécution de requêtes où seulement les tuples du résultat qui ne sont pas dominés par d'autres sont renvoyés.

En ce qui concerne l'*approche*, il s'agit d'une approche de personnalisation de requête. La requête est exécutée sans personnalisation, puis les préférences sont exploitées pour déterminer les tuples du résultat qui sont meilleurs que tous les autres. Plus précisément, l'exécution de la requête suit une démarche non proactive qui évalue d'une manière incrémentale des requêtes enrichies par des combinaisons de préférences jusqu'à trouver une requête qui renvoie un résultat non vide.

En ce qui concerne *l'algorithme* de personnalisation, il prend en entrée une requête, un ensemble de préférences et l'instance de la BDM. Il produit en sortie l'ensemble des tuples non dominés qui représente un sous-ensemble du résultat généré sans personnalisation. L'algorithme se déroule en deux étapes : la construction du graphe de dominance de préférences, puis le parcours du graphe.

Le graphe de dominance est construit à partir des préférences exprimées dans la requête. Chaque préférence induit le partitionnement des tuples du résultat en plusieurs classes regroupant chacune des tuples équivalents. Chaque nœud du graphe correspond à un ensemble de classes qui ne sont pas dominées par d'autres ; et est représenté par un prédicat permettant de sélectionner les tuples de ses classes. Les arcs traduisent les liens de dominance entre les classes des nœuds. Le parcours du graphe est effectué selon l'ordre de dominance des nœuds. Pour chaque nœud visité, une requête MDX est générée par l'expansion de la requête initiale

avec le prédicat correspondant au nœud. Si l'exécution de la requête enrichie génère un résultat non vide, alors ce résultat est renvoyé à l'usager, sinon le nœud suivant est visité.

D'un point de vue *système*, l'implantation de cette approche impose l'utilisation du langage MDX pour renvoyer des n-uplets avec différents niveaux d'agrégation. Cette implantation a nécessité l'extension du moteur de requête afin de prendre en compte le modèle BMO (Biondi et al., 2011). Il faut noter que l'application de ce modèle pose un problème de performance (Endres et Kießling, 2008; Hafenrichter et Kießling, 2005) à cause de l'exécution de plusieurs requêtes pour rechercher les tuples non dominés. Cependant, Golfarelli et al (2011) ne traitent pas le problème de performance de leur approche.

En conclusion, ces travaux ont les points positifs suivant : la réduction du volume important du résultat et l'expressivité du modèle de préférences. Les points négatifs majeurs sont l'effort manuel et cogintif de formulation des préférences et la nécessité d'exécuter plusieurs requêtes intermédiaires afin de répondre à une seule requête utilisateur.

Exemple. L'usager définit la requête Q_E^1 en l'annotant par les préférences P_1 (CONTAIN[6](AUTEURS,Equipe)) et P_2 (POS[7](Année, 2011)) afin d'indiquer son intérêt au nombre de publications de l'année courante par équipe. La Figure 5 montre le graphe de dominance des préférences où chaque nœud est représenté par le prédicat de la classe qui lui correspond.

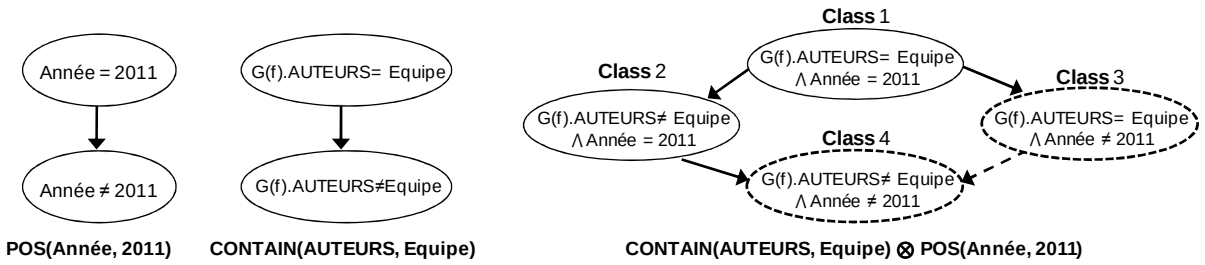


Figure 5. Exemple de graphe de dominance des préférences

La matrice d'utilité est déduite du graphe de dominance et est décrite comme suit, où une cellule (i; j) correspond à l'ordre de priorité de la classe j pour la requête i (Q_E^1):

U	Class 1	Class 2	Class 3	Class 4
Q_E^1	1	2	3	4

La requête Q_E^1 est étendue par le prédicat de *Class 1*. Si le résultat de la requête étendue Q_E^{1*} n'est pas vide, elle est renvoyée à l'usager. Sinon, le prédicat de *Class 2* est intégré dans Q_E^1, etc.

[6] CONTAIN(D_i,a_i) indique que les valeurs de mesure qui sont agrégées suivant l'attribut a_i (et éventuellement d'autres attributs) de la dimension D_i sont préférées aux autres valeurs.

[7] POS(a_i,val_i) indique que les valeurs de mesure qui sont agrégées suivant la valeur v_i de l'attribut a_i sont préférées aux autres valeurs.

SELECT [AUTEURS].[Ville].Members ON COLUMNS, [DATES].[Année].Members ON ROWS FROM PUBLICATIONS WHERE [Measures].NB_PUBLIS PREFERRING **AUTEURS CONTAIN Equipe** AND **Année = 2011**	SELECT [AUTEURS].[**Equipe**].Members ON COLUMNS, {[DATES**].[Année].2011**} ON ROWS FROM PUBLICATIONS WHERE [Measures].NB_PUBLIS
(a) Requête Q_E^1 définie par l'usager	*(b) Requête Q_E^{1*} personnalisée*

Figure 6. Exemple de personnalisation de requête selon (Golfarelli et al, 2011)

5.2.2 Proposition de Bellatreche et *al.*

L'approche de Bellatreche et al. (2005, 2006) permet de personnaliser la requête ainsi que la visualisation du résultat. Nous nous intéresserons uniquement à la partie de ces travaux consacrée à la personnalisation de requête.

Du point de vue de *l'approche*, la personnalisation est centrée sur l'utilisateur. Elle consiste à étendre la requête de l'usager par des prédicats issus du profil. Cette approche s'applique à une requête ponctuelle sans considérer les requêtes précédentes de l'analyse courante. La modélisation de l'analyse OLAP n'est pas étudiée.

Du point de vue de *l'algorithme* de personnalisation, il prend en entrée la requête Q de l'usager, le profil et une contrainte de visualisation. Il génère une requête $Q' \subseteq Q$ au sens de l'inclusion syntaxique des requêtes MDX. Les prédicats qui sont intégrés dans la requête initiale sont de la forme « $A_i = v_i$ » où v_i est un élément d'un ordre strict partiel sur les valeurs de A_i. La recherche des éléments v_i est effectuée en deux étapes. D'abord, les préférences stockées sont comparées à la requête (appariement de type Requête×Utilisateur). Les préférences sélectionnées sont utilisées pour définir un ordre sur les références[8] de la requête. Puis, les meilleures références sont utilisées d'une manière ascendante pour construire la requête personnalisée. Pour chaque référence (r_1, …, r_n) considérée, un prédicat « $R_i = r_i$ » est inséré dans la requête initiale, où r_i est une valeur de l'attribut de dimension R_i.

Concernant le niveau *système*, l'implantation de cette approche est effectuée en dehors du moteur de requête OLAP puisque la requête étendue est envoyée au moteur de requête et est exécutée à la place de la première.

L'avantage majeur de cette approche est le non accès aux tables fait durant le processus de personnalisation. Cependant, l'inconvénient est le chargement en mémoire des tables dimension durant ce processus, ce qui pose un problème de performance. Par ailleurs, ces travaux se limitent au filtrage des valeurs des attributs affichés par la requête. De plus, seules les préférences sous forme de prédicats égalitaires sont prises en compte. Les préférences du type « Nombre de publications > 10 » ne sont pas supportées. Ceci représente une limite pour la personnalisation des valeurs des attributs numériques, notamment les mesures.

Exemple. Considérons le même scénario d'analyse précédent. Supposons que le profil de l'usager stocke les préférences suivantes, $>_p$ étant une relation d'ordre :

P_1 : Auteurs$>_p$Dates ; P_2 : 'Toulouse'$>_p$'Paris'$>_p$'Lyon' ; P_3 : 2011$>_p$2010$>_p$2009$>_p$2008 ; P_4 : IRIT$>_p$LAAS$>_p$ERIC.

[8] Une référence est un N-uplet (r_1, …, r_n) où r_1, …, r_n sont des valeurs des attributs d'agrégation de la requête.

P_4 est rejetée puisqu'elle porte sur l'attribut *institut* qui n'est pas affichée par la requête. Dans la matrice d'appariement présentée ci-dessous, une cellule (i; j) est nulle si la préférence j n'est pas définie sur une dimension ou un attribut de la requête i (Q_E^1).

M	P_1	P_2	P_3	P_4
Q_E^1	1	1	1	0

La matrice d'utilité est traduite comme suit (une cellule (i; j) correspond au classement de la référence j par rapport aux références de la requête):

U	('Toulouse', 2011)	('Toulouse', 2010)	('Paris', 2011)	('Paris', 2010)
Q_E^1	1	2	3	4

Supposons que la contrainte de visualisation limite le nombre de valeurs pour chaque axe à 2. La requête Q_E^{1*} est générée par l'insertion des prédicats « *Ville='Toulouse'* », « *Ville='Paris'* », « *Année=2010* », et « *Année=2011* ».

```
SELECT [AUTEURS].[Ville].Members ON
COLUNMS,
[DATES].[Année].Members ON ROWS
FROM PUBLICATIONS
WHERE [Measures].NB_PUBLIS
```

(a) Requête Q_E^1 définie par l'usager

```
SELECT {[AUTEURS].[Ville].Toulouse,
[AUTEURS].[Ville].Paris} ON COLUNMS,
{[DATES].[Année].2010, [DATES].[Année].2011}
ON ROWS
FROM PUBLICATIONS
WHERE [Measures].NB_PUBLIS
```

(b) Requête Q_E^{1*} personnalisée

Figure 7. Exemple de personnalisation de requête selon (Bellatreche et al., 2005, 2006)

5.2.3 Proposition de Sarawagi et *al.*

A travers une série de travaux (Sarawagi et al., 1998 ; Sarawagi 1999, 2000; Sathe et Sarawagi, 2001), Sarawagi a étudié comment assister l'usager par l'automatisation de l'exploration des données OLAP. Ces travaux considèrent qu'un usager effectue une analyse afin de chercher ou d'expliquer des anomalies au sein des données d'un cube OLAP. La recherche de n-uplets constituant une anomalie ou une explication se fait d'une manière progressive en appariant des sous-cubes de différents niveaux de détail (approche *centrée sur les données*). Le résultat de l'appariement d'une étape détermine s'il faut explorer un sous-cube où un cube plus général dans l'étape suivante (approche *non proactive*).

Du point de vue *algorithme*, nous considérons que les opérateurs proposés sont complémentaires. Nous étudions leurs algorithmes d'une manière intégrée. L'algorithme global prend en entrée le schéma et une instance de la BDM, la requête, et un profil utilisateur stockant les parties de la BDM les plus visitées. Chaque opérateur renvoie à l'usager un ensemble de n-uplets qui peut servir d'entrée à l'opérateur suivant.

- L'opérateur INFORM (Sarawagi, 2000) renvoie les n-uplets qui contiennent des anomalies : les valeurs largement différentes de celles estimées à partir du profil. L'algorithme de cet opérateur est basé sur le principe de l'entropie maximale.
- L'opérateur RELAX (Sathe et Sarawagi, 2001) effectue des forages vers le haut (roll-up) afin de vérifier si une anomalie détectée est confirmée à un niveau plus général.

- L'opérateur DIFF (Sarawagi, 1999) effectue des forages vers le bas (drill-down) et renvoie les n-uplets où les différences de valeurs traduisant l'anomalie sont les plus importantes.

Dans notre contexte, chaque opérateur permet de recommander une ou plusieurs requêtes permettant de retrouver les n-uplets qui constituent l'anomalie ou l'explication.

Concernant le *système*, ces travaux se limitent à l'interrogation de la BDM à l'aide des opérateurs proposés et à l'affichage des données suivant une table bi-dimensionnelle. Le moteur de requête OLAP est étendu afin de prendre en charge ces nouveaux opérateurs. Le fonctionnement de l'opérateur INFORM est coûteux puisqu'il nécessite la comparaison de l'ensemble du cube de données avec les sous-cubes stockés dans le profil. De même, les opérateurs RELAX et DIFF induisent un coût important suite à l'évaluation de plusieurs sous-cubes (vers le haut ou vers le bas) afin de détecter ou d'expliquer les anomalies.

D'un point de vue utilisateur, l'utilisation du système nécessite parfois la définition manuelle des anomalies et la sélection des parties visitées d'une BDM qui sont les plus informatives. Bien que cette approche permette de resencer des anomalies dans un volume important de données agrégées, elle ne traite pas la détection d'anomalies au sein des données non agrégées.

5.2.4 Proposition de Ravat et *al.*

Les travaux de Ravat et al. (Ravat et al., 2007a, Ravat et Teste, 2008) se situent dans le cadre de la personnalisation des requêtes par expansion. Malgré la proposition d'une algèbre d'opérateurs OLAP, cette approche n'est pas basée sur un modèle d'analyse OLAP. La personnalisation d'une requête est indépendante des requêtes déjà lancées au cours de l'analyse courante.

L'algorithme de personnalisation prend en entrée la requête de l'usager, le schéma de la BDM et un profil de l'usager comportant un ensemble de règles. Il génère une requête étendue. Une règle détermine les attributs de dimension à afficher. Ainsi, les règles avec un score supérieur à un seuil sont utilisées pour enrichir la requête.

Du point de vue *système*, il est demandé à l'usager de définir les règles et de spécifier manuellement un seuil de personnalisation. L'implantation de cette approche est effectuée au niveau applicatif par l'ajout d'un module permettant d'analyser la requête et de l'enrichir en fonction des règles.

L'inconvénient majeur de cette approche réside dans la subjectivité de la précision du seuil de la requête qui déterminera les attributs de dimension à afficher. Par ailleurs, les règles sont limitées au niveau des structures d'une BDM. L'approche ne permet pas par exemple de focaliser l'analyse sur l'année en cours.

Exemple. Supposons qu'après l'observation du résultat de Q_E^1, l'usager définit une opération de forage suivant l'axe *Auteurs* afin d'afficher les données par *Institut* (Q_E^2) : DRILLDOWN(T_{RES1}, AUTEURS, Institut, 0.7), T_{RES1} étant la table multidimensionnelle qui résulte de Q_E^1, 0.7 est le seuil de personnalisation.

La requête Q_E^2 est appariée aux règles stockées dans le profil. Une règle est sélectionnée si elle est définie sur les dimensions *Auteurs* et *Dates* de Q_E^1 avec un score qui est supérieur à 0.7. Par exemple, R_1, et R_2 sont définies comme suit :

CREATE RULE R1 ON Auteurs
WHEN displayed
THEN priority(Auteurs.Equipe, 0.9);

CREATE RULE R2 ON Dates
WHEN displayed
THEN priority(Dates.Année, 0.6);

La valeur d'une cellule (i; j) de la matrice d'appariement est égale à 1 si la règle j est sélectionnée par rapport au seuil de la requête i (0.7). Une partie de cette matrice est :

U	R_1	R_2	R_3	R_4
Q_E^2	1	0	0	0

Le système affiche le paramètre *Equipe* suivant l'axe *Auteurs* en plus du paramètre *Institut* qui est explicitement défini dans l'opération. Selon cette approche, toutes les règles sélectionnées sont utilisées dans le processus de personnalisation. Aucun tri des règles n'est effectué.

Remarque. Bien que cette approche permette de recommander des attributs de dimension, elle ne représente pas une approche d'assistance à la formulation de requête. En effet, elle ne permet pas de traiter des requêtes incomplètes. Conformément à notre définition (*cf.* section 4.2.1), il s'agit d'une approche de personnalisation de requête.

5.2.5 Proposition de Garrigós et *al.*

Contrairement aux approches précédentes où la personnalisation est effectuée après la mise en œuvre du système OLAP, le mécanisme de personnalisation selon Garrigós et al. (2009) débute depuis l'étape de conception de la BDM. Durant cette étape, un profil de l'usager est défini ainsi qu'un ensemble de règles ECA précisant les actions de personnalisation à effectuer en ligne. La deuxième étape de ce mécanisme est effectuée au moment de l'interrogation de la BDM. Elle permet de générer une vue du schéma en fonction du profil de l'usager courant afin de faciliter la tâche de formulation de requête. Il s'agit d'une approche d'assistance à la définition de requête.

En ce qui concerne l'*algorithme*, les auteurs proposent de générer une partie du schéma de la BDM (approche descendante) à partir du profil de l'usager et d'un ensemble de règles. Contrairement à (Ravat et al., 2007a) où l'ensemble des règles ECA constitue le profil de l'usager, les règles proposées dans ce travail permettent de mettre à jour le profil et d'adapter le schéma de la BDM en fonction du profil. Un appariement entre les règles et le profil permet de sélectionner celles dont les conditions de déclenchement sont satisfaites.

En ce qui concerne le *système*, les auteurs ne donnent pas d'indications sur l'implantation et la performance de leur approche. Il faut noter qu'aucune extension du moteur de requête n'est obligatoire puisque la personnalisation concerne la consultation du schéma. Cependant, un langage de définition des règles est utilisé.

En conclusion, les auteurs donnent les grandes lignes d'une approche de personnalisation qui est mieux étudiée dans son niveau conceptuel. Malgré la définition des informations que doit comporter un profil (préférences, rôle, contexte, …), les auteurs ne définissent ni le modèle de stockage du profil ni l'algorithme de son exploitation pour la personnalisation.

Exemple. Considérons le même scénario de l'exemple précédent en supposant que l'usager n'a pas encore décidé du niveau de forage. Supposons que la dimension *Auteurs* comporte un

nombre important d'attributs (nom de l'auteur, équipe, institut, ville, région, pays, continent). Le profil comporte le rôle de l'usager (directeur du laboratoire). Soient les règles suivantes :

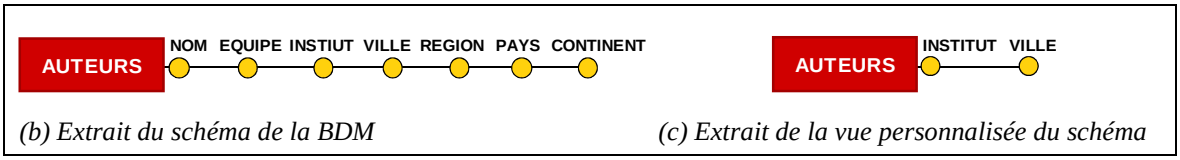
Rule: R1
When Roullup('Publications', 'Auteurs.Institut') do
hideBase(Publications.Auteurs.Ville),
hideBase(Publications.Auteurs.Region),
hideBase(Publications.Auteurs.Continent)
endWhen

Rule: R2
When Display('Publications', 'Auteurs') do
hideBase(Publications.Auteurs.Nom),
hideBase(Publications.Auteurs.Equipe),
hideBase(Publications.Auteurs.Ville),
hideBase(Publications.Auteurs.Region),
hideBase(Publications.Auteurs.Pays),
hideBase(Publications.Auteurs.Continent)
endWhen

Suite à la première requête Q_E^1 qui affiche la dimension *Auteurs*, la règle R_2 est déclenchée afin de spécifier les attributs pertinents. Une partie de la matrice d'appariement suite à la requête Q_E^1 est la suivante, sachant que les règles R_3 et R_4 sont associées au rôle étudiant.

M	R_1	R_2	R_3	R_4
Q_E^1	0	1	0	0

Lorsque l'usager consulte le schéma pour définir la requête de forage, le système lui génére une vue personnalisée (un extrait est présenté en Figure 8). Ainsi, l'usager pourra facilement identifier le niveau de forage *institut* afin de définir la requête Q_E^2 de l'exemple précédent.



(b) Extrait du schéma de la BDM *(c) Extrait de la vue personnalisée du schéma*

Figure 8. Exemple de personnalisation selon (Garrigós et al., 2009)

5.2.6 Proposition de Giacometti et *al.*

L'*approche* de Giacometti et al. (2008, 2009) permet de recommander la requête suivante durant une analyse OLAP. Une analyse est considérée comme une succession de requêtes MDX, où chaque requête est représentée par un ensemble de références. Malgré son exploitation de l'historique des requêtes de tous les utilisateurs, cette approche n'est pas qualifié de collaborative. En effet, aucun appariement d'utilisateurs n'est effectué afin de déterminer les recommandations. Il s'agit d'une approche *centrée historique* qui sélectionne d'une façon *descendante* une requête à recommander à partir de l'ensemble des requêtes du log.

Du point de vue de *l'algorithme*, il prend en entrée la session courante et un log des analyses effectuées sur la BDM. Il génère un ensemble de requêtes à recommander. L'algorithme sélectionne l'ensemble des analyses du log où apparaît l'analyse courante ou une légère modification de cette analyse. L'appariement des analyses est effectué par le calcul de la distance de Levenshtein (Levenshtein, 1966) entre l'analyse courante et chaque analyse du log. La requête qui succède la requête courante de l'usager dans chaque analyse sélectionnée est considérée comme candidate à la recommandation. Les requêtes candidates sont

ordonnées en fonction de leur proximité avec la requête courante de l'usager. Cette proximité est calculée à l'aide de la distance entre les références des requêtes. L'ordonnancement des requêtes peut être également effectué à l'aide du profil utilisateur en s'inspirant des travaux de (Bellatreche et al., 2006).

En ce qui concerne le *système*, l'algorithme de génération de recommandation est implanté au niveau applicatif. Cependant, cet algorithme nécessite l'appariement de l'analyse courante avec toutes les analyses du log, ce qui induit un temps de calcul important.

L'inconvénient majeur de cette approche est l'absence de la prise en compte de l'utilisateur lors de la génération des recommandations. L'ensemble des requêtes candidates est le même quel que soit l'utilisateur. Par ailleurs, cette approche se limite à la recommandation de requêtes suivant des scénarios d'analyse précédents et ne permet pas de recommander de nouvelles requêtes.

Exemple. Considérons le même scénario de l'exemple précédent. $S_c = <Q_E^1 , Q_E^2>$ représente l'analyse courante. Supposons que les analyses stockées dans le log sont les suivantes: $S_1 = <Q_1^1 , Q_1^2>$; $S_2 = <Q_E^1 , Q_2^2, Q_E^2, Q_2^4>$; et $S_3 = <Q_3^1, Q_E^1, Q_E^2, Q_3^3>$.

Dans la matrice d'appariement présentée ci-dessous, une cellule (i; j) indique la distance entre l'analyse i et l'analyse j. 0 indique que l'analyse i n'apparait pas dans l'analyse j.

M	S_1	S_2	S_3
Sc	0	9	4

Les requêtes qui succèdent Q_E^2 dans chaque analyse sont triées en fonction de la distance de Hamming (Hamming, 1950) entre leurs références. Une cellule (i,j) de la matrice d'utilité suivante indique la distance entre la requête courante Q_E^2 et la requête candidate j. Conformément à cette matrice, l'ensemble ordonné des recommandations est $<Q_3^3,Q_2^4>$.

U	Q_2^4	Q_3^3
Q_E^2	17	11

5.3 Synthèse des approches de personnalisation de l'interrogation des BDM

Dans cette section nous dressons un tableau comparatif des travaux examinés précédemment (*cf.* Tableau 2). Ce tableau montre qu'aucun travail ne couvre l'ensemble des critères définis dans la section 2 de ce chapitre. L'expression «N.D.» signifie qu'aucun élément n'a été défini par rapport au critère énoncé. A partir de ce tableau, nous pouvons constater les insuffisances suivantes :

En ce qui concerne *l'approche*, la proposition d'une approche de personnalisation générique n'est pas traitée. Nous avons constaté l'absence d'une approche mixte qui intègre les deux fonctionnalités de personnalisation de requête et de recommandation pour les analyses OLAP. Une telle approche permettrait d'adapter le résultat de la requête à l'usager, d'une part, et de l'assister à définir la requête suivante, d'autre part. Les travaux de (Giacometti et al., 2009) permettent de personnaliser une requête recommandée sans personnaliser la requête initiale de

l'usager. Nous avons également noté l'absence de travaux sur l'assistance à la formulation des requêtes OLAP.

Concernant *l'algorithme* de personnalisation, aucun travail ne considère les préférences pour générer les recommandations. Par ailleurs, les approches d'appriement proposées ne permettent pas un appariement détaillé de fragments de requêtes.

En ce qui concerne *le système* de personnalisation, à l'exception de (Bellatreche et al., 2005), tous les travaux supportent un seul langage de requête et sont dépendants de la structure de visualisation.

6 Bilan de l'état de l'art

6.1 Conclusion

Dans ce chapitre nous avons présenté un panorama des travaux portant sur la personnalisation des systèmes OLAP. Nous avons pu observer que certains problèmes ont été traités et que d'autres problèmes existent encore. Cependant, nous avons pu remarquer plusieurs insuffisances qui nous paraissent importantes.

Absence d'un modèle générique des analyses OLAP

Les approches de représentation des analyses OLAP actuelles souffrent de l'absence de formalisme et de modèle globaux :

- Un état de l'analyse est réduit soit aux données du résultat, soit aux éléments de la requête de laquelle il résulte. Les modèles d'état d'analyse représentant la requête n'intègrent pas toutes les structures multidimensionnelles d'une requête OLAP.
- Les transitions entre les états d'analyse sont parfois limitées à certaines manipulations OLAP. De plus, des contraintes sur ces transitions limitent les analyses modélisées à des cas particuliers. Par exemple, la première requête dans certains travaux doit explorer les données d'une seule dimension (Dittrich et al., 2005) ou du niveau de granularité le plus général (Thalhammer et al., 2001 ; Sarawagi, 2000).

Par ailleurs, les approches actuelles de représentation des analyses OLAP ne sont pas suffisamment génériques. Certaines restrictions sont imposées sur le langage utilisé, tel que MDX, et sur la structure de visualisation des données, telle que le tableau croisé (Dittrich et al., 2005). Ceci met en question l'intégration d'un modèle d'analyse surtout dans un contexte OLAP où les outils de manipulation souffrent d'une grande hétérogénéité par rapport aux langages proposés et sont caractérisés par une diversité des structures de visualisation des données.

Modélisation partielle des préférences OLAP

Les modèles de préférences actuels ne portent pas sur toutes les structures de BDM et les valeurs en même temps, ce qui limite en conséquence les possibilités de leur usage. Par ailleurs, aucune approche ne permet la modélisation de préférences contextuelles.

			Golfarelli et al.	Bellatreche et al.	Sarawagi et al.	Ravat et al.	Garrigós et al.	Giacometti et al.
Approche	Objectif	Personnalisation de requête	×	×		×		
		Recommandation de requêtes			×			×
		Assistance à la définition de requête					×	
	Catégorie		Centrée utilisateur	Centrée utilisateur	Centrée données	Centrée utilisateur	Centrée utilisateur	Centrée historique
	Modèle analyse OLAP	Sens de navigation	Du plus général au plus particulier	N.D.	Du plus général au plus particulier	N.D.	N.D.	N.D.
		Etat d'une analyse	Table bi-dimensionnelle	N.D.	Table bi-dimensionnelle	Table bi-dimensionnelle	N.D.	{Références de requête MDX}
		Modélisation générique	N.D.	N.D.	Non	Non	N.D.	Non
	Proactivité		Non	Oui	Non	Oui	Oui	Oui
Algorithme	Entrée		- Requête *Q* - Préférences - Instance BDM	- Requête *Q* - Profil - Contrainte - Instance BDM	- Requête *Q* - Profil - Schéma BDM - Instance BDM	- Requête *Q* - Profil - Schéma BDM	- Schéma BDM (*S*) - Profil - Règles ECA	- Schéma BDM - Instance BDM - Log d'analyses - Analyse courante - Requête *Q*
	Sortie		Tuples	*Q'*⊑ *Q*	Tuples	Requête *Q'*	Schéma (*S'*)	Requête *Q'*
	Timing		Après l'évaluation de requête	Avant l'évaluation de requête	Après l'évaluation de requête	Avant l'évaluation de requête	Durant conception, durant définition de requête	Après l'évaluation de requête
	Méthode de génération		Ascendante	Ascendante	Descendante	Ascendante	Descendante	Descendante

	Appariement	Matrice d'appariement	Requête×Tuples	Requête×Utilisateur	Utilisateur×Tuples	Requête ×Utilisateur	Requête ×Utilisateur	Session×Sessions
		Technique	- modèle BMO - graphe de dominance	Calcul de relation d'ordre	Entropie maximale	ECA	ECA	- Distance de Hamming - Distance de Levenshtein
	Matrice d'utilité		Utilisateur× Utilisateur	Référence× Référence	Tuples×Tuples	N.D.	N.D.	Requête×Requête
Système	Implication de l'usager		Semi-automatique	Automatique	Semi-automatique	Manuelle	Manuelle	Automatique
	Implantation du système	Performance	Exécution requêtes intermédiaires	Surcharge mémoire	Exécution requêtes intermédiaires	N.D.	N.D.	Voume très important du log
		Généricité	Dépendance du langage MDX	Générique	- Dépendance du langage - Affichage selon deux dimensions	- Application au niveau algébrique	N.D.	Application aux requêtes MDX
		Extesion du moteur de requête	Oui	Non	Oui	Non	Non	Non
Particularités des travaux	Points positifs		- Expressivité du modèle de préférences	- Réduction du volume du résultat	- Réduction de la charge utilisateur - Recommandation de parties non visitées de la BDM	- Réduction de la charge de l'usager - Simplicité de l'implantation	Réduction de l'espace multidimensionnel à explorer	Aspect collaboratif
	Points négatifs		- Effort manuel et congnitif - Exécution de plusieurs requêtes intermédiaires pour chaque requête utilisateur	- Pas de personnalisation ni des valeurs agrégées ni des valeurs des dimensions non affichées	- Coût de calcul important - Limitée aux données agrégées	- Subjectivité du choix du seuil - Pas de personnalisation des valeurs	- Pas de personnalisation des valeurs	- Génération non personnalisée des recommandations - Pas de recommandation de nouvelles requêtes

Tableau 2. Synthèse des travaux sur la personnalisation de l'interrogation des BDM

Absence d'une approche globale de personnalisation

L'une des limites des approches existantes est le manque de capacité de fournir des services de personnalisation à différents niveaux de détail (une partie d'une requête, une requête complète, …) et à des moments différents de la phase d'analyse (lors de la définition de la requête, après la définition de la requête, après la génération du résultat). Nous avons constaté l'absence d'une approche d'assistance à la définition des requêtes OLAP ainsi que l'absence d'un mécanisme global combinant les fonctionnalités de personnalisation et de recommandation.

6.2 Objectifs de la thèse

Modélisation de l'analyse OLAP

Dans le cadre de nos travaux, nous proposons une modélisation globale des analyses OLAP. D'abord, l'analyse OLAP est effectuée sur un schéma en constellation regroupant plusieurs faits et dimensions. Nous définissons le concept de contexte d'analyse afin de modéliser un état donné de l'analyse OLAP. Ce concept intègre les éléments de la requête ainsi que les données du résultat. Le passage d'un contexte d'analyse à un autre n'est pas restreint à un ensemble de manipulations OLAP, ni à un sens de navigation spécifique.

Afin de rendre tout mécanisme se basant sur le modèle d'analyse OLAP indépendant des choix d'implantation, notre modélisation est indépendante du langage de requête et des structures de visualisation des données. Les transitions sont exprimées à un niveau d'abstraction qui permet la prise en compte de tous les langages et opérations OLAP. Une représentation interne des données est utilisée pour la manipulation et l'affichage des données, rendant possible le choix de toute structure d'affichage.

Modélisation des préférences de l'usager

Afin de répondre à la problématique de modélisation des usagers, nous proposons un modèle de préférence global portant sur les structures et les valeurs d'une BDM. Notre modèle associe chaque préférence à un contexte interne représentant l'état courant de l'analyse.

Personnalisation de l'analyse OLAP

Nous envisageons de définir une démarche de personnalisation globale qui couvre toutes les étapes de l'analyse OLAP. Plus particulièrement, notre démarche permettra d'assister l'usager à formuler une requête OLAP, de personnaliser des requêtes définies par l'usager et de renvoyer, en plus du résultat de requêtes, des recommandations. Divers scénarios de recommandation sont proposés. La recommandation par anticipation permet de guider l'usager vers l'étape suivante d'analyse. La recommandation d'alternative permet de guider l'usager vers des étapes jugées nouvelles dans son analyse. Notre démarche est basée sur des structures et sur des algorithmes qui sont indépendants des langages de requête et des structures de visualisation.

Références

Abelló, A., Samos, J., Saltor, F. (2003). Implementing operations to navigate semantic star schema. Intl. Workshop on Data Warehousing and OLAP (DOLAP), ACM Press, pages 56–62.

Abelló, A., Samos, J., Saltor, F. (2006). YAM2: a multidimensional conceptual model extending UML, Information Systems, Vol. 31, No. 6, pages 541–567.

Adomavicius, G., Tuzhilin, A. (2001). Multidimensional Recommender Systems: A Data Warehousing Approach. Intl. Workshop on Electronic Commerce (WELCOM), Lecture Notes in Computer Science, pages 180–192

Adomavicius, G., Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6, pages 734–749.

Agrawal, R., Gupta, A., Sarawagi, S. (1997). Modeling Multidimensional Databases. Intl. Conf. on Data Engineering (ICDE), pages 232–243.

Agrawal, R., Rantzau, R., and Terzi, E. (2006). Context-sensitive ranking. ACM SIGMOD Intl. Conf. on Management of Data (SIGMOD), pages 383–394.

Agrawal, R., & Wimmers, E. L. (2000). A Framework for Expressing and Combining Preferences. ACM SIGMOD Intl. Conf.on Management of Data (SIGMOD), ACM Press, pages 297–306.

Akbarnejad, J., Chatzopoulou, G., Eirinaki, M., Koshy, S., Mittal, S., On, D., Polyzotis, N., Varman, J.S.V. (2010). SQL QueRIE Recommendations. PVLDB, Vol. 3, No. 2, pages 1597–1600.

Aligon, J., Golfarelli, M., Marcel, P., Rizzi, S., Turricchia, E. (2011). Mining Preferences from OLAP Query Logs for Proactive Personalization. Conf. on Advances in Databases and Information Systems (ADBIS), Springer-Verlag, pages 84-97.

Bellatreche, L., Giacometti, A., Marcel, P., Mouloudi, H., Laurent, D. (2005). A personalization framework for OLAP queries. Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 9–18.

Bellatreche, L., Giacometti, A., Marcel, P., Mouloudi, H. (2006). Personalization of MDX Queries. Journées Bases de Données Avancées (BDA).

Biondi, P., Golfarelli, M., Rizzi, S. (2011). Preference-based datacube analysis with MYOLAP. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 1328–1331.

Blaschka, M. Sapia, C. Höfling, G. (1999). On Schema Evolution in Multidimensional Databases. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer, pages 153–164.

Boutilier, C., Brafman, R. I., Domshlak, C., Hoos, H. H., Poole, D. (2004). CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. Journal of Artificial Intelligence Research, Vol. 21, pages 135–191.

Brown, P., Bovey, J., Chen, X. (1997). Context-aware applications: From the laboratory to the marketplace. IEEE Personal Communications, Vol. 4, No. 5, pages 58–64.

Börzsönyi, S., kossmann, D., Stocker, K. (2001). The skyline operator. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 421-432

Bunningen, A. H., Feng, L., and Apers, P. M. G. (2006). A context-aware preference model for database querying in an ambient intelligent environment. Intl. Conf. on Database and Expert Systems Applications (DEXA), pages 33–43.

Cabanac, G., Chevalier, M., Ravat, F., Teste, O. (2007). An Annotation Management System for Multidimensional Databases. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), pages 89–98.

Cabibbo, L., Torlone, R. (1997). Querying Multidimensional Databases. Intl. Workshop Database Programming Languages (DBPL), pages 319–335.

Cabibbo, L., Torlone, R. (1998). A Logical Approach to Multidimensional Databases. Intl Conf. on Extending Database Technology (EDBT), pages 183–197.

Cabibbo, L., Torlone, R. (2000). The Design and Development of a Logical System for OLAP. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), pages 1–10.

Chatzopoulou, G., Eirinaki, M., Polyzotis, N. (2009). Query recommendations for interactive database exploration. Intl. Conf. on Scientific and Statistical Database Management (SSDBM), IEEE Computer Society, pages 3–18.

Chomicki, J. (2003). Preference formulas in relational queries. ACM Trans. Database Syst. 28, 4, pages 427–466.

Cuppens, F., Demolombe, R. (1991). Extending answers to neighbour entities in a cooperative answering context. Journal of Decision Support Systems, Vol. 7, No. 1, pages 1–11.

Datta, A., Thomas, H. (1999). The cube data model: A conceptual model and algebra for on-line analytical processing in data warehouses. Decision Support Systems (DSS), Vol. 27, No. 3, pages 289–301.

Dey, A. K. (2001). Understanding and using context. Personal and Ubiquitous Computing, Vol. 5, No. 1, pages 4–7.

Dittrich, J-P, Kossmann, D., Kreutz, A. (2005). Bridging the gap between OLAP and SQL. Intl. Conf. on Very Large Data Bases (VLDB), pages 1031–1042.

Endres, M. Kießling, W. (2008). Optimization of Preference Queries with Multiple Constraints. Intl. Workshop on Personalized Access, Profile Management, and Context Awareness (PersDB), VLDB Workshops, pages 25–32.

Fan, J., Li, G., Zhou, L. (2011). Interactive SQL query suggestion: Making databases user-friendly. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 351–362.

Favre, C., Bentayeb, F., & Boussaid, O. (2007). Evolution of Data Warehouses' Optimization: a Workload Perspective. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer, pages 13–22.

Franconi, E., Kamble, A. (2004). A Data Warehouse Conceptual Data Model. Intl. Conf. on Scientific and Statistical Database Management (SSDBM), pages 435–436.

Garrigós, I., Pardillo, J., Mazón, J., Trujillo, J. (2009). A Conceptual Modeling Approach for OLAP Personalization. Intl. Conf. on Conceptual Modeling (ER), pages 401–414.

Giacometti, A, Marcel, P., Negre, E. (2009). Recommending Multidimensional Queries. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer, pages 453–466.

Giacometti, A., Marcel, P., Negre, E. (2008). A Framework for Recommending OLAP Queries. Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 73–80.

Golfarelli, M., Maio, D., et Rizzi, S. (1998). Conceptual design of data warehouses from E/R schemes, Intl. Conf. on System Sciences.

Golfarelli, M. et Rizzi, S. (2009). Expressing OLAP Preferences. Intl. Conf. on Scientific and Statistical Database Management (SSDBM), IEEE Computer Society, pages 83–91.

Golfarelli, M. (2010). OLAP Query Personalization. Journées sur les Entrepôts de Données et l'Analyse en ligne (EDA).

Golfarelli, M., Rizzi, S., Biondi, P. (2011). myOLAP: An Approach to Express and Evaluate OLAP Preferences. IEEE Transactions on Knowledge and Data Engineering (TKDE), Vol. 23, No. 7, pages 1050–1064.

Gray, J., Bosworth, A., Layman, A., Pirahesh, H. (1996). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 152–159.

Gyssens, M., Lakshmanan, L.V.S. (1997). A foundation for multi-dimensional databases. Intl. Conf. on Very Large Data Bases (VLDB), pages 106–115.

Hafenrichter, B., Kießling, W. (2005). Optimization of Relational Preference Queries. Australasian Database Conference (ADC), pages 175–184.

Hamming, R. (1950). Error-detecting and error-correcting codes. Bell System Technical Journal, Vol. 26, pages 147–160.

Holland, S., Ester, M., Kießling,W. (2003). Preference mining: A novel approach on mining user preferences for personalized applications. European Conference on Principles and practice of Knowledge Discovery in Databases (PKDD), pages 204–216.

Holland, S., Kießling,W. (2004). Situated preferences and preference repositories for personalized database applications. In Intl. Conf. on Conceptual Modeling (ER), pages 511–523.

Hurtado, C. A., Mendelzon, A. O., Vaisman, A. A. (1999). Maintaining Data Cubes under Dimension Updates. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 346–355.

Ilyas, I. F., Beskales, G., and Soliman, M. A. (2008). A survey of top-k query processing techniques in relational database systems. ACM Computing Surveys, Vol. 40, No. 4.

Ioannidis, Y., Koutrika, G. (2005). Personalized systems: models and methods from an IR and DB perspective. Intl. Conf. on Very Large Data Bases (VLDB), pages 1365–1365.

Jerbi, H. *(2007). Mémoire d'expertises décisionnelles à base d'annotations. Mémoire Master 2 Recherche, Université Paul Sabatier, Toulouse III, Juin 2007.*

Khoussainova, N., Balazinska, M., Gatterbauer, W., Kwon, Y., Suciu, D. (2009). A case for a collaborative query management system. Biennial Conference on Innovative Data Systems Research (CIDR) .

Kießling, W. (2002). Foundations of preferences in database systems. Intl. Conf. on Very Large Data Bases (VLDB), pages 311–322.

Kimball, R. (1996). The data warehouse toolkit: Practical Techniques for Building Dimensional Data Warehouses, John Wiley and Sons, ISBN : 0-471-15337-0, 1996, 2ème

ed. : Ralph Kimball, Margaery Ross, The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd Edition, John Wiley & Sons, 2002.

Koutrika, G. Ioannidis, Y. E. (2004). Personalization of queries in database systems. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 597–608.

Koutrika, G., Ioannidis, Y. E. (2005a). Personalized queries under a generalized preference model. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 841–852.

Kumar, N., Gangopadhyay, A., Karabatis, G., Bapna, S., Chen, Z. (2006). Navigation Rules for Exploring Large Multidimensional Data Cubes. Intl. Journal of Data Warehousing & Mining (IJDWM), Vol. 2, No. 4, pages 27–48.

Levenshtein, V.I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. Rapport technique, 1966.

Li, C., Wang, X.S. (1996). A Data Model for Supporting On-Line Analytical Processing. Intl. Conf. on Information and Knowledge Management (CIKM), pages 81–88.

Li, C., Chang, K. C.-C., Ilyas, I. F. (2006). Supporting ad-hoc ranking aggregates. ACM SIGMOD Intl. Conf.on Management of Data (SIGMOD), ACM Press, pages 61–72.

Li, C., Wang, M., Lim, L., Wang, H., Chang, K. C.-C. (2007a). Supporting ranking and clustering as generalized order-by and group-by. ACM SIGMOD Intl. Conf.on Management of Data (SIGMOD), ACM Press, pages 127–138.

Li, H-G., Yu, H., Agrawal, D., El Abbadi, A. (2007b). Progressive ranking of range aggregates. and Knowledge Engineering (DKE), Elsevier Science Publishers, Vol. 63, No. 1, pages 4–25.

Loh, Z.X., Ling, T.W., Ang, C-H. Lee, S.Y. (2002). Adaptive Method for Range Top- k Queries in OLAP Data Cubes. Intl. Conf. on Database and Expert Systems Applications (DEXA), pages 648–657.

Marcel, P., Negre, E. (2011). A survey of query recommendation techniques for datawarehouse exploration. Journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA).

Pedersen, T.B., Jensen, C.S., Dyreson, C. E. (2001). A foundation for capturing and querying complex multidimensional data. Information Systems (IS), Vol. 26, No. 5, Elsevier, pages 383–423.

Pitkow, J. E., Schutze, H., Cass, T. A., Cooley, R., Turnbull, D., Edmonds, A., Adar, E., Breuel, T. M. (2002). Personalized Search. Communications of the ACM, Vol. 45, No. 9, pages 50–55.

Rafanelli, M. (2003). Operators for Multidimensional Aggregate Data. Chapitre V, Multidimensional Databases: Problems and Solutions, IGI Publishing Group, ISBN 1-59140-053-8, pages 116–165.

Ravat F., Teste O., Zurfluh G. (2007a). Personnalisation de bases de données multidimensionnelles. Congrès INFormatique des ORganisations et Systèmes d'Information et de Décision (INFORSID), pages 121–136.

Ravat F., Teste O., Tournier R., Zurfluh G. (2007b). Querying Multidimensional Databases. Conf. on Advances in Databases and Information Systems (ADBIS), Springer-Verlag, pages 298–313.

Ravat F., Teste O. (2008). Personalization and OLAP Databases. Annals of Information Systems, Vol. 3, Numéro spécial "New Trends in Data Warehousing and Data Analysis", pages 1–22.

Ravat F., Teste O., Tournier R., Zurfluh G. (2008). Algebraic and graphic languages for OLAP manipulations. Intl. Journal of Data Warehousing and Mining (IJDWM), Vol. 4, No. 1, pages 17–46.

Rizzi S. (2007). OLAP preferences: a research agenda. Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 99–100.

Rizzi, S., Abelló, A., Lechtenbörger, J., Trujillo, J. (2006). Research in data warehouse modeling and design: dead or alive? Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 3–10.

Romero, O., Abelló, A. (2007). On the Need of a Reference Algebra for OLAP. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer, pages 99–110.

Sapia, C. (2000). PROMISE : Predicting Query Behavior to Enable Predictive Caching Strategies for OLAP Systems. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer, pages 224–233.

Sapia, C. (1999). On Modeling and Predicting Query Behavior in OLAP Systems. Intl. Workshop on Design and Management of Data Warehouses (DMDW), CAISE Workshops.

Sarawagi, S. (1999). Explaining differences in multidimensional aggregates. Intl. Conf. on Very Large Data Bases (VLDB), pages 42–53.

Sarawagi, S. (2000). User-adaptive exploration of multidimensional data. Intl. Conf. on Very Large Data Bases (VLDB), pages 307–316.

Sarawagi, S., Agrawal, R., Megiddo, N. (1998). Discovery-driven exploration of OLAP data cubes. Intl. Conf. on Extending Database Technology (EDBT), pages 168–182.

Sathe, G., Sarawagi, S. (2001). Intelligent rollups in multidimensional OLAP data. Intl. Conf. on Very Large Data Bases (VLDB), pages 531–540.

Schmidt, A., Aidoo, A. K., Takaluoma, A., Tuomela, U., Laerhoven, K., and de Velde, M. (1999). Advanced interaction in context. Intl. Symposium on Handheld and Ubiquitous Computing, pages 89–101.

Schneider, M. (2003). Well-formed data warehouse structures. Intl. Workshop on Design and Management of Data Warehouses (DMDW), CAISE Workshops.

Stolte, C. (2003). Query, Analysis, and Visualization of Multidimensional Databases, Thèse de doctorat, Université de Stanford (Etats-Unis), Juin 2003.

Stefanidis, K., Pitoura, E., Vassiliadis, P. (2007). Adding context to preferences. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 846–855.

Stefanidis, K., Drosou, M., Pitoura, E. (2009). "You May Also Like" Results in Relational Databases. Intl. Workshop on Personalized Access, Profile Management and Context Awareness in Databases (PersDB), VLDB Workshops.

Stefanidis, K., Koutrika, G., Pitoura, E. (2011). A Survey on Representation, Composition and Application of Preferences in Database Systems. ACM Transactions on Database Systems (TODS), Vol. 36, No. 3.

Thalhammer, T., Schrefl, M., Mohania, M. (2001). Active DataWarehouses. Complement-ing OLAP with Analysis Rules, Data and Knowledge Engineering (DKE), Elsevier Science Publishers, Vol. 39, No. 3, pages 241–269.

Torlone, R. (2003). Conceptual Multidimensional Models. Chapitre 3 de l'ouvrage Multidimensional Databases: Problems and Solutions, IGI Publishing Group(IGP), ISBN 1-59140-053-8, pages 69–90.

Xin, D., Han, J. (2008). P-cube: Answering preference queries in multidimensional space. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 1092–1100.

Chapitre 3

Prise en compte de l'utilisateur dans les analyses OLAP

Sommaire

1 Introduction

Ce chapitre présente un cadre pour la prise en compte de l'usager dans les analyses OLAP.

Exemple illustratif

Afin d'illustrer le problème, nous allons prendre l'exemple de deux responsables d'équipes de recherche. Le premier s'intéresse essentiellement aux publications dans des conférences *IEEE* et *ACM*. Afin de suivre les performances de leurs équipes, les deux responsables souhaitent analyser le nombre de publications de l'année en cours par trimestre et par catégorie de publication. En demandant l'aide de la secrétaire du laboratoire, elle fournit à chacun un rapport qui est conforme à ses besoins spécifiques. Elle se base sur sa connaissance de chacun et sur ses discussions précédentes avec chacun. Cependant, en utilisant un outil d'analyse décisionnelle, le système produit le même résultat aux deux responsables. Le premier responsable est donc obligé de chercher dans l'espace multidimensionnel résultat les données qui correspondent aux catégories *IEEE* et *ACM*. Il faut noter que l'année de publication représente pour le premier responsable un besoin instantané qu'il exprime explicitement dans sa requête au niveau du système. Cependant, les catégories de la publication représentent une information à long terme traduisant un centre d'intérêt du responsable qui le distingue des autres chercheurs de son laboratoire. Par ailleurs, la secrétaire remarque en imprimant le rapport que le nombre de publications est très faible. Grâce à sa collaboration durant plusieurs années avec le deuxième responsable, elle sait qu'il cherchera le groupe de chercheurs inactifs afin d'organiser des réunions de travail. Elle lui fournit donc, en plus du rapport demandé, un rapport détaillant le nombre de publications par chercheur.

Les systèmes OLAP actuels sont malheureusement incapables d'adopter le comportement de recommandation de la secrétaire car ils considèrent généralement seulement la requête de l'usager pour la restitution du résultat. Le système permettrait de personnaliser l'analyse des données OLAP grâce à la connaissance préalable de l'utilisateur.

Objectifs et organisation du chapitre

La définition d'un cadre pour la prise en compte de l'usager dans l'analyse OLAP nécessite en première étape de préciser qu'est-ce qu'on entend par une analyse (personnaliser quoi), donc définir la notion d'analyse OLAP au vu des limites des différentes définitions dans les travaux existants. En deuxième étape, nous envisageons de présenter notre modélisation de l'usager dans le domaine OLAP afin d'expliquer sur quelle base la personnalisation sera effectuée (par quel biais personnaliser). En troisième étape, nous désirons préciser comment pourrait-on personnaliser cette analyse (comment personnaliser).

Premièrement, nous souhaitons que notre modèle d'analyse OLAP prenne en compte une notion très importante dans le domaine OLAP qui est la navigation. Vu l'absence d'un langage standard pour la manipulation des données OLAP d'une part (Romero et Abelló, 2007) et la diversité des formes d'affichage des résultats, d'autre part, notre modèle d'analyse OLAP devra être indépendant des langages de manipulation ainsi que des structures de visualisation des données.

Deuxièmement, la modélisation de l'usager doit répondre à deux problématiques essentielles. Le modèle doit permettre l'intégration de préférences aussi bien sur le schéma de la BDM que

sur ses valeurs. De plus, ce modèle doit permettre la définition de préférences contextuelles afin de traduire au mieux les besoins spécifiques du décideur.

Enfin, une définition du concept de personnalisation en OLAP doit être fournie afin de limiter ses fonctions et de la distinguer de la variété de mécanismes de personnalisation qui ont été proposés dans le domaine des systèmes d'information en général.

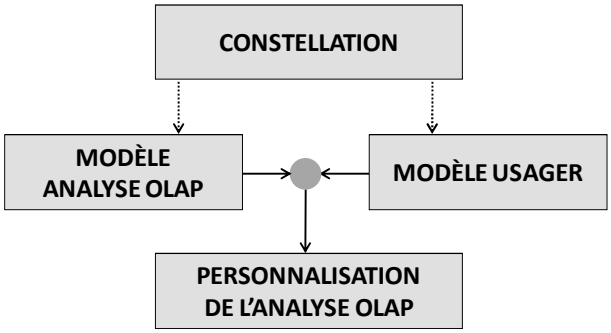


Figure 9. Positionnement de la personnalisation des analyses OLAP

Plan du chapitre. La section 2 présente la modélisation en constellation des données OLAP ainsi que la modélisation des analyses OLAP. La section 3 décrit le modèle de préférences permettant de personnaliser les données OLAP, puis montre un cadre générique pour la personnalisation d'une analyse OLAP.

2 De la constellation à son analyse

Le modèle que nous proposons dans la section 2.1 constitue une généralisation du modèle en étoile (Kimball, 1996). Il s'agit d'un modèle en constellation qui regroupe un ensemble de faits associés à des dimensions qui sont munies de hiérarchies multiples. Cette modélisation permet l'expression d'analyses OLAP multidimensionnelles décrites dans la section 2.2.

2.1 Modélisation des données OLAP

Cette section explicite les concepts OLAP de modélisation des données.

2.1.1 Modèle en constellation

Nous proposons de modéliser les données dans une BDM par extension du concept de constellation (Ravat *et al*., 2008) afin de permettre sa personnalisation. L'extension consiste à intégrer un ensemble de préférences utilisateur. Une constellation est donc composée de faits (F^{CS}) et de dimensions (D^{CS}) interconnectés ($Star^{CS}$) ainsi que de préférences (P^{CS}).

Définition. Une *constellation CS* est définie par (N^{CS}, F^{CS}, D^{CS}, $Star^{CS}$, P^{CS}) où

- N^{CS} est le nom de la constellation,
- $F^{CS} = \{F_1, \ldots, F_n\}$ est un ensemble de faits,
- $D^{CS} = \{D_1, \ldots, D_m\}$ est un ensemble de dimensions,
- $Star^{CS} : F^{CS} \rightarrow 2^{D^{CS}}$ est une fonction associant les faits aux dimensions, et

- $P^{CS} = \{P_1, \ldots, P_p\}$ est un ensemble de préférences de personnalisation.

Une dimension modélise un axe d'analyse. Elle est caractérisée par des attributs organisés au sein d'une ou plusieurs hiérarchies.

Définition. $\forall i \in [1..m]$, une *dimension* D_i est définie par $(N^{Di}, A^{Di}, H^{Di}, I^{Di})$ où

- N^{Di} est le nom identifiant la dimension dans la constellation,
- $A^{Di} = \{Id_i, All\} \cup P_i \cup W_i$ est l'ensemble des attributs de la dimension. On distingue les paramètres P_i représentant les graduations possibles, des attributs faibles W_i représentant des informations additionnelles associées aux paramètres,
- $H^{Di} = \{H^{Di}_1, \ldots, H^{Di}_r\}$ est l'ensemble des hiérarchies, et
- $I^{Di} = \{I^{Di}_1, \ldots, I^{Di}_s\}$ est l'ensemble des instances de D_i.

Une hiérarchie modélise l'organisation des différents niveaux de granularité, à savoir, une vision particulière de la graduation de l'axe.

Définition. $\forall\ H^{Di}_j \in H^{Di}$ une *hiérarchie* H^{Di}_j (notée plus simplement H_j) est définie par $(N^{Hj}, P^{Hj}, \prec_{Hj}, Weak^{Hj})$ où

- N^{Hj} est le nom identifiant la hiérarchie dans la constellation,
- $P^{Hj} = \{p_1, \ldots, p_y\} \subseteq P_i$ est l'ensemble des paramètres de la hiérarchie,
- $\prec_{Hj}$ est une relation d'ordre sur P^{Hj} telle que
  - l'ordonnancement des paramètres suit un ordre total $\forall p_{k1} \in P^{Hj}, p_{k2} \in P^{Hj}, k_1 \neq k_2, p_{k1} \prec_{Hj} p_{k2} \vee p_{k2} \prec_{Hj} p_{k1}$
  - il existe un paramètre racine $\forall p_{k1} \in P^{Hj}, Id_i \prec_{Hj} p_{k1}$
  - il existe un paramètre extrémité $\forall p_{k1} \in P^{Hj}, p_{k1} \prec_{Hj} All$
- $Weak^{Hj}: P^{Hj} \rightarrow 2^{WHj}$ est une application qui associe les paramètres à un ensemble d'attributs faibles, W^{Hj} étant l'ensemble des attributs faibles de la hiérarchie H_j.

Un fait regroupe un ensemble d'indicateurs relatifs à un sujet d'analyse. Il est modélisé au travers de mesures représentant ces indicateurs.

Définition. Un *fait* F_i est défini par le quadruplet $(N^{Fi}, M^{Fi}, I^{Fi}, IStar^{Fi})$ où

- N^{Fi} est le nom identifiant le fait dans la constellation,
- $M^{Fi} = \{m_1, \ldots, m_u\}$ est un ensemble de mesures (ou indicateurs) pouvant être agrégées selon une fonction $f_i \in \{AVG, SUM, MAX, MIN, COUNT, \ldots\}$,
- $I^{Fi} = \{I^{Fi}_1, \ldots, I^{Fi}_w\}$ est l'ensemble des instances de F_i, et
- $IStar^{Fi}: I^{Fi} \rightarrow I^{D1} x \ldots x I^{Dy}$ $(\forall k \in [1..y], D_k \in Star^{CS}(F_i))$, est une fonction associant les instances de I^{Fi} aux instances correspondantes des dimensions liées au fait.

Notre modèle supporte les faits sans mesures ($M^{Fi} = \varnothing$). Dans ce cas, seule l'analyse du nombre d'occurrences est possible.

2.1.2 Cas d'étude

Dans la suite du mémoire, nous considérons l'exemple de base de données des publications d'un laboratoire de recherche comme une étude de cas illustrant nos différentes propositions. Cette BDM permet l'analyse des *publications* ainsi que le suivi des *missions* de recherche des membres du laboratoire selon les axes d'analyse *Dates*, *Manifestations* et *Auteurs*. Les *publications* peuvent être analysées en plus selon les *domaines* de recherche. Chaque

publication est associée un *domaine principal* de recherche (par exemple, système d'information) et à un *sous-domaine* (par exemple, base de données). Au niveau de la dimension *Manifestations*, l'attribut *Catégorie* représente la catégorie de la publication (IEEE, ACM, ….), alors que le *Type* détermine si la manifestation est une conférence nationale, une conférence internationale, une revue nationale, une revue internationale, ... Une manifestation peut être de *Niveau* national ou international. La hiérarchie *HPOS* de la dimension *Auteurs* permet l'analyse des publications ainsi que le suivi des missions selon les *statuts* (permanent, non permanent) ou les *Postes* des auteurs (professeur, maître de conférences, doctorant, …).

Le schéma en constellation de cette BDM est représenté dans la Figure 10 selon nos formalismes graphiques que nous définissons par extension des notations introduites dans (Golfarelli et al., 1998). Un fait est représenté par un rectangle vert (gris clair) comprenant les noms de ses mesures. Il est lié à ses dimensions. Chaque dimension est représentée par un rectangle rouge (gris foncé). Un paramètre est représenté par un rond jaune (gris clair) accompagné de son nom et un attribut faible est représenté par un segment rattachant son intitulé au paramètre auquel il est associé. *All* étant un paramètre système, n'est pas représenté graphiquement.

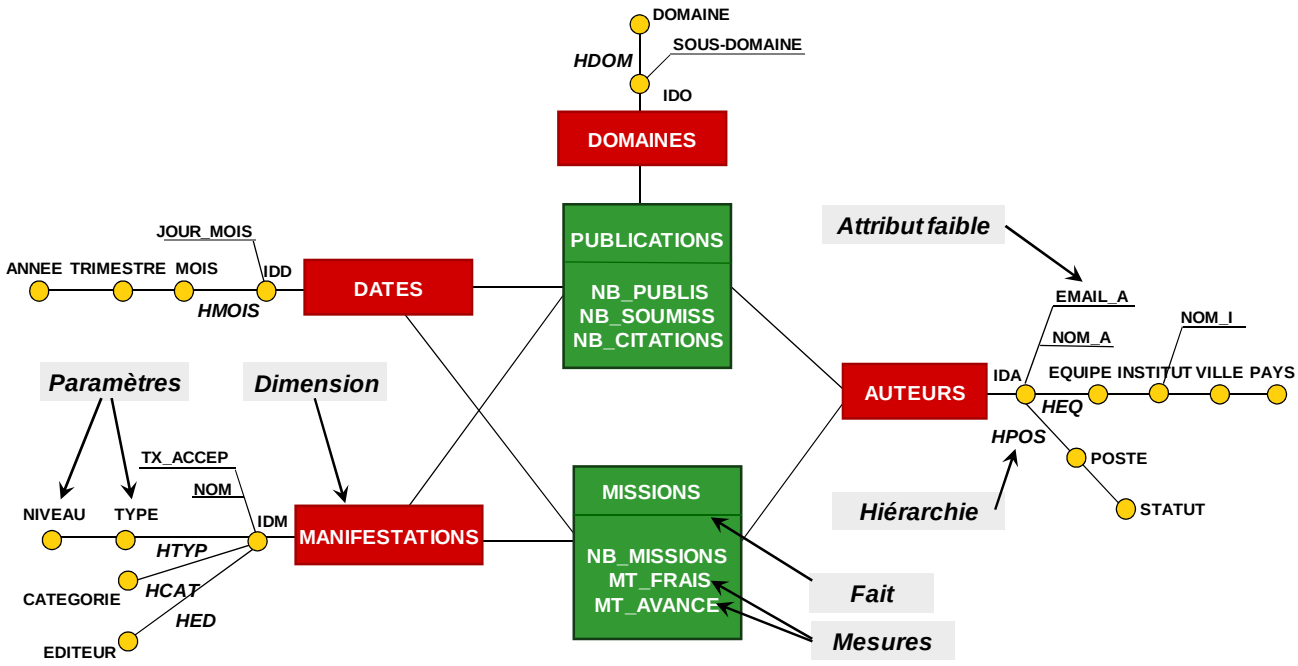


Figure 10. Exemple de schéma en constellation d'une base de données multidimensionnelle

Cette constellation *RECH* comporte deux faits et trois dimensions. Elle est définie par (N^{RECH}, F^{RECH}, D^{RECH}, $Star^{RECH}$, P^{RECH}) où :

- N^{RECH} = 'RECH' ;
- $F^{RECH} = \{F^{PUBLI}, F^{MISS}\}$;
- $D^{RECH} = \{D^{DAT}, D^{AUT}, D^{MANIF}, D^{DOM}\}$;
- $Star^{RECH}(F^{PUBLI}) \rightarrow \{D^{DAT}, D^{AUT}, D^{MANIF}, D^{DOM}\}$; $Star^{RECH}(F^{MISS}) \rightarrow \{D^{DAT}, D^{AUT}, D^{MANIF}\}$;
- $P^{RECH} = \{ P^{RECH}_1, \ldots, P^{RECH}_m \}$.

La dimension D^{AUT}, appelée *Auteurs*, est définie par ($N^{D^{AUT}}$, $A^{D^{AUT}}$, $H^{D^{AUT}}$, $I^{D^{AUT}}$) où

- $N^{D^{AUT}}$ = 'Auteurs' ;
- $A^{D^{AUT}}$ = {Equipe, Nom_A, Email_A, Institut, Nom_I, Ville, Pays, Poste, Statut} ∪ {IdA, All} ;
- $H^{D^{AUT}}$ = {H^{POS}, H^{EQ}} ;
- $I^{D^{AUT}}$ = {i^{AUT}_1,…, i^{AUT}_n}.

Cette dimension est composée des hiérarchies H^{POS} et H^{EQ}. La hiérarchie H^{EQ} est définie par :

- N^{HEQ} = 'HEq' ;
- P^{HEQ} = {IdA, Equipe, Institut, Ville, Pays, All} ;
- IdA $\prec_{HEQ}$ Equipe $\prec_{HEQ}$ Institut $\prec_{HEQ}$ Ville $\prec_{HEQ}$ Pays $\prec_{HEQ}$ All ;
- $Weak^{HEQ}$: IdA→{Nom_A, Email_A} ; Institut→{Nom_I}.

Les publications des chercheurs du laboratoire peuvent être étudiées selon le fait F^{PUBLI} = (N^{FPUBLI}, M^{FPUBLI}, I^{FPUBLI}, $IStar^{FPUBLI}$) où :

- $N^{F^{PUBLI}}$ = 'Publications' ;
- $M^{F^{PUBLI}}$ = {Nb_Publis, Nb_Soumiss, Nb_Citations} ;
- $I^{F^{PUBLI}}$ = {i^{PUBLI}_1,…, i^{PUBLI}_y} ;
- $IStar^{F^{PUBLI}}$ = {$i^{PUBLI}_k \rightarrow (i^{AUT}_{k_i}, i^{DAT}_{k_j}, i^{MANIF}_{k_p}, i^{DOM}_{k_l}) \mid \forall k \in [1..y], i^{PUBLI}_k \in I^{F^{PUBLI}} \wedge \exists\, i^{AUT}_{k_i} \in I^{D^{AUT}} \wedge \exists\, i^{DAT}_{k_j} \in I^{D^{DAT}} \wedge \exists\, i^{MANIF}_{k_p} \in I^{D^{MANIF}} \wedge \exists\, i^{DOM}_{k_l} \in I^{D^{DOM}}$}.

2.2 Analyse en ligne OLAP

L'analyse en ligne OLAP consiste à explorer interactivement l'espace multidimensionnel d'une constellation afin d'obtenir ou d'expliquer des résultats. Elle s'apparente à une navigation où chaque étape représente un état de l'analyse et la transition d'un état à un autre est effectuée par une opération de manipulation OLAP (*cf.* chapitre 1, section 2.3.1). Nous définissons une représentation de l'analyse OLAP générique qui admet les avantages suivants :

- Chaque état de l'analyse est traduit par le concept de contexte d'analyse qui regroupe les éléments de la requête utilisateur et son résultat. L'ensemble est représenté par une vue interne qui est indépendante des structures de visualisation.
- Les transitions d'un état d'analyse à un autre sont assurées par des opérations de transformation des données définies indépendamment du choix de l'algèbre et du langage pour lesquels il n'existe pas encore de standard.

2.2.1 Modélisation de l'analyse OLAP

Le décideur procède à différentes analyses OLAP au cours du temps. Le but de chaque analyse est de répondre à un besoin décisionnel particulier.

Exemple: Supposons qu'un décideur explore la BDM présentée en Figure 10 afin de répondre à la question suivante : « Quels sont les catégories de chercheurs en baisse d'activité de recherche et plus particulièrement, dans quel type de manifestations ? ». Il procède aux étapes suivantes présentées en Figure 11 (l'algèbre OLAP utilisée est présentée en Annexe 1).

- En affichant le nombre de publications par type de poste du premier auteur et par année, le décideur remarque que ce nombre est faible pour les maîtres de conférences durant les deux dernières années.
- Il se focalise alors sur les publications des maîtres de conférences et observe le nombre de publications durant les deux dernières années.
- Ensuite, il analyse le nombre détaillé par trimestre. Le décideur remarque que les publications des maîtres de conférences sont nettement moins nombreuses durant le troisième trimestre.
- Il demande d'afficher leur nombre de publications par trimestre par niveau de manifestations.
- Le nombre étant faible pour les manifestations nationales, il détaille l'analyse par type pour voir si cette tendance se retrouve quelque soit le type de publication.
- En observant le dernier résultat, le décideur peut conclure que les chercheurs les moins actifs sont les maîtres de conférences, plus particulièrement en termes de publication dans les ateliers et conférences nationaux.

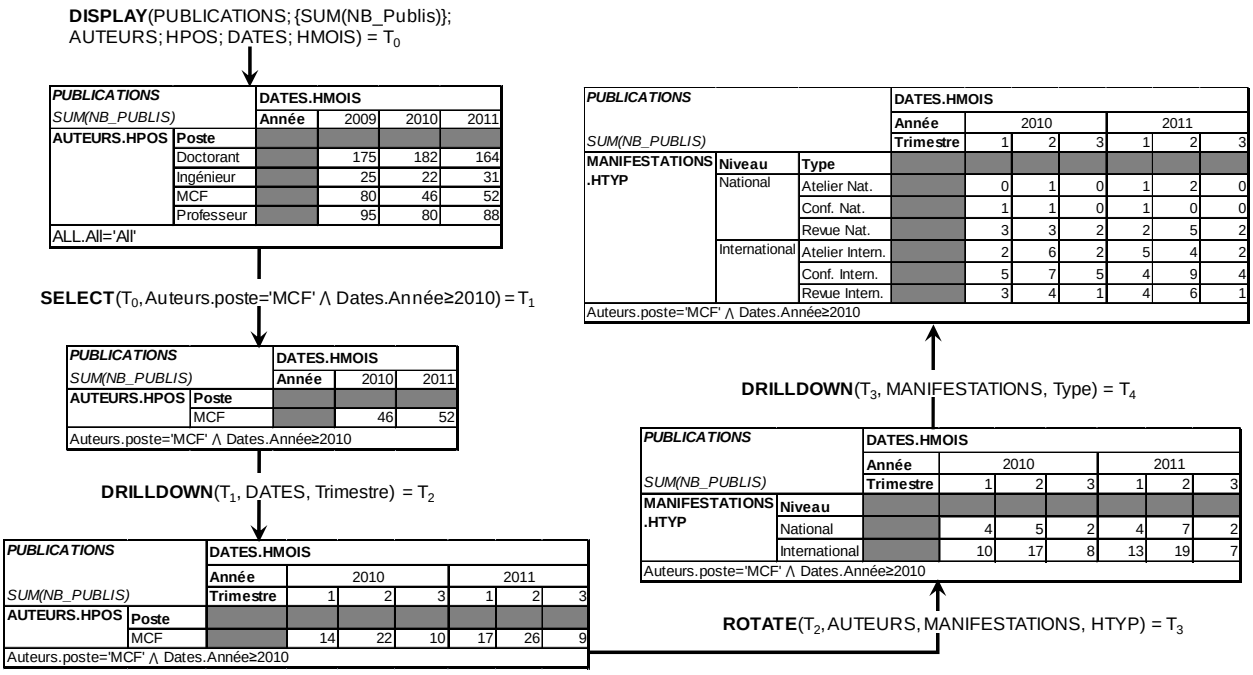


Figure 11. Exemple d'analyse OLAP

Graphe d'analyse

Nous pouvons modéliser une analyse OLAP menée par l'usager par un graphe où chaque nœud représente l'état courant de l'analyse et les arcs représentent les transitions entre ces différents états (*cf.* Figure 12).

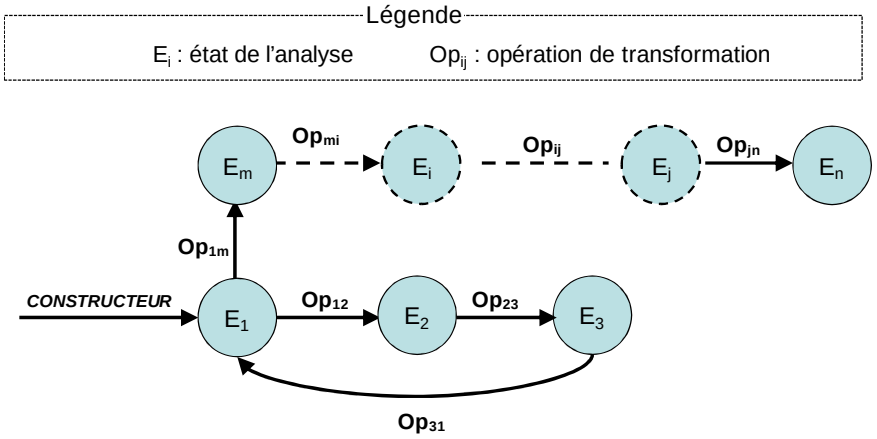


Figure 12. Graphe d'une analyse OLAP

Les transitions entre les nœuds sont assurées par des opérations de transformation de données, telles que la rotation, le forage, la sélection, etc. Il est à noter qu'une suite d'opérations en départ d'un état d'analyse donné peut aboutir à un même état résultat qui est produit par une autre séquence d'opérations à partir d'un autre état. De plus, à partir d'un même état E_p, plusieurs séquences d'opérations sont parfois possibles pour aboutir à un état E_q.

Le résultat d'une opération utilisateur peut prendre la forme d'une table multidimensionnelle, d'une courbe, d'un histogramme, etc. Certaines structures de visualisation imposent des contraintes sur les analyses de données. Par exemple, un histogramme permet d'afficher les données d'une seule mesure selon un seul niveau de détail par dimension et une TM permet d'analyser les données en fonction de deux dimensions seulement. Afin de rendre notre approche indépendante des structures de visualisation, nous définissons le concept de *contexte d'analyse* par une description arborescente traduisant la représentation interne des données.

2.2.2 Concept de contexte d'analyse OLAP

Un *contexte d'analyse* représente l'état courant d'une analyse qui résulte de l'application d'une opération de transformation. Il regroupe des structures multidimensionnelles (le fait, les mesures, les dimensions et les attributs de dimensions) ainsi que des valeurs (les instances des mesures agrégées et des dimensions analysées).

Définition. Soit une constellation *CS*. Un contexte d'analyse *CA* est défini par (C^F; C^D; C^R) où

- $C^F = N^{Fi}$ [/ [f(m_j)∈]?{[val_{j_k}]$^+$}]$^+$ représente le sujet d'analyse en cours au travers du fait F_i, des mesures affichées m_j, ainsi que des valeurs agrégées, où $val_{j_k} \in Type(m_j)$[9],
- $C^D = \{C^{D1};\ldots; C^{Du}\}$ représente les axes de l'analyse en cours avec : $\forall i \in [1..u]$, $C^{Di} = N^{Di}[.N^{Hj}]$ [/ $(p_{k_1},p_{k_2}) \in \{ [(val_{k_1},val_{k_2})]^+ \}]^+$ où $p_{k_1} \in A^{Di}$, $p_{k_2} \in A^{Di}$ et $(val_{k_1},val_{k_2}) \in dom(p_{k_1}) x dom(p_{k_2})$[10],
- $C^R = \{pred^{Fi}, pred^{D1},\ldots, pred^{Du}\}$ est l'ensemble des prédicats définissant les restrictions sur les valeurs analysées du fait F_i ou des attributs d'une dimension D_j.

[9] Type(A) définit toutes les valeurs possibles de l'attribut A

[10] dom(A) définit l'ensemble des valeurs val_i de l'attribut A, ($dom(A) \subseteq Type(A)$).

Le signe '?' indique que l'élément peut être absent. En effet, il est possible d'analyser un fait sans mesures (calcul du nombre des occurrences du fait). Le signe '$^+$' désigne la présence de plusieurs éléments : un fait peut être analysé selon plusieurs mesures simultanément et plusieurs paramètres peuvent être affichés suivant une dimension.

Le signe '/' traduit une relation de dépendance entre les composants du contexte d'analyse : l'élément à droite ne peut être représenté en l'absence de celui de gauche. Ceci implique qu'un paramètre doit être toujours rattaché à une dimension. De même, une mesure doit être toujours associée au fait analysé.

Remarque. Nous considérons le cas d'analyse d'un seul fait. L'analyse simultanée de plusieurs faits est modélisée au travers de plusieurs contextes d'analyse.

Réellement, un nombre très important de contextes d'analyse peut être généré à partir d'une constellation. En effet, chaque combinaison de fait, d'une ou plusieurs dimensions, d'un ou plusieurs paramètres par dimension, en changeant leur ordre d'affichage, et éventuellement de prédicats de restriction, peut constituer une structure d'un contexte d'analyse. Cette même structure correspond à un nombre important d'instances en fonction de l'état des valeurs dans la BDM.

Typologie des contextes d'analyse. Un contexte d'analyse est un ensemble multidimensionnel multi-hiérarchisé constitué de composants de la constellation.

- Un contexte d'analyse est dit *complet* s'il comporte au moins un fait, deux dimensions avec un paramètre pour chacune, une valeur par paramètre, et une valeur des mesures agrégées pour chaque combinaison des valeurs des paramètres.
- Un contexte d'analyse est dit *partiel* s'il comporte au moins un fait ou une dimension. Ce contexte n'est pas complet, il n'est pas affichable. Si le contexte partiel comporte les différentes structures (fait, mesure, dimensions et paramètres) sans les valeurs correspondantes, il est dit non-évalué.

Remarque. Selon notre formalisme, les éléments vides d'un contexte partiel sont notés '∅'. Par exemple, CA_1 = (PUBLICATIONS.NB_PUBLIS ; ∅ ; ∅) est le contexte d'analyse du nombre des publications qui manque la spécification des axes courants de l'analyse.

Notation. Le contexte CA = (∅; ∅ ; ∅) est un contexte vide. Il est noté ∅.

Exemple. Suivant les spécifications formelles, le contexte d'analyse du nombre des publications durant les deux dernières années par équipe de recherche par type de manifestation est défini par l'expression (C^{F1}; C^{D1}; C^{R1}) où :

- C^{F1}= PUBLICATIONS/SUM(NB_PUBLIS)∈ {15;10;48;25;12;8}
- C^{D1}={MANIFESTATIONS.HTYP/(All;TYPE)∈ {(All,Atelier); (All,Conférence); (All,Revue)}; AUTEURS.HEQ/(All;ÉQUIPE)∈ {(All,SIG) ; (All,SMAC)}}
- C^{R1}={DATES.ANNEE≥2010}

Arbre de contexte d'analyse

Afin de faciliter la gestion des contextes d'analyse, nous proposons de représenter ces derniers à l'aide d'une forme arborescente qui traduit la disposition multidimensionnelle et hiérarchique de leurs éléments.

Propriétés de l'arbre de contexte

L'arbre de contexte est un arbre enraciné, ordonné et étiqueté[11].

L'arbre est dirigé vers le bas. Les flèches ne sont pas présentées dans notre formalisme graphique. Rappelons brièvement qu'un arbre ordonné est un arbre où les enfants de chaque nœud ont un ordre désigné (pas nécessairement en fonction de leur valeur). Il existe donc un nouvel ordre « horizontal » qui s'ajoute à l'ordre « vertical » déterminé à partir de la racine.

Racine

La racine d'un arbre de contexte d'analyse complet est un nœud modélisant un fait. La racine d'un contexte d'analyse partiel est un fait ou une dimension.

Nœuds (ou sommets)

L'arbre comporte des *nœuds de structure* représentant les composants structurels (fait, mesure, dimension avec la hiérarchie courante, paramètre et attribut faible) ainsi que des *nœuds de type valeur* correspondant aux valeurs des mesures et des paramètres. Chaque nœud est étiqueté par le nom de la structure ou de la valeur qu'il représente. Les étiquettes des nœuds de type structure sont uniques.

Chaque niveau de granularité (un paramètre $param_i$ et les attributs faibles $attrib_1$, $attrib_2$, … qui le décrivent) est représenté par un seul nœud d'étiquette *$param_i$ ($attrib_1$, $attrib_2$, …)*. De même, chaque instance d'un paramètre est représentée par un seul nœud avec les instances des attributs faibles correspondants.

Le prédicat de restriction du domaine des valeurs d'une mesure est représenté par le même nœud de cette mesure. Une conjonction des prédicats sur les paramètres et sur les attributs faibles d'une dimension est représentée par le nœud qui modélise cette dimension. Nous distinguons donc deux catégories de nœuds :

- des nœuds simples $\mathcal{N}_i$ modélisant un composant structurel ou une valeur.
- des nœuds $\mathcal{N}_j$ composés d'un attribut (dimension ou mesure) et d'une conjonction de disjonctions de prédicats ($pred_{j1} \wedge \ldots \wedge pred_{jn}$) définis sur cet attribut. Nous notons $\mathcal{N}_j$.Attribut (par défaut $\mathcal{N}_j$) le nom de l'attribut et $\mathcal{N}_j$.Pred la conjonction des prédicats modélisés par le nœud. Les prédicats par défaut de type « Attribut. All = 'ALL' » ne sont pas représentés sur l'arbre.

Arcs

Les arcs de l'arbre traduisent quatre types de liens :

- des liens de type « est analysé selon » reliant d'une part un nœud de fait à des nœuds modélisant les dimensions et les mesures suivant lesquelles il est analysé, et d'autre part, un nœud de dimension aux nœuds modélisant ses paramètres ;
- des liens de type « est plus général que » traduisant l'ordre d'affichage des attributs de la hiérarchie en cours d'une dimension ;
- des liens de type « est une instance de » reliant un paramètre à l'une de ses valeurs ou une mesure et l'une de ses valeurs agrégées ;

[11] Un arbre étiqueté est un arbre dans lequel on associe à chaque sommet une information (une étiquette).

- des liens de type « est imbriqué dans » représentant l'imbrication hiérarchique des valeurs des paramètres.

Profondeur et largeur

La profondeur d'un arbre de contexte est le nombre maximum de nœuds entre le nœud racine et les feuilles. Elle dépend essentiellement du nombre des paramètres affichés et du nombre des mesures analysées. La largeur de l'arbre dépend du nombre de valeurs des paramètres et des mesures affichées.

Cardinalité

La cardinalité d'un arbre de contexte est définie par : $|A^{CA}| = n + b$, où n est le nombre de nœuds de l'arbre ; b = 1 si l'arbre comporte au moins un nœud composé, 0 sinon.

Remarque. Nous distinguons les dimensions de filtrage des dimensions affichées.

- Les dimensions de filtrage sont des dimensions qui ne sont pas affichées dans le contexte d'analyse, mais auxquelles sont associées des restrictions sur les valeurs de paramètres. Elles sont représentées par les nœuds de dimension qui n'admettent pas de descendants.
- Une dimension affichée est une dimension suivant laquelle les données sont analysées. Elle est représentée par un nœud possédant un sous arbre modélisant les paramètres affichés ainsi que leurs valeurs.

Formalisme graphique. Afin de représenter de manière explicite l'arbre d'un contexte d'analyse, nous proposons un formalisme graphique (*cf.* légende de la Figure 13). Le fait analysé est représenté par un rectangle blanc avec son nom à l'intérieur. Toutes les autres structures sont représentées en noir, les valeurs étant en blanc. Une dimension, avec la hiérarchie courante s'il s'agit d'une dimension affichée, est représentée par un rectangle noir. Une mesure est représentée par un losange noir. Les attributs de la dimension et la fonction d'agrégation d'une mesure sont représentés par des ronds noirs. Les valeurs des paramètres et des mesures sont représentées par des ronds blancs. Au dessous de la racine, nous plaçons dans l'ordre de la gauche vers la droite les dimensions affichées, puis les dimensions de filtrage, puis les mesures. Chaque dimension (resp. mesure) est liée au paramètre le plus général (resp. fonction d'agrégation) et à ses valeurs. Les autres paramètres sont liés chacun au paramètre qui est plus détaillé. Chaque valeur de paramètre est liée aux valeurs qui lui sont imbriquées.

Remarque. Afin de simplifier la représentation graphique, seul le nom de l'attribut figurera sur les nœuds composés. Pour les distinguer des nœuds simples, nous représentons leurs étiquettes entre {}. Ainsi, une dimension affichée munie de prédicats de restriction sur ses valeurs est représentée par un nœud d'étiquette *{Nom dimension.Nom hiérarchie en cours}*. Par contre le nœud d'une dimension de filtrage est étiqueté par *{Nom dimension}*.

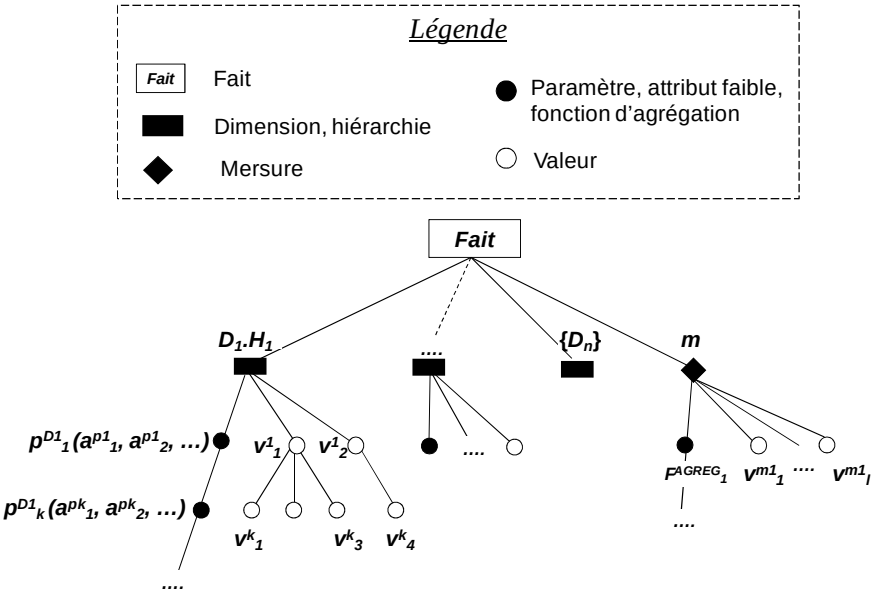


Figure 13. Représentation graphique d'un contexte d'analyse.

Exemple. La figure suivante présente l'arbre du contexte d'analyse de l'exemple précédent ainsi que son affichage suivant une TM.

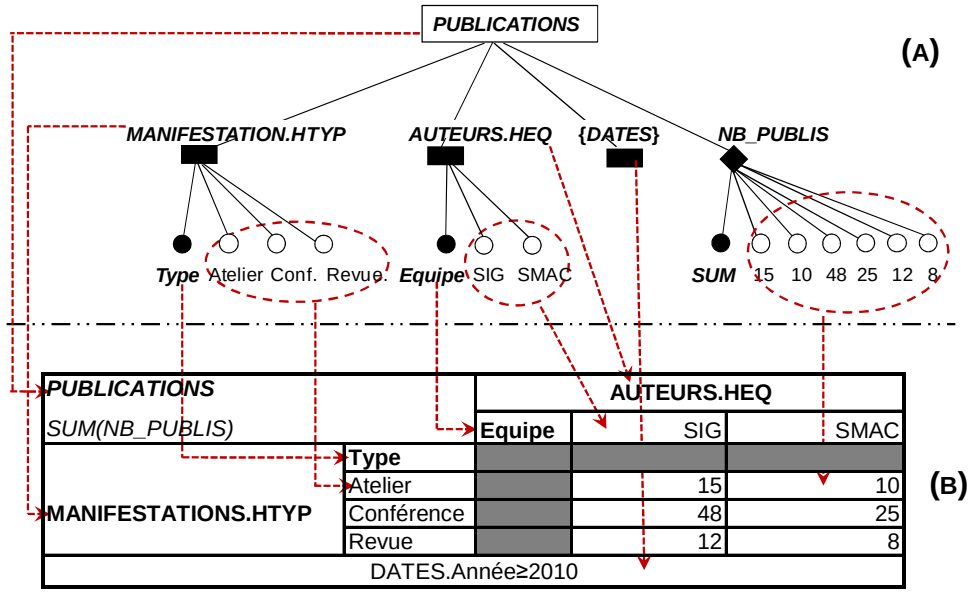


Figure 14. Exemple d'arbre de contexte d'analyse (A) avec la TM correspondante (B)

Opérations de transformation d'arbre de contexte d'analyse

Le passage d'un contexte d'analyse à un autre se fait à l'aide d'une opération OLAP. Dans notre cadre formel, le passage d'un arbre de contexte à un autre est effectué par une ou plusieurs opérations de transformation d'arbre de contexte. Notre arbre de contexte traduit plusieurs relations spécifiques par rapport à un arbre classique, telles que la dépendance du paramètre de la dimension, l'imbrication des valeurs d'un paramètre, la relation entre les valeurs du paramètre le plus détaillé et les valeurs agrégées de la mesure, etc. Par conséquent, les opérations de transformation d'un arbre classique doivent être adaptées pour être appliquées à un arbre de contexte d'analyse.

Les opérations de transformation prennent en entrée un arbre de contexte d'analyse et fournissent en sortie un autre arbre de contexte. Nous distinguons les opérations de modification de la structure de l'arbre de contexte (*InsertN* et *DeleteN*) des opérations de mise à jour des nœuds (*AddP*, *SubstractP*, *UpdateE*).

- **Insertion d'un nœud de contexte**

Syntaxe. $InsertN(A^C, att_i, N_{i-1}[,\{N^1_{i+1}, N^2_{i+1}, \ldots\}]) = A^{C'}$

L'opération *InsertN* permet d'insérer dans l'arbre de contexte A^C un nouveau nœud $\mathcal{N}$ d'attribut att_i en tant que fils du nœud N_{i-1}. Un nœud peut être inséré de différentes façons en fonction du paramètre $\{N^1_{i+1}, N^2_{i+1}, \ldots\}$:

- $\{N^1_{i+1}\}$. Le nœud $\mathcal{N}$ devient le nœud père de la séquence des nœuds consécutifs précédemment fils de N^1_{i+1} (*cf.* Figure 15 (b)).
- $\{N^1_{i+1}, N^2_{i+1}, \ldots\}$. Les nœuds $N^1_{i+1}, N^2_{i+1}, \ldots$ sont des fils directs du nouveau nœud $\mathcal{N}$ (*cf.* Figure 15 (c)).
- $\{N^1_{i+1}, N^2_{i+1}, \ldots\}$ n'est pas renseigné. Le nœud $\mathcal{N}$ est inséré en tant que feuille (*cf.* Figure 15 (d)).

Conditions. Les nœuds $N^1_{i+1}, N^2_{i+1}, \ldots$ sont des fils directs du nœud d'attribut N_{i-1}. De plus, l'attribut du nœud inséré (att_i) doit correspondre à celui du nœud père N_{i-1} (att_{i-1}).

- att_{i-1} est un fait $F \in \mathrm{F}^{CS}$: att_i est une dimension ($att_i \in \mathrm{Star}^{CS}(F)$) ou att_i est une mesure ($att_i \in \mathrm{M}^F$).
- att_{i-1} est une mesure m : att_i est une valeur ($att_i \in \mathrm{Type}(m)$) ou une fonction d'agrégation {SUM, AVG, …}.
- att_{i-1} est une dimension $D \in \mathrm{D}^{CS}$: $att_i \in \mathrm{A}^D$.
- att_{i-1} est un attribut de la hiérarchie $\mathrm{H_j}$: $att_i \in \mathrm{P}^{Hj}$, avec $att_i \prec_{\mathrm{Hj}} att_{i-1}$.

Exemple. Considérons l'exemple de contexte d'analyse présenté en Figure 14. La figure suivante montre plusieurs façons d'insertion de nœuds dans l'arbre A^{C0} de ce contexte suite à la séquence d'opérations suivantes :

- Op_1 : $\mathrm{InsertN}(\mathrm{A}^{C0}, \mathrm{Institut}, \mathrm{N}^{\mathrm{AUTEURS.HEQ}}, \mathrm{N}^{\mathrm{Equipe}}) = \mathrm{A}^{C1}$
- Op_2 : $\mathrm{InsertN}(\mathrm{A}^{C1}, \text{'IRIT'}, \mathrm{N}^{\mathrm{AUTEURS.HEQ}}, \{\mathrm{N}^{\mathrm{SIG}}, \mathrm{N}^{\mathrm{SMAC}}\}) = \mathrm{A}^{C2}$
- Op_3 : $\mathrm{InsertN}(\mathrm{A}^{C2}, \text{'PYRAMIDE'}, \mathrm{N}^{\mathrm{IRIT}}) = \mathrm{A}^{C3}$

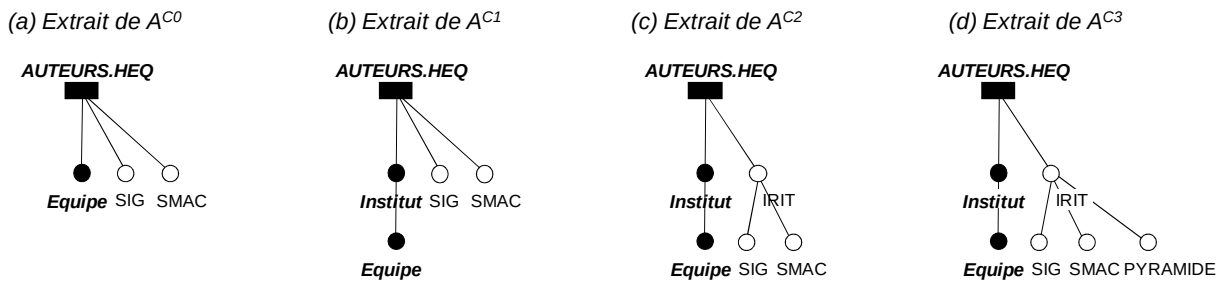


Figure 15. Différentes possibilités d'insertion d'un nœud dans un arbre de contexte

- **Suppression d'un nœud de contexte**

Syntaxe. $DeleteN(A^C, N) = A^{C'}$

Cette opération permet de supprimer un nœud de l'arbre (une dimension, une mesure, un paramètre, ou une valeur). Certaines opérations sont effectuées en conséquence suivant les liens impliqués par le nœud supprimé :

- La suppression d'un nœud dimension ou mesure est suivie de la suppression de son sous-arbre (lien de type « est analysé selon »).
- La suppression d'un nœud paramètre est suivie de la suppression des nœuds valeur correspondants (lien de type « est une instance de »).
- Si le nœud *N* supprimé représente la valeur d'un paramètre intermédiaire, les nœuds fils de *N* deviennent des fils du nœud père de *N* (lien de type « est imbriqué dans »). Cependant, la suppression du nœud représentant la valeur du paramètre le plus détaillé entraine la suppression des nœuds de valeurs de mesures dépendantes.

Exemple. Considérons l'exemple précédent. L'application de la séquence d'opérations suivantes sur l'arbre de contexte final A^{C3} permet de rétablir l'arbre de contexte initial A^{C0}.

- Op_4 : DeleteN(A^{C3},$N^{PYRAMIDE}$) = A^{C2}
- Op_5 : DeleteN(A^{C2},N^{IRIT}) = A^{C1}
- Op_6 : DeleteN(A^{C1},$N^{Institut}$) = A^{C0}

- **Ajout d'un prédicat**

Syntaxe. $AddP(A^C, N_i, pred_i) = A^{C'}$

AddP permet de modifier le prédicat d'un nœud. Il s'agit d'ajouter le prédicat $pred_i$ au nœud N_i (N_i.Pred = $pred_i$). Si N_i est composé, cette opération permet de mettre à jour le nœud en ajoutant $pred_i$ en conjonction avec N_i.Pred.

- **Suppression d'un prédicat**

Syntaxe. $SubstractP(A^C, N_i, pred_i) = A^{C'}$

SubstractP effectue l'opération inverse de *AddP*. Elle permet de supprimer le prédicat $pred_i$ du nœud N_i. Si N_i est composé, cette opération permet d'éliminer $pred_i$ de la conjonction des prédicats associée à N_i : soit N_i.Pred = $pred_i \wedge pred_j$, le nœud modifié est tel que N_i.Pred = $pred_j$.

Suite à l'insertion ou à la suppression d'un prédicat, des opérations d'ajout ou de suppression de nœuds valeur de paramètre et/ou mesure sont effectuées afin d'assurer la conformité des nouveaux prédicats de restriction par rapport aux nœuds valeur.

- **Substitution d'un nœud valeur**

Syntaxe. $UpdateE\ (A^C, N_i, att^*_i) = A^{C'}$

Cette opération consiste à mettre à jour l'étiquette d'un nœud sans changer la structure de l'arbre. L'étiquette du nœud N_i est substituée par att^*_i.

Certaines opérations de transformation de l'arbre induisent le re-calcul des valeurs de mesures agrégées. Elles sont donc suivies d'opérations *UpdateE* afin de mettre à jour les nœuds valeur de mesure. Il s'agit du cas de :

- l'insertion ou la suppression d'un nœud représentant le paramètre le plus détaillé d'une dimension,
- l'insertion ou la suppression d'un prédicat d'un nœud représentant une dimension de filtrage.

Les opérations de manipulation d'arbre de contexte sont indépendantes des mécanismes des opérations OLAP pour lesquelles il n'existe pas un langage standard. A chaque opération OLAP exprimée par l'utilisateur correspond une ou plusieurs opérations de transformation de l'arbre. Par exemple, une opération de rotation de dimension (Ravat et al., 2008) correspond à une suppression du nœud dimension suivie de l'insertion d'un sous arbre représentant la nouvelle dimension (nœud dimension, nœuds de paramètres et nœuds de valeurs des paramètres).

2.2.3 Appariement de contextes d'analyse

Au cours de leurs différentes analyses menées sur la BDM, les utilisateurs peuvent visiter des contextes d'analyse très variés, parfois plus ou moins proches l'un de l'autre. Afin de pouvoir illustrer les différentes relations qui peuvent exister entre deux contextes d'analyse, nous allons considérer le contexte d'analyse C_1^A du nombre de publications dans le domaine des entrepôts des données par année et par équipe. Le contexte d'analyse C_2^A du nombre de publications par année et par équipe est plus général que C_1^A vu qu'il considère tous les domaines de recherche, tandis que le contexte C_3^A d'analyse du nombre de publications en recherche d'information est proche de C_1^A sans partager avec lui tous ses détails. En se basant sur leur représentation en arbre, la relation entre deux contextes d'analyse se traduit par l'appariement entre leurs arbres correspondants. Par exemple, l'arbre de C_2^A est un sous-arbre de celui de C_1^A, alors que les arbres de C_1^A et C_3^A sont intersectés.

La similarité sémantique entre les contextes d'analyse est au-delà des limites de cette thèse. Nous allons nous focaliser uniquement sur l'appariement structurel des contextes d'analyse. Il s'agit de comparer les contextes d'analyse en raison de leurs composants structurels et de leurs valeurs.

Appariement d'arbres de contexte

L'appariement de deux contextes d'analyse consiste à parcourir simultanément leurs arbres et à comparer les couples de nœuds de même position. L'algorithme de parcours d'un arbre de contexte est inspiré de l'algorithme de parcours en largeur des arbres étiquetés et ordonnés (Okasaki, 2000).

Le parcours en largeur des arbres classiques correspond à un parcours par niveau des nœuds. Dans chaque niveau, les nœuds sont parcourus de la gauche vers la droite. Ainsi, tous les nœuds d'une profondeur donnée sont traités avant de passer aux nœuds plus profonds. Or, un niveau donné de l'arbre de contexte peut comporter des nœuds de type structure, d'autres de type valeur. De plus, les liens d'instanciation entre les nœuds doivent être considérés dans le parcours des arbres de contexte. Intuitivement, une comparaison de deux contextes d'analyse confrontera d'abord leurs structures, représentées par le fait, les dimensions et les paramètres de chacun, puis les instances correspondantes (valeurs des mesures et des paramètres de dimensions). Par conséquent, l'imposition d'un ordre de parcours alterné, qui considère d'abord les nœuds structure puis les nœuds valeur, est nécessaire afin d'adapter les algorithmes de parcours en largeur aux spécificités des arbres de contexte d'analyse. Ainsi, l'ordre de visite des nœuds pour un contexte d'analyse complet est : fait ; dimensions puis mesures ; paramètres puis fonctions d'agrégation de niveau 1; … ; paramètres puis fonctions d'agrégation de niveau n; valeurs des paramètres puis valeurs des mesures de niveau 1; … ; valeurs des paramètres puis valeurs des mesures de niveau n.

Définition. Le parcours d'un arbre de contexte est un parcours en largeur *alterné*.

Exemple. L'ordre de visite des nœuds de l'arbre de contexte de la Figure 14 est le suivant : PUBLICATIONS ; MANIFESTATION.HTYP ; AUTEURS.HEQ ; DATES.Année≥2010 ; NB_PUBLIS ; 'Type' ; 'Equipe' ; SUM ; 'Atelier' ; 'Conférence' ; 'Revue' ;'SIG' ; 'SMAC' ; '15' ; '10' ; '48' ; '25' ; '12' ; '8'.

Soient A^C_1 et A^C_2 deux arbres de contexte. L'appariement de A^C_1 et A^C_2 consiste à parcourir A^C_1 et à comparer chaque nœud avec son homologue dans A^C_2. Ceci risque de négliger des disproportions des nœuds valeur admettant la même étiquette, mais qui reflètent des liens d'instanciation ou d'imbrication différents. Par exemple, deux nœuds admettant la même étiquette '*2000*' (*cf.* Figure 16 (a)) seront considérés semblables bien que l'un représente la valeur du paramètre *année* et l'autre représente la valeur de la mesure *Sum(Nb_Pulis)*. De même, deux nœuds d'étiquette '*SIG*' peuvent être imbriqués aux nœuds '*IRIT*' (nom d'un institut) et '*Conférence*' (type de manifestation) représentant respectivement le nom d'une équipe de recherche et le nom d'une conférence (*cf.* Figure 16 (b)). Afin de prendre en compte les différents liens sémantiques entre les nœuds lors de l'appariement de deux arbres de contexte, il est nécessaire d'examiner les ancêtres des nœuds. Pour comparer deux nœuds N_1 de A^C_1 et N_2 de A^C_2, il faut comparer les arcs (N_1,Père(N_1)) et (N_2,Père(N_2)), où Père (N_i) est le nœud père de N_i.

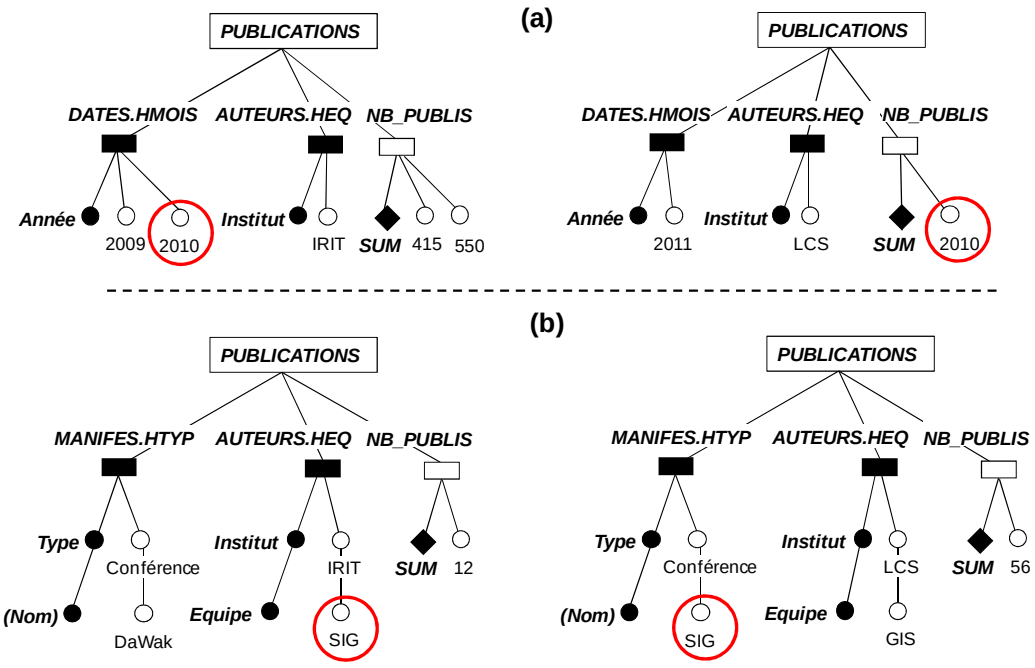


Figure 16. Liens d'instanciation différents pour des nœuds de valeurs équivalents

Définition. L'appariement de deux arbres de contexte A^C_1 et A^C_2 consiste à effectuer un parcours en largeur alterné de A^C_1. Pour chaque niveau de A^C_1, chaque nœud N_i est comparé à son homologue N_j de A^C_2 en appariant les couples (N_i,Père(N_i)) et (N_j,Père(N_j)).

Appariement de nœuds

La comparaison des nœuds des arbres de contexte tient compte, en plus de leurs étiquettes, de:

- leur type : des nœuds de type fait, dimension, …

- leur différence de composition : des nœuds sont composés, d'autres sont simples
- leurs prédicats

Soient deux nœuds $\mathcal{N}_1$ et $\mathcal{N}_2$, avec $\mathcal{N}_1$.Pred = $pred^1_1 \wedge \ldots pred^1_m$ et $\mathcal{N}_2$.Pred = $pred^2_1 \wedge \ldots pred^2_n$), où $pred^x_y = pred_1 \vee \ldots \vee pred_p$, avec $pred_i = A_i\ op_i\ a_i$ ($pred_i$ est dit *prédicat élémentaire*). Rappelons que A_i représente un attribut de dimension ou une mesure associée, éventuellement, avec une fonction d'agrégation, $op_i \in \{=,<,>,\leq,\geq, \neq\}$ pour les attributs numériques et $op_i \in \{=, \neq\}$ pour les autres attributs.

Les nœuds d'un arbre de contexte sont typés. La première étape est de la comparaison de deux nœuds $\mathcal{N}_1$ et $\mathcal{N}_2$ est de vérifier s'ils sont de même type, c'est-à-dire s'ils modélisent le même type de composant de la constellation ou une instance. Autrement, les nœuds sont incompatibles même s'ils partagent la même étiquette. Ensuite, une étape de confrontation des étiquettes des nœuds $\mathcal{N}_1$ et $\mathcal{N}_2$ est nécessaire en vérifiant si $\mathcal{N}_1$.Attribut = $\mathcal{N}_2$.Attribut. Dans le cas de nœuds composés, les prédicats correspondants $\mathcal{N}_1$.Pred et $\mathcal{N}_2$.Pred sont également comparés. La comparaison entre $\mathcal{N}_1$.Pred et $\mathcal{N}_2$.Pred (notés dans la suite $Pred_1$ et $Pred_2$ pour simplifier) est une comparaison entre deux conjonctions de disjonctions de prédicats élémentaires qui est assurée par les règles de logique des prédicats. Cette comparaison prend en compte également les spécificités d'un contexte d'analyse.

Définition. Soient deux prédicats élémentaires $pred_i$ et $pred_j$. $pred_i$ est inclus dans $pred_j$, noté $pred_i \subset pred_j$, dans deux cas :

- $pred_i$ et $pred_j$ sont sous la forme de $A \leq a_i$ (resp. $A < a_i$) et $A \leq a_j$, (resp. $A < a_j$) avec $a_i \leq a_j$.
- $pred_i$ et $pred_j$ sont sous la forme de $A \geq a_i$ (resp. $A > a_i$) et $A \geq a_j$ (resp. $A > a_j$), avec $a_i \geq a_j$.

Exemple. Le prédicat *Année=2010* est inclus dans le prédicat *Année>2008*, tandis que le prédicat *Année≤2010* n'est pas inclus dans le prédicat *Année>2008*.

Propriété. Soient $pred^x_y$ et $pred^z_w$ deux disjonctions de prédicats élémentaires ($pred^x_y = pred_{y_1} \vee \ldots \vee pred_{y_k}$ et $pred^z_w = pred_{w_1} \vee \ldots \vee pred_{w_j}$), $pred^x_y \subset pred^z_w$ si et seulement si $\forall\ i \in [1..k]$, $pred_{y_i} \subset pred^z_w$.

Définition. Soient deux prédicats $Pred^1 = pred^1_1 \wedge \ldots \wedge pred^1_m$ et $Pred^2 = pred^2_1 \wedge \ldots \wedge pred^2_n$, $Pred^1 \subset Pred^2$ si et seulement si, $\forall\ i \in [1..m]$ et $\forall j \in [1..n]$, $(pred^1_i \subset pred^2_j) \vee (pred^1_i = pred^2_j)$.

Nous identifions quatre relations possibles entre deux nœuds de contexte.

- $\mathcal{N}_1$ et $\mathcal{N}_2$ sont incompatibles si et seulement si $\mathcal{N}_1$.Attribut ≠ $\mathcal{N}_2$.Attribut
- $\mathcal{N}_1$ et $\mathcal{N}_2$ sont égaux si et seulement si
  - $\mathcal{N}_1$.Attribut = $\mathcal{N}_2$.Attribut,
  - $Pred_1 = Pred_2$ ou $Pred_1$ est une forme simplifiée de $Pred_2$ si $\mathcal{N}_1$ et $\mathcal{N}_2$ sont des nœuds composés.
- $\mathcal{N}_1$ est inclus dans $\mathcal{N}_2$, noté $\mathcal{N}_1 \subset \mathcal{N}_2$, si et seulement si
  - $\mathcal{N}_1$.Attribut = $\mathcal{N}_2$.Attribut,
  - $Pred_1 \subset Pred_2$
- $\mathcal{N}_1$ et $\mathcal{N}_2$ sont dits intersectés si et seulement si:
  - $\mathcal{N}_1$.Attribut = $\mathcal{N}_2$.Attribut,
  - $Pred_1 \neq Pred_2$.

Relations entre contextes

Nous identifions quatre principales relations possibles entre deux contextes d'analyse C^A_i et C^A_j : la disjonction, l'intersection, la dominance et l'équivalence. Afin de détailler ces relations, nous allons utiliser la notation suivante pour un contexte d'analyse C^A_k:

- A^{CAk} est l'arbre de contexte correspondant au contexte C^A_k,
- Structure(A^{CAk}) est l'ensemble des nœuds de structure au sein de l'arbre A^{CAk},
- Val(A^{CAk}) est l'ensemble des nœuds valeur de l'arbre A^{CAk}.

Afin de traiter le cas le plus général, C^A_i et/ou C^A_j peuvent être des contextes partiels.

Notation. Nous notons le *nœud d'ancrage* de deux arbres le premier nœud structure commun rencontré en parcourant les deux arbres selon un parcours en largeur alterné.

Disjonction de contextes

Deux contextes d'analyse C^A_i et C^A_j sont disjoints (noté $C^A_i \cap C^A_j = \emptyset$) s'ils n'ont aucun élément en commun.

Définition. Soient C^A_i et C^A_j deux contextes d'analyse.

Disjoint (C^A_i,C^A_j) = VRAI si $\forall (\mathcal{N}_i, \mathcal{N}_j) \in$ Structure(A^{CAi})×Structure(A^{CAj}), $\mathcal{N}_i$ et $\mathcal{N}_j$ sont incompatibles

FAUX sinon.

Si l'intersection de deux contextes se réduit seulement à des valeurs, $A^{CAi} \cap A^{CAj} \subset$ Val(A^{CAi}) alors $C^A_i \cap C^A_j = \emptyset$. En effet, si les deux arbres partagent uniquement des valeurs, leur fonction d'appariement *Disjoint* renvoie FAUX vu que l'algorithme effectue un parcours en largeur alterné qui considère les structures en première étape. En absence de structures communes, l'algorithme s'arrête.

Algorithme

- Rechercher le nœud $\mathcal{N}^a$ d'ancrage de A^{CAi} et A^{CAj}.
- Si $\mathcal{N}^a$ n'existe pas, les deux contextes sont disjoints.

Dominance de contextes

Un contexte d'analyse C^A_i domine un autre contexte C^A_j ($C^A_j \subset C^A_i$) si tous les nœuds de A^{CAj} sont inclus ou égaux aux nœuds homologues de A^{CAi}.

Définition. Soient deux contextes d'analyse C^A_i et C^A_j

Englobe (C^A_i,C^A_j) = VRAI si $\forall \mathcal{N}_j \in A^{CAj}, \exists \mathcal{N}_i \in A^{CAi}$, tel que $\mathcal{N}_j \subset \mathcal{N}_i$ ou $\mathcal{N}_i$ et $\mathcal{N}_j$ sont égaux

FAUX sinon.

Algorithme

- Rechercher le nœud $\mathcal{N}^a$ d'ancrage de A^{CAj} et A^{CAi}.
- Si $\mathcal{N}^a$ est racine de A^{CAj}, vérifier si A^{CAj} est inclus dans le sous-arbre de $\mathcal{N}^a$ dans A^{CAi}.
- L'algorithme s'arrête lorsqu'il rencontre un nœud de A^{CAj} qui est n'est pas inclus ou égal à son homologue dans A^{CAi}.

Exemple. Soit C^A_1 le contexte d'analyse du nombre de publications par année par équipe, C^A_2 le contexte d'analyse du nombre de publications par année de l'équipe SIG. C^A_1 domine C^A_2.

Equivalence de contextes

Deux contextes d'analyse sont équivalents (noté $C^A_i \equiv C^A_j$) s'ils ont des arbres de contexte identiques.

Définition. Soient deux contextes d'analyse C^A_i et C^A_j

Egal (C^A_i,C^A_j) = VRAI si $\forall \mathcal{N}_i \in A^{CAi}$, $\exists \mathcal{N}_j \in A^{CAj}$, tel que $\mathcal{N}_i$ et $\mathcal{N}_j$ sont égaux

FAUX sinon.

Algorithme

- Rechercher le nœud $\mathcal{N}^a$ d'ancrage de A^{CAi} et A^{CAj}.
- Si $\mathcal{N}^a$ est racine de A^{CAi} et $\mathcal{N}^a$ est racine de A^{CAj}, parcourir A^{CAi} et vérifier si chaque nœud est égal à son homologue dans A^{CAj}.
- L'algorithme s'arrête lorsqu'il rencontre un nœud de A^{CAi} qui n'est pas égal à son homologue dans A^{CAj}.

Remarque. Dans le cas de dominance ou d'équivalence, l'appariement de contextes est dit *total*.

Intersection de contextes

Deux contextes d'analyse sont intersectés (noté $C^A_i \cap C^A_i \neq \emptyset$) si leurs arbres admettent au moins deux nœuds homologues qui sont égaux, intersectés ou l'un qui est inclus dans l'autre.

Définition. Soient deux contextes d'analyse C^A_i et C^A_j

Chevauche (C^A_i,C^A_j) = VRAI si $\exists \mathcal{N}_i \in A^{CAi}$, $\exists \mathcal{N}_j \in A^{CAj}$, tel que $\mathcal{N}_i$ et $\mathcal{N}_j$ sont égaux ou $\mathcal{N}_i$ et $\mathcal{N}_i$ sont intersectés ou $\mathcal{N}_i \subset \mathcal{N}_j$

FAUX sinon.

Algorithme

- Rechercher le nœud $\mathcal{N}^a$ d'ancrage de A^{CAi} et A^{CAj}.
- Comparer chaque nœud du sous-arbre de $\mathcal{N}^a$ dans A^{CAi} avec son homologue dans le sous-arbre de $\mathcal{N}^a$ dans A^{CAj}.
  - Afin de respecter les liens d'instances et d'imbrications dans les arbres de contexte, si un nœud $\mathcal{N}_i$ d'un arbre est incompatible avec son homologue de l'autre arbre, tout le sous-arbre $\mathcal{N}_i$ est rejeté, donc ne sera pas visité.
  - Si aucun nœud structure n'est en commun, arrêter le parcours.
- L'algorithme s'arrête lorsqu'il ne reste aucun nœud à visiter dans le sous-arbre de $\mathcal{N}^a$.
- Si le seul nœud en commun est $\mathcal{N}^a$, alors Rejoint(C^A_i,C^A_i).

Remarque. Dans le cas d'intersection, l'appariement de contextes est dit *partiel*.

Exemple. Considérons l'exemple précédent. Soit C^A_3 le contexte d'analyse des publications par année à la conférence DaWak. Nous avons Chevauche(C^A_1,C^A_3) et Chevauche(C^A_2,C^A_3).

Le tableau suivant résume les différentes relations entre deux contextes d'analyse. La partie commune entre les deux contextes est représentée en noir.

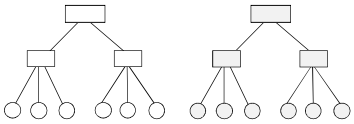
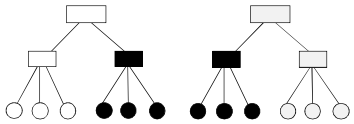
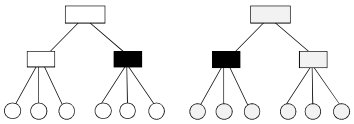
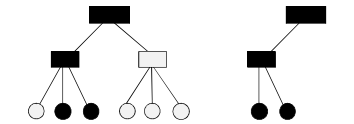
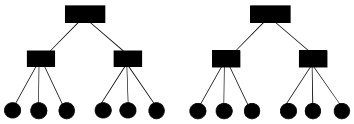
Relation	Schéma	Fonction d'appariement
Disjonction		Disjoint (C_1,C_2) $C_1 \cap C_2 = \emptyset$
Intersection		Chevauche(C_1,C_2) $C_1 \cap C_2 \neq \emptyset$
		Rejoint(C_1,C_2) $C_1 \cap C_2 = \{e\}$
Dominance		Englobe(C_1,C_2) $C_1 \cap C_2 = \{C_2\}$
Equivalence		Egal(C_1,C_2) $C_1 \equiv C_2$

Tableau 3. Relations entre deux contextes d'analyse

3 Personnalisation de l'analyse OLAP

Dans cette section, nous présentons notre modèle de préférences, puis nous décrivons les fonctionnalités de personnalisation que nous proposons.

3.1 Modélisation des préférences de l'usager

Nous proposons de définir des préférences sur le schéma de la BDM pour faciliter sa navigation. Par ailleurs, l'usager admet également des préférences sur les valeurs d'une BDM (Golfarelli et Rizzi, 2009).

Notre modèle de préférences répond à trois propriétés principales.

1. Les préférences sont définies sur les éléments d'une constellation et/ou sur ses valeurs

Nous considérons en particulier les préférences sur les mesures, sur les dimensions et sur les attributs de dimensions, ainsi que sur les instances de la BDM.

2. Les préférences sont exprimées sur des domaines numériques et textuels

Les préférences peuvent être associées à des éléments qui sont composés de données numériques telles que des nombres de publications, des années…, ainsi qu'à des attributs textuels composés de données alphanumériques tels que des noms de manifestations, des

adresses… Il est évident que les préférences sur des structures de la constellation portent sur des domaines textuels puisqu'elles font référence à l'élément par son nom. Les préférences sur les valeurs, par contre, peuvent être exprimées sur des domaines numériques ou textuels. Il est à noter que, conformément à notre modèle de constellation, les préférences sur les mesures ont des domaines numériques.

3. Les préférences sont sensibles au contexte d'analyse

Nous proposons de capturer la variation des préférences de l'usager en fonction du contexte par leur association à des contextes d'analyse particuliers. De telles préférences sont qualifiées de *contextuelles*. Notons qu'une préférence peut être indépendante de tout contexte. Il s'agit de préférences *absolues* qui sont associées au contexte vide ∅.

3.1.1 Préférences contextuelles

Définition. Etant donné une constellation *CS*, une préférence OLAP contextuelle est définie par P = (*E*; *θ*; *cp*), où

- *E* est un élément de structure de *CS* ou un prédicat sur les valeurs de *CS*.
- *θ* est le score de la préférence. Il représente un nombre réel entre 0 et 1 indiquant le degré d'intérêt de l'usager à l'attribut *E*. Selon la valeur de *θ*, on distingue entre des préférences plus au moins fortes. *θ*=0 indique l'absence d'intérêt à *E* et *θ*=1 indique un intérêt maximum.
- *cp* est le contexte de la préférence précisant le cadre de son application.

Nous distinguons cinq types de préférences en fonction du type de *E*. Afin d'illustrer ces types, nous considèrerons des exemples de préférences absolues.

1. *E est une mesure* ($E \in M^F$) associée éventuellement avec une fonction d'agrégation. P est une **préférence sur les mesures**.

   Une telle préférence indique que l'usager préfère l'analyse des données du fait F selon l'indicateur *E*, indépendamment des axes d'analyse.

   Exemple. P = (MIN(MT_FRAIS); 0.6; ∅) indique une préférence des montants minimums de frais de missions.

2. *E est une dimension* ($E \in D^{CS}$). P est une **préférence sur les dimensions**.

   Une telle préférence indique que l'usager préfère l'analyse des données qui sont agrégées selon la dimension *E*, indépendamment du fait analysé.

   Exemple. Selon la préférence P = ('*Dates*'; 0.9; ∅), les données qui sont agrégées selon l'axe temporel sont fortement préférées.

3. *E est un attribut de dimension* (un paramètre ou un attribut faible, $E \in A^D$). P est une **préférence sur les paramètres**.

   P précise que les données agrégées selon le niveau de granularité *a* de la dimension *D* sont préférées, indépendamment de leur niveau d'agrégation selon les autres dimensions.

   Exemple. P = (*Dates.Année*; 0.7; ∅) indique une préférence de données par année.

4. *E est une condition sur les valeurs d'un paramètre* ($E = a\ op\ v_i$, $a \in A^D$, $v_i \in Type(a)$). P est une **préférence sur les valeurs des paramètres**.

Ce type de préférences permet de décrire les valeurs du paramètre a (satisfaisant la condition $a\ op\ v_i$) suivant lesquelles les données pertinentes sont agrégées, indépendamment du fait analysé et des autres niveaux d'agrégation. Le score de préférence θ indique le degré d'intérêt de l'usager aux valeurs de *a* qui satisfont la condition *E*.

Exemple. P = (*TX_ACCEP*≤0.35; 0.9; ∅) précise que les données des manifestations avec un taux d'acceptation moins de 35% sont préférées.

5. *E est une condition sur les valeurs d'une mesure* brute (E = $m\ op\ v_i$, $m \in M^{Fi}$, $v_i \in$ Type(*m*)) ou agrégée (E = $f^{AGREG}(m)\ op\ v_i$, $f^{AGREG} \in$ {AVG, SUM, COUNT,...}). P est **une préférence sur les valeurs des mesures**.

   Selon P, les données de la mesure *m* satisfaisant la condition $f(m)\ op\ v_i$ sont préférées, indépendamment des niveaux d'agrégation. Le score θ indique le degré d'intérêt de l'utilisateur aux valeurs de la mesure *m* qui sont générées par le prédicat $f(m)\ op\ v_i$.

   Exemple. Selon la préférence P = (*Sum(NB_PUBLIS)*>5; 1; ∅), l'usager est intéressé par une somme des nombres de publications qui dépasse 5.

Notation. doi(P) désignera le degré d'intérêt d'une préférence P, soit θ.

Remarque. Les préférences sur les structures sont des préférences directes. Elles définissent l'intérêt de l'usager sur une structure particulière. Cependant, les préférences sur les valeurs sont indirectes car elles traduisent l'intérêt de l'usager sur l'ensemble des valeurs qui résultent d'un prédicat. Il n'est possible de définir cet ensemble que lors de l'évaluation de la préférence.

Les préférences sont rattachées à des contextes d'analyse de différents nombres de dimensions et de différents niveaux de détail. Une préférence peut dépendre du sujet d'analyse seulement, par exemple, l'analyse des publications.

Propriété. Le contexte d'une préférence est un contexte d'analyse partiel.

3.1.2 Profil utilisateur

Définition. Un profil utilisateur $\mathcal{U} = \{P_1, \ldots, P_n\}$ est défini par l'ensemble des préférences contextuelles P_i.

Conceptuellement, le profil utilisateur est modélisé par une association de type (m,n) entre les éléments de préférences et les contextes. La Figure 17 présente un schéma simplifié d'un profil utilisateur. Pour des mesures de simplification, la classe *META_CONTEXT* n'est pas détaillée ici. Dans le cas où plusieurs préférences sont associées au même contexte, le contexte est stocké une seule fois.

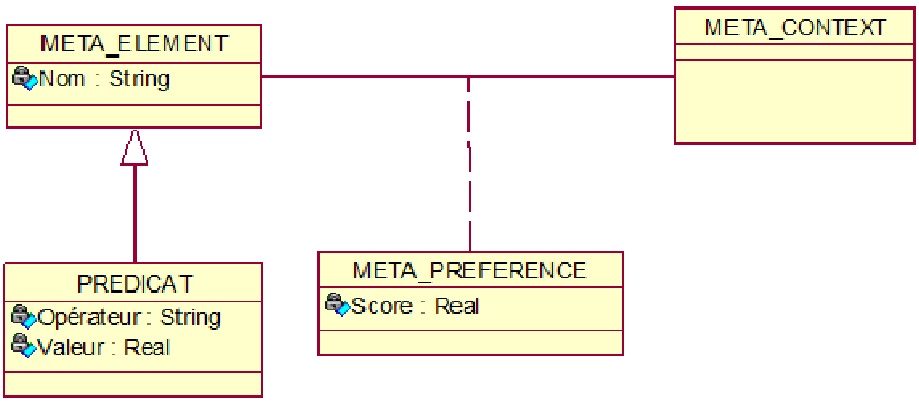


Figure 17. Modèle d'un profil utilisateur (représenté au format UML)

Exemple. En plus de son intérêt aux publications de catégories ACM et IEEE (P_1), le décideur admet les préférences contextuelles suivantes :

- Il préfère toujours voir les données par date durant l'analyse du nombre des missions de recherche (P_2)
- Il est fortement intéressé par les données par trimestre dans le contexte d'analyse des publications des doctorants (P_3)
- Il préfère généralement visualiser les instituts de rattachement des chercheurs durant l'analyse des données des permanents (P_4)
- L'usager souhaite ne pas considérer les manifestations de type ateliers dans le contexte d'analyse du nombre des missions de recherche (P_5)
- Il préfère se focaliser sur les deux dernières années pour l'analyse des données des permanents (P_6)
- Il n'aime pas les données des manifestations de type atelier lors de l'analyse de la somme du nombre de publications (P_7)

Ces préférences sont définies de la manière suivante :

- P_1 : (Catégorie='IEEE' ∨ Catégorie='ACM';0.75 ; ∅)
- P_2 : (Dates; 0.7 ; cp_1) ; cp_1 = (Missions/Nb_Missions; ∅; ∅)
- P_3 : (Dates.Trimestre;0.8 ; cp_2) ;
  cp_2 = (Publications;∅ ; {Auteurs.Poste = 'Doctorants'})
- P_4 : (Auteurs.Institut; 0.6 ; cp_3) ; cp_3 = (∅; ∅ ; {Auteurs.Statut = 'Permanents'})
- P_5 : (Manifestations.Type≠'Atelier'; 0.85 ; cp_1)
- P_6 : (Dates.Année= 2010 ∨ Date.Année= 2011; 0.9 ; cp_3)
- P_7 : (Manifestations.Type≠'Atelier';1; cp_4) ;
  cp_4 = (Publications/SUM(Nb_Publis); ∅ ; ∅))

Les préférences P_2, P_3 et P_4 portent sur le schéma, alors que les autres préférences sont relatives aux valeurs. La figure suivante représente le profil du décideur. Il faut noter qu'un attribut de préférence peut être associé à différents contextes d'analyse (par exemple *'Manifestations.Type≠'Atelier'*) éventuellement selon différents degrés d'intérêt. De même, un contexte d'analyse peut être associé à plusieurs attributs de préférence (par exemple cp_1 et cp_3).

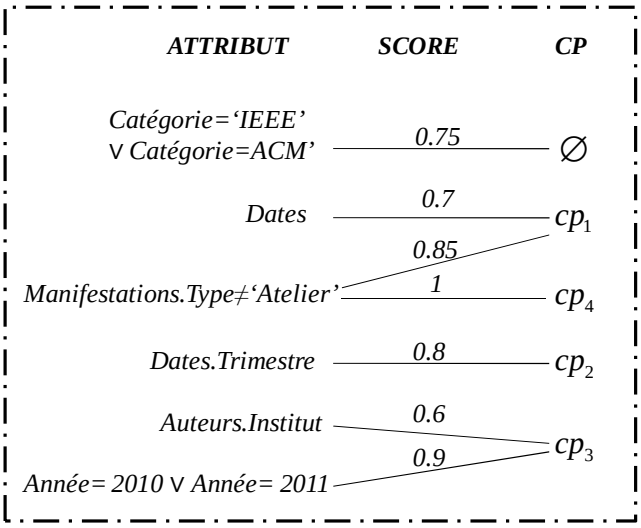


Figure 18. Exemple d'un profil utilisateur

Les profils sont définis au moment de la conception (Garrigós et al., 2009), comme ils peuvent être créés après la mise en place de la BDM (Golfarelli et Rizzi, 2009). Ils sont mis à jour pour suivre l'évolution des besoins spécifiques des décideurs dans le temps. Dans le cadre de cette thèse, nous ne traitons pas le processus d'acquisition et de mise à jour automatiques des profils. Nous nous focalisons plutôt sur leur exploitation pour la personnalisation de l'analyse de l'usager. Plus précisément, l'approche que nous décrivons dans la suite du mémoire se situe à un instant t où les profils utilisateurs sont dans un état E(t).

3.2 Cadre générique de la personnalisation

Après avoir introduit les données que nous utiliserons pour la personnalisation de l'analyse OLAP, nous présentons dans cette section comment nous souhaitons intégrer la fonctionnalité de personnalisation dans un système OLAP.

Afin de mieux positionner notre approche, nous proposons une classification de l'espace d'une analyse OLAP personnalisée en trois dimensions (Garrigós et al., 2009).

- Actions de la personnalisation : Une analyse OLAP est vue comme une navigation d'une BDM où à chaque étape de la navigation, un résultat d'une requête est renvoyé à l'utilisateur. Nous prévoyons deux actions de personnalisation : personnaliser la navigation et personnaliser le résultat d'une requête.
- Critères de la personnalisation : il s'agit de tous les éléments susceptibles d'influencer la personnalisation d'une analyse de l'usager. Nous considérons en particulier les préférences de l'usager afin d'augmenter son degré de satisfaction. De plus, afin d'assurer cette satisfaction, la restitution des données doit tenir compte du contexte d'analyse courant de l'utilisateur. Finalement, la personnalisation est souvent définie comme l'optimisation des intérêts de l'utilisateur en fonction de certaines contraintes. La prise en compte de contraintes rend le mécanisme de personnalisation configurable.
- Types de la personnalisation : la personnalisation de l'analyse OLAP peut être implicite à l'utilisateur ou explicite. Elle peut être statique, en préparant des données

stables qui sont proposées d'une façon constante à l'utilisateur, ou dynamique en développant les données à proposer à l'utilisateur au moment de l'exécution. Nous nous intéressons dans le cadre de ce travail à des mécanismes de personnalisation dynamique qui relèvent plus de défis de recherche. De plus, une personnalisation dynamique éviterait la redondance des magasins de données (un seul magasin partagé par les décideurs) et la mise en place de plusieurs processus ETL en conséquence.

Les sections suivantes présentent les deux actions de personnalisation assurées par notre approche.

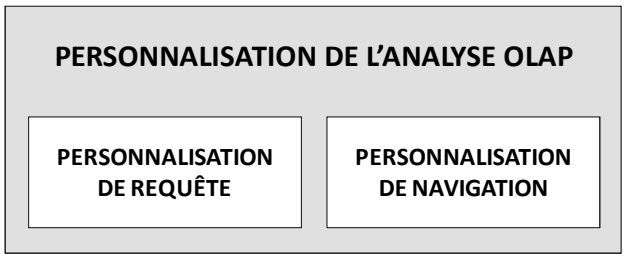


Figure 19. Actions de personnalisation de l'analyse OLAP

Personnalisation des requêtes

Par rapport au modèle d'analyse OLAP, le mécanisme de personnalisation que nous décrivons ici se situe au niveau du passage d'un contexte d'analyse à un autre. Le processus de personnalisation de requête est appliqué au contexte d'analyse courant et permet de générer une version personnalisée du contexte d'analyse résultant (*cf.* Figure 12).

Chaque opération utilisateur produit un nouveau contexte d'analyse. Cette opération est réellement traduite en une requête qui est appliquée à la BDM. La personnalisation d'une requête vise à satisfaire les préférences de l'utilisateur afin d'augmenter le degré d'intérêt du contexte d'analyse généré (*cf.* Figure 20). Ainsi, la prise en compte des préférences de l'usager permettra un passage de la notion de « *pertinence collective* », où le résultat est produit seulement en fonction de la requête et est supposé pertinent pour tous les usagers, vers la notion de « *pertinence personnelle* » où la pertinence du résultat est évaluée en fonction des besoins spécifiques de chacun.

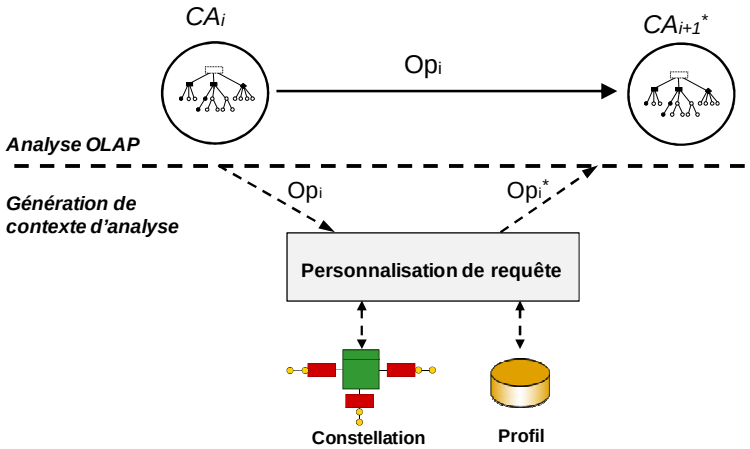


Figure 20. Action de personnalisation de requête

Nous distinguons entre les profils utilisateurs, stockés en tant qu'une couche enrichissant la BDM, et les requêtes utilisateurs : un profil est un modèle utilisateur qui décrit les préférences d'un utilisateur qui le différencie des autres, alors qu'une requête est un besoin utilisateur qui est exprimé par un ordre explicite, et dont l'évaluation doit tenir compte de son profil. Au moment d'exécution d'une requête, le processus de personnalisation reformule la requête afin de prendre en compte les éléments du profil de l'utilisateur en cours. La requête reformulée est exécutée sur la BDM générant un résultat personnalisé.

Personnalisation de la navigation

La personnalisation de l'analyse en ligne doit aller au-delà de la personnalisation du résultat des requêtes. Afin de faciliter la tâche de l'utilisateur, nous proposons de personnaliser la navigation de la BDM à travers la technique de recommandation. En nous basant sur le modèle de l'analyse OLAP en graphe, nous intégrons trois scénarios de recommandation :

- une partie d'un nœud du graphe d'analyse (par exemple le niveau de granularité des données) : le système aide l'usager à construire son rapport. Ceci constitue une assistance interactive à la composition de requête (*cf.* Figure 21, étape (1)).
- un nœud du graphe d'analyse que l'utilisateur pourrait visiter dans les étapes suivantes de l'analyse. Ceci permet d'anticiper la stratégie de navigation de l'usager (*cf.* Figure 21, étape (2)). Il s'agit de recommandation par anticipation.
- des nœuds d'analyse qui n'appartiennent pas nécessairement aux nœuds visités mais qui sont utiles pour l'utilisateur (*cf.* Figure 21, étape (3)). Il s'agit de recommandation d'alternatives.

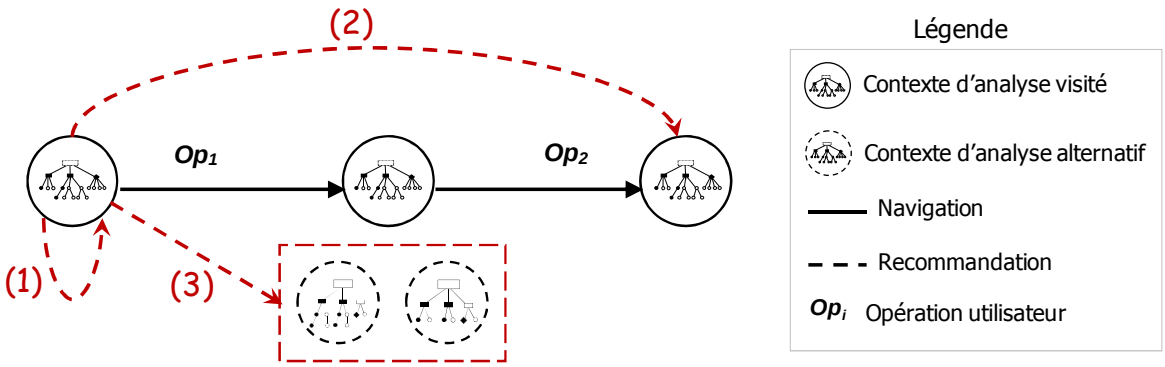


Figure 21. Personnalisation de la navigation par recommandation

4 Bilan

En conclusion, nous avons proposé un cadre générique pour la prise en compte de l'usager dans l'analyse des données OLAP.

D'abord, les données décisionnelles sont modélisées en constellation au travers de faits (sujets d'analyse) et de dimensions (axes d'analyse), généralisant le modèle en étoile.

L'analyse OLAP représente l'exploration interactive de l'espace multidimensionnel d'une constellation par une succession d'opérations de manipulation OLAP afin d'obtenir le résultat souhaité. Elle est modélisée par un graphe où chaque nœud représente un contexte d'analyse

et chaque arc correspond à une opération utilisateur qui permet de passer d'un contexte à un autre (Jerbi et al., 2009b, 2009c). Afin de faciliter la gestion des contextes d'analyse, nous proposons de représenter ces derniers à l'aide d'un arbre spécifique (Jerbi et al., 2008, 2009a, 2009c). Un intérêt de cette structure arborescente est son indépendance vis-à-vis du choix de visualisation. Ainsi, cette représentation interne est affichée à l'utilisateur sous diverses formes (tableau, diagramme, etc).

Afin de personnaliser l'analyse des données OLAP, nous avons proposé un modèle de préférences contextuelles (Jerbi et al., 2009c, 2010c) portant sur le schéma ainsi que les valeurs d'une constellation. Notre modèle de préférence est basé sur une approche quantitative qui attribue à une préférence un score traduisant le degré d'intérêt de l'usager. Chaque préférence est associée avec un contexte d'analyse qui précise son cadre d'application (contexte interne). Le profil utilisateur est constitué d'un ensemble de préférences contextuelles qui sont stockées en extension de la BDM.

Nous avons proposé un cadre multidimensionnel pour la personnalisation d'une analyse OLAP, où la personnalisation dépend de trois axes, à savoir l'axe des actions (navigation, résultat de la requête), l'axe des types (implicite ou explicite, statique ou dynamique) et l'axe des critères. Dans le cadre de cette thèse, nous nous intéressons à la personnalisation dynamique qui exploite des profils utilisateurs afin d'adapter le schéma et le contenu de cette BDM à la perception de chaque usager, évitant de stocker statiquement une BDM individuelle pour chaque usager. Les critères essentiels de notre approche de personnalisation sont les préférences de l'usager ainsi que des contraintes de personnalisation. Finalement, notre vision de la personnalisation englobe aussi bien la personnalisation de la navigation de l'usager que celle du résultat des requêtes.

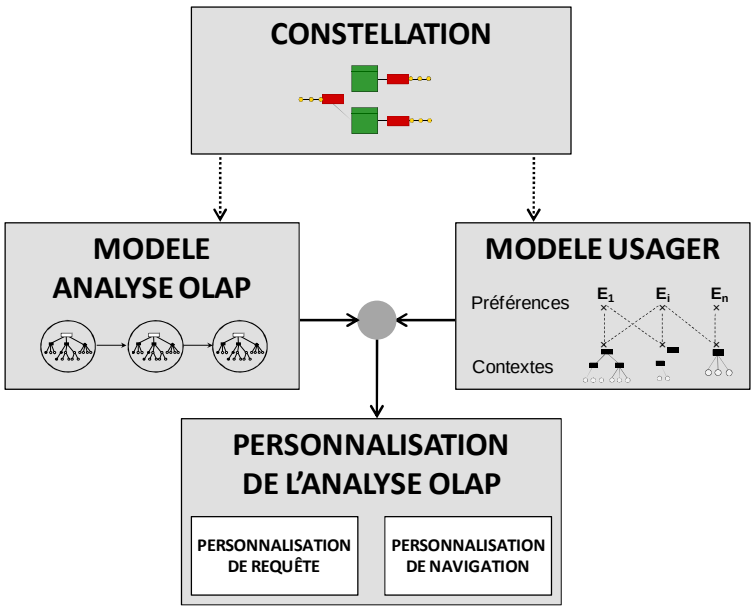


Figure 22. Cadre de personnalisation des analyses OLAP

Le chapitre suivant présente la première action de la personnalisation qui est la personnalisation des requêtes.

Références

Garrigós, I., Pardillo, J., Mazón, J., Trujillo, J. (2009). A Conceptual Modeling Approach for OLAP Personalization. Intl. Conf. on Conceptual Modeling (ER), pages 401–414.

Golfarelli, M., Maio, D., et Rizzi, S. (1998). Conceptual design of data warehouses from E/R schemes, Intl. Conf. on System Sciences.

Golfarelli, M. et Rizzi, S. (2009). Expressing OLAP Preferences. Intl. Conf. on Scientific and Statistical Database Management (SSDBM), IEEE Computer Society, pages 83–91.

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2008). Management of Context-aware Preferences in Multidimensional Databases. Intl. Conf. on Digital Information Management (ICDIM), IEEE, pages 669–675.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2009a). Modèle de Préférences Contextuelles pour les Analyses OLAP. Journées Francophones Extraction et Gestion de Connaissances (EGC), pages 253–258.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2009b). Applying Recommendation Technology in OLAP Systems. Intl. Conf. on Enterprise Information Systems (ICEIS), Springer, LNBIP 24, pages 220–233.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G (2009c). Preference-Based Recommendations for OLAP Analysis. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer-Verlag, pages 467–478.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2010c). A Framework for OLAP Content Personalization. East European Conf. on Advances in Databases and Information Systems (ADBIS), Springer-Verlag, pages 262–277.*

Kimball, R. (1996). The data warehouse toolkit: Practical Techniques for Building Dimensional Data Warehouses, John Wiley and Sons, ISBN : 0-471-15337-0, 1996, 2ème ed. : Ralph Kimball, Margaery Ross, The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd Edition, John Wiley & Sons, 2002.

Okasaki, C. (2000). Breadth-first numbering: lessons from a small exercise in algorithm design. In Proceedings of the 2000 ACM SIGPLAN Intl. Conf on Functional Programming, Vol. 35, No. 9, pages 131–136.

Ravat F., Teste O., Tournier R., Zurfluh G. (2008). Algebraic and graphic languages for OLAP manipulations. Intl. Journal of Data Warehousing and Mining (IJDWM), Vol. 4, No. 1, pages 17–46.

Romero, O., Abelló, A. (2007). On the Need of a Reference Algebra for OLAP. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer, pages 99–110.

Chapitre 4

Personnalisation des requêtes OLAP

Sommaire

1 Introduction

Ce chapitre présente la première action de la personnalisation, à savoir la personnalisation des requêtes.

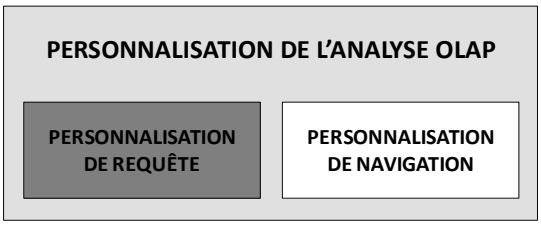


Figure 23. Positionnement de la personnalisation de requête

Le chapitre précédent présente un modèle de préférences utilisateur. Dans ce chapitre, nous montrons comment ces préférences peuvent être exploitées afin de restituer un résultat personnalisé. Concrètement, il s'agit d'intégrer ces préférences au sein de la requête initiale afin de réduire le résultat à l'ensemble des données pertinentes (*cf.* Figure 24). Ce résultat étant composé de structures multidimensionnelles et de valeurs de ces structures, nous nous focalisons dans ce chapitre sur la personnalisation des valeurs. La personnalisation des structures de la requête (par exemple par changement de l'attribut d'une dimension) conduit à produire une nouvelle requête qui sera recommandée à l'usager afin de personnaliser sa navigation de la BDM (*cf.* chapitre 5).

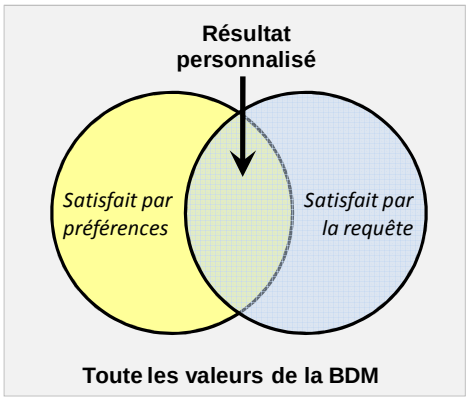


Figure 24. Personnalisation de la requête par restriction du résultat

La personnalisation de requêtes OLAP est le processus d'enrichissement dynamique d'une requête avec les préférences de l'usager stockées dans un profil dans le but de fournir un résultat personnalisé.

Comme dans toute approche d'enrichissement de requête à partir d'un profil utilisateur, deux axes importants gagnent à être explorés : 1) Comment déterminer les préférences de l'usager à utiliser ? et 2) Comment réécrire une requête OLAP en fonction des préférences ?

Plusieurs problèmes se posent par rapport à ces deux axes. La personnalisation ne doit pas engendrer des résultats vides (Rizzi, 2007) ou bloquer le traitement de la requête. Ainsi, les sources de conflits potentiels doivent être identifiées et un mécanisme de gestion de ces

conflits doit être mis en place. De plus, le processus de personnalisation doit être indépendant du langage de définition des requêtes et de la forme d'affichage du résultat.

D'autre part, l'emploi des préférences peut impliquer deux perspectives que le processus de personnalisation doit considérer:

- une perspective utilisateur suivant laquelle l'intégration des préférences permet de générer un résultat qui répond au plus aux besoins spécifiques de l'usager, et
- une perspective système, où l'utilisation des préférences doit être liée à des contraintes.

Plan du chapitre. La suite de ce chapitre est organisée comme suit : la section 2 décrit les différents traitements des préférences durant le mécanisme de personnalisation des requêtes ; la section 3 présente une approche naïve de personnalisation qui correspond à la perspective de l'utilisateur, alors que la section 4 introduit une approche avancée prenant en compte des contraintes du système.

2 Gestion de préférences

Généralement, dans les approches quantitatives, la recherche des préférences est effectuée soit en fonction de leur pertinence par rapport à la requête (Koutrika et Ioannidis, 2004, 2005a), soit selon un processus d'appariement de contexte qui vérifie si le contexte de la préférence apparie avec celui de la requête (Stefanidis et al., 2007). Nous proposons une approche combinée qui considère ces deux approches pour déterminer les préférences à retenir dans le processus de personnalisation.

Cette section fixe la terminologie des préférences et présente leur traitement durant le processus de personnalisation.

Notations. Soit Q une requête OLAP. Nous notons par F^Q le fait considéré par Q ($F^Q \in F^{CS}$), $M^Q = \{f_1(m^Q_1),\ldots, f_n(m^Q_n)\}$ ($\forall i \in [1..n]$, $m^Q_i \in M^{FQ}$) l'ensemble des mesures agrégées par Q selon les fonctions $f_1,\ldots, f_n$ (SUM, AVG, …), $D^Q = \{d^Q_1,\ldots, d^Q_x\}$ l'ensemble des dimensions d'agrégation ($\forall j \in [1..x]$, $d^Q_j \in Star^{CS}(F^Q)$), $A^Q = \{a^Q_1,\ldots, a^Q_y\}$ les niveaux d'agrégation selon ces dimensions ($\forall k \in [1..y]$, $a^Q_k \in A^{dj}$), et $PRED^Q = \{pred^Q_1,\ldots, pred^Q_u\}$ l'ensemble des prédicats de restriction de Q.

2.1 Préférences actives

L'objectif du processus de personnalisation des requêtes est de restituer un contenu personnalisé à l'utilisateur. Conformément à notre modèle de préférences, ce processus considère particulièrement les préférences sur les valeurs $P_i = (pred_i; \theta_i ; cp_i)$.

Selon la valeur de doi(P), les préférences peuvent être relativement ou extrêmement faibles ou fortes. Il est évident qu'une préférence avec un score proche de 1 soit forte, et qu'une autre préférence avec un score proche de 0 soit faible. Pour d'autres scores, le jugement de l'importance d'une préférence reste intuitif. La définition d'un score référence permet de définir un seuil à partir duquel une préférence est considérée dans le processus de personnalisation. Ce seuil pourrait être défini par l'administrateur du système de personnalisation, ou approuvé par l'usager au moment de l'expression de la requête. Les

préférences qui sont peu importantes sont maintenues dans le profil de l'utilisateur vu que leurs scores peuvent varier dans le temps en fonction de l'évolution des besoins de l'usager. La même préférence peut passer donc d'un état « activée » à l'état « désactivée ».

Définition. Une préférence ($pred_i$; θ_i; cp_i) est relative à la requête *Q* si le prédicat de la préférence n'existe pas déjà dans *Q* et il est pertinent par rapport au résultat ou à une partie du résultat de *Q*.

Parmi les préférences sur les valeurs stockées dans le profil, certaines peuvent être *relatives à la requête Q*. Il s'agit :

- des préférences qui sont rattachées à une mesure agrégée par Q, ou aux attributs d'agrégation de Q.
- des préférences qui sont définies sur d'autres attributs de dimensions qui sont connectées au fait F^Q. Ces dernières permettent de filtrer les données avant leur agrégation par la requête. Par exemple, pour une requête qui renvoie le nombre des publications par année (Dates) et par nom d'auteur (Auteurs), la préférence P_1 de l'usager (P_1 = (Catégorie='IEEE'∨Catégorie='ACM';0.75 ; ∅)) permet de calculer Q sur la base des données des manifestations IEEE et ACM seulement.

En conclusion, les préférences relatives à une requête *Q* sans être redondantes avec l'un des éléments de la requête, et dont le score dépasse un seuil de pertinence sont prises en compte pour l'exécution de *Q*. Ces préférences sont qualifiées de préférences *actives* P^{ACTIV}.

Définition (Préférences actives). Soit *Q* une requête OLAP et $P_i = (A_i\ op_i\ a_i;\ \theta_i;\ cp_i)$ une préférence sur les valeurs. P_i est une préférence active par rapport à *Q* et à un seuil de pertinence λ si

- A_i op $a_i \notin PRED^Q$,
- $\theta_i \geq \lambda$, et
- $A_i \in \{m_i, f_i(m_i)\}$, où $f_i(m_i) \in M^Q$, ou $A_i \in A^{Dj}$, où $D_j \in Star^{CS}(F^Q)$.

Les préférences actives peuvent porter sur des mesures agrégées ou non agrégées. Une préférence qui est associée à une mesure agrégée $f_i(m_i) \in M^Q$ permet de ne restituer que les valeurs agrégées qui satisfont la préférence. Par contre, une préférence associée à une mesure m_i est prise en compte avant l'agrégation des données et permet de définir les valeurs des mesures à agréger. Par exemple, pour une requête qui affiche le nombre total de publications, les préférences portant sur *Nb_publis* ($pred_i$ est de la forme *Nb_publis* op_i val_i) et celles portant sur *Sum(Nb_publis)* ($pred_j$ est de la forme *Sum(Nb_publis)* op_j val_j) sont relatives à cette requête.

Exemple. Reprenons la requête de l'usager du chapitre précédent (*cf.* chapitre3, section 2.2.2), soit Q_1 : « nombre de publications des deux dernières années par équipe de recherche et par type de manifestation ». Considérons également les préférences sur les valeurs P_1, P_5, P_6 et P_7 stockées dans le profil de l'usager (*cf.* chapitre3, section 3.1.2). Supposons que l'usager admette les préférences supplémentaires suivantes:

L'usager est intéressé par l'analyse d'un nombre de soumissions qui est supérieur ou égal à 5 lors de la focalisation de l'analyse sur l'année en cours.

- P_8 = (Sum(Nb_Soumiss)≥5 ; 0.9; (∅;∅;{Dates.Année=2011})).

L'usager préfère souvent se focaliser sur les chercheurs toulousains lors de l'analyse des publications de l'année en cours par équipe d'auteur.

- P_9= (Auteurs.Ville='Toulouse'; 0.9; (Publications; {Auteurs.Heq/(All;Équipe)} ; {Dates.Année=2011})).

Il préfère considérer les manifestations dont le taux d'acceptation ne dépasse pas 35%.

- P_{10}= (Manifestations.Tx_Accep≤0.35 ; 0.6; ∅).

Il s'intéresse aux manifestations avec un taux d'acceptation inférieur à 25% lors d'une analyse du nombre annuel des publications des maîtres de conférences.

- P_{11} = (Manifestations.Tx_Accep<0.25; 0.75 ; (Publications/SUM(Nb_Publis); {Dates.Hmois/(All;Année)} ; {Auteurs.Poste='MCF'})).

P_8 n'est pas active car elle est définie sur le nombre de soumissions alors que la requête porte sur le nombre de publications. Bien que P_6 (de prédicat « *Année*=2010∨*Année*=2011 ») soit définie sur les valeurs de la dimension *Dates* qui est connectée au fait de la requête, elle n'est pas active car elle est redondante avec une condition de la requête. Les préférences P_1, P_5, P_7, P_{10} et P_{11} sont actives par rapport à Q_1 puisqu'elles portent sur les valeurs des attributs *Catégorie*, *Type* et *Tx_Accep* de la dimension *Manifestations* de Q_1. De même, P_9 portant sur les valeurs de la dimension *Auteurs* de la requête est active.

2.2 Préférences candidates sur les valeurs

Dans la suite, afin de rendre notre approche indépendante du langage de définition de la requête, nous ne considérons pas l'expression syntaxique de la requête mais plutôt son contexte d'analyse induit qui représente *le contexte d'analyse courant*. Comme le processus de personnalisation est effectué avant l'évaluation de la requête *Q*, le contexte d'analyse induit par *Q* qui est considéré par ce processus ne comporte pas les valeurs des structures. Il englobe les différents éléments structurels d'un contexte d'analyse complet : le fait, les dimensions, les attributs et les prédicats de restriction.

Définition (contexte d'analyse induit par une requête). Soit une requête OLAP *Q*, le contexte d'analyse courant induit par *Q* est défini par : $CAC_Q = (C^{FQ}; C^{DQ}; C^{RQ})$ où :

- $C^{FQ} = F^Q[/f_i(m^Q_i)]+$,
- $C^{DQ} = \{C^{D1},\ldots,C^{Dx}\}$, $\forall j \in [1..x]$, $C^{Dj} = d^Q_j[/(a^Q_{k1},a^Q_{k2})]^+$ où $(a^Q_{k1},a^Q_{k2}) \in A^{Dj} \times A^{Dj} \cup \{All\}$,
- $C^{RQ} = PRED^Q$.

Propriété. CAC_Q est un contexte d'analyse non évalué.

Exemple. Le contexte d'analyse courant induit par Q_1 est défini par $CAC_{Q1}=(CAC_{Q1}{}^F, CAC_{Q1}{}^D, CAC_{Q1}{}^R)$, où :

- $CAC_{Q1}{}^F$= PUBLICATIONS/SUM(NB_PUBLIS)
- $CAC_{Q1}{}^D$={MANIFESTATIONS.HTYP/(All;Type) ; AUTEURS.HEQ/(All;Équipe)}
- $CAC_{Q1}{}^R$={DATES.Année≥2010}

Son arbre de contexte est représenté dans la figure suivante. Le nœud N^{DATES} d'étiquette {*Dates*} est composé avec N^{DATES}.Pred = {DATES.Année≥2010}.

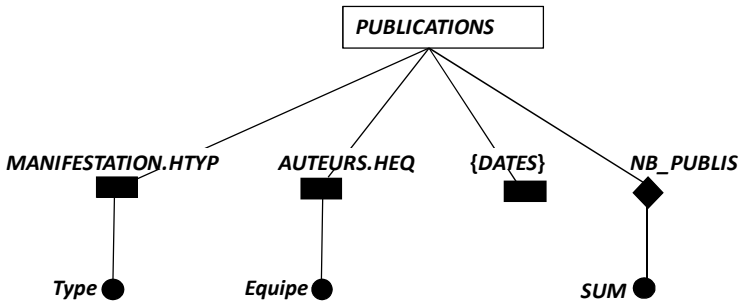


Figure 25. Contexte d'analyse induit par la requête Q_1

La solution la plus directe pour personnaliser une requête serait de considérer les préférences de l'usager qui sont relatives à cette requête (Koutrika et Ioannidis, 2004, 2005a) (les préférences actives). Toutefois, cette solution néglige une caractéristique très importante des préférences OLAP, à savoir la sensibilité au contexte d'analyse. En effet, une préférence est associée à un contexte d'analyse particulier, qui pourrait ne pas correspondre à celui de la requête. Par exemple, la préférence active P_5 ne doit pas être prise en compte dans la personnalisation de Q_1 puisqu'elle est associée à l'analyse des missions, alors que Q_1 correspond à l'analyse des publications. Par conséquent, afin de personnaliser une requête, il convient de trouver, parmi les préférences actives, celles qui sont valides sous le contexte d'analyse induit par la requête. Il s'agit de l'ensemble des préférences candidates sur les valeurs P_V^{CAND} ($P_V^{CAND} \subseteq P^{ACTIV}$).

Définition. Une préférence est candidate pour la personnalisation d'une requête Q si elle est associée au contexte d'analyse induit par Q.

Les préférences absolues étant indépendantes de contextes particuliers, elles sont pertinentes dans tout contexte d'analyse. Ainsi, parmi les préférences actives, toutes les préférences absolues ($P_V^A = \{P_i \in P^{ACTIV} \mid cp_i = \emptyset\}$) sont pertinentes pour le contexte d'analyse courant. Néanmoins, certaines préférences actives contextuelles sont écartées à cause de l'incohérence de leurs contextes avec le contexte d'analyse courant CAC_Q induit par Q. Plus précisément, la personnalisation de Q ne considère parmi les préférences actives contextuelles que celles associées à CAC_Q ($cp = CAC_Q$). Il s'agit de l'ensemble des préférences contextuelles candidates sur les valeurs P_V^C.

Intuitivement, seules les préférences qui sont associées directement à CAC_Q seront prises en compte pour personnaliser Q. Cependant, certaines préférences qui sont associées à une partie du CAC_Q restent aussi valables.

Discussion. La recherche des préférences candidates se ramène donc à une tâche de résolution de contextes. Conformément aux différents types d'appariement de contextes d'analyse (*cf.* chapitre 3, section 2.2.3), les préférences dont les contextes n'apparient pas ou apparient partiellement avec le contexte d'analyse courant sont rejetées. En effet, les préférences associées avec des contextes disjoints avec CAC_Q traduisent les besoins de l'usager dans des contextes d'analyse totalement différents du contexte courant. D'autre part, la prise en compte des préférences de contextes intersectés avec CAC_Q induira des approximations considérables. Par ailleurs, l'emploi de ces préférences va au-delà de la simple personnalisation de la requête en cours, jusqu'à la génération d'une requête qui traduit des besoins différents de ceux exprimés par Q. Ce cas sera étudié dans le chapitre suivant dans le cadre de la recommandation de requêtes.

Par conséquent, les contextes des préférences candidates sont ceux qui apparient totalement avec le contexte courant : la relation entre ces contextes et CAC_Q est une relation de dominance ($cp_i \subset CAC_Q$) ou d'équivalence ($cp_i \equiv CAC_Q$).

Définition (Préférences contextuelles candidates sur les valeurs). Soit une requête *Q*, l'ensemble des préférences contextuelles candidates sur les valeurs pour *Q* est défini par : $P_V^C = \{P_i = (E_i; \theta_i ; cp_i) \in P^{ACTIV}$ tel que *Egal*(cp_i, CAC_Q) ou *Englobe*(CAC_Q, cp_i)$\}$.

Notation. $CP^{CAND} = \{cp^{cand}_i \mid P_i = (E_i; \theta_i ; cp^{cand}_i) \in P_V^C\}$ représente l'ensemble des contextes des préférences candidates.

Définition (Préférences candidates sur les valeurs). L'ensemble des préférences candidates sur les valeurs est défini par : $P_V^{CAND} = P_V^C \cup P_V^A$.

Exemple. Reprenons l'exemple précédent. Étant Disjoint((Missions.Nb_Missions;∅;∅), CAC_{Q1}), P_5 n'est pas candidate pour Q_1. Par ailleurs, étant Chevauche(CAC_{Q1}, (Publications/SUM(Nb_Publis); {Dates.Hmois/(All;Année)} ; {Auteurs.Poste='MCF'})), P_{11} est aussi non candidate pour Q_1.

Conformément au profil de l'usager et à la requête Q_1, $P_V^{CAND} = \{P_1, P_7, P_9, P_{10}\}$.

2.3 Conflits de préférences

L'objectif de la personnalisation d'une requête est de satisfaire les préférences des utilisateurs. Les conflits qui peuvent survenir lors de l'acquisition ou du traitement en ligne des préférences ne doivent pas bloquer l'exécution de la requête.

Définition (conflit). Soient $\mathcal{U}$ un profil utilisateur, $P_i \in \mathcal{U}$ une préférence et *Q* une requête. Un conflit $\mathcal{C}(P_i, X)$ survient entre P_i et X lorsque l'exécution de la requête *Q*, combinée conjonctivement avec la préférence P_i engendre un résultat vide indépendamment des données stockées (X étant la requête *Q* ou une ou plusieurs préférences), ou lorsqu'il existe une incohérence de degrés d'intérêt de l'usager (X étant une préférence).

Chaque conflit est identifié par un tuple (*domaine*, *niveau*, *cause*). Des règles associent à chaque conflit une manière de réagir :

(domaine, niveau, cause) → *manière de réagir*

Domaine

Le domaine d'un conflit définit la situation dans laquelle il est détecté. Nous identifions deux domaines de conflits potentiels : des conflits qui surviennent entre deux ou plusieurs préférences, appelés conflits *préférence-préférence*, et des conflits entre une préférence et une requête, notés par des conflits *préférence-requête*.

Niveau

Lors de la gestion des préférences, des conflits surviennent généralement au niveau syntaxique. Ces conflits sont détectés à l'aide des règles de la logique des prédicats. Par contre, la résolution syntaxique ne permet pas d'identifier certains conflits qui surviennent au niveau sémantique. Afin de décider si une préférence « peut être satisfaite » sur le plan sémantique, le système de gestion de préférences doit stocker dans une métabase de conflits

sémantiques pré-identifiés. Dans la suite, nous étudieront principalement les conflits syntaxiques. Mais, nous présenterons des exemples de conflits sémantiques et nous montrerons comment les résoudre.

Cause

La cause d'un conflit identifie les préférences qui sont à l'origine du conflit.

Manière de réagir

Afin de résoudre les conflits, le système de gestion des préférences réagit suivant des *politiques de résolution de conflits*, en tenant compte des priorités établies entre les éléments à l'origine du conflit. Par exemple, les éléments de la requête ont la priorité par rapport aux préférences car ils sont demandés explicitement par l'utilisateur.

Les politiques de résolution sont définies selon le domaine et la cause du conflit. Elles varient entre rejeter les préférences qui causent problème et privilégier celle la plus importante ou la plus récente.

2.3.1 Conflits hors-ligne

Certains conflits préférence-préférence apparaissent lors de la mise à jour des préférences. La détection et le traitement de ces conflits s'effectuent hors-ligne.

Les conflits hors-ligne sont occasionnés par l'ajout d'une préférence qui est associée au même contexte d'une ancienne préférence et qui est définie sur la même mesure ou sur le même attribut de dimension. La nouvelle préférence est source de conflit si elle définit un prédicat incompatible avec le prédicat de l'ancienne préférence (leur conjonction logique est fausse) ou si elle attribue un score différent au même prédicat de la préférence existante.

Définition (conflit préférence-préférence hors ligne). Soient $P_i = (pred_i; \theta_i ; cp_i)$ et $P_j = (pred_j; \theta_j ; cp_j)$ deux préférences sur le même attribut, avec $cp_i = cp_j$. P_i et P_j sont en conflit si :

- $pred_i \wedge pred_j = F$ ou
- $pred_i = pred_j$ et $\theta_i \neq \theta_j$

Politique 1. En considérant des préférences comme indicateurs d'intérêts positifs, le conflit hors-ligne entre préférences privilégie celle qui admet le score maximum.

Exemple. L'ajout de la préférence (Manifestations.Type='Atelier';0.75; (PUBLICATIONS/SUM(NB_PUBLIS); ∅ ; ∅)) induit un conflit avec la préférence P_7 déjà stockée dans le profil de l'usager. Cette nouvelle préférence n'est pas enregistrée conformément à la politique 1.

2.3.2 Conflits en ligne

Un conflit en ligne est occasionné au moment d'exécution de la requête. Certains conflits préférence-préférence surviennent en ligne. Ils sont détectés suite au processus de résolution de contextes entre des préférences candidates par rapport à une requête. Il s'agit de préférences qui admettent des prédicats incompatibles dont la conjonction logique est fausse. Par ailleurs, une préférence candidate est en conflit avec une requête si son prédicat est incompatible avec les prédicats de la requête. Il faut noter que, bien que certaines préférences

ne soient pas en conflit mutuel avec la requête, elles peuvent occasionner ensemble un conflit avec cette requête. Par exemple, chacun des prédicats de préférences « *Année*=2009 » et « *Ville*='Paris' » est compatible mutuellement avec le prédicat de requête « *Ville*='Toulouse'∨ *Année*='2010' ». Mais, les trois prédicats sont incompatibles ensemble puisque la conjonction logique des trois est fausse.

Définition (Conflit en ligne). Soient Q une requête utilisateur et P_1, …, P_n des préférences candidates par rapport à Q, $\forall i \in [1..n]$, $P_i = (pred_i; \theta_i ; cp_i) \in P_V^{CAND}$.

- P_1, …, P_n sont en conflit si et seulement si $pred_1 \wedge \ldots \wedge pred_n = F$
- P_i est en conflit avec Q si et seulement si $pred_i \wedge pred^Q{}_1 \wedge \ldots \wedge pred^Q{}_u = F$, où $PRED^Q = \{pred^Q{}_1, \ldots, pred^Q{}_u\}$.

Politique 2. Toutes les préférences qui sont à l'origine d'un conflit préférences-préférences en ligne sont rejetées.

Politique 3. Toutes les préférences qui sont en conflit avec une requête sont rejetées.

Exemple. Etant (Année≥2010 ∧ Année=2009) = F, la préférence (Dates.Année=2009; 0.6; ∅), qui est candidate pour Q_1, est en conflit avec la requête. Cette préférence est rejetée conformément à la politique 3.

Contrairement au niveau syntaxique, les conflits sémantiques dépendent de la BDM analysée. Des conflits prédéfinis sont déterminés en fonction de l'instance de la BDM considérée. Parmi les conflits d'ordre sémantique possibles, nous pouvons mentionner les conflits prédéfinis suivants :

- Conflits dû à des règles internes de gestion. Par exemple, dans un laboratoire de recherche, un budget annuel limite de 10000 euros est fixé pour le financement des missions de chaque équipe. Supposons que l'usager admet les préférences P_x = (*Equipe*='SIG'; θ_x ; cp_x) et P_y = (*Année*=2010; θ_y ; cp_y). Si $cp_x = cp_y$, l'ajout d'une préférence P_z = (*SUM(MT_FRAIS)*≥20000; θ_z ; cp_x) associée au même contexte d'analyse occasionne un conflit entre préférences . Par ailleurs, si P_x et P_y sont candidates par rapport à une requête portant sur la somme des frais de missions supérieur à 10000, alors P_x et P_y sont en conflit avec la requête.
- Incohérence sémantique entre deux éléments. Par exemple, une préférence de prédicat « *Année*=2010 » est en conflit avec la requête portant sur la conférence *CIDR* (admet une condition « *Nom='CIDR'* »), tel que *CIDR* est le nom d'une conférence qui a lieu dans des années impaires.

La résolution des conflits au niveau sémantique est effectuée selon la politique 2 si le conflit survient entre des préférences, et suivant la politique 3 s'il s'agit d'un conflit préférence-requête.

Nous ne traitons pas les conflits pouvant survenir au niveau sémantique entre les préférences et les contraintes définies sur le schéma de la BDM (Ghozzi 2004). Nous supposons que les préférences usager sont bien formulées par rapport à ces contraintes.

Le tableau suivant résume les cas de conflits possibles survenant lors du traitement de préférences OLAP contextuelles. La colonne *Autre* correspond au cas des conflits sémantiques prédéfinis. $Pred^Q$ représente la conjonction de tous les prédicats de la requête Q.

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="4">Préférence-Préférence
$\mathcal{C}(P_i, P_j)$</td><td colspan="2">Préférence-Requête
$\mathcal{C}(P_i, Q)$</td></tr>
<tr><td colspan="2" rowspan="2">Cas possibles
Critères</td><td colspan="2">$cp_i = cp_j$</td><td colspan="2">$cp_i \neq cp_j$</td><td colspan="2">$Egal(cp_i, CAC_Q)$
$Englobe(CAC_Q, cp_i)$</td></tr>
<tr><td>$pred_i \wedge pred_j = F$</td><td>$\theta_j \neq \theta_j$
$(pred_i = pred_j)$</td><td>$pred_i \wedge \ldots \wedge pred_j = F$</td><td>Autre</td><td>$pred_i \wedge Pred^Q = F$</td><td>Autre</td></tr>
<tr><td rowspan="2">Timing</td><td>Hors-ligne</td><td>×</td><td>×</td><td></td><td>×</td><td></td><td></td></tr>
<tr><td>En ligne</td><td></td><td></td><td>×</td><td>×</td><td>×</td><td>×</td></tr>
<tr><td rowspan="2">Niveau</td><td>Syntaxique</td><td>×</td><td>×</td><td>×</td><td></td><td>×</td><td></td></tr>
<tr><td>Sémantique</td><td></td><td></td><td></td><td>×</td><td></td><td>×</td></tr>
</table>

Tableau 4. Conflits survenant durant le processus de personnalisation OLAP

Le système de gestion des conflits prend en entrée un ensemble de politiques de résolution des conflits et produit en résultat un ensemble de préférences qui sont compatibles mutuellement (en ligne et hors-ligne) et avec la requête (en ligne). Ces préférences sont dites **homogènes**.

Nous décrivons dans la section suivante comment les préférences sont employées pour personnaliser les requêtes de l'usager.

3 Approche naïve

La BDM est étendue par des profils stockant les préférences des utilisateurs. Le processus de personnalisation de requête consiste à enrichir une requête pour adapter son résultat au mieux au profil de l'utilisateur.

Étant donné une préférence sur les valeurs $P_i = (pred_i; \theta_i; cp_i)$, θ_i traduit l'intérêt de l'utilisateur à considérer la condition $pred_i$ pour personnaliser une requête impliquant le contexte d'analyse cp_i. La personnalisation d'une requête revient donc à intégrer les prédicats des préférences associées au contexte d'analyse induit par la requête, soit les prédicats des préférences candidates.

La Figure 26 illustre le principe général du processus de personnalisation de requête. Étant donné un profil utilisateur $\mathcal{U}$ et une requête utilisateur *Q* qui est exprimée sur une étoile de la BDM, la personnalisation de la requête procède comme suit : l'analyse de *Q* permet de déterminer les préférences actives parmi celles sur les valeurs, puis, seules les préférences candidates homogènes sont sélectionnées pour enrichir la requête *Q*. La requête produite *Q'* permet de restituer un résultat qui satisfait les préférences utilisateur. Les étapes les plus importantes du processus de personnalisation de requête sont : a) sélection des préférences, puis b) intégration des préférences.

Dans la suite de la section 3, nous traitons le cas le plus simple où toutes les préférences candidates sont prises en compte. Nous présentons dans la section 4 les spécificités d'une personnalisation avancée qui tient compte de paramètres supplémentaires.

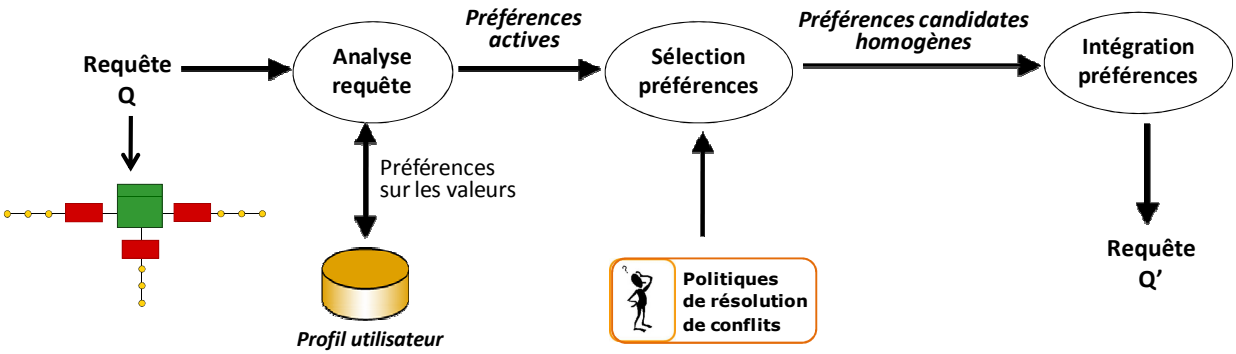


Figure 26. Processus de personnalisation de requête

3.1 Modes de personnalisation

En fonction des préférences utilisées dans la personnalisation d'une requête OLAP *Q*, nous distinguons trois modes de personnalisation : filtrage du résultat, redressement du résultat et le mode combiné.

Filtrage du résultat. La personnalisation élimine une partie du résultat de la requête initiale. Plus précisément, elle élimine (1) des valeurs de paramètres et/ou (2) des valeurs de mesures affichées dans le résultat classique.

- Le premier cas correspond à l'insertion dans la requête initiale de prédicats sur les attributs d'agrégation des dimensions affichées ou sur des attributs plus globaux.
- Dans le deuxième cas, des prédicats sur les mesures agrégées $f(m_i)$ sont insérées dans *Q*.

Au niveau de la restitution, ceci entraine l'élimination de tranches (respectivement de cellules) d'un cube, de lignes ou de colonnes (resp. de cellules) d'une TM, ou de graduations des axes (resp. de points) d'une courbe.

Propriété. Selon le mode filtrage du résultat, le résultat de *Q'* est inclus dans celui de *Q*.

Recalcul du résultat. Ce mode a pour effet d'insérer dans la requête:

- des préférences sur des paramètres de dimensions de filtrage (non affichées), ou
- des préférences sur des paramètres de dimensions affichées, mais de niveaux de granularité inférieurs aux attributs d'agrégation ou d'une autre hiérarchie, ou
- des préférences sur des mesures non agrégées.

Mode combiné. Toutes les préférences candidates sont intégrées dans la requête *Q*.

La sélection des préférences actives prend en compte le mode de personnalisation. Dans la suite, le mode de personnalisation *MPers* sera utilisé comme paramètre pour la configuration du processus de personnalisation d'une requête. S'il n'est pas renseigné, la personnalisation sera effectuée par défaut selon le mode combiné.

3.2 Sélection des préférences

L'étape de sélection de préférences vise à déterminer les préférences candidates par rapport à une requête.

Algorithme

L'algorithme EPCV (Extraction de Préférences Candidates sur les Valeurs) permet de sélectionner les préférences candidates non conflictuelles. L'algorithme prend en entrée une requête utilisateur *Q*, un profil utilisateur $\mathcal{U}$, et éventuellement un seuil λ et un mode de personnalisation *MPers*. Il produit en sortie un ensemble de préférences candidates homogènes sur les valeurs: P_V^{CAND}.

L'algorithme emploie les fonctions et procédures suivantes :

- Active(Q, λ, MPers,$\mathcal{U}$) est une fonction qui cherche les préférences actives par rapport à Q, au seuil λ et au mode MPers.
- BuildCTree(C) est une fonction qui construit l'arbre du contexte d'analyse C conformément aux propriétés présentées dans la section 2.2.2 du chapitre 3.
- GestConflit(P,Q,$\mathcal{R}$) est une procédure qui permet de gérer les conflits en ligne. Elle prend en entrée, en plus de la requête Q et d'un ensemble de préférences P, un ensemble de politiques de résolution $\mathcal{R}$. Elle élimine de P les préférences qui sont mutuellement en conflit ou qui sont en conflit avec Q.

```
Algorithme EPCV (Q,U,λ,MPers)

DEBUT

P^ACTIV←Ø, P_v^CAND←Ø
P^ACTIV←Active(Q,λ, MPers,U)
Si P^ACTIV ≠ Ø Alors
   BuildCTree(CAC_Q)
   Tant que (∃P_i dans P^ACTIV) Faire
      BuildCTree(cp_i)
      Si(Egal(cp_i,CAC_Q) OU Englobe(CAC_Q,cp_i)) Alors
         P_v^CAND← P_v^CAND ∪ P_i
      Finsi
   FinFaire
   Si P_v^CAND ≠ Ø Alors
      GestConflit(P_v^CAND,Q,R)
   Finsi
Finsi
Retourner (P_v^CAND)

FIN EPCV
```

Exemple. Poursuivons l'exemple précédent. P_V^{CAND}= {P_7, P_9} en mode de filtrage du résultat, P_V^{CAND}= {P_1, P_{10}} en mode de redressement du résultat, et P_V^{CAND}={P_1, P_7, P_9, P_{10}} en mode combiné.

Le résultat de l'étape de sélection de préférences est un ensemble de préférences homogènes candidates pour la requête *Q*. Si cet ensemble est vide, le contenu de la BDM représente un contenu qui est parfaitement adapté aux préférences de l'utilisateur en cours sous le contexte d'analyse courant *CAC*. Il s'agit du cas où les préférences ont été définies explicitement dans la requête, ou du cas où l'utilisateur n'admet pas de préférences valides non-conflictuelles qui correspondent à *Q*. Par conséquent, le système exécute la requête d'une manière ordinaire. Autrement, les préférences homogènes seront intégrées dans *Q*.

3.3 Intégration des préférences

L'objectif de l'étape d'intégration de préférences est d'enrichir la requête utilisateur par les préférences sélectionnées et de générer ainsi une nouvelle requête *Q'* qui renvoie un résultat personnalisé tenant compte des préférences de l'utilisateur.

Nous présentons comment intégrer les préférences en gardant le processus de personnalisation de requête indépendant des langages de manipulation. Dans ce cas, cette étape agit sur le contexte d'analyse courant CAC_Q induit par la requête. Il s'agit d'enrichir CAC_Q par les prédicats des préférences sélectionnées, soit de modifier la composante C^{RQ} de CAC_Q. Le résultat est un contexte d'analyse transformé CAC_Q', tel que : $CAC_Q \subset CAC_Q$'. Plus précisément, cette étape prend en entrée :

- le contexte courant $CAC_Q = (C^{FQ}; C^{DQ}; C^{RQ})$, et
- un ensemble de préférences candidates homogènes $P_V^{CAND} = \{(pred_i; \theta_i ; cp_i)\}$.

Elle produit en sortie le contexte d'analyse $CAC_Q' = (C^{FQ}; C^{DQ}; C^{RQ'})$, où $C^{RQ'} = C^{RQ} \cup_{Pi \in P_V^{CAND}} pred_i$.

L'enrichissement de CAC_Q par un prédicat de préférence $pred_i$ (e_i op val_i) consiste à ajouter ou à mettre à jour un nœud composé:

- Ajouter un nouveau nœud composé $\mathcal{N}$ représentant la dimension D_i de l'attribut du prédicat ($e_i \in A^{Di}$) : $InsertN(A^{CACQ}, D_i, N^{Fi}) = CAC_Q'$; avec Fi est le fait de CAC_Q ($D_i \in Star^{CS}(F^i)$), s'il n'existe pas déjà un nœud dans A^{CACQ} représentant D_i.
- Mettre à jour le nœud $\mathcal{N}$: $AddP(A^{CACQ}, N^{Di}, pred_i) = CAC_Q'$, en ajoutant $pred_i$ en conjonction aux prédicats de $\mathcal{N}$, $\mathcal{N}$ étant le nœud représentant le paramètre ou la mesure de la préférence.

Propriété. CAC_Q' est un contexte d'analyse non évalué

L'évaluation de $CAC_{Q'}$ (l'extension de l'arbre de contexte par l'ajout des valeurs des paramètres et celles des mesures correspondantes) génère un contexte d'analyse complet qui représente un résultat personnalisé intégrant les préférences de l'usager.

Exemple. L'intégration des préférences candidates dans le contexte d'analyse induit par Q_1 (*cf.* Figure 25) est assurée par la suite des opérations suivantes :

- $AddP(A^{CACQ1}, N^{Manifestations}, Type \neq \text{'Atelier'}) = A^{CACQ1}{}_1$;
- $AddP(A^{CACQ1}{}_1, N^{Manifestations}, Tx\_Accep \leq 0.35) = A^{CACQ1}{}_2$;
- $AddP(A^{CACQ1}{}_2, N^{Auteurs}, Ville = \text{'Toulouse'}) = A^{CACQ1'}$

Le contexte d'analyse résultant de l'enrichissement de CAC_{Q1} est $CAC_{Q1}' = (CAC_{Q1}{}^F, CAC_{Q1}{}^D, CAC_{Q1}{}^{R'})$, où :

- $CAC_{Q1'}^{F'} = CAC_{Q1}^{F}$
- $CAC_{Q1}^{D'} = CAC_{Q1}^{D}$
- $CAC_{Q1}^{R'}$={Dates.Année≥2010; Auteurs.Ville='Toulouse' ; Manifestations.Type≠'Atelier'∧ Manifestations.Tx_Accep ≤0.35}

3.4 Exemple

(a) Résultat classique de la requête Q_1

PUBLICATIONS *SUM(NB_PUBLIS)*		MANIFESTATIONS.HTYP			
		Type	Atelier	Conférence	Revue
DOMAINES.IDO	**Sous-domaine**				
	Bases de Données		12	14	16
	Entrepôts de données		10	13	18
	Fouille de données		12	10	10
	Recherche d'information		14	16	13
	Traitement d'images		11	8	10
	Systèmes embarquées		20	22	32
	Réseaux sans fils		28	25	25
(Auteurs.Institut='IRIT')					

(b) Résultat filtré avec des préférences sur les dimensions affichées

PUBLICATIONS *SUM(NB_PUBLIS)*		MANIFESTATIONS.HTYP		
		Type	Conférence	Revue
DOMAINES.IDO	**Sous-domaine**			
	Bases de Données		14	16
	Entrepôts de données		13	18
	Fouille de données		10	10
	Recherche d'information		16	13
(Auteurs.Institut='IRIT') **AND** (Domaines.Domaine='Systèmes d'information') **AND** (Manifestations.Type≠'Atelier')				

(c) Résultat redressé avec des préférences sur d'autres dimensions

PUBLICATIONS *SUM(NB_PUBLIS)*		MANIFESTATIONS.HTYP			
		Type	Atelier	Conférence	Revue
DOMAINES.IDO	**Sous-domaine**				
	Bases de Données		10	14	12
	Entrepôts de données		10	12	14
	Fouille de données		8	9	7
	Recherche d'information		14	14	10
	Traitement d'images		11	8	10
	Systèmes embarquées		19	22	26
	Réseaux sans fils		20	22	23
(Auteurs.Institut='IRIT') **AND** (Manifestations.Tx_Accep≤0.35)					

Figure 27. Exemples de résultats personnalisés d'une requête OLAP

Considérons une requête qui affiche le nombre de publications des chercheurs de l'IRIT par type de manifestation et par domaine de recherche secondaire. La Figure 27 (a) montre le résultat de cette requête sans personnalisation. La Figure 27 (b) montre un résultat personnalisé à l'aide de la préférence P_7 (de prédicat « *Manifestations.Type*≠'Atelier' ») et d'une préférence sur le domaine principal (de prédicat « *Domaines.Domaine*='Système d'information' »). Ceci permet de se focaliser sur les valeurs d'attributs des dimensions affichées *Domaines* et *Auteurs*. Le résultat représente un sous-ensemble du résultat classique, c'est-à-dire un résultat filtré. La Figure 27 (c) présente un résultat personnalisé intégrant la préférence P_{10} (de prédicat « *Manifestations.Tx_Accep* ≤0.35 »). Le nombre des publications

est calculé seulement par rapport aux manifestations avec un taux d'acceptation qui ne dépasse pas 35%.

Dans une approche de personnalisation naïve, toutes les préférences candidates sont prises en compte pour personnaliser une requête. Dans la section suivante, nous montrons comment prendre en compte des scénarios plus complexes.

4 Approche avancée

Personnaliser vise à satisfaire au mieux les besoins de l'utilisateur. Théoriquement, il s'agit d'un problème d'optimisation : pour une requête *Q* et un utilisateur donnés, l'objectif est de trouver les éléments du profil $\mathcal{U}$ de l'utilisateur qui, une fois combinés avec *Q*, pourraient maximiser l'intérêt de l'utilisateur en *Q*. Certains considèrent que la personnalisation implique également l'optimisation des intérêts de l'utilisateur en fonction de certaines contraintes comme la charge d'exécution de la requête, la taille limite de son résultat, …Il est donc nécessaire de pouvoir indiquer ces contraintes comme des critères de personnalisation. Ainsi, en fonction des contraintes de personnalisation en cours, des réponses différentes peuvent être parfois produites au même usager et pour la même requête.

Dans cette section, nous décrivons comment peut-on prendre en compte des contraintes extérieures dans le processus de personnalisation.

4.1 Principe

Un mécanisme de personnalisation peut être soumis à des contraintes qui permettent de contrôler la taille du résultat personnalisé et/ou le coût de la personnalisation de requête (Koutrika et Ioannidis, 2005b). Imposer un nombre K limite de préférences que le résultat doit prendre en compte est un moyen pour la mise en œuvre de ces contraintes de personnalisation. Les préférences étant intégrées en tant que des contraintes fortes, le système peut automatiquement déterminer les valeurs appropriées de K en fonction de la contrainte à satisfaire. Par exemple, si le résultat est affiché suivant une courbe, il serait mieux de considérer le plus de préférences afin d'obtenir un résultat peu volumineux, et en même temps efficace puisqu'il intègre un nombre important de préférences. Par contre, si l'usager envisage d'imprimer le résultat sur papier, un nombre K plus restreint peut être choisi. Ainsi, le paramètre K permet de rendre le mécanisme de personnalisation configurable. Il faut noter que le cas par défaut, où K n'est pas renseigné, correspond à une personnalisation naïve tenant compte de toutes les préférences de l'usager.

Dans un processus de personnalisation avancée, les préférences sont triées (*cf.* section 4.2), puis, les K meilleures préférences sont intégrées dans la requête initiale. L'étape de sélection des préférences comprend deux étapes supplémentaires par rapport au processus naïf : le tri des préférences et la détermination du paramètre K à partir d'une contrainte de personnalisation. La Figure 28 met en évidence cette étape.

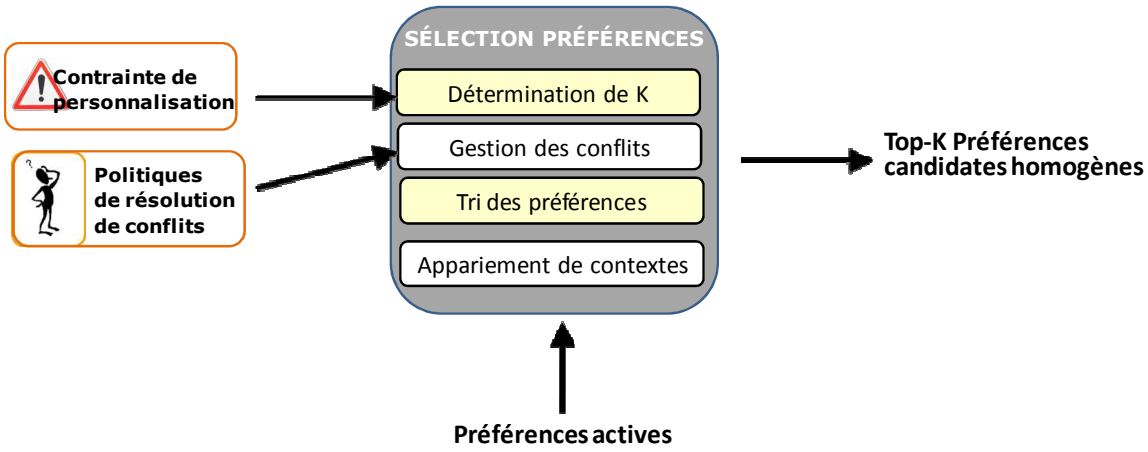


Figure 28. Sélection de préférences dans un processus de personnalisation avancée

4.2 Tri des préférences

Discussion. Intuitivement, les préférences peuvent être triées en fonction de leur degré d'intérêt (traduisant leur pertinence par rapport à l'usager) ou selon le degré d'intérêt de leur contexte par rapport au contexte d'analyse courant (représentant leur pertinence par rapport à la requête) (Stefanidis et al., 2011). Le tri des préférences selon leur degré d'intérêt seulement néglige leur contextualisation. Ainsi, deux préférences de même score seront de même niveau d'importance sous le contexte d'analyse courant bien qu'elles soient associées à des contextes différents. Par ailleurs, une préférence de score 0.8 qui est associée à un contexte d'analyse minimal sera considérée plus importante qu'une préférence de score 0.75 mais qui est associée directement au contexte d'analyse courant. D'autre part, le choix d'un tri qui se base uniquement sur le degré d'intérêt des contextes des préférences n'est pas rationnel. Une préférence admettant un très fort score sera égalisée à une préférence de faible score si elles sont associées au même contexte d'analyse. De plus, ce choix ne permettrait pas de compenser l'écart entre les scores de préférences par celui entre leurs contextes.

Dans la suite, nous présentons un algorithme de tri qui combine les deux modes de tri: les préférences sont triées selon leur pertinence par rapport à l'usager dans le contexte d'analyse courant.

4.2.1 Score de contextes de préférences

Etant tous inclus dans CAC_Q, les contextes de préférences candidates cp^{cand}_i de CP^{CAND} sont plus ou moins proches de CAC_Q : plus cp^{cand}_i couvre une partie importante de CAC_Q, plus il lui est proche. Afin de mesurer ces degrés de similarité, nous définissons le degré de couverture de cp^{cand}_i par rapport à CAC_Q qui est déterminé par le nombre d'éléments en commun entre cp^{cand}_i et CAC_Q. Comme $cp^{cand}_i \subseteq CAC_Q$, ce taux est un réel entre 0 et 1.

Définition (degré de couverture). Soit $cp_i \in CP^{CAND}$, le *degré de couverture* de cp_i par rapport à CAC_Q est défini par :

$$\partial^{CACQ}(cpi) = \frac{Card(cp_i)}{Card(CAC^Q)} \in [0,1] \qquad \textbf{(1)}$$

Card(cp_i) (respectivement *Card(CAC_Q)*) est la cardinalité de l'arbre de contexte de cp_i (resp. CAC_Q) (*cf.* chapitre 3).

Remarque. Par définition, les préférences absolues sont pertinentes dans tous les contextes d'analyse. Elles sont relatives en l'occurrence à CAC_Q. Par conséquent, le degré de couverture du contexte vide ∅ est égal à 1 quelque soit le contexte d'analyse courant CAC_Q.

Exemple. Le contexte d'analyse de la somme des nombres de publications (contexte de P_7) couvre le contexte d'analyse courant CAC_{Q1} à 37,5%, alors que le contexte d'analyse des publications de l'année en cours par équipe d'auteur (contexte de P_9) couvre CAC_Q à 50%.

4.2.2 Relaxation des scores des préférences

L'importance d'une préférence contextuelle varie d'un contexte d'analyse à un autre. Son score initial demeure invariable si le contexte d'analyse courant est son contexte de rattachement ($CAC_Q = cp_i$). Par contre, si le contexte d'analyse est différent de cp_i, son degré d'intérêt serait différent de son score sous cp_i. Par exemple, une préférence de score 0.7 dans le contexte d'analyse des publications serait différent sous le contexte d'analyse des publications des doctorants par mois et par type de manifestation.

Il est donc nécessaire d'effectuer une relaxation des scores des préférences afin de les arrondir au niveau de CAC_Q. Comme le contexte d'une préférence candidate cp_i est inclus ou égal à CAC_Q, le degré d'intérêt d'une préférence sous CAC_Q doit être une fonction du degré d'intérêt de la préférence sous cp_i.

Propriété. Soient $P_i = (pred_i; \theta_i ; cp_i)$ une préférence candidate par rapport à une requête Q et $doi(P_i)^{CACQ}$ son score dans le contexte d'analyse courant CAC_Q, $doi(P_i)^{CACQ} = f(\theta_i)$.

Nous pouvons imaginer plusieurs fonctions de relaxation de score. Cependant, chacune de ces fonctions doit satisfaire la condition suivante, pour être intuitive: $doi(P_i)^{CACQ} = f(\theta_i) \leq \theta_i$. En d'autres termes, le degré d'intérêt d'une préférence diminue puisqu'elle elle est liée à un contexte plus général que CAC_Q. Ainsi, plus le contexte d'une préférence couvre CAC_Q, plus son score sous CAC_Q est proche de son score initial. $doi(P_i)^{CACQ}$ est proportionnel au degré de couverture de cp_i par rapport à CAC_Q.

Définition. Soit $P_i = (pred_i; \theta_i ; cp_i)$, tel que Egal($cp_i$, CAC_Q) ou Englobe(CAC_Q,cp_i), le score de P_i sous CAC_Q est défini par :

$$F_P^{RANK}: [0,1]\times[0,1]\rightarrow[0,1]$$
$$F_P^{RANK}(P_i) = doi(P_i)^{CACQ} = \theta_i \times \partial^{CACQ}(cpi) \text{ ; Englobe}(CAC_Q, cp_i) \quad \textbf{(2)}$$

Comme le montre la figure suivante, $doi(P_i)^{CACQ}$ est une fonction croissante. Pour le même score de préférence θ_i, plus le contexte de préférence couvre CAC_Q, plus le score sous CAC_Q est important. D'autre part, pour des préférences associées au même contexte, plus le score de préférence est important plus le score sous CAC_Q est important.

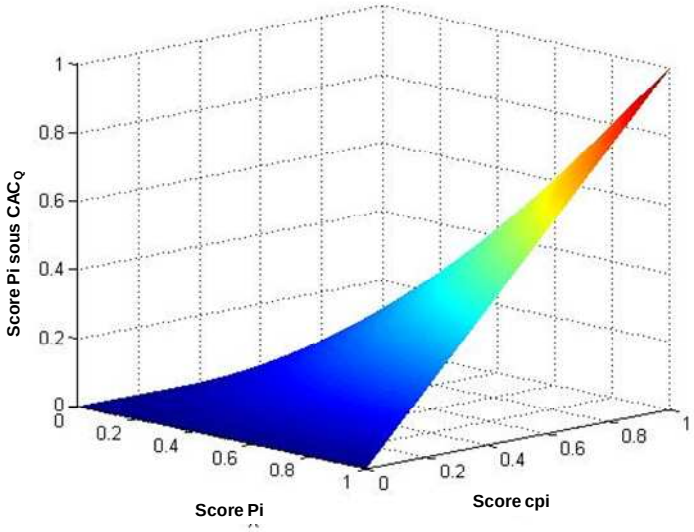


Figure 29. Courbe de relaxation de score des préférences (cas d'appariement total)

Les préférences admettant le même score relaxé sont triées de la plus récente vers la plus ancienne. Ainsi, l'ensemble P_V^{CAND} est totalement ordonné suite à la relaxation des scores. L'élément maximal que peut avoir cet ensemble est P_{max} = (E ; 1 ; CAC_Q), et l'élément minimal possible est P_{max} = (E ; θ_{min} ; cp_{min}), avec $\theta_{min} \rightarrow 0$ et cp_{min} est un contexte d'analyse minimal (constitué d'un seul composant: un fait ou une dimension).

Définition. Pour $P_i \in P_V^{CAND}$ et $P_j \in P_V^{CAND}$, P_i est plus importante que P_j si et seulement si :

- $doi(P_i)^{CACQ} > doi(P_j)^{CACQ}$, ou
- $doi(P_i)^{CACQ} = doi(P_j)^{CACQ}$, avec P_i plus récente que P_j

Exemple. Conformément aux formules précédentes, $doi(P_7)^{CACQ1}$ = 1*0.375 = 0.37 ; $doi(P_9)^{CACQ1}$ = 0.9*0.5 = 0.45 ; et $doi(P_1)^{CACQ1}$=0.75*1 = 0.75.

4.3 Sélection des Top-K préférences

L'algorithme `EPCV-K` permet de sélectionner les K meilleures (Top-K) préférences candidates homogènes sur les valeurs $P^{CAND\text{-}K}$. Il représente une redéfinition de l'algorithme `EPCV`. Il prend en entrée un paramètre *K* en plus des éléments d'`EPCV`. Les préférences sont triées selon leur score sous CAC_Q.

Intuitivement, l'algorithme doit suivre les même étapes de l'algorithme `EPCV`, puis trie les préférences candidates renvoyées et en retourne les K meilleures. Afin de réduire le nombre d'itérations d'appariement de contextes, `EPCV-K` trie les préférences actives avant d'effectuer l'appariement, puis parcourt l'ensemble des préférences actives dans l'ordre décroissant. Il s'arrête lorsque la $K^{ième}$ préférence active dont le contexte apparie avec CAC_Q est sélectionnée. Les K préférences sélectionnées représentent les K meilleures préférences candidates homogènes.

L'algorithme `EPCV-K` utilise autrement la procédure `GestConflit` et emploie une nouvelle fonction `Rank-P`:

- `GestConflit` est appliquée sur les préférences actives. Parmi les préférences non éliminées par cette procédure, celles dont les contextes apparient avec CAC_Q représentent les préférences candidates homogènes.
- `Rank-P(`$\{P_1, P_2, ...\}, C$`)` est une fonction qui permet de trier les préférences P_1, P_2, … selon $doi(P_i)^C$

Algorithme EPCV-K(Q,U,λ,MPers, K)

```
DEBUT

P^ACTIV←Ø, P_v^CAND-K←Ø, p←0
P^ACTIV←Active(Q,λ, MPers,U)
GestConflit(P^ACTIV,Q,R)
Si P^ACTIV ≠ Ø Alors
   Calculer doi(P_i)^CACQ pour chaque P_i dans P^ACTIV
   P^ACTIV ←Rank-P(P^ACTIV,C)
   BuildCTree(CAC_Q)
   Tant que (∃P_i dans P^ACTIV) ET (p ≤ K) Faire
      BuildCTree(cp_i)
      Si(Egal(cp_i,CAC_Q) OU Englobe(CAC_Q,cp_i)) Alors
         P_v^CAND-K ← P_v^CAND-K ∪ P_i
         p←p+1
      Finsi
   FinFaire
Finsi
Retourner (P_v^CAND-K)

FIN EPCV-K
```

Propriété. Le nombre d'itérations d'appariement de contextes dans un processus de personnalisation avancée est plus réduit que celui d'une personnalisation naïve, indépendamment du type de la contrainte de personnalisation à satisfaire.

Preuve. Soit n le nombre des préférences candidates. Le nombre d'itérations d'appariement de contextes nécessaires pour une personnalisation naïve est égal à n. Soit m le nombre d'itérations pour une personnalisation avancée.

1er cas. K≤n: Comme les préférences actives sont pré-triées et traitées dans l'ordre décroissant de leurs scores relaxés, l'appariement de contextes est effectué seulement pour un sous-ensemble de ces préférences jusqu'à l'extraction de K préférences candidates. K≤m≤n. Il faut noter que m n'est pas connu à l'avance : plus les contextes des préférences actives les mieux ordonnées sont appariés avec CAC_Q, moins d'itérations d'appariement sont nécessaires.

2ème cas. K>n: Les n préférences candidates sont sélectionnées. Comme pour la personnalisation naïve, l'appariement de contexte est effectué pour toutes les préférences actives. m = n.

Exemple. Considérons la requête Q_1 et les préférences de l'exemple initial. Rappelons que P^{ACTIV} = {P_1, P_5, P_7, P_9, P_{10} et P_{11}} et P_V^{CAND}={P_1, P_7, P_9, P_{10}}. L'ensemble ordonné des préférences actives est : < P_1, P_{10}, P_{11}, P_9, P_7, P_5>. Supposons que K est fixé à 3,

P_v^{CAND-3} = {P_1, P_{10}, P_9}. L'algorithme de sélection des préférences s'arrête après quatre itérations d'appariement de contextes. Dans le cas où K=4 (toutes les préférences candidates sont sélectionnées), l'algorithme `EPCV-K` effectue 5 itérations d'appariement au lieu de 6 itérations pour un algorithme de sélection dans le cadre d'une personnalisation naïve.

Remarque. L'étape d'intégration des préférences dans un processus de personnalisation avancée est pareille à celle d'une personnalisation naïve. Elle prend en entrée l'ensemble des K meilleures préférences candidates homogènes P_V^{CAND-K}.

5 Bilan

L'objectif de ce chapitre est de proposer un processus de personnalisation des requêtes OLAP en fonction de préférences contextuelles stockées dans un profil utilisateur (Jerbi et al., 2010b, 2010c). Les préférences permettent d'enrichir la requête initiale afin de générer une requête susceptible de renvoyer un résultat personnalisé. Ce processus comporte deux phases majeures : la sélection, puis l'intégration des préférences.

La sélection des préférences à employer pour la personnalisation suit plusieurs étapes :

- La détermination des préférences qui sont relatives à la requête.
- L'extraction parmi ces préférences de celles liées au contexte d'analyse courant. Ceci est effectué par un appariement total entre les contextes de préférences et le contexte induit par la requête.
- La gestion des conflits pouvant survenir selon différentes politiques de résolution.

Selon une perspective utilisateur, toutes les préférences retenues sont intégrées dans la requête afin de maximiser l'intérêt de l'usager au résultat. Dans certains cas, seules les K meilleures préférences doivent être utilisées. Ceci correspond à un mécanisme de personnalisation avancée correspondant à une perspective système, où la personnalisation est soumise à des contraintes. Dans ce cas, une étape supplémentaire est nécessaire pour trier les préférences afin de considérer les meilleures. Contrairement aux travaux où les préférences contextuelles sont triées en fonction de leur degré d'intérêt (Koutrika et Ioannidis, 2004, 2005a) ou selon le degré d'intérêt de leur contexte (Stefanidis et al., 2011), nous avons proposé la fonction de score F_P^{RANK} qui combine entre ces deux critères: les préférences sont triées selon leur pertinence par rapport à l'usager dans le contexte d'analyse courant.

La personnalisation admet différents impacts sur le résultat de la requête qui varient entre le filtrage, le redressement du résultat et la combinaison de deux.

L'emploi de préférences contextuelles permet de personnaliser, d'une part, le résultat pour différents usagers (des résultats différents pour une même requête appliquée par différents usagers), et d'autre part, le résultat pour un même usager lorsqu'il change de contexte d'analyse (affichage de toutes les années lors de l'analyse des publications et de l'année en cours lors de l'analyse des missions).

Pour assurer la flexibilité de la personnalisation, la sélection des préférences et l'enrichissement de la requête sont indépendants du langage de définition des requêtes et de la forme d'affichage des résultats. Le processus de personnalisation de requête agit sur une représentation arborescente interne des requêtes et produit le résultat sous la même forme générique.

Le processus de personnalisation admet quelques limites :

- La résolution des conflits sémantiques doit être étudiée en détail afin d'élargir les domaines de conflits pris en compte.
- Il convient de définir un mécanisme de détermination du paramètre K en fonction de contraintes système pour les différents types de préférences.

Références

Ghozzi, F. (2004). Conception et manipulation de bases de données dimensionnelles à contraintes, Thèse de doctorat, Université Paul Sabatier, Toulouse 3 (France), novembre 2004.

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2010b). Personnalisation du contenu des bases de données multidimensionnelles. Journées Francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA), pages 5–20.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2010c). A Framework for OLAP Content Personalization. East European Conf. on Advances in Databases and Information Systems (ADBIS), Springer-Verlag, pages 262–277.*

Koutrika, G. Ioannidis, Y. E. (2004). Personalization of queries in database systems. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 597–608.

Koutrika, G., Ioannidis, Y. E. (2005a). Personalized queries under a generalized preference model. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 841–852.

Koutrika, G., Ioannidis, Y. (2005b). Constrained Optimalities in Query Personalization. ACM SIGMOD Intl. Conf.on Management of Data (SIGMOD), ACM Press, pages 73–64.

Rizzi S. (2007). OLAP preferences: a research agenda. Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 99–100.

Stefanidis, K., Pitoura, E., Vassiliadis, P. (2007). Adding context to preferences. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 846–855.

Stefanidis, K., Koutrika, G., Pitoura, E. (2011). A Survey on Representation, Composition and Application of Preferences in Database Systems. ACM Transactions on Database Systems (TODS), Vol. 36, No. 3.

Chapitre 5

Personnalisation de la navigation OLAP

Sommaire

1 Introduction

Ce chapitre présente la deuxième action de la personnalisation des analyses OLAP, à savoir la personnalisation de la navigation.

Le mécanisme de personnalisation de requêtes du chapitre précédent agit sur la requête de l'usager avant son exécution et permet de l'enrichir afin de personnaliser son résultat. Dans ce chapitre, nous montrons comment faciliter les analyses OLAP en intégrant un mécanisme de recommandation de requêtes.

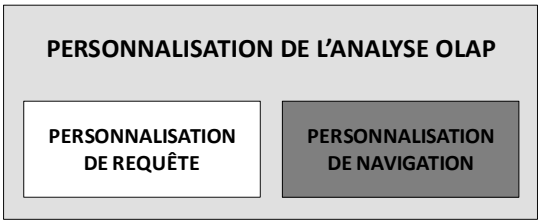


Figure 30. Positionnement de la recommandation de requêtes

Afin de répondre à la multiplicité des systèmes OLAP, le processus de recommandation doit être indépendant du langage de définition de requête et de la structure d'affichage du résultat. Par ailleurs, pour limiter les manipulations de l'usager, ce processus doit être flexible afin de permettre des recommandations suivant différents niveaux d'action de l'usager dans le processus de requêtage. Ces recommandations doivent être adaptées à l'usager afin de faciliter sa tâche. L'étude d'un processus de recommandation OLAP flexible et personnalisé n'a pas été traitée dans les travaux existants.

Plan du chapitre. La section 2 expose le cadre général de recommandation OLAP ainsi que nos différents choix de recommandation. La section 3 présente les étapes de génération des recommandations candidates. L'algorithme de recommandation global est présenté dans la section 4.

2 Cadre de recommandations OLAP

Nous avons adopté le modèle de navigation basée sur l'objectif (Dittrich et al., 2005; Thalhammer et al., 2001). Selon ce modèle, à chaque étape de l'analyse, l'usager définit une requête, observe le résultat, puis définit la requête suivante en conséquence.

Afin d'assister l'usager au cours de l'analyse OLAP, le système de recommandation reproduit le même principe de navigation de l'usager en simulant son comportement durant la construction d'une requête, puis lors de la recherche de la requête suivante. Les recommandations sont générées aux différents niveaux en fonction du profil de l'usager.

Afin de formaliser le problème des recommandations OLAP, nous définissons la fonction ***ORecommend*** *(CAC,scénario,* $\mathcal{U}$ *[,options])* qui génère une ou plusieurs recommandations en fonction du profil $\mathcal{U}$ de l'usager suivant le scénario de recommandation et d'autres options.

2.1 Recommandations flexibles

L'intervention du système de recommandation OLAP dans l'analyse de l'usager varie de la proposition d'une partie de la requête en cours de formulation à la suggestion d'une ou de plusieurs requêtes après l'application d'une requête complète. Conformément à notre modélisation de contexte d'analyse, une requête OLAP constitue un contexte d'analyse non-évalué et son résultat est un contexte d'analyse complet. Le problème de recommandation se réduit à la recherche d'un contexte d'analyse non-évalué à partir du contexte d'analyse courant *CAC* qui est induit par la requête.

Assistance interactive à la définition de requêtes OLAP

Le scénario de recommandation par assistance interactive (scénario 1) a lieu au cours de la formulation de la requête. Le processus de recommandation prend en entrée le contexte d'analyse induit par la requête en cours de formulation. Dans ce cas, *CAC* est un contexte d'analyse partiel.

Muni d'une potentialité de recommandation, le système OLAP est en mesure de traiter des requêtes textuelles incomplètes qu'il enrichit automatiquement. Dans ce cas, la requête de l'usager doit comporter au moins le fait et une dimension de l'analyse. La recommandation est assurée par l'appel de la fonction *ORecommend* (*CAC*,1, $\mathcal{U}$). Par ailleurs, dans le cas où la requête est définie graphiquement d'une manière incrémentale (Ravat et al., 2008), le système peut proposer un élément de la requête à chaque étape. Ceci correspond à l'appel de la fonction *ORecommend* (*CAC*,1, $\mathcal{U}$, *type*), où *type* décrit l'élément à proposer (une mesure, une dimension, un attribut de dimension ou un prédicat de restriction).

Exemple. Supposons que l'usager utilise un langage graphique pour définir sa requête (Ravat et al., 2007b ; 2008). Le système de recommandation complète interactivement la requête en fonction des sélections de l'usager. L'usager sélectionne à partir de la constellation les éléments de la requête d'une manière incrémentale : le fait *Publications* (❶), puis la mesure *Nb_pulis* (❷), puis la dimension *Auteurs* (❸), puis le paramètre *Poste* (❹). A ce niveau, le système recommande l'autre axe de l'analyse (*Dates*) ainsi que le niveau de détail correspondant (*Année*). Le système propose également une restriction de l'analyse (*Manifestations.Type≠'Atelier Nat.'*). Les éléments recommandés sont présentés en couleur distinguée dans la Figure 31. Après la validation de la recommandation par l'usager, le système évalue la requête correspondante et affiche le résultat.

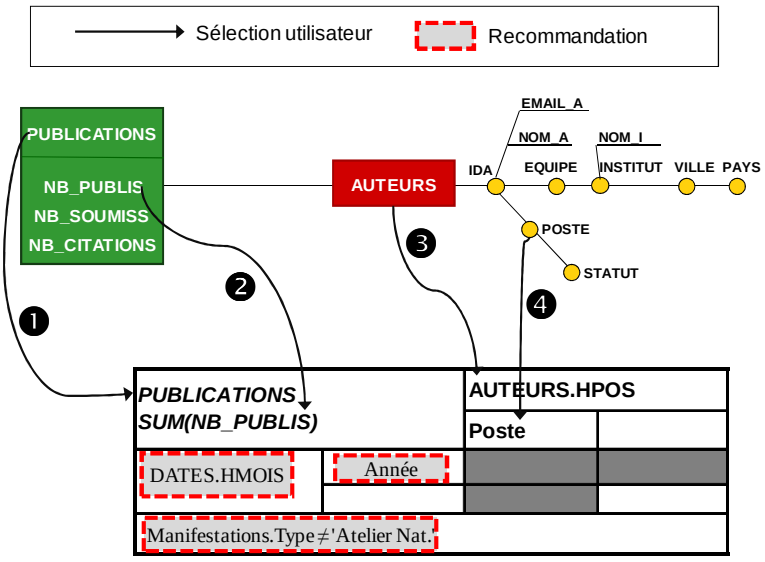


Figure 31. Exemple d'assistance interactive à la définition d'une requête graphique

Recommandation de requêtes

Après l'application d'une requête à partir du contexte d'analyse CA_i, le système de recommandation détermine des contextes d'analyse qui pourraient aider l'usager dans sa navigation de la BDM (*cf.* Figure 21):

- par anticipation en déterminant un futur contexte d'analyse CA_{i+j} que l'usager est susceptible de construire (scénario 2),
- par alternatives en suggérant les contextes d'analyse CA_k alternatifs à la navigation de l'usager (scénario 3).

Le contexte d'analyse courant *CAC* en entrée du processus de recommandation est le résultat de la requête de l'usager. *CAC* est un contexte d'analyse complet.

2.2 Recommandation en fonction de profil

Par définition, les préférences contextuelles permettent de renseigner sur les besoins spécifiques de l'usager dans des contextes particuliers (quoi analyser ? et quand ?). Considérons une préférence qui est rattachée au contexte d'analyse représentant le résultat d'une requête *Q*. Cette préférence précise ce que l'usager demanderait d'analyser quand il observe le résultat de *Q*. Ainsi, la prise en compte de cette préférence dans la transformation du résultat de *Q* permettrait de produire un contexte d'analyse qui correspond aux besoins de l'usager. De même, une préférence associée avec une partie d'une requête permet d'indiquer les éléments à insérer dans cette requête.

Dans notre approche, les recommandations sont générées suite à l'appariement du contexte d'analyse induit par la requête avec les contextes de préférences. Par rapport aux travaux existants, nos recommandations sont obtenues à partir d'une matrice (Requête∪Résultat)×Fragments de requêtes.

Recommandation et appariement

Une préférence $P_i = (E_i; \theta_i; cp_i)$ traduit l'intérêt de l'usager à sélectionner l'élément E_i dans le contexte d'analyse cp_i. L'élément E_i est exploité pour générer une recommandation en fonction de la relation entre cp_i et le contexte courant *CAC*.

- *cp_i et CAC apparient totalement* (*cf.* chapitre 3, section 2.2.3): *Egal*(cp_i,*CAC*) ou *Englobe*(*CAC*,cp_i). À partir du contexte d'analyse cp_i, l'insertion de l'élément E_i permet de passer à un nouveau contexte cp_{i+1} que l'usager devrait construire durant son processus d'analyse. Plus généralement, l'emploi de l'élément E_i dans la transformation d'un contexte qui est égal à cp_i ou qui est plus général que cp_i permet de produire un nouveau contexte d'analyse qui répond à un besoin anticipé de l'usager. Dans le cadre de nos travaux, nous proposons d'employer les préférences dont les contextes apparient totalement avec le contexte d'analyse courant pour la génération d'une recommandation par anticipation.
- *cp_i et CAC apparient partiellement* (*cf.* chapitre 3, section 2.2.3) : *Chevauche*(cp_i,*CAC*). L'objectif de la recommandation d'alternatives est de proposer des contextes d'analyse considérés comme nouveaux dans l'analyse de l'usager. Une solution envisageable est de proposer à l'usager des éléments qu'il aurait sélectionnés dans des contextes proches du contexte courant. La recommandation d'alternatives est basée sur un appariement partiel de contextes d'analyse.
- *cp_i et CAC n'apparient pas* (*cf.* chapitre 3, section 2.2.3): *Disjoint*(cp_i, *CAC*). Aucun lien n'existe entre cp_i et *CAC*. Les contextes de préférence répondant à cette propriété sont rejetés.

3 Génération de recommandations candidates

But. L'objectif de cette section est de spécifier un mécanisme de génération des différents types recommandations de la section 2.1 en fonction des préférences du décideur.

Exemple de référence. Afin d'illustrer le mécanisme de génération de recommadations, nous considérons dans la suite la requête Q_2 qui permet d'afficher le nombre des publications par année et par niveau de manifestation.

La génération d'une recommandation à partir du contexte d'analyse courant est effectuée d'une manière progressive.

Etape de redressement

Une première transformation permet le redressement du contexte d'analyse courant à travers le changement des indicateurs de l'analyse, la réorientation de l'analyse, le changement de sa précision, ou la focalisation de l'analyse. Cette transformation produit un contexte d'analyse partiel, éventuellement non-évalué. Il est considéré en tant que contexte intermédiaire.

Etape d'accomplissement

Si le contexte d'analyse redressé est partiel, le système l'enrichit jusqu'à la construction d'un contexte partiel non-évalué. Chaque itération prend en entrée le contexte d'analyse intermédiaire produit par l'itération précédente. Les différents types de transformation de contexte sont détaillés dans la section 3.2.

Etape d'affinement

Le contexte d'analyse intermédiaire partiel non-évalué est affiné afin de générer le contexte d'analyse à recommander. Ceci consiste à enrichir le contexte par des prédicats de restriction sur les valeurs. Il s'agit d'appliquer le mécanisme de personnalisation des requêtes détaillé dans le chapitre 4 au contexte d'analyse non-évalué généré par l'étape d'accomplissement.

Notation. $CA^{Rec} = \{CA^{Rec}_0, CA^{Rec}_1, \ldots\}$ désigne l'ensemble des contextes d'analyse à recommander. *CAC* représente le contexte d'analyse courant et CA^i est un contexte d'analyse intermédiaire.

Il faut noter que le processus de génération de recommandation peut se raffiner en quelques sous-processus qui, chacun, produit une recommandation candidate. Dans ce cas, les recommandations sont triées, puis filtrées avant d'être retournées à l'usager.

La Figure 32 présente les deux étapes principales de la construction des recommandations OLAP : la génération des recommandations candidates, puis le tri et le filtrage des recommandations (section 4.2).

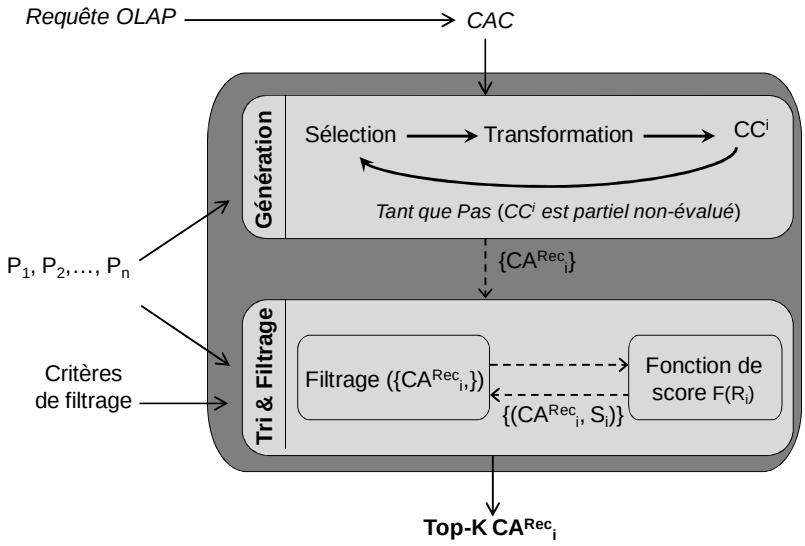


Figure 32. Mécanisme de construction des recommandations OLAP

Les deux sous-sections suivantes présentent respectivement l'étape de sélection de préférences et l'étape de transformation d'un contexte d'analyse en fonction des préférences.

3.1 Sélection des préférences

L'objectif de l'étape de sélection des préférences est de déterminer les éléments à utiliser lors des différentes étapes de génération des contextes d'analyse à recommander. Il s'agit de rechercher les préférences qui sont associées au contexte d'analyse en construction. Comme un contexte d'analyse est composé d'éléments structurels ainsi que de valeurs, la sélection peut concerner les préférences sur les structures ainsi que celles sur les valeurs.

Le processus de sélection des préférences est déclenché à chaque itération de la génération des recommandations et renvoie à chaque fois les préférences admettant le score le plus

important dans le contexte d'analyse courant. Il faut noter que les préférences intégrées dans le contexte d'analyse ne sont plus prises en compte dans les itérations suivantes.

Le processus de sélection de préférences varie selon le scénario de recommandation. En effet, en plus du type d'appariement de contextes qui change d'un scénario à un autre, le mode de calcul des scores des préférences varie également en conséquence.

3.1.1 Score de préférence

Le calcul du score relaxé d'une préférence P_i sous le contexte d'analyse courant *CAC* dépend de la relation entre le contexte de préférence cp_i et *CAC*. Lorsque cp_i apparie totalement avec *CAC*, le score relaxé de P_i est calculé en fonction du degré de couverture de cp_i (*cf.* section 4.2.1 du chapitre 4). Cependant, la relaxation de score est différente dans le cas d'appariement partiel où cp_i et *CAC* sont intersectés. Dans ce cas, moins cp_i est proche de *CAC*, plus le score de P_i sous *CAC* diminue. Ainsi, $doi(P_i)^{CAC}$ est proportionnel au *degré de rapprochement* de cp_i par rapport à *CAC*.

Distance d'édition de deux contextes d'analyse intersectés. La distance d'édition est une extension de la distance de chaînes de caractères (Navarro, 2001) dans le domaine des graphes. La distance d'édition de deux graphes *G1* et *G2* permet les suppressions, les insertions et les substitutions de nœuds afin de transformer *G1* en *G2*. A chaque opération d'édition est associé un coût (une valeur réelle positive). La distance d'édition entre *G1* et *G2* est le coût minimal de toute la séquence d'opérations transformant *G1* en *G2*. Plus cette distance est petite, plus deux graphes sont similaires. Cette distance convient donc pour mesurer la similarité de deux arbres de contexte intersectés.

Les opérations autorisées afin de comparer deux arbres de contexte A^{C1} et A^{C2} sont les opérations élémentaires de transformation d'arbre. Toutes les opérations contribuent avec un côut qui est égal à 1. Une raison intuitive pour ce choix est de tenir compte du niveau de hiérarchie impliqué. Par exemple, l'insertion d'une dimension, qui est suivie de l'insertion de paramètres, sera plus chère que l'insertion d'un paramètre seulement. D'autre part, ce choix permet de valoriser les opérations effectuées en fonction des liens impliqués par le nœud inséré ou mis à jour. Par exemple, la suppression d'un paramètre sera valorisée en fonction du nombre de nœuds valeur qui lui correspondent.

Définition. Soient deux contextes d'analyse intersectés C_1 et C_2 représentés respectivement par les arbres de contexte A^{C1} et A^{C2}. La distance entre les deux contextes C_1 et C_2, notée $d_{ca}(A^{C1},A^{C2})$, est le nombre minimum d'opérations élémentaires nécessaires pour transformer A^{C1} en A^{C2}.

Exemple. Considérons la requête Q_2. Soient les deux contextes suivants :

- C_{11} : Analyse du nombre des publications par trimestre
- C_{12} : Analyse des missions par niveau de manifestation et par année par mois

La Figure 33 présente les arbres de C_{11}, C_{12} et CAC_{Q2} (le contexte d'analyse résultat de Q_2). Pour des mesures de simplification, les nœuds valeur de CAC_{Q2} ne sont pas présentés. Une séquence d'opérations qui transforme C_{11} en CAC_{Q2} est : mettre à jour l'étiquette du nœud « *Trimestre* » par « *Année* » ; insérer les nœuds « *Manifestations* » et « *Niveau* ». Sachant que CAC_{Q2} comporte 29 nœuds valeur, le coût de cette transformation est 32. La séquence d'opérations transformant C_{12} en CAC_{Q2} est : mettre à jour l'étiquette du fait « *Mission* » ;

supprimer le nœud « *Mois* », insérer la mesure *NB_Publis* ainsi que la fonction *SUM* . Le côut de cette transformation s'élève à 33.

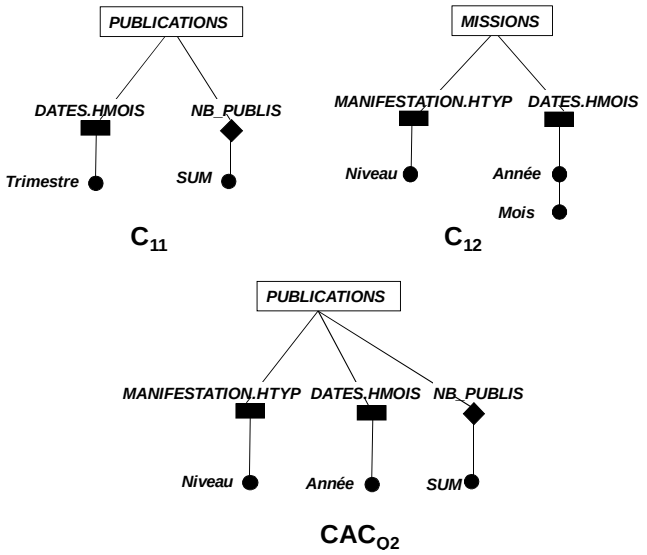


Figure 33. Exemples de contextes d'analyse intersectés

Définition. Soient deux contextes d'analyse intersectés C_1 et C_2 représentés respectivement par les arbres de contexte A^{C1} et A^{C2}. Le degré de rapprochement de C_1 à C_2 est défini par :

$$\chi^{C1}(C_2) = \frac{|A^{C1}|}{|A^{C1}| + d_{ca}(A^{C2}, A^{C1})} \in [0,1] \quad \textbf{(3)}$$

Ainsi, le score d'une préférence $P_i = (E_i; \theta_i; cp_i)$ sous *CAC*, en cas de chevauchement entre cp_i et *CAC*, est défini par :

$$F_P^{RANK}: [0,1]\times[0,1]\rightarrow[0,1]$$
$$F_P^{RANK}(P_i) = \text{doi}(P_i)^{CAC} = \theta_i \times \chi^{CAC}(cp_i)\ ;\ \text{Chevauche}(cp_i, CAC) \quad \textbf{(4)}$$

3.1.2 « EPC-ByMatch »

La fonction `EPC-ByMatch` utilisée dans l'algorithme de génération des recommandations permet de trouver une préférence candidate dont le contexte apparie avec le contexte d'analyse courant. Seules les préférences avec le score relaxé maximal sont retournées.

Définition. Soient un contexte d'analyse courant *CAC* et un profil utilisateur $\mathcal{U}$. $P_i \in \mathcal{U}$ est candidate par rapport à *CAC* si et seulement si cp_i apparie avec *CAC* et $\text{doi}(P_i)^{CAC}$ est maximal.

Syntaxe. `EPC-ByMatch`(*CA*, $\mathcal{U}$, *Cible* [,*Mode*]), où :

- CA est un contexte d'analyse
- $\mathcal{U}$ est un profil utilisateur
- *Cible* renseigne sur le type de la préférence recherchée : une mesure associée au fait *NomF* (*NomF*:m), une dimension connectée au fait *NomF* (*NomF*:D), un attribut d'une dimension *NomD* (*NomD*:a). Si le paramètre *Cible* est Null, la recherche

concerne tous les types de préférences dont les prédicats sur les valeurs des paramètres de dimensions ou sur les valeurs d'une mesure agrégée $f^{Agreg}(m)$.

- *Mode* représente le mode d'appariement des contextes (total ou partiel).

Cette fonction renvoie un ensemble $P^{CAND} = \{P_1, P_2, \ldots\}$ de préférences candidates pour le contexte *CA*. Si plusieurs préférences admettent le score relaxé maximal, elles sont toutes retenues. En effet, il est difficile de savoir a priori quelle préférence contribuera au mieux à la construction de la recommandation finale. Ainsi, nous choisissons de ne donner à cette étape aucune priorité à des éléments par rapport à d'autres, puisque la pertinence du choix serait effective lors de la construction du contexte entier à la fin du processus progressif. Il est noté que le nombre de préférences partageant le même score n'est probablement pas important vu la relaxation des scores qu'elles subissent.

Sélection de l'élément pivot

La sélection de l'élément pivot correspond à la recherche de l'élément qui va redresser le contexte d'analyse courant. Aucune contrainte sur le type de cet élément n'est imposée. Ceci correspond à l'intuition de l'usager qui peut, à l'observation du résultat de la requête, modifier ou ajouter une mesure, une dimension, un niveau de détail ou une restriction sur les valeurs. L'élément sélectionné est celui le plus important suivant le contexte d'analyse courant *CAC* traduisant le résultat de la requête. La relaxation du score au niveau de *CAC* tient compte du scénario de recommandation.

La sélection de l'élément pivot est assurée par l'appel de la fonction `EPC-ByMatch` (*CAC*, $\mathcal{U}$, Null, *Mode*), où *CAC* représente un contexte d'analyse complet et *Mode* indique si l'appariement est total (pour les scénarios de recommandation 1 et 2) ou partiel (pour le scénario 3).

Condition. Le fait courant n'est pas remplacé. Ainsi seules les préférences qui sont associées à ce fait sont considérées. Soit F_i le fait de *CAC*, une préférence $P_i = (E_i; \theta_i; cp_i)$ est sélectionné si E_i est:

- une mesure m_k de F_i ($m_k \in M^{Fi}$),
- une dimension D_k qui est connectée à F_i ($D_k \in Star^{CS}(F_i)$),
- un paramètre a_k d'une dimension connectée à F_i ($a_k \in A^{Dk}$, avec $D_k \in Star^{CS}(F_i)$),
- une condition sur les valeurs d'un paramètre a_k d'une dimension connectée à F_i ($a_k \in A^{Dk}$, avec $D_k \in Star^{CS}(F_i)$),
- une condition sur les valeurs d'une mesure m_k ou une mesure agrégée $f_k(m_k)$, tel que $f_k(m_k) \in \in M^{Fi}$.

Sélection des éléments d'accomplissement

La sélection des préférences lors d'une itération d'accomplissement vise à déterminer des éléments structurels afin de compléter un contexte d'analyse intermédiaire. Il s'agit de rechercher une dimension ou un attribut de dimension. Contrairement à l'itération de redressement où *CAC* est apparié totalement ou partiellement avec les contextes des préférences, la recherche d'un élément d'accomplissement est effectuée par appariement total.

La sélection d'un élément d'accomplissement est assurée par l'appel de la fonction `EPC-ByMatch`(*CA*^{i}, $\mathcal{U}$, *Cible*,*'total'*), où :

- *CA*^{i} représente un contexte d'analyse intermédiaire partiel

- *Cible* = *NomF*:D s'il est nécessaire de déterminer une dimension pour le fait *NomF*, *Cible* = *NomD*:a si l'élément recherché est un attribut de la dimension *NomD*. Il faut noter qu'aucune sélection de mesure n'est nécessaire lors de cette itération.

Pour l'étape d'affinement, CA^n étant un contexte partiel non-évalué, la sélection des préférences concerne les préférences sur les valeurs qui sont candidates pour CA^n. Il s'agit de faire appel à l'algorithme EPCV (*cf.* chapitre 4, section 3.2).

3.2 Transformation de contexte d'analyse

La transformation d'un contexte d'analyse revient à modifier son arbre. Deux modes de transformation sont possibles:

Enrichissement du contexte d'analyse par un nouveau composant,

Substitution d'un composant du contexte par un ou plusieurs autres composants.

Soit $P^{CAND} = \{P_1, P_2, \ldots\}$ l'ensemble des préférences candidates pour *CAC*. S'il existe plusieurs préférences candidates, chaque préférence est intégrée séparément dans le contexte courant afin de générer un contexte d'analyse intermédiaire. Ainsi, plusieurs contextes d'analyse candidats à la recommandation sont ensuite générés. Toutefois, l'ensemble des préférences candidates peut être vide. Dans le cas d'absence de préférences correspondant à l'élément pivot, aucune recommandation n'est retournée. Par contre, des règles d'accomplissement par défaut sont employées pour pallier l'absence des préférences dans la phase d'accomplissement du contexte d'analyse.

Afin de pouvoir spécifier les modes d'intégration des préférences dans un contexte d'analyse, nous introduisons deux opérateurs de transformation de contexte:

- l'opérateur « $\oplus$ » : $P_i \oplus CA$ signifie que l'élément de la préférence P_i est intégré dans le contexte d'analyse CA par enrichissement
- l'opérateur « $\otimes$ » : $P_i \otimes CA$ signifie que l'élément de P_i est intégré dans le contexte d'analyse CA par substitution

Comme pour la sélection des préférences, l'intégration est différente entre l'itération de redressement et celles d'accomplissement. Lors du redressement, l'intégration de l'élément pivot est effectuée suivant son type. Cependant, les éléments sont intégrés par enrichissement afin d'accomplir un contexte d'analyse.

Nous discutons dans la suite les différents cas d'intégration de l'élément pivot dans le contexte d'analyse courant *CAC*.

Intégration d'une mesure

L'intégration d'une nouvelle mesure agrégée est toujours effectuée par enrichissement. La conservation des mesures du contexte précédent n'a pas d'influence sur le calcul des valeurs de la nouvelle mesure. Ceci produit un contexte d'analyse intermédiaire CA^i qui est partiel.

Règle 1. Si la mesure sélectionnée est non agrégée, le système l'associe avec la fonction d'agrégation SUM.

Transformation de l'arbre de contexte A^{CAC}. L'intégration de la préférence $(f^{Agreg}(m_i); \theta_i; cp_i)$ dans le contexte d'analyse *CAC* se traduit par l'opération d'édition d'arbre suivante : $InsertN(A^{CAC}, f^{Agreg}(m_i), N^F)$, où N^F est le nœud représentant le fait courant *F*.

Exemple. La figure suivante montre la transformation du contexte résultat de Q_2 suite à l'insertion de la nouvelle mesure moyenne du nombre de soumissions. Il s'agit d'ajouter dans l'arbre de CAC_{Q2} deux nœuds représentant la mesure *Nb_Soumiss* et la fonction d'agrégation *AVG*. L'objectif étant de déterminer un contexte d'analyse non-évalué, les nœuds valeur de la nouvelle mesure ne sont pas ajoutés.

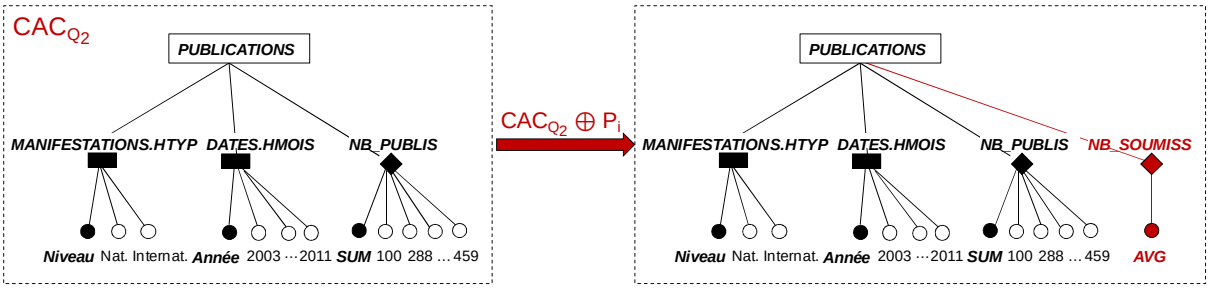


Figure 34. Exemple de transformation de contexte d'analyse par intégration de mesure

Intégration d'une dimension

L'ajout d'une nouvelle dimension affichée dépend de la composition du contexte de préférence. Supposons que l'usager préfère observer des données selon la dimension D_i dans le contexte d'analyse des données suivant les dimensions D_1, …, D_k. Ceci implique que l'usager est intéressé par des données qui sont agrégées suivant D_1, …, D_k et D_i. Tout contexte d'analyse agrégeant les données suivant d'autres dimensions n'est pas conforme à la préférence. Il convient donc, lors de la transformation de *CAC*, de ne pas maintenir les dimensions qui n'existent pas dans le contexte de préférence. Ainsi, l'intégration d'une dimension dans ce cas est par substitution.

Définition. Soit $P_i = (D_k; \theta_i; cp_i)$ une préférence candidate pour *CAC*, D^{cpi} et D^{CAC} les ensembles des dimensions dans cp_i et *CAC*. $CA^i = P_i \otimes CAC$ si et seulement si $D^{cpi}/D^{CAC12} \neq \emptyset$.

Dans les autres cas, une dimension est intégrée par enrichissement. Ainsi, si P_i est une préférence absolue, D_i est toujours intégrée par enrichissement à *CAC*.

L'intégration d'une dimension D_i est suivie d'une itération d'accomplissement qui insère un paramètre de D_i à partir des préférences de l'usager. Ce paramètre est sélectionnée à l'aide de la fonction `EPC-ByMatch`(*CAC*, $\mathcal{U}$, D_i,*'total'*).

Règle 2. En cas d'absence d'une préférence sur les paramètres de D_i, le système sélectionne par défaut le paramètre de plus haut niveau de granularité dans une perspective d'analyse du plus général au plus détaillé. Dans le cas où la dimension D_i est multi-hiérarchies, le système sélectionne le paramètre le plus général de chaque hiérarchie et génère pour chacun un contexte d'analyse intermédiaire différent.

Transformation de l'arbre de contexte A^{CAC}. L'intégration de la préférence $(D_i; \theta_i; cp_i)$ dans le contexte *CAC* se traduit par l'opération d'édition d'arbre suivante : InsertN(A^{CAC}, D_i, N^F), où N^F est le nœud représentant le fait courant *F*. Si l'intégration est par substitution, cette opération est précédée par les opérations suivantes : $\forall D_j \in D^{cpi}/D^{CAC}$, DeleteN($A^{CAC}$,$N^{Dj}$).

[12] E_1/E_2 est l'ensemble des éléments de E_1 qui n'appartiennent pas à E_2

Exemple. Considérons la requête Q_2. La figure suivante illustre l'intégration par enrichissement d'une préférence de la dimension *Domaines*. Conformément à la Règle 2, le paramètre le plus général *Domaine* est ajouté à CAC_{Q2}. Cette intégration engendre la suppression des valeurs de mesure puisqu'elles ne correspondent plus aux attributs des dimensions affichées.

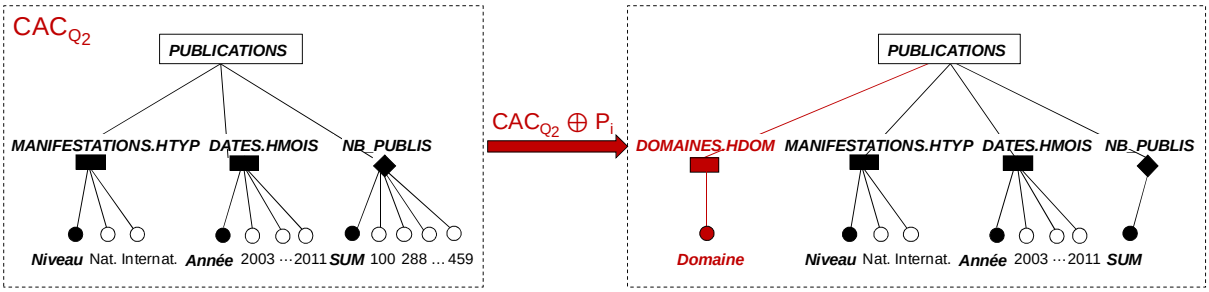


Figure 35. Exemple de transformation de contexte d'analyse par intégration de dimension

Intégration d'un attribut de dimension

L'intégration d'un attribut de dimension dans le contexte d'analyse courant vise à changer le niveau de détail des données.

Soit la préférence sur les paramètres $P_i = (a_k; \theta_i; cp_i)$ avec $a_k \in A^{Di}$; D_i est une dimension affichée dans *CAC* ; H_i est la hiérarchie courante de D_i ; a_1, …, a_x sont les paramètres affichés au sein de la dimension D_i, tel que $a_1 \prec_{Hi} \ldots \prec_{Hi} a_x$.

1er cas. Si a_k n'appartient pas à la hiérarchie H_i, il est intégré dans *CAC* par substitution à tous les attributs de la dimension D_i. La hiérarchie courante de D_i est désormais celle de a_k.

Nous distinguons deux cas d'insertion d'un attribut appartenant à la hiérarchie H_i. Afin d'illustrer ces cas, nous considérons un contexte d'analyse CA_1 affichant les données suivant l'axe *Auteurs* par *Ville* par *Equipe*. L'ensemble des paramètres affichés est : <*Equipe*, *Ville*>.

2ème cas. $a_k \prec_{Hi} a_1$. a_k est ajouté par enrichissement à *CAC*. L'ajout de l'attribut a_k permet d'analyser les données suivant D_i avec un niveau plus fin. L'ensemble des attributs affichés dans le nouveau contexte suivant la hiérarchie H_i est : <a_k, a_1, …, a_x>.

Exemple. L'insertion de l'attribut *Nom_A* dans CA_1 est effectuée par enrichissement. L'ensemble des attributs de la dimension *Auteurs* devient <*Nom_A*, *Equipe*, *Ville* >.

3ème cas. $a_1 \prec_{Hi} a_k$. a_k est intégré par substitution de manière à constituer le niveau de détail le plus fin dans D_i. Ainsi, les attributs qui sont de niveau de granularité plus détaillé que a_K sont éliminés. L'ensemble des attributs dans le nouveau contexte d'analyse est : <a_k, …, a_x>.

Exemple. Supposons que l'attribut *Institut* est intégré dans le contexte d'analyse CA_1. L'ensemble des attributs affichés selon l'axe *Auteurs* devient < *Institut*, *Ville* >.

Un cas particulier consiste à insérer un attribut de niveau de granularité plus haut que tous les attributs affichés. Dans ce cas, tous les attributs affichés sont éliminés. Par exemple, suite à l'intégration de l'attribut *Pays* au sein de CA_1, l'ensemble des attributs de la dimension *Auteurs* du contexte transformé se limite à < *Pays* >.

Remarque. Si la dimension à laquelle le paramètre est rattaché n'est pas affichée dans *CAC* ($a_k \in A^{Dk}$ et $D_k \notin D^{CAC}$), l'insertion du paramètre est précédée par l'insertion de sa dimension.

Transformation de l'arbre de contexte A^{CAC}. L'intégration d'un attribut a_k se traduit par l'insertion d'un nœud ou d'un fragment d'arbre si une dimension est insérée en conséquence:

- 1^{er} cas. $\forall a_i \in P^{Hi}$, DeleteN(A^{CAC},N^{ai}) = A^{CAC1}; UpdateE (A^{CAC1},$N^{Di.Hi}$, $D_i.H_k$) = A^{CAC2}, avec $a_k \in P^{Hk}$; InsertN(A^{CAC2}, a_k, $N^{Di.Hk}$) = A^{CAC3}
- $2^{ème}$ cas. InsertN(A^{CAC}, a_k, N^{a1}) = A^{CAC1}
- $3^{ème}$ cas. DeleteN(A^{CAC},a_m) = A^{CAC1} ; InsertN(A^{CAC1},a_k,N^{ap}) = A^{CAC2}, avec $a_m \prec_{Hi} a_k \prec_{Hi} a_p \prec_{Hi} a_x$

Intégration d'un prédicat de restriction.

En absence de conflit entre le prédicat de la préférence et les prédicats du contexte courant, le prédicat de préférence est intégré par enrichissement. Il est inséré en conjonction avec les prédicats existants de la dimension ou la mesure à laquelle il correspond. En cas de conflit, le prédicat de préférence est inséré par substitution aux prédicats existants qui sont à la source de conflit.

Transformation de l'arbre de contexte A^{CAC}. L'intégration d'un prédicat $pred_k$ sur les valeurs d'une mesure m_i ou d'une dimension D_i se traduit par la mise à jour du nœud composé *N* représentant la mesure ou la dimension.

- AddP(A^{CAC}, *N*, $pred_k$).

Exemple. Considérons la requête Q_2. Supposons que l'usager se focalise ensuite sur l'analyse des publications des chercheurs de l'IRIT. Supposons aussi qu'il existe deux préférences sur les valeurs de prédicats *Auteurs.Institut≠'IRIT'* et *Manifestations.Tx_accep < 0.3*. La première préférence étant en conflit avec le prédicat de la requête (*Auteurs.Institut='IRIT'),* la conjonction des deux préférences est intégrée dans CAC_{Q2} par substitution. Le contexte d'analyse intermédiaire concerne les publications de chercheurs qui ne sont pas de l'IRIT dans des manifestations avec un taux d'acceptation inférieur à 30%.

4 Algorithme de recommandation

4.1 ORecommend

Cette section présente l'algorithme de la fonction *ORecommend* qui génère, en fonction du profil de l'usager, un ensemble de contextes d'analyse candidats en se basant sur une requête incomplète ou sur le résultat d'une requête.

L'algorithme se déroule en trois étapes : redressement, accomplissement et affinement. A chaque étape, un ou plusieurs éléments sont sélectionnés à partir des préférences de l'usager puis intégrés dans le contexte en construction. Durant l'étape d'accomplissement, les règles d'accomplissement par défaut sont employées afin de sélectionner les éléments à insérer en l'absence de préférences correspondantes. A la fin de cette étape, les nœuds valeur sont supprimés de manière à ce que le contexte intermédiaire CA^i constitue un contexte d'analyse partiel non-évalué. Lors de l'étape d'affinement, les préférences sur les valeurs sélectionnées selon le mode de personnalisation combiné sont intégrées dans le contexte d'analyse avant d'être renvoyé.

Le fonctionnement de l'algorithme varie suivant le scénario de recommandation. Dans le cas du scénario d'assistance à la définition de requête, l'étape de redressement n'est pas évoquée. Dans l'algorithme suivant, nous supposons que le paramètre *Type* n'est pas renseigné ; *ORecommend* permet d'enrichir une requête incomplète. Notons que dans le cas où *Type* est renseigné, la fonction *ORecommend* permettrait de recommander un élément de requête correpondant à *Type*. Elle se réduirait à l'appel : EPC-ByMatch(*CAC, 𝒰, Type, 'total'*).

La différence du fonctionnement entre les scénarios d'anticipation et d'alternative réside essentiellement dans le choix de l'élément pivot et dans le calcul de proximité des contextes des préférences lors du tri des préférences.

Si toutes les transformations sont effectuées par enrichissement, le contexte d'analyse induit par la requête CAC_Q est inclus dans le contexte d'analyse généré CA^{Rec}_i. Si au moins une transformation est effectuée par substitution, alors CA^{Rec}_i et CAC_Q sont intersectés. CA^{Rec}_i et CAC_Q ne sont en aucun cas disjoints car ils partagent au moins le fait qui n'est pas recommandé.

Propriété. Soit CAC_Q le contexte d'analyse induit par la requête *Q*, et CA^{Rec}_i un contexte d'analyse candidat à la recommandation. La relation entre CA^{Rec}_i et CAC_Q est définie par : Englobe(CA^{Rec}_i, CAC_Q) ou Chevauche(CA^{Rec}_i, CAC_Q).

Algorithme ORecommend(*CAC* , Scénario, *U*, NULL)
Entrée :
CAC : contexte d'analyse courant (contexte partiel ou contexte résultat de la requête)
U : profil de l'usager
Scénario (1 : assistance à la définition de requête ; 2 : recommandation par anticipation ; 3 : recommandation d'alternative)
Λ : seuil de score de préférences
Sortie :
CA^{Rec} un ensemble de contextes d'analyse non-évalués candidats à la recommandation

DEBUT

$P^{PIVOT} \leftarrow \emptyset$, $P_v^{CAND} \leftarrow \emptyset$, $CA_{interm} \leftarrow \emptyset$
Si scénario = 3 Alors
Mode = « partiel »
Sinon
Mode = « total»
Finsi
Si scénario ≠ 1 Alors
//Etape de redressement
$P^{PIVOT} \leftarrow$ *EPC-ByMatch*(*CAC*,*U*, Null, *Mode*)
Pour chaque $P_i^{PIVOT} \in P^{PIVOT}$ Faire
$CA^i \leftarrow P_i^{PIVOT} \oplus CAC$ ou $CA^i \leftarrow P_i^{PIVOT} \otimes CAC$
$CA_{interm} \leftarrow CA_{interm} \cup CA^i$
FinPour
Sinon
$CA_{interm} \leftarrow \{CAC\}$
Finsi

```
Pour chaque CA^i ∈ CA_interm Faire
//Etape d'accomplissement
P^ACCOMP ← ∅
Tant que PAS(CA^i est partiel non-évalué) Faire
Type ← Déterminer le niveau de rupture de CA^i
P^ACCOMP ← EPC-ByMatch(CA^i, U, Type, 'total')
Si P^ACCOMP est vide
P^ACCOMP ← élément selon règle d'accomplissement par défaut
Finsi
CA^i ← P^ACCOMP ⊕ CA^i
Fin Tant que
//Etape d'affinement
Supprimer les nœuds valeur de CA^i
P_v^CAND ← EPCV (CA^i, U, λ, NULL)
CA^i ← CA^i ⊕ P_v^CAND
CA^Rec ← CA^Rec U CA^i
Fin Pour
Renvoyer CA^Rec

FIN ORecommend
```

Exemple. Considérons la requête Q_2 de l'exemple de référence de la section 3. Soit l'ensemble des préférences suivantes :

- P_{21} = (Auteurs; 0.8; (Publications; {Manifestations.HTyp/(All;Niveau)} ; ∅))
- P_{22} = (Manifestations.Editeur; 0.6; cp_{22})
- P_{23} = (« Niveau = international »; 0.9; cp_{23})
- P_{24} = (« Pays = France »; 0.6; cp_{24})

La figure suivante montre les étapes de génération des contextes candidats à la recommandation (scénario anticipation) à partir du contexte CAC_{Q2} résultat de Q_2 et des préférences. Pour des mesures de simplification, nous ne présentons pas les nœuds valeur. Les contextes d'analyse candidats générés par ce processus sont CA^{111} et CA^{121}.

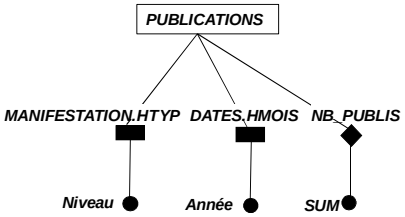
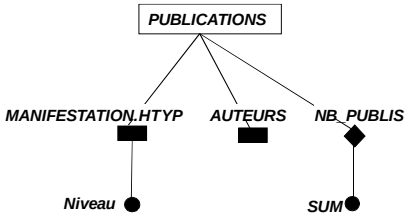
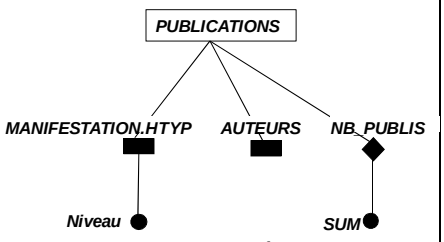
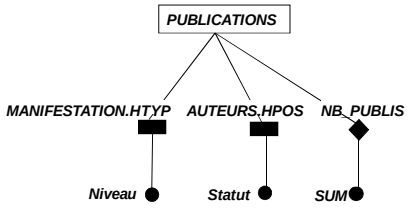
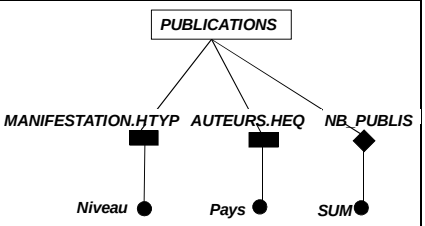
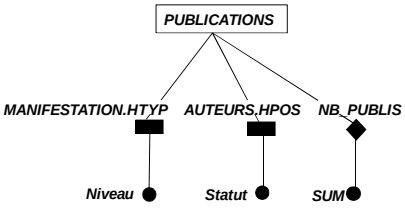
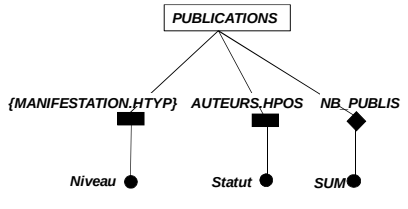
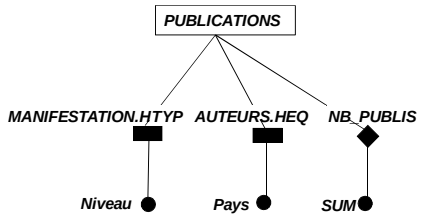
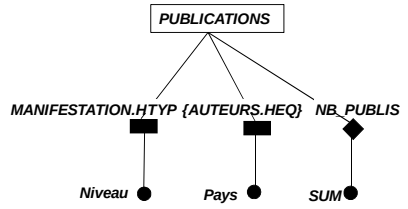
	Contexte d'analyse en entrée	Préférences candidates Englobe(CA^i, cp_i)	Contexte d'analyse produit
Redressement	 **CAC_{Q_2}**	P^{CAND} : - P_{21} - P_{22} Meilleure préférence suivant $\partial^{CAi}(cpi)$: P_{21}	 **CA^1** $CA^1 = CAC_{Q_2} \otimes P_{21}$
Accomplissement	 **CA^1**	Suivant Règle 2, P^{ACCOMP} : - « Statut » - « Pays »	 **CA^{11}** $CA^{11} = CA^1 \oplus$ « Statut »  **CA^{12}** $CA^{12} = CA^1 \oplus$ « Pays »
Affinement	 **CA^{11}**	P^{CAND} : - P_{23}	 **CA^{111}** $CA^{111} = CA^{11} \oplus P_{23}$
	 **CA^{12}**	P^{CAND} : - P_{24}	 **CA^{121}** $CA^{121} = CA^{12} \oplus P_{24}$

Figure 36. Exemple de génération de recommandations par anticipation

4.2 Tri et filtrage des recommandations candidates

Comme plusieurs recommandations candidates peuvent être générées à partir du contexte d'analyse courant, nous étudions comment filtrer et trier ces recommandations pour proposer les plus pertinentes pour l'usager.

Tri des recommandations candidates

Plusieurs approches peuvent être adoptées à cette fin. Nous pouvons citer :

- Trier les recommandations candidates en fonction de leur adéquation au profil de l'usager.
- Trier les recommandations candidates en fonction du taux des composants qui sont nouveaux par rapport aux requêtes précédentes de l'analyse en cours.
- Trier les recommandations candidates en fonction du nombre de leurs sélections durant les analyses précédentes.

Suivant le deuxième critère, une recommandation est favorisée si elle permet de découvrir de nouvelles données décisionnelles. Cependant, le troisième critère donne la priorité aux requêtes qui ont été déjà demandées par l'usager. Afin de favoriser les recommandations personnalisées aidant à découvrir des nouveautés, nous adoptons une méthode de tri des recommandations qui combine les deux premiers critères. Plus la recommandation est riche en préférences et intègre moins d'éléments de la requête précédente, plus elle est pertinente. Dans cette perspective, un score est attribué à chaque contexte d'analyse à recommander. Plus le contexte d'analyse est construit à partir des éléments les plus pertinents, plus son score est important. Ainsi, le degré d'intérêt global d'un contexte d'analyse est une fonction des degrés d'intérêt de ses composants.

Rappelons qu'une recommandation est composée d'un ensemble d'éléments qui dérivent de la requête de l'usager, noté *NQ*, d'un ensemble d'éléments qui dérivent des préférences de l'usager, noté *NP*, et d'un ensemble d'éléments qui sont choisis d'une manière arbitraire, noté *NA*. Le calcul du score d'une recommandation $CA^{Rec}{}_i$ tient compte des hypothèses suivantes :

(1) Chaque élément structurel (mesure, dimension, paramètre) dérivé d'une préférence P_k contribue au score global avec le score relaxé $doi(P_k)^{CAReci}$ de P_k par rapport au contexte d'analyse candidat. Les prédicats de restriction de $CA^{Rec}{}_i$ participent avec un score unifié qui est la moyenne de leurs scores relaxés par rapport à $CA^{Rec}{}_i$.

(2) Les éléments qui ont été précisés dans la requête contribuent au score total avec une constante β entre 0 et 1. Une valeur très basse de β maximise le score des recommandations qui ne comportent pas des éléments de la requête. Ceci minimise le nombre d'éléments communs entre la requête et les meilleures recommandations. D'autre part, une valeur assez haute de β favorise les éléments de la requête par rapport aux nouveaux éléments induits par les préférences. L'ajustement de cette valeur est présenté dans les expérimentations décrites dans le chapitre 6 (*cf.* Figure 52).

(3) Les éléments qui sont rajoutés arbitrairement sont pondérés avec des scores nuls.

(4) Le fait ne contribue pas au score global puisqu'il ne représente pas un élément recommandé.

Plus précisément, le score d'une recommandation CA^{Rec}_{i} est le score de son arbre de contexte:

- Le score d'un nœud (simple ou composé) est le score de son attribut.
- Le score des prédicats des nœuds est la moyenne des scores des préférences de type prédicat.

La fonction de score F_{CA}^{RANK} associe un score entre 0 et 1 à un arbre de contexte A^{CA}:

F_{CA}^{RANK}: $A^{CA} \longrightarrow [0,1]$

$$F_{CA}^{RANK}(A^{CA}) = \frac{\sum_{i=1}^{n} f(N_i) + AVG^{Pred}}{n} \; ; f(N_i) = \begin{cases} \beta \; si \; N_i.\text{Attribut} \in NQ, \beta \in [0,1] \\ doi(P_i)^{CA} \; si \; N_i.\text{Attribut} \in NP \\ 0 \; si \; N_i.\text{Attribut} \in NA \end{cases} \quad \textbf{(5)}$$

- n est la cardinalité de l'arbre de contexte qui ne considère pas le nœud fait : n = $|A^{CA}|$-1.
- $doi(P_i)^{CA}$ est le score relaxé de la préférence P_i par rapport au contexte *CA*. Rappelons que le calcul de ce sore varie en fonction du scénario de recommandation. (*cf.* section 3.1.1).
- AVG^{Pred} est la moyenne des scores de tous les prédicats rattachés aux nœuds de l'arbre A^{CA}.

Ainsi les contextes d'analyse candidats à la recommandation CA^{Rec}_{i} sont triés selon $F_{CA}^{RANK}(A^{CAReci})$.

Remarque. Nos recommandations sont triées à partir d'une matrice d'utilité Utilisateurs×Requêtes, en mesurant l'utilité d'une requête pour un usager donnée, et en retournant la (les) requête(s) qui maximise(nt) cette utilité.

Filtrage de recommandations

Afin d'éviter la génération abusive de recommandations d'une part, et de maintenir un seuil de pertinence raisonnable des recommandations, d'autre part, un mécanisme de filtrage est appliqué au cours ou après la génération des recommandations. Plusieurs critères de filtrage peuvent être utilisés :

- Sélectionner les recommandations qui ont été activées au moins K fois dans le passé.
- Sélectionner les recommandations étant donné un critère de satisfaction.
- Sélectionner au maximum *N* recommandations.

Nous avons choisi d'adopter les deux derniers critères de filtrage. Parmi les recommandations qui ont un score supérieur à un seuil *S*, seules les *N* meilleures recommandations sont renvoyées. Le seuil *S* et le nombre *N* sont fixés arbitrairement au début, puis évoluent en fonction de l'interaction de l'usager avec le système de recommandation, en fonction du seuil moyen et du nombre moyen des recommandations activées par l'usager.

5 Bilan

Afin de faciliter les analyses OLAP, nous avons proposé un mécanisme de recommandation de requête. Conformément à notre cadre de personnalisation OLAP (*cf.* chapitre 3, section 3.2), le mécanisme de recommandation proposé correspond à l'*action* de personnalisation de

la navigation. Il s'agit d'une personnalisation de *type* explicite et dynamique, dont les *critères de la personnalisation* sont les préférences de l'usager et les paramètres de filtrage et de tri.

Notre mécanisme de recommandation intègre trois scénarios (Jerbi et al., 2009b) : assistance à la définition de requête (Jerbi et al., 2008, 2009a), recommandation de requêtes anticipées (Jerbi et al., 2009c) et recommandation de requêtes alternatives (Jerbi et al., 2010a, 2011).

Afin d'aboutir à une approche adaptée aux différents langages et structures d'affichage, notre mécanisme de recommandation se base sur une représentation interne de la requête et de son résultat sous forme d'arbre de contexte. Le processus de recommandation est traduit par un mécanisme de transformation d'arbres de contexte.

Le processus de génération de recommandation se décompose en trois étapes :

- Redressement du contexte d'analyse induit par la requête en simulant des manipulations de l'usager.
- Accomplissement du contexte produit. C'est un processus itératif permettant d'insérer des éléments de contexte jusqu'à la génération d'un contexte d'analyse non-évalué.
- Affinement du contexte d'anlyse accompli par l'insertion des conditions de restriction.

A chaque étape, les éléments à intégrer sont déterminés à partir du profil de l'usager suite à un appariement de contexte, suivi de la mesure de l'utilité des différents éléments pour la requête à recommander.

Les recommandations générées sont triées en fonction du profil de l'usager afin de renvoyer les meilleures (Jerbi et al., 2009c).

Notre processus de recommandation est ascendant. Il permet de recommander de nouvelles requêtes à l'usager.

Toutefois, nous identifions les limites suivantes du processus de recommandation :

- Le mécanisme d'appariement tient compte seulement de la dernière requête appliquée par l'usager. Les autres requêtes précédemment posées dans l'analyse ne sont pas considérées.
- Le calcul des degrés de couverture et de rapprochement des contextes d'analyse basé sur le nombre d'éléments donne la priorité aux préférences associées aux contextes les plus détaillées, sans distinguer entre les types des composants des contextes d'analyse. La définition d'une politique de priorité sur les composants des contextes et l'introduction des aspects sémantiques dans l'appariement de contextes permettrait d'améliorer le processus de recommandation.

Le chapitre suivant montre comment implanter les mécanismes de personnalisation et de recommandation de requêtes dans un système unifié et présente les résultats d'expérimentation des algorithmes proposés.

Références

Dittrich, J-P, Kossmann, D., Kreutz, A. (2005). Bridging the gap between OLAP and SQL. Intl. Conf. on Very Large Data Bases (VLDB), pages 1031–1042.

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2008). Management of Context-aware Preferences in Multidimensional Databases. Intl. Conf. on Digital Information Management (ICDIM), IEEE, pages 669–675.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2009a). Modèle de Préférences Contextuelles pour les Analyses OLAP. Journées Francophones Extraction et Gestion de Connaissances (EGC), pages 253–258.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2009b). Applying Recommendation Technology in OLAP Systems. Intl. Conf. on Enterprise Information Systems (ICEIS), Springer, LNBIP 24, pages 220–233.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G (2009c). Preference-Based Recommendations for OLAP Analysis. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer-Verlag, pages 467–478.*

***Jerbi, H.**, Pujolle, G., Ravat, F., Teste, O. (2010a). Personnalisation de systèmes OLAP annotés. Congrès Informatique des Organisations et Systèmes d'Information et de Décision (INFORSID), pages 327–344.*

***Jerbi, H.**, Pujolle, G., Ravat, F., Teste, O. (2011). Recommandation de requêtes dans les bases de données multidimensionnelles annotées. Revue des Sciences et Technologies de l'Information, série Ingénierie des Systèmes d'Information, Vol. 16, No. 1/2011, pages 113–138.*

Navarro, G. (2001). A guided tour to approximate string matching. ACM Comput. Surv., Vol. 33, No. 1, pages 31–88.

Ravat F., Teste O., Tournier R., Zurfluh G. (2007b). Querying Multidimensional Databases. Conf. on Advances in Databases and Information Systems (ADBIS), Springer-Verlag, pages 298–313.

Ravat F., Teste O., Tournier R., Zurfluh G. (2008). Algebraic and graphic languages for OLAP manipulations. Intl. Journal of Data Warehousing and Mining (IJDWM), Vol. 4, No. 1, pages 17–46.

Thalhammer, T., Schrefl, M., Mohania, M. (2001). Active DataWarehouses. Complement-ing OLAP with Analysis Rules, Data and Knowledge Engineering (DKE), Elsevier Science Publishers, Vol. 39, No. 3, pages 241–269.

Chapitre 6

Implantation et expérimentation

Sommaire

1 Introduction

Ce chapitre présente le prototype OLAPers afin de valider nos propositions. L'objectif est de fournir un système d'analyses OLAP personnalisées qui fournit les mécanismes de personnalisation présentés dans les chapitres 4 et 5 et qui repose sur le modèle de préférences contextuelles décrit dans le chapitre 3.

Plan du chapitre. Ce chapitre s'articule comme suit ; la section suivante présente une description générale du prototype développé ; la section 3 décrit comment les modèles de constellation et de préférences contextuelles sont implantés ; puis la section 4 montre comment interroger et afficher les données d'une manière personnalisée. La section 5 détaille l'implantation des différents algorithmes de personnalisation. La section 6 présente une étude expérimentale de nos solutions.

2 Description générale du système

Fonctionnement global du système de personnalisation

L'objectif du système implanté est d'intégrer la personnalisation et la recommandation de requêtes au cours des manipulations de l'usager. Pour chaque requête, le système personnalise la requête et renvoie le résultat en plus de requêtes recommandées. La Figure 37 présente une vision globale de l'évolution du processus de construction d'une requête en fonction des interactions de l'usager. Le système de personnalisation intervient à différents niveaux et permet de:

- assister l'usager à construire la requête conformément au premier scénario de recommandation.
- personnaliser la requête de l'usager une fois elle est complète et validée par l'usager, puis l'exécuter afin de renvoyer son résultat.
- exploiter le résultat de la requête afin de calculer des requêtes recommandées conformément au deuxième et au troisième scénario de recommandation. Les requêtes recommandées sont personnalisées avant d'être retournées.

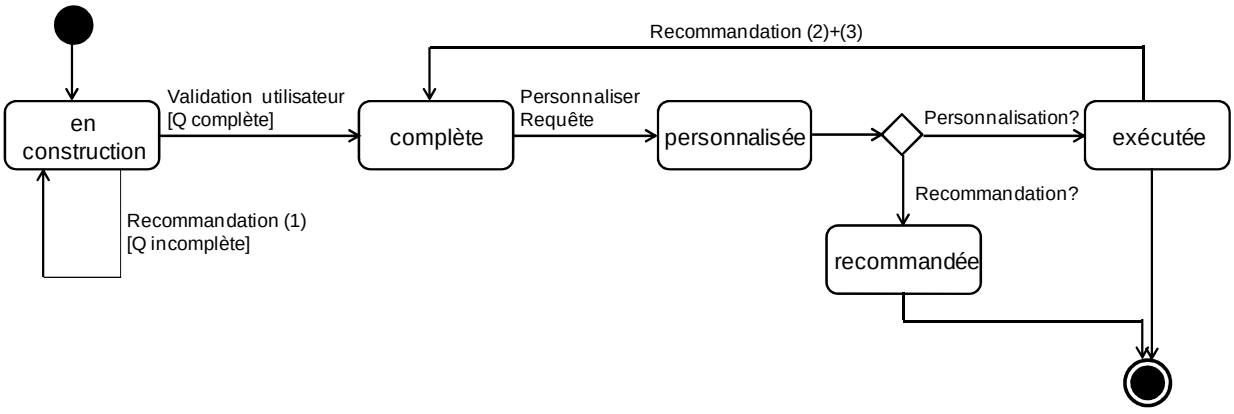


Figure 37. Evolution de la construction d'une requête dans OLAPers

Configuration

Un module de configuration de l'interrogation de la BDM permet aux usagers et à l'administrateur du système de définir des paramètres de la personnalisation des analyses :

- nature du service de personnalisation attendu. Ce paramètre permet de spécifier les actions de personnalisation assurées par le système : personnalisation de requête (filtrage du résultat suivant toutes les dimensions ou selon les dimensions affichées seulement), assistance à la formulation de requête, recommandation de requêtes anticipées et/ou alternatives.
- score minimal λ des préférences sélectionnées lors d'un processus de personnalisation (par `EPCV-K`) de requête ou de recommandation (par `EPC-ByMatch`).
- nombre maximal K de préférences sélectionnées à chaque itération du processus de personnalisation. Dans le cas de recommandation, K correspond au nombre de préférences admettant le score relaxé maximal qui sont retenues.
- paramètres de recommandation d'alternatives. L'administrateur peut spécifier le(s) type(s) requis et le nombre minimal nb_{min} de nœuds communs entre les contextes intersectés. Ainsi, la fonction d'appariement partiel implantée est :

  Chevauche (C_1, C_2, {t_1, t_2, …}, nb_{min}), où t_1, t_2, … sont les types de nœuds structurels communs entre C_1 et C_2 (fait, dimension, paramètre, mesure, prédicat).

Architecture

Le prototype OLAPers repose principalement sur une BDM implantée au dessus du système de gestion de bases de données relationnel Oracle 11g. La BDM est interrogée par une application cliente java qui permet la spécification de manipulations OLAP et la restitution auprès de l'usager. La Figure 38 illustre l'architecture de notre système.

- Le niveau « stockage » représente l'implantation du schéma et des valeurs de la BDM. Ces données sont étendues par les profils des utilisateurs qui sont composées de préférences contextuelles.
- Le niveau « interaction utilisateur » offre un ensemble d'interfaces permettant à l'utilisateur d'interagir avec la BDM. Un éditeur de commandes permet de saisir des requêtes selon le langage OLAP/SQL personnalisé et d'obtenir des réponses par rapport au statut de l'exécution de la requête. Le résultat d'une requête est affiché dans une table multidimensionnelle personnalisée. L'usager peut consulter le schéma multidimensionnel de la BDM via un visualiseur interactif.
- Le niveau « exécution de requête » regroupe quatre modules. Chaque requête OLAP/SQL textuelle ou graphique est analysée afin d'être validée. Le moteur de personnalisation calcule la version personnalisée d'une requête valide qui est récupérée par le moteur OLAP/SQL qui est chargé des transcriptions vers le SGBD relationnel de stockage. Les requêtes OLAP/SQL sont évaluées et leurs résultats sont renvoyés dans une première étape au moteur de personnalisation afin d'en déduire des recommandations, puis au générateur graphique chargé de l'affichage sous forme de table multidimensionnelle.

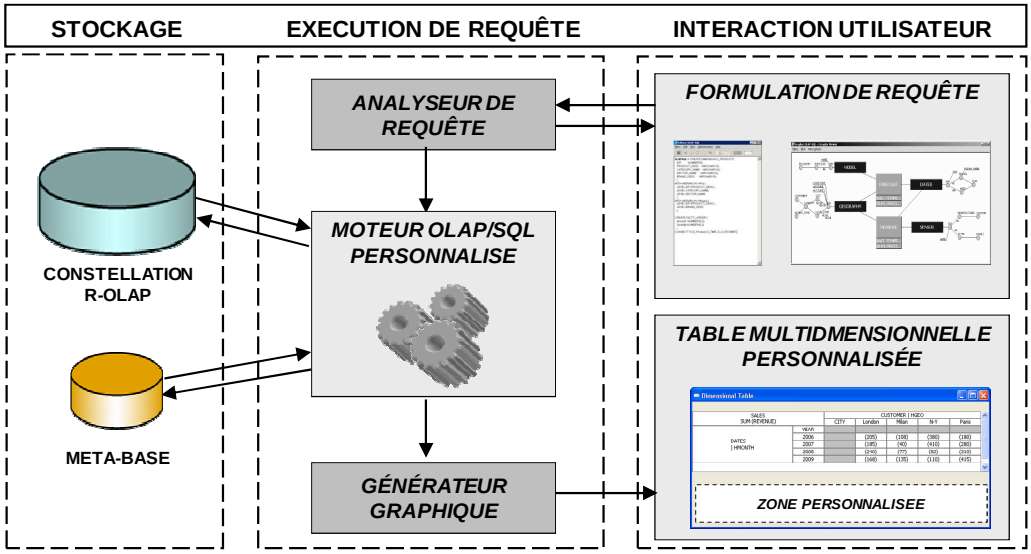


Figure 38. Architecture de OLAPers

3 Stockage de constellation personnalisée

La BDM des publications décrite dans le chapitre 3 (*cf.* Figure 10) est stockée sous le SGBD Oracle 11g et repose sur une architecture R-OLAP. Elle est répartie sur deux bases de données : une méta-base et une base R-OLAP.

3.1 Méta-base

La méta-base permet de décrire les structures multidimensionnelles de la constellation ainsi que les préférences contextuelles des utilisateurs.

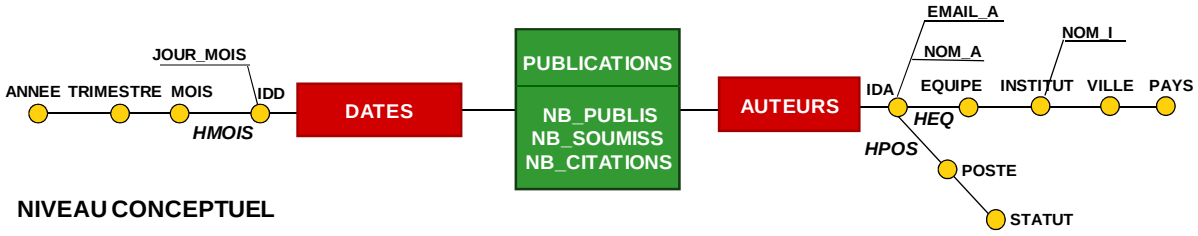
3.1.1 Stockage des structures multidimensionnelles

Un schéma de l'extrait de la méta-base représentant les structures de la constellation est représenté en Annexe 2. Concrètement, cette partie est composée d'une classe générale à partir de laquelle dérivent des classes correspondant aux différents composants de la constellation (fait, dimension, mesure, paramètre, attribut faible et hiérarchie) et un ensemble d'associations traduisant les différents liens existant entre ces composants ($Star^C$,M^{Fi},H^{Di},P^{Hj},$\prec_{Hj}$,$Weak^{Hj}$et H^{Di}).

Ce schéma conceptuel est implanté sous la forme de relations. Dans l'implantation relationnelle décrite ci-dessous, la relation *meta_element* traduit tous les composants d'une constellation qui sont distingués à l'aide du champ *type* qui décrit s'il s'agit d'un fait (F), d'une dimension (D), …. ; meta_star représente les liaisons entre les faits et les dimensions (la fonction $Star^C$) ; meta_measure permet de déterminer l'ensemble de mesures correspondantes à chaque fait (M^{Fi}) ; et *meta_hierarchy* permet d'établir la correspondance entre les hiérarchies et les dimensions (H^{Di}). La relation *meta_level* permet de décrire les paramètres des dimensions (P^{Hj}), les attributs faibles et leur association aux paramètres ($Weak^{Hj}$), ainsi que la hiérarchisation des paramètres et des attributs faible au sein des hiérarchies ($\prec_{Hj}$).

meta_element (id, nom, type)
meta_star (idf#, idd#)
meta_measure (idm#, idf#)
meta_hierarchy (idh#, idd#)
meta_level (idh#, ida#, position, parameter?)

Exemple. La Figure 39 présente un exemple du contenu de la méta-base correspondant au fait *publications* et aux dimensions *dates* et *auteurs*. Il s'agit particulièrement du contenu de la relation *meta_élément* (l'ensemble des structures multidimensionnelles) et du contenu des relations décrivant les liens entre ces structures (*meta_star*, *meta_measure*, *meta_hierarchy* et *meta_level*).



META-BASE

META_STAR

IDF#	IDD#
1	6
1	13
...	...

META_MEASURE

IDM#	IDF#
3	1
4	1
5	1
...	...

META_HIERARCHY

IDH#	IDD#
7	6
14	13
22	13
...	...

META_ELEMENT

ID	Nom	Type
1	PUBLICATIONS	F
2	MISSIONS	F
3	NB_PUBLIS	M
4	NB_SOUMISS	M
5	NB_CITATIONS	M
6	DATES	D
7	HMOIS	H
8	IDD	A
9	MOIS	A
10	TRIMESTRE	A
11	ANNEE	A
12	JOURS_MOIS	A
13	AUTEURS	D
14	HEQ	H
15	IDA	A
16	EQUIPE	A
17	INSTITUT	A
18	VILLE	A
19	PAYS	A
20	EMAIL_A	A
21	NOM_A	A
22	HPOS	H
23	POSTE	A
24	STATUT	A
25	...	...

META_LEVEL

IDH#	IDA#	Position	Parameter?
7	8	1	1
7	9	2	1
7	10	3	1
7	11	4	1
7	12	1	0
14	15	1	1
14	16	2	1
14	17	3	1
14	18	4	1
14	19	5	1
14	20	1	0
14	21	1	0
22	15	1	1
22	23	2	1
22	24	3	1
...	...	...	...

Figure 39. Extrait de la méta-base décrivant les structures de la constellation

3.1.2 Stockage des préférences et des contextes

Afin de décrire les préférences contextuelles, deux relations sont ajoutées à la méta-base.

La relation *meta_preference* décrit l'élément de la constellation (un *meta_element*) ou le prédicat préféré, le score de la préférence ainsi que son contexte associé (NULL dans le cas d'une préférence absolue)

La relation *meta_context* décrit l'arborescence des composants du contexte. Chaque n-uplet représente un nœud de l'arbre de contexte : *eid* traduit l'élément de la constellation (un *meta_element*) avec un prédicat pour les nœuds composés ; et un lien vers le nœud père.

Exemple. La préférence P_9 est définie par : P_9= (Auteurs.Ville='Toulouse'; 0.9; cp_9) ; cp_9 = (Publications; {Auteurs.Heq/(All;Équipe)} ; {Dates.Année=2011})

Les tables suivantes décrivent le stockage de cette préférence. Les champs *eid* dans les tables meta_preference et meta_contexte font référence au champ *id* de la table meta_element.

META_ELEMENT

ID	Nom	Type
1	PUBLICATIONS	F
...	...	...
6	DATES	D
...	...	...
14	HEQ	H
...	...	...
16	EQUIPE	A
...	...	...
18	VILLE	A
...	...	...

META_PREFERENCE

PID	EID	OP	VAL	SCORE	CID
P9	18	=	Toulouse	0.9	C9

META_CONTEXT

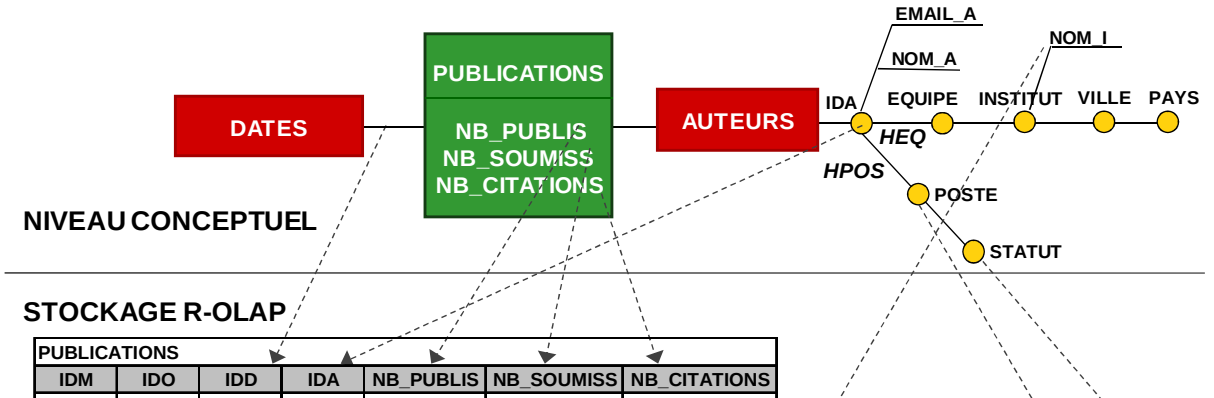
CID	EID	PRED	PERE_ID
C9	1		NULL
C91	14		C9
C911	16		C91
C92	6	ANNEE=2011	C9

Figure 40. Exemple de stockage d'une préférence contextuelle

3.2 Stockage des instances de la constellation

La base R-OLAP contient les instances de la constellation qui sont stockées en tables de fait et de dimension.

La Figure 41 illustre le stockage du fait *publications* et de la dimension *auteurs*. La table est composée de champs correspondant aux mesures ainsi que les paramètres racines des dimensions qui lui sont associées. Chaque attribut de dimension est traduit par un champ au niveau de la table de dimension *auteurs*. La clé primaire de cette table est le paramètre identifiant de la dimension.



PUBLICATIONS

IDM	IDO	IDD	IDA	NB_PUBLIS	NB_SOUMISS	NB_CITATIONS
m1	do1	d1	a1	1	0	2
m2	do2	d2	a2	0	1	3
...						

AUTEURS

IDA	EQUIPE	NOM_A	EMAIL_A	INSTITUT	NOM_I	VILLE	PAYS	POSTE	STATUT
a1	SIG	Jerbi	jerbi@irit.fr	UT1	Université Toulouse 1	Toulouse	France	Doctorant	NP
a2	SIG	Zurfluh	zurfluh@irit.fr	UT1	Université Toulouse 1	Toulouse	France	Professeur	P
...	...	...	...	...	...	...	...	...	...

Figure 41. Extrait de la BD R-OLAP

4 Requêtage et restitution des données

Cette section présente le niveau « interaction utilisateur » d'OLAPers.

4.1 Langage de définition des préférences

Les préférences de l'usager sont acquises explicitement par le système au travers d'un langage textuel. La préférence P = (E; θ; cp) est définie par l'ordre suivant :

```
StarPreference E
With degree = θ
When cp;
```

4.2 Table multidimensionnelle personnalisée

Le prototype OLAPers affiche les données sous forme de table multidimensionnelle personnalisée.

Une table multidimensionnelle T est définie par (S, L, C, R) où

- S = (F, {$f_1(m_1)$, $f_2(m_2)$,…}) représente le sujet d'analyse relatif au fait F et ses mesures observées {m_1, m_2,…} agrégées à l'aide de fonctions d'agrégationf_1, f_2,…
- $L = (D^L, H^L, P^L)$ représente l'axe d'analyse en ligne de T au travers d'une dimension courante D^L, d'une hiérarchie courante H^L et d'une liste ordonnée de paramètres affichés $P^L = <All, p^{HL}_{max},…p^{HL}_{min}>$, $H^L \in H^{DL}$ et $D^L \in Star^{CS}(F)$
- $C = (D^C, H^C, P^C)$ représente l'axe d'analyse en colonne de la table T au travers d'une dimension courante D^C, d'une hiérarchie courante H^C et d'une liste ordonnée de paramètres affichés $P^C = <All, p^{HC}_{max},…p^{HC}_{min}>$, $H^C \in H^{DC}$ et $D^C \in Star^{CS}(F)$
- $R = pred_1 \wedge … \wedge pred_v$ est le prédicat de restriction composé d'une conjonction de prédicats normalisés portant sur les dimensions et/ou sur les mesures {m_1, m_2,…}.

Cette table multidimensionnelle est dotée d'une *zone de notification* permettant d'afficher un message en fonction du service de personnalisation courant.

- Dans le cas de personnalisation de requête, une notification de l'extension de la requête initiale est affichée.
- Dans le cas du premier et du deuxième scénario de recommandation, les requêtes recommandées sont affichées dans cette zone.
- Si l'usager a soumis une requête incomplète, un message sera affiché dans cette zone afin d'expliquer les choix d'enrichissement de la requête.

Exemple. Reprenons l'exemple de génération de recommandations de la section 4.1 du chapitre 5 (*cf.* Figure 36). La figure suivante montre le résultat de l'exécution de la requête « *nombre de publications par année et par niveau de manifestation* » sous OLAPers. Le résultat de la requête est affiché sous forme d'une TM. Les requêtes recommandées par anticipation sont affichées dans la zone personnalisée sous forme de liens. Le système exécute une requête recommandée suite à l'activation du lien correspondant.

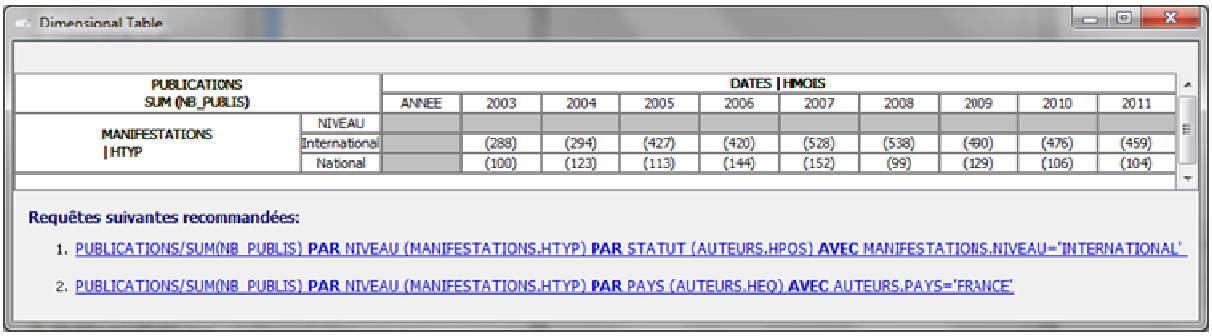


Figure 42. Exemple de table multidimensionnelle personnalisée (avec recommandations)

4.3 Assistance à la définition de requêtes

Afin de faciliter la tâche de formulation de requêtes (1er scénario de recommandation), nous avons défini un langage de manipulations OLAP personnalisées que l'usager saisit textuellement au travers de l'éditeur de commande.

Langage de manipulations OLAP personnalisées

Nous avons étendu le langage OLAP (Ravat et al., 2008) décrit dans Annexe 1 afin de faciliter l'expression des opérations de construction de table multidimensionnelle (DISPLAY), de rotation (ROTATE), de forages (DRILL-DOWN et ROLL-UP) et de restriction des données (SELECT). La saisie de certains paramètres de ces opérations devient facultative (voir « paramètres facultatifs » dans Tableau 5). Le système de recommandation les complète à partir du profil de l'usager. Tableau 5 récapitule les propriétés et les paramètres des opérations proposées.

Ainsi, la manipulation des données au sein d'OLAPers consiste à construire une première table multidimensionnelle à l'aide du constructeur DISPLAY, puis à appliquer un ensemble d'opérations sur la table multidimensionnelle pour transformer les données visualisées provenant de la constellation. Chaque opération porte en entrée sur une TM source notée T_{SRC} = (S_{SRC} ; L_{SRC}; C_{SRC} ; R_{SRC}), et produit en sortie une TM résultat notée T_{RES} = (S_{RES} ; L_{RES}; C_{RES} ; R_{RES}).

Opérateur	Entrée		Sortie
	Paramètres obligatoires	*Paramètres facultatifs*	
DISPLAY	F, D^L, H^L	$\{f_1(m_1),...\}$,D^C, H^C	T_{RES}=(S_{RES}, L_{RES}, C_{RES}, R_{RES}) – S_{RES} = (F, M), M={*f*1(m1),…, *f*x(mx)}, ∀i∈[1..x], mi∈M, – L_{RES} = (D^L, H^L, <All, p^{HL}_{max}>), – C_{RES} =(DC, H^C, <All, p^{HC}_{max}>), – R_{RES} = $\bigwedge_{\forall i, D_i \in Star^C(F^S)} D_i.ALL = 'all'$.
DRILLDOWN	T_{SRC}, D – D∈{D^C_{SRC}, D^L_{SRC}} – p_{inf}∈$Param^{Hi}$, H_i∈H^D	p_{inf}	T_{RES}=(S_{SRC}, L_{RES}, C_{RES}, R_{SRC}) Si D=D^L_{SRC} Alors – L_{RES}= (D^L_{SRC}, H^L_{SRC}, P^L_{SRC}+<p_{inf}>), – C_{RES} = C_{SRC}, Si D=D^L_{SRC} Alors – L_{RES} = L_{SRC}, – C_{RES}= (D^C_{SRC}, H^C_{SRC}, P^C_{SRC}+<p_{inf}>).
ROLLUP	T_{SRC}, D – D∈{D^C_{SRC}, D^L_{SRC}} – P_{sup}∈$Param^{Hi}$, H_i∈H^D	p_{sup}	T_{RES}=(S_{SRC}, L_{RES}, C_{RES}, R_{SRC}) Si D=D^L_{SRC} Alors – L_{RES}= (D^L_{SRC}, H^L_{SRC}, <All, p^{HL}_{max}>,…, p_{sup}>), – C_{RES} = C_{SRC}, Si D=DL_{SRC} Alors – L_{RES} = L_{SRC}, – C_{RES}= (D^C_{SRC}, H^C_{SRC}, <All, p^{HC}_{max}>,…, p_{sup}>).
ROTATE	T_{SRC}, D_{old} – D_{old}∈{D^C_{SRC}, D^L_{SRC}} – D_{new}∈Star(F_{SRC}) – H^D_{new}∈H^{Dnew}	D_{new}, H^D_{new}	T_{RES}=(S_{SRC}, L_{RES}, C_{RES}, R_{SRC}) Si D_{old}=D^L_{SRC} Alors – L_{RES} = (D_{new}, H^D_{new}, <All, p^{HDnew}_{max}>), – C_{RES} = C_{SRC}, Si D_{old}=D^C_{SRC} Alors – L_{RES} = L_{SRC}, – C_{RES} =(D_{new}, H^D_{new}, <All, p^{HDnew}_{max}>).
SELECT	T_{SRC}, *A* – *A*∈ M^F∪A^D, D∈$Star^{CS}$(F)	op *val* – op∈{= ;< ;≤ ; > ; ≥ ;≠} – *val* ∈dom(*A*)	T_{RES}=(S_{SRC}, L_{SRC}, C_{SRC}, R_{RES}) – R_{RES} = R_{SRC} ∧ *A* op *val*

Tableau 5. Opérations de manipulation OLAP personnalisées

5 Moteur de requête personnalisé

Nous avons adopté les choix suivants afin d'implanter notre cadre de personnalisation générique :

- Comme la structure de visualisation utilisée est une table à deux dimensions, les requêtes définies permettent d'afficher les données suivant deux dimensions. Ainsi, les contextes courants induits par les requêtes et les contextes d'analyse recommandés admettent deux dimensions d'affichage. Cependant, il n'y a pas de contrainte sur le nombre de dimensions pour les contextes de préférences.
- Deux recommandations par anticipation et une recommandation d'alternative sont générées pour chaque requête utilisateur.

Les contextes d'analyse sont représentés en mémoire sous forme de structure interne qui traduit l'arbre de contexte.

Le moteur OLAP/SQL personnalisé se compose de quatre modules.

Générateur de requête. Les commandes OLAP sont transcrites en mémoire sous forme de structure interne. Comme les requêtes sont appliquées à la BDM sous forme de requêtes SQL du type SELECT-GROUP-BY-HAVING, le générateur de requête permet également de transcrire la structure interne générée vers le SGBD relationnel.

Gestionnaire de contexte. Ce module intervient durant les phases de sélection et d'intégration de préférence. Dans la première phase, il permet d'effectuer l'appariement entre contextes et de calculer la distance d'édition entre deux contextes d'analyse dans le cas d'appariement partiel. Dans un processus de personnalisation de requête, il prend en entrée une liste de prédicats des K préférences candidates et produit en sortie une structure interne représentant la requête enrichie. Dans un processus de recommandation, il génère à chaque itération un contexte d'analyse transformé suite à l'insertion des préférences. Il assure également le tri des contextes d'analyse candidats à la recommandation et la sélection des meilleures.

Gestionnaire de préférences. Ce module permet de déterminer les préférences actives et renvoie leurs contextes au gestionnaire de contexte. Il permet de trier les préférences et de les regrouper en classes de K préférences.

Détecteur de conflits. Le prototype implanté traite les conflits qui surviennent au niveau syntaxique au moment de la définition de préférences et de la sélection de préférences. Les conflits sont résolus suivant les politiques 1, 2 et 3 du chapitre 4 (*cf.* chapitre 4, section 2.3).

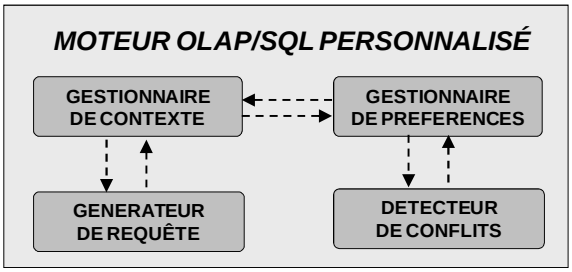


Figure 43. Moteur de requête personnalisé

5.1 Implantation de l'algorithme de sélection de préférences

L'algorithme de sélection de préférences consiste à exécuter l'algorithme `EPCV-K` ou `EPC-ByMatch` en fonction du service de personnalisation demandé. Il prend en entrée la structure interne correspondant à la requête ou au contexte d'analyse courant *CAC*, le mode de personnalisation (filtrage ou global) ou de recommandation (total ou partiel) et le seuil de préférences actives à considérer ainsi que le nombre maximal K. Dans le cas de l'itération de redressement du processus de recommandation, l'élément cible constitue également un paramètre de cet algorithme.

L'algorithme se déroule en deux phases :

- une phase d'accès à la base de données afin d'extraire les préférences actives, puis les contextes de préférences,

- une phase de traitement en mémoire ; il s'agit des étapes d'analyse de requête, de gestion de conflits, d'appariement de contextes, de relaxation de score et de tri de préférences et de recommandations.

L'algorithme de sélection procède de manières différentes lors de l'appariement total et partiel avec *CAC* en fonction des propriétés de chacun.

Dans le cas d'appariement total (voir algorithme de vérification de dominance de contextes dans chapitre 3, section 2.2.3), le système effectue un premier accès à la méta-base afin d'extraire les préférences actives dont tous les nœuds de contexte appartiennent à l'arbre du contexte courant *CAC*. Ainsi, plusieurs préférences actives sont rejetées depuis le premier accès à la méta-base. Ces préférences sont triées suivant leurs scores relaxés « fictifs » sous *CAC*. Ces sores ne sont pas tous réels car leurs contextes n'ont pas été totalement appariés avec CAC. A ce niveau, il n'est pas possible de déterminer si tout l'arbre d'un contexte de préférence est inclus dans celui de *CAC*. Les préférences restantes sont classées dans l'ordre de leurs scores relaxés dans des classes de K préférences. A chaque tour i, le système charge en mémoire les contextes de préférences complets de la $i^{ème}$ classe et les apparie avec *CAC*. L'algorithme s'arrête lorsque K contextes de préférences qui apparient totalement avec *CAC* sont retrouvés.

Dans le cas d'appariement partiel (voir algorithme de vérification d'intersection de contextes, dans chapitre 3, section 2.2.3), les contextes de préférences cp_i sont appariés suivant la fonction Chevauche (*CAC*, cp_i, {t_1, t_2, …}, nb_{min}). Un premier accès à la méta-base permet d'extraire les préférences, rattachées éventuellement à la cible de `EPC-ByMatch`, et associées à des contextes partageant nb_{min} nœuds avec *CAC* dont au moins un nœud de type t_1, t_2, … Les contextes sont chargés en mémoire. Pour chacun, le système procède à la vérification de l'existence de nœuds correspondant à tous les types t_1, t_2, …, au calcul de la distance d'édition et au tri de préférences.

5.2 Enrichissement de requête OLAP

Afin de personnaliser les requêtes de l'usager, les commandes OLAP sont d'abord décomposées dans une représentation interne arborescente. Une transcription SQL du résultat de l'algorithme de personnalisation permet de générer la requête SQL *Q'* à exécuter. Les différences entre la requête initiale *Q* et la requête personnalisée *Q'* sont :

- insertion dans la clause WHERE des prédicats de préférences sur les paramètres et sur les mesures non agrégées (Nb_Publis >10) en conjonction avec ceux de Q,
- insertion dans la clause HAVING des prédicats de préférences sur les mesures agrégées (SUM(Nb_Publis) >10) en conjonction avec ceux de Q,
- ajout des tables dimensions correspondant aux prédicats des dimensions non affichées par la requête (clause FROM), et
- ajouter des jointures correspondantes (clause WHERE).

Exemple. Reprenons l'exemple de la section 4.2. L'enrichissement de la requête en mode combiné (filtrage du résultat suivant toutes les dimensions) permet de générer la requête Q_2' suivante. La partie grisée représente les éléments ajoutés par rapport à la requête initiale Q_2.

```
SELECT NIVEAU, ANNEE, SUM (NB_PUBLIS)
FROM PUBLICATIONS AS P, MANIFESTATIONS AS M, DATES AS D,AUTEURS AS A
```

```
WHERE P.IDA = A.IDA AND P.IDM=M.IDM AND P.IDD = D.IDD
AND D.ANNEE ≥ 2009
AND A.POSTE='Doctorant'
GROUP BY NIVEAU, ANNEE;
```

L'exécution personnalisée de la requête Q_2 sous OLAPers permet d'afficher la TM suivante.

Figure 44. Exemple de résultat d'une requête personnalisée

6 Etude expérimentale

Nous avons utilisé dans nos différentes expérimentations la BDM de publications de notre laboratoire. Cette BDM contient 3 millions d'instances de faits et 500 instances de dimensions réparties sur les quatre dimensions *Manifestations*, *Auteurs*, *Dates* et *Domaines*.

6.1 Stockage des profils

Dans cette première expérience, nous avons mesuré le changement de la taille d'une BDM suite à son extension par des profils utilisateurs. Nous avons utilisé des profils synthétiques produits automatiquement par un générateur de profils. Rappelons qu'un élément de préférence (défini par les attributs *eid*, *op* et *val* de la relation *meta_preference*) peut être associé à plusieurs contextes et inversement. Nous avons utilisé 100 profils avec différents nombres d'éléments de préférence et différents nombres de contextes par élément de préférence. Les contextes de préférences générés sont des contextes d'analyse non évalués (sans valeurs). La Figure 45 montre le pourcentage de la taille de stockage des profils par rapport à la taille de la BDM pour des intervalles de 200 éléments de préférence et différents nombres moyens de contextes par élément de préférence. La taille de stockage de profils composés seulement de préférences absolues ne dépasse pas 0.5% de la taille de la BDM initiale. Ceci est expliqué par leur faible coût de stockage puisqu'elles comprennent seulement des références vers des éléments de la constellation, des prédicats (chaines de caractères) et des scores (nombres réels entre 0 et 1).

Pour le même nombre d'éléments de préférence, la taille de stockage des profils est plus importante lorsque ces éléments sont associés à plus de contextes. La différence de taille résulte du stockage des contextes. La Figure 45 montre une augmentation importante de la taille de stockage des profils pour un nombre d'éléments de préférence entre 0 et 600. Par

contre, nous observons une légère augmentation à partir de 800 éléments. En effet, le nombre total de contextes, dont la taille constitue la partie majeure de celle des profils, n'évolue pas trop à partir de ce point. Par conséquent, les nouvelles préférences sont associées avec des contextes existants (seuls le lien à la table *meta_contexte* est ajouté).

En conclusion, cette expérience a montré l'utilité de la personnalisation dynamique des données OLAP en terme de coût de stockage (au maximum 5% de la taille de la BDM). La génération de différentes versions de la BDM pour différents types d'utilisateurs (personnalisation statique) entraine un coût de stockage très important, alors que dans notre approche seuls les profils sont stockés.

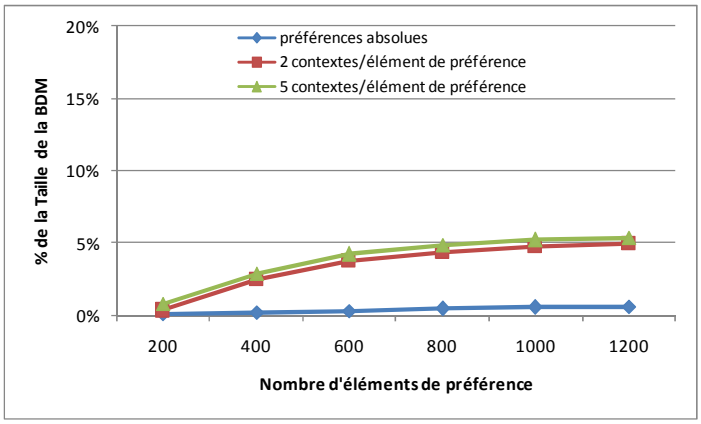


Figure 45. Impact du stockage des profils

Remarque. Dans la suite, la taille d'un profil représentera le nombre de préférences du profil.

6.2 Etude des performances

Cette série d'expériences permet d'évaluer la performance de la personnalisation des analyses. Nous avons étudié les processus de personnalisation et de recommandation de requêtes séparément afin d'identifier les critères qui affectent les performances de chacun.

Afin de tester nos algorithmes, nous avons considéré un ensemble de 10 requêtes de différents nombres de dimensions et attributs avec et sans prédicats de restriction. Nous avons utilisé des profils synthétiques produits automatiquement par un générateur de profils en fonction des exigences de chaque expérience.

Les durées sont exprimées en millisecondes.

6.2.1 Sélection des préférences

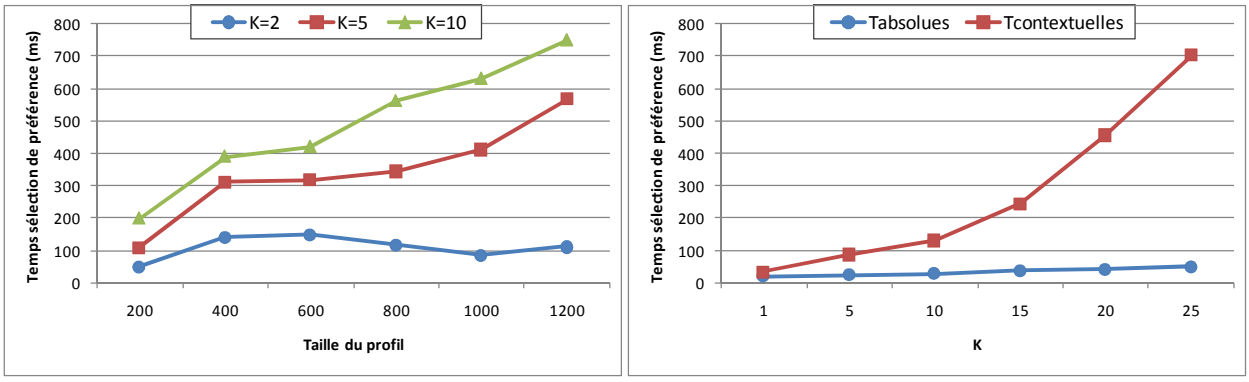
Cette expérince est une étude de la phase de sélection de préférences sur laquelle repose la personnalisation.

Coût de l'algorithme de sélection des préférences

Afin d'évaluer le temps de sélection de préférences, nous avons exécuté alternativement les algorithmes `EPCV-K` et `EPC-ByMatch`. Notons que dans le cas de `EPC-ByMatch`, ce qui est mesuré ici est le temps nécessaire à la recherche des K meilleurs éléments pivots.

Nous avons mesuré pour chaque requête le temps de sélection de préférences sur un profil synthétique de taille 200 (U_{200}). Nous avons reporté le temps moyen des différentes requêtes pour chaque valeur de K (K=2, K=5 et K=10). Puis, nous avons mesuré ce temps en étendant à chaque fois le profil par 200 nouvelles préférences (U_{400}, U_{600}, U_{800}, U_{1000}, U_{1200}). Selon le fonctionnement de l'algorithme de sélection, le nombre d'accès à la méta-base (chargement de contextes de préférences) dépend du paramètre K et le temps de traitement en mémoire (appariement de contexte et tri des préférences) est lié à la taille du profil.

La Figure 46 (a) montre que, quelque soit la taille du profil, le temps nécessaire à la sélection des préférences augmente suivant K. Rappelons que les préférences actives sont triées suivant leurs scores fictifs et l'algorithme s'arrête à la k^{ième} préférence dont le contexte apparie avec *CAC*. Pour K donné, l'évolution du temps suite à l'ajout de préférences à un profil U_i dépend des nouvelles préférences (actives ou non) et de leurs scores. Plus les contextes des préférences actives les mieux classées apparient avec *CAC*, moins d'accès à la méta-base et d'appariements sont nécessaires, et par conséquent moins de temps est nécessaire pour la sélection des préférences. Ainsi, lorsque toutes les nouvelles préférences actives sont classées après les K meilleures préférences de U_i, le temps de sélection augmente très légèrement ou reste invariant (par exemple pour K=5 et taille de 400 à 600) car le nombre d'accès à la méta-base est le même. Cependant, le temps de sélection diminue si l'intégration des nouvelles préférences actives change l'ordonnancement des préférences candidates et les rend à la tête de la liste (par exemple pour K=2 et taille de 600 à 800). Dans ce cas, moins d'accès à la méta-base sont nécessaires. Sinon, le temps de sélection augmente à cause d'un nombre plus important d'accès à la méta-base afin de rechercher les K meilleures préférences candidates. Nous pouvons observer que le temps augmente légèrement pour K=2 et taille de 200 à 600 et il évolue remarquablement pour K=10 et taille de 600 à 1200.



(a) Temps de sélection des préférences contextuelles en fonction de la taille de profils

(b) Temps de sélection des préférences contextuelles ($T_{contextuelles}$) et absolues ($T_{absloues}$) suivant K

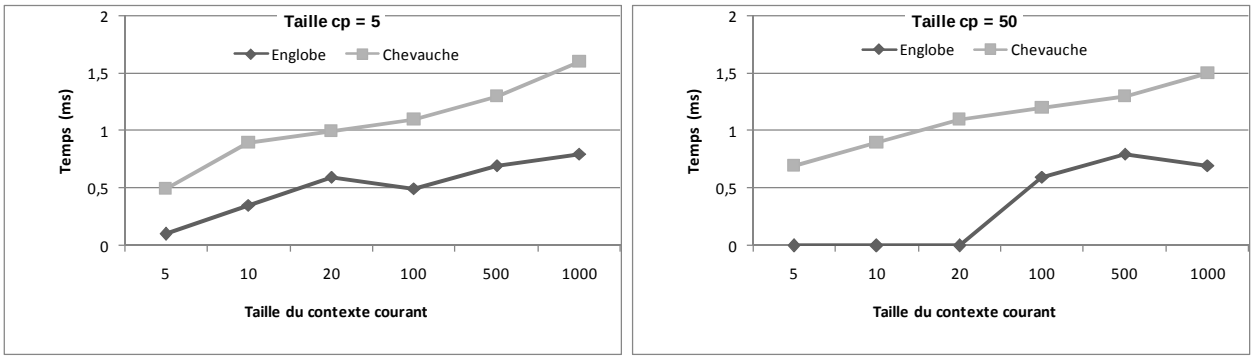
Figure 46. Performance de la sélection des préférences

Afin de confirmer cette tendance, nous avons mesuré le temps de sélection des préférences contextuelles ($T_{contextuelles}$) et des préférences absolues ($T_{absolues}$) à partir de profils de tailles différentes. La Figure 46 (b) présente le temps moyen de sélection en fonction de K. Comme aucun appariement n'est effectué pour la sélection des préférences absolues, $T_{absolues}$ n'évolue pas suivant K. Cependant, $T_{contextuelles}$ est une courbe croissante en fonction de K à cause de plus d'appariements de contexte qui sont effectués. $T_{contextuelles}$ reste acceptable et ne dépasse pas 1 seconde. Notons que l'écart entre les courbes de $T_{contextuelles}$ et $T_{absolues}$ correspond au

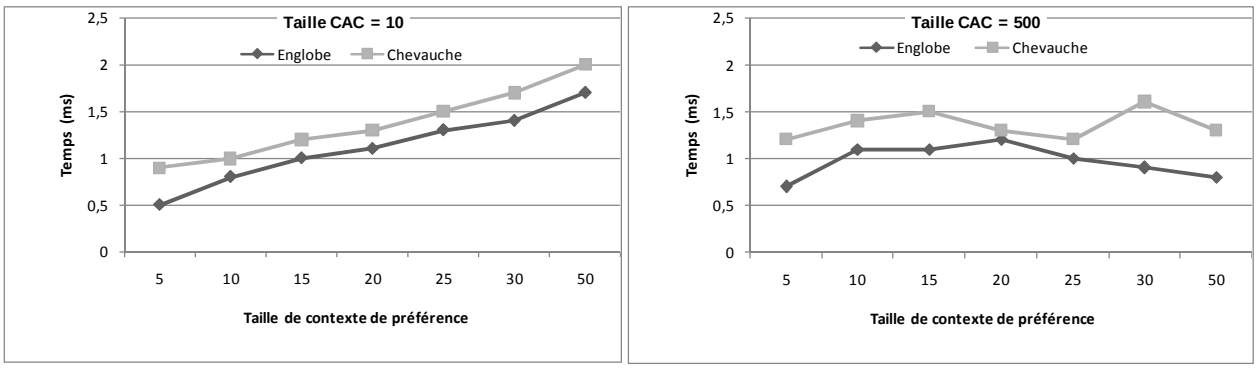
temps d'appariement de contexte (chargement des contextes de préférence à partir de la méta-base puis appariement avec *CAC* en mémoire).

Coût d'appariement de contextes

Cette expérience vise à analyser le temps nécessaire à l'appariement de contexte en fonction de la taille de *CAC* puis de la taille moyenne des contextes de préférences (la taille d'un contexte est le nombre de nœuds de son arbre. *CAC* représente un contexte non évalué (tailles 5, 10 et 20) ou un contexte complet (tailles 100, 500 et 1000). Nous avons calculé le temps nécessaire à l'appariement total (Englobe (*CAC*, cp_i)) puis partiel (Chevauche (*CAC*, cp_i, NULL, NULL)) de *CAC* avec les contextes cp_i. Notons que le temps considéré ici concerne les algorithmes de vérification de relation de contexte (dominance et intersection de contextes) effectuant un parcours en largeur alterné des arbres de contexte pré-chargés en mémoire. Le temps de chargement des contextes à partir de la méta-base n'est pas pris en compte. La Figure 47 montre que le temps moyen d'appariement total évolue d'une manière croissante en fonction de la taille de *CAC*. Cependant, il reste inférieur au temps d'appariement partiel. En effet, lors de l'appariement partiel, au moins tous les nœuds structure de chaque cp_i sont comparés à ceux de *CAC* afin de déterminer les nœuds en communs. Cependant, l'algorithme d'appariement total s'arrête lors du premier nœud structure ou valeur de cp_i qui n'appartient pas à *CAC*.



(a) Temps d'appariements total et partiel en fonction de la taille du contexte courant



(b) Temps d'appariements total et partiel en fonction de la taille du contexte de préférence

Figure 47. Comparaison des performances de l'appariement total et partiel de contexte

La Figure 47 (a) montre que le temps d'appariement total et partiel évoluent légèrement lorsque *CAC* est non-évalué (taille entre 5 et 20). Cependant, le temps n'évolue pas beaucoup

pour des tailles plus importantes de *CAC* (taille entre 100 et 1000). En effet, il s'agit du cas où *CAC* est évalué. Seuls les nœuds valeur correspondant à des nœuds structures communs sont comparés. Ainsi, ce temps peut diminuer lorsque le premier nœud structure différent est retrouvé plus vite lors de l'appariement total ou quand le nombre de nœuds structure communs est plus petit dans le cas de l'appariement partiel.

La Figure 47 (b) montre que le temps d'appariement évolue moins en fonction de la taille de cp_i. L'écart entre l'appariement partiel et total est petit lorsque *CAC* est non-évalué (taille = 10). Cet écart est plus important pour *CAC* évalué (taille = 500). Il faut noter que le temps d'appariement partiel ne diminue pas lorsque *CAC* est non-évalué car les nœuds de *CAC*, qui sont tous des nœuds structure, sont visités.

En conclusion, le temps d'appariement reste très faible (ne dépasse pas 3 milliseconde). Nous pouvons conclure que la variation du temps de sélection de préférences est due essentiellement au temps d'accès à la méta-base pour l'extraction des contextes de préférences.

6.2.2 Personnalisation de requête

Nous avons identifié deux critères critiques qui affectent la performance de la personnalisation de requête : le nombre maximal de préférences intégrées dans la requête (K) ; et le seuil du score de préférences considérées (λ)

La première expérience vise à étudier le surcoût engendré par la personnalisation de requête. Nous avons effectué une série de tests de personnalisation d'une requête par rapport à 10 profils synthétiques en variant K. Pour chaque valeur de K, nous avons calculé le temps d'exécution de la requête initiale (sans personnalisation) et la moyenne du temps d'exécution de la requête personnalisée. Le temps d'exécution de la requête personnalisée est la somme du temps de personnalisation de la requête (sélection et intégration de préférences) et du temps de calcul du résultat de la requête personnalisée. La Figure 48 montre que le temps d'exécution de la requête personnalisée est inférieur à celui de la requête initiale lorsque K est entre 1 et 30. En effet, comme les préférences sont intégrées sous forme de prédicats de restriction, l'exécution de la requête enrichie renvoie moins de n-uplets et nécessite moins de temps. La diminution du temps de calcul du résultat permet de récompenser relativement le temps de réponse global qui aurait augmenté à cause du temps de personnalisation. Le temps d'exécution de la requête personnalisée dépasse celui de la requête initiale lorsque K est assez important (K=40). Dans ce cas, le temps de personnalisation de requête est élevé.

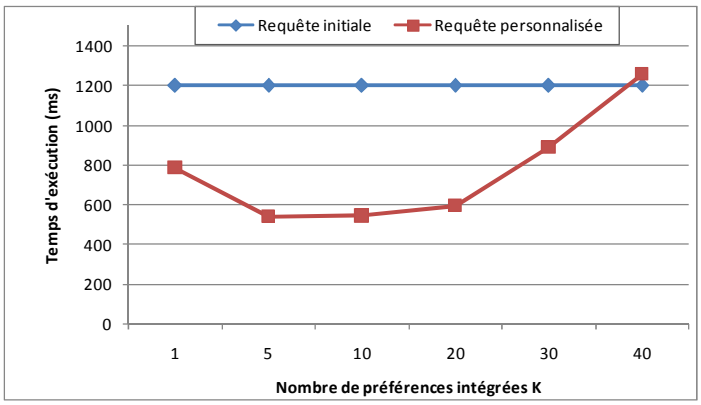


Figure 48. Performance de la personnalisation de requête suivant K

La deuxième expérience permet de comparer les coûts des processus de personnalisation naïve et avancée de requête. Nous avons mesuré pour la même requête précédente le temps d'exécution suivant la personnalisation naïve ($T_{naïve}$) puis avancée avec K = 2 ($T_{K=2}$) et K=10 ($T_{K=10}$) en variant le seuil des scores de préférences λ. La Figure 49 montre que le temps nécessaire pour exécuter la requête personnalisée est inférieur ou légèrement supérieur à celui de la requête initiale.

$T_{naïve}$ diminue en fonction du seuil λ et reste toujours supérieur à $T_{K=2}$ et $T_{K=10}$. En effet, afin d'extraire toutes les préférences candidates, l'algorithme de personnalisation parcourt toutes les préférences actives qui sont plus nombreuses lorsque le seuil λ est plus petit. Ce temps est réduit dans l'approche avancée où seules les K premières préférences pré-ordonnées sont extraites. Lorsque le seuil λ est égal à 0.4, $T_{naïve}$ est très proche du temps de requête sans personnalisation. Cette valeur du seuil λ représente un point d'équilibre entre le surcoût engendré par la personnalisation de requête, d'une part, et le gain en temps de calcul du résultat d'une requête plus restrictive, d'autre part.

Concernant l'approche avancée, nous pouvons observer que plus K est grand, plus le temps d'exécution de requête personnalisée est proche de celui de l'approche naïve. $T_{K=2}$ est inférieur $T_{K=10}$ lorsque le seuil λ est entre 0.1 et 0.7. Bien que le temps de calcul du résultat avec K=10 soit intuitivement inférieur au temps avec K=2, le temps d'exécution global de requête personnalisée $T_{K=10}$ reste largement supérieur à $T_{K=2}$. Ceci peut être expliqué par la différence plus importante du temps de personnalisation de requête (surtout le temps de sélection des préférences). Cette tendance change partir de la valeur 0.8 du seuil λ où le temps de personnalisation avec K=2 devient proche du temps correspondant à K=10. Dans ce cas, le nombre des préférences actives devient très petit (seuil important). Par conséquent, $T_{K=10}$ devient inférieur à $T_{K=10}$ grâce à son temps de calcul du résultat qui est moins important.

Finalement, nous pouvons observer que, quelle que soit la méthode de personnalisation utilisée, le temps d'exécution de requête personnalisée dépasse le temps de requête initiale lorsque le seuil est égal à 1. En l'absence de préférences candidates avec un score égal à 1, le temps d'exécution de requête personnalisée est égal au temps de requête initiale augmenté du temps de personnalisation de requête.

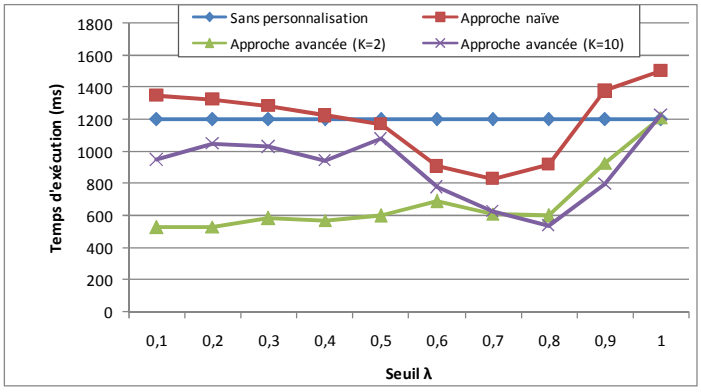


Figure 49. Comparaison des performances de la personnalisation naïve et avancée de requête

Les résultats des deux dernières expériences sont conformes aux résultats attendus vu la nature du rôle de K qui permet de contrôler le coût du processus de personnalisation en fonction de contraintes spécifiques : la personnalisation est plus performante avec K.

6.2.3 Recommandation

Coût du processus de recommandation

Cette expérience évalue le temps nécessaire pour générer une recommandation (fonction ORecommend) sans personnalisation en fonction du seuil des préférences utilisées. Nous avons calculé le temps de génération lorsque le nombre de paramètres par dimension n'est pas spécifié, puis quand le système génère une requête avec un seul paramètre par dimension. Rappelons que dans notre cas, chaque requête comprend deux dimensions affichées. Nous avons reporté le temps moyen de génération de recommandation par anticipation puis d'alternatives pour nos requêtes de test.

La Figure 50 montre que, quelle que soit la méthode utilisée, le temps nécessaire pour générer une recommandation diminue linéairement avec le seuil des préférences. Le temps reste acceptable sauf pour la génération d'alternatives avec un seuil faible (λ entre 0.1 et 0.3) où il atteint 12 secondes et devra être optimisé. Notons que ce cas est plutôt théorique car le seuil doit être configuré à de hautes valeurs afin d'avoir des résultats satisfaisants. Le temps de génération d'alternatives est globalement supérieur à celui des recommandations par anticipation. Ceci peut être expliqué par un nombre d'itérations (redressement et accomplissement) plus important dans le cas des contextes intersectés puisque le nombre de contextes qui apparient partiellement avec *CAC* est toujours supérieur à celui des contextes qui apparient totalement. De plus, pour une seule itération, la sélection des préférences par appariement partiel nécessite plus de temps à cause du calcul de la distance d'édition pour le tri des préférences.

Nous pouvons remarquer que le temps diminue lorsque les contextes recommandés comportent un seul paramètre par dimension. Le temps de génération avec et sans contrainte sur le nombre de paramètres par dimension ne varie pas à partir de la valeur 0.9 du seuil λ. Ceci peut être expliqué par l'absence de préférences candidates sur les paramètres ayant un score supérieur ou égal à 0.9.

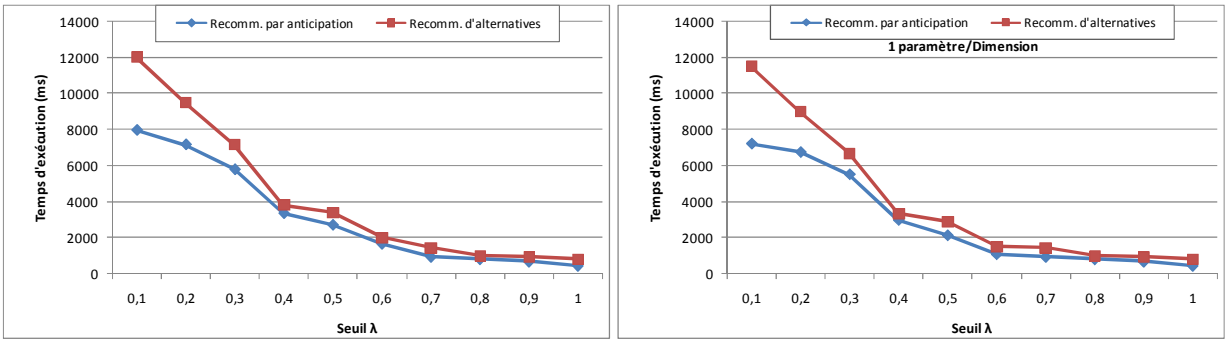


Figure 50. Performance de la recommandation de requêtes

6.3 Etude de l'efficacité de la personnalisation

Nous présentons dans cette section notre seconde expérimentation que nous avons menée avec des utilisateurs réels afin d'étudier l'utilisabilité de notre système de personnalisation. 15 utilisateurs ont participé à cette expérimentation dont 5 connaissent le principe de l'analyse en ligne (OLAP) des données (experts).

Nous avons demandé à chaque usager de spécifier 100 préférences par rapport à notre base de test :

- en sélectionnant un profil parmi une liste de profils prédéfinis, puis en le changeant afin de l'adapter et
- en associant des préférences à des contextes d'analyse prédéfinis avec différents niveaux de détail.

Cette expérimentation se déroule suivant un scénario proactif reposant sur un retour utilisateur implicite, puis suivant un scénario interactif se basant sur un retour d'expérience utilisateur.

6.3.1 Evaluation proactive

Notre première série d'expériences permet d'analyser l'efficacité des processus de personnalisation et de recommandation de requête en terme de rappel et de précision.

Après une explication du schéma de la BDM, chaque usager a été demandé d'effectuer une suite d'opérations OLAP afin répondre à la question suivante :

« Quelles sont les catégories de chercheurs en baisse d'activité de recherche et plus particulièrement, dans quel type de manifestations ? »

La suite d'opérations définies constitue une analyse OLAP. Les analyses effectuées comportent en moyenne 9 opérations. Pour chaque opération, le système affiche le résultat de la requête personnalisée (sans avertir l'usager) et génère les recommandations qu'il garde en interne :

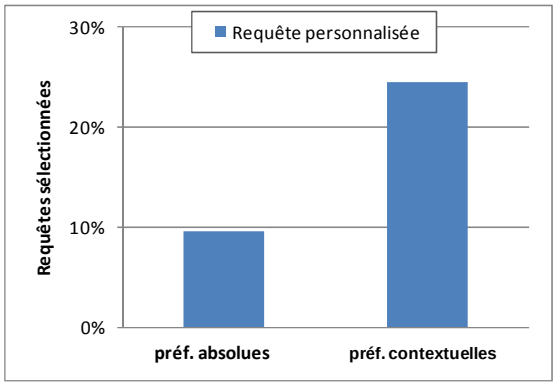
- Une requête personnalisée est jugée pertinente si l'usager réutilise son résultat dans l'étape suivante, c'est-à-dire, si la TM résultat de la requête personnalisée représente la TM en entrée (T_{SRC}) de l'opération suivante.
- Une requête recommandée est jugée pertinente si elle est lancée dans les étapes suivantes de l'analyse.

Précision du processus de personnalisation de requête

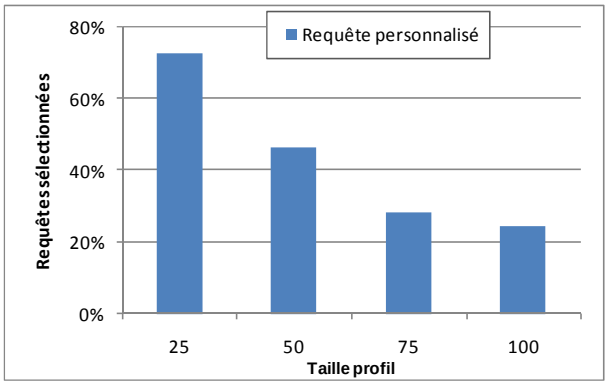
Nous demandons à l'usager d'effectuer l'analyse cinq fois. Lors de la première analyse, les requêtes sont enrichies par des préférences absolues. Durant les quatre analyses suivantes, les requêtes sont personnalisées à l'aide de profils (comprenant des préférences contextuelles) de tailles différentes (25, 50, 75 et 100 préférences). La personnalisation est effectuée suivant une approche naïve (toutes les préférences candidates sont intégrées dans la requête). Afin d'évaluer la précision des requêtes personnalisées, nous avons calculé le pourcentage des TM personnalisées qui ont été réutilisées par l'usager dans l'étape suivante.

La Figure 51 (a) montre que les usagers sélectionnent plus les requêtes personnalisées à l'aide de préférences contextuelles. Ainsi, la contextualisation permet d'améliorer la qualité de la personnalisation de requête. Par ailleurs, la Figure 51 (b) montre que pour des profils de taille 25, 73% des requêtes personnalisées ont été sélectionnées par l'usager, alors que seulement 24% de requêtes personnalisées à partir de profils de taille 100 ont été sélectionnées. La personnalisation de requête induit des résultats moins bons lorsqu'elle est basée sur des profils de tailles plus importantes. Ceci peut être expliqué par l'intégration d'un nombre important de préférences dont la conjonction n'est pas appropriée à l'usager. En effet, des préférences supplémentaires sont intégrées dans la requête si le profil exploité est de plus grande taille. Ainsi, la définition d'un nombre maximal de préférences à intégrer (paramètre K) permettrait

d'améliorer l'efficacité du processus de personnalisation de requête. Ceci rejoint les résultats de notre expérimentation de la performance de la personnalisation de requête (*cf.* Figure 48).



(a) Comparaison entre la personnalisation de requête à partir de préférences absolues et contextuelles



(b) Efficacité de la personnalisation de requêtes en fonction de la taille du profil

Figure 51. Efficacité du processus de personnalisation de requête

Rappel/Précision du processus de recommandation par anticipation

Cette expérience est effectuée pour *S*=0 et *N*=2.

Avant d'évaluer l'efficacité du processus de recommandation, nous avons mené une expérience pour déterminer la valeur de β permettant d'obtenir les meilleurs précision et rappel (β étant le paramètre de la fonction F_{CA}^{RANK} définissant le score des éléments de la requête de l'usager dans la requête recommandée). Les deux meilleures recommandations sont déterminées à l'aide de la fonction de score F_{CA}^{RANK}. La Figure 52 montre qu'une faible précision est obtenue lorsque β est proche de 0 ou de 1. La précision et le rappel maximaux sont obtenus lorsque β vaut 0.6. Ainsi, dans les expérimentations suivantes du processus de recommandation, la valeur de β est fixée à 0.6.

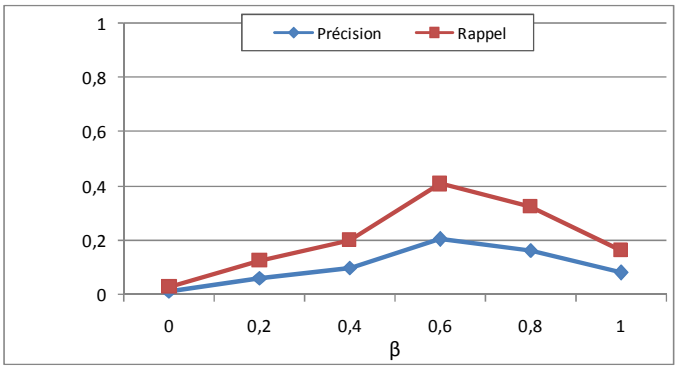


Figure 52. Précision et Rappel du processus de recommandation en fonction de β

Nous avons évalué le rappel et la précision des recommandations par anticipation en utilisant les formules suivantes :

$$- Précision(ORecommend) = \frac{|\{Q_{analyse} \cap Q_{Rec}\}|}{|\{Q_{Rec}\}|}$$

$$- Rappel(ORecommend) = \frac{|\{Q_{analyse} \cap Q_{Rec}\}|}{|\{Q_{analyse}\}|}$$

où Q_{Rec} est l'ensemble des meilleures requêtes générées par la fonction *ORecommend* et $Q_{analyse}$ est l'ensemble des requêtes lancées par l'usager dans une étape ultérieure de son analyse.

Les recommandations candidates sont générées à partir de profils de tailles différentes en considérant d'abord 20 préférences, puis en rajoutant à chaque fois 20 préférences supplémentaires. La Figure 53 montre que la taille du profil a une influence sur la qualité des recommandations générées. Contrairement à la personnalisation de requête, plus les profils sont volumineux, plus les recommandations sont de bonne qualité. La précision maximale obtenue est égale à 0.4, ce qui représente un bon résultat puisque 45% des contextes de préférence saisis par les usagers ne comportent pas de valeurs. Ainsi, l'appariement de ces contextes avec la requête est basé principalement sur les composants structurels sans exploiter le résultat. Il faut noter que la précision est inférieure au rappel car le nombre de requêtes recommadées est supérieur à celui des requêtes lancées durant l'analyse de l'usager.

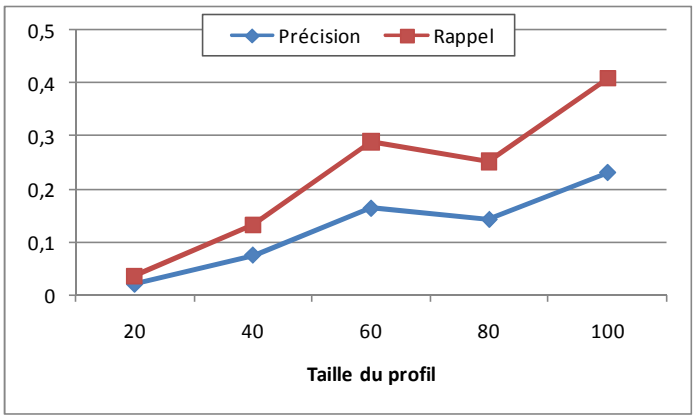


Figure 53. Précision et Rappel du processus de recommandation par anticipation

Simplification de l'analyse OLAP

L'objectif de cette expérience est d'étudier la contribution des recommandations par anticipation à la réduction de la charge de navigation de l'usager. Cette expérience correspond au cas de génération de recommandations par anticipation à partir de profil de taille 100. Nous avons calculé le nombre d'opérations utilisateur anticpées par les recommandations petinentes. Dans la Figure 54, un point *(x,y)* du graphique indique que *x%* des requêtes pertinentes recommandées par anticipation ont été lancées par l'usager après *y* opérations. La Figure 54 montre que 40% des recommandations permettent d'anticiper une opération de l'utilisateur. Cependant, seulement 3% des recommndations permettent de proposer la 6ème opération de l'usager. Les résultats ont confirmé que l'intégration du mécanisme de recommandation permet de faciliter les analyses des usagers. Mais, ce mécanisme ne permet pas d'anticiper les opérations de l'usager à un niveau détaillé de l'analyse, ce qui peut être expliqué par la génération d'une recommandation par une seule itération de la fonction *ORecommend*. En effet, afin de recommander des opérations correspondant à des niveaux plus détaillés de l'analyse OLAP, il faut répéter d'une manière récursive l'algorithme de génération de recommandation pour la même requête de l'usager. La première itération

prendrait en entrée le contexte induit à la requête de l'usager. Puis, chaque itération prendrait en entrée le contexte d'analyse candidat généré par l'itération précédente. Nous avons rejeté ce choix à cause des problèmes de performance induits.

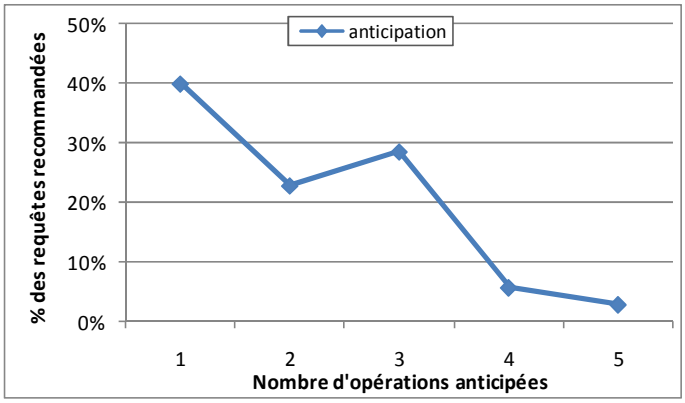


Figure 54. Etude de la réduction de l'effort d'analyse OLAP

6.3.2 Evaluation avec retour d'expérience utilisateur

La deuxième série d'expériences portant sur l'utilisabilité du système permet d'évaluer la fonction de score des recommandations F_{CA}^{RANK}.

Nous avons demandé à chaque utilisateur de définir trois requêtes différentes.

Pour chaque requête, le système calcule les recommandations par anticipation et les alternatives. Afin de générer un nombre important de recommandations, tous les éléments pivots sont sélectionnés ; une recommandation est générée pour chacun. Les scores des recommandations candidates sont calculés suivant F_{CA}^{RANK} pour différentes tailles du profil. Le système affiche toutes les recommandations par anticipation candidates, en les complétant éventuellement par des requêtes arbitraires pour atteindre un ensemble comportant 8 requêtes. Nous demandons à l'usager d'indiquer les 3 meilleures recommandations à réprésenter sa prochaine requête. De même, l'usager est demandé d'indiquer, parmi les recomandations alternatives candidates, les 3 meilleures qui sont utiles pour découvrir de nouvelles données. En se basant sur les résultats de ce test, nous avons évalué la précision de la fonction de score F_{CA}^{RANK} suivant la formule suivante :

$$- Précision(F_{CA}^{RANK}) = \frac{\left|\{Top\text{–}3(Q)_{User} \cap Top\text{–}3(Q)_{F^{RANK}}\}\right|}{|\{Top\text{–}3(Q)_{F^{RANK}}\}|}$$

où $Top\text{–}3(Q)_{User}$ est l'ensemble des trois meilleures requêtes sélectionnées par chaque usager et $Top\text{–}3(Q)_{F^{RANK}}$ est l'ensemble des trois requêtes ayant les meilleurs scores suivant F_{CA}^{RANK}.

Notons que pour cette expérience, le rappel est égal à la précision car le nombre des requêtes sélectionnées par l'usager est égal au nombre de requêtes ordonnées par F_{CA}^{RANK}.

La Figure 55 montre que le tri effectué à l'aide de la fonction F_{CA}^{RANK} est plus précis lorsque les scores sont canculés à partir de profils de tailles plus importantes. La précision du tri suivant F_{CA}^{RANK} est de 60% dans le cas d'anticipation à partir des profils de taille 100. Afin d'améliorer la performance du tri de recommandations, le calcul de score doit être basé sur des profils de tailles importantes. D'après la Figure 55, F_{CA}^{RANK} obtient de moins bons résultats dans le cas des recommandations alternatives. Enfin, cette figure montre que la précision peut

diminuer avec des profils plus volumineux, ce qui peut être expliqué par le choix arbitraire de certains éléments de la recommandation rendant la requête recommandée inadéquate pour l'usager.

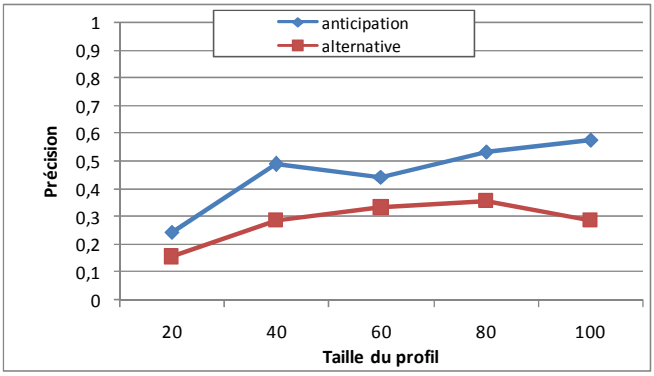


Figure 55. Précision de la fonction de score F_{CA}^{RANK}

7 Bilan

Dans ce chapitre, nous avons présenté notre prototype de système de personnalisation des analyses OLAP. Ce système implémente le modèle de préférences contextuelles et les processus génériques de personnalisation (Jerbi et al., 2010b, 2010c) et de recommandation de requêtes (Jerbi et al., 2009b, 2009c) que nous avons proposés. Les contextes d'analyse manipulés par notre prototype comportent deux dimensions affichées et sont visualisées sous forme de tables multidimensionnelles.

Nous avons proposé un langage de manipulations OLAP personnalisées (Jerbi et al., 2008) afin de valider le premier scénario de recommandation. Les scénarios de recommandation 2 et 3 sont traduits par l'affichage de requêtes recommandées en plus du résultat de la requête.

Nous avons mené des expérimentations de nos différents algorithmes de personnalisation de requêtes et de recommandation. Les résultats que nous avons obtenus sont satisfaisants. L'étude des performances a montré que le coût de personnalisation dépend essentiellement du temps d'appariement de contextes, qui doit être optimisé lorsque les tailles des profils sont importantes. Par ailleurs, le coût de stockage de profils de moyennes tailles (jusqu'à 1200 préférences) est faible. Cependant, il serait nécessaire d'optimiser le stockage des contextes de grandes tailles, en regroupant par exemple les contextes dans des classes et en stockant le contexte représentatif de chaque classe.

Lors de l'étude de l'efficacité de nos algorithmes de personnalisation, nous avons observé que la qualité des requêtes personnalisées et des requêtes recommandées est acceptable malgé l'utilisation de profils de petites et moyennes tailles. Les résultats ont montré que les processus de personnalisation et de recommandation sont plus efficaces lorsqu'ils sont basés sur des conaissances plus approfondies de l'usager (profils de tailles importantes et contextes de préférences détaillés). L'étude de la précision du processus de personnalisation de requêtes a démontré la nécessité de l'imposition d'une contrainte sur le nombre de préférences utilisées, ce qui rejoint les résultats de l'étude de performance de ce processus. Cependant, afin d'améliorer la qualité des requêtes personnalisées et des requêtes recommandées, les

contextes de préférences appariés doivent être plus détaillés et doivent intégrer plus de valeurs. De tels contextes ne peuvent pas être facilement définis par l'usager.

Références

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2008). Management of Context-aware Preferences in Multidimensional Databases. Intl. Conf. on Digital Information Management (ICDIM), IEEE, pages 669–675.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2009b). Applying Recommendation Technology in OLAP Systems. Intl. Conf. on Enterprise Information Systems (ICEIS), Springer, LNBIP 24, pages 220–233.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G (2009c). Preference-Based Recommendations for OLAP Analysis. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer-Verlag, pages 467–478.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2010b). Personnalisation du contenu des bases de données multidimensionnelles. Journées Francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA), pages 5–20.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2010c). A Framework for OLAP Content Personalization. East European Conf. on Advances in Databases and Information Systems (ADBIS), Springer-Verlag, pages 262–277.*

Ravat F., Teste O., Tournier R., Zurfluh G. (2008). Algebraic and graphic languages for OLAP manipulations. Intl. Journal of Data Warehousing and Mining (IJDWM), Vol. 4, No. 1, pages 17–46.

Chapitre 7

Conclusion et perspectives

Sommaire

1 Bilan général

Les travaux de recherche présentés dans ce mémoire de thèse s'inscrivent dans le cadre des systèmes d'aide à la décision. Ces systèmes se basent sur un processus d'analyse en ligne (OLAP) de données organisées de manière multidimensionnelle. Jusqu'à présent, peu de travaux se sont intéressés à la personnalisation des analyses OLAP. Nous avons constaté l'absence de solutions globales pour la personnalisation de ces analyses. De plus, les modèles de préférences de l'usager ne sont pas très développés dans le contexte OLAP. Sur la base de ces constatations, nous avons proposé une approche globale de personnalisation des analyses OLAP basée sur les préférences. Plus précisément, notre contribution comprend :

Un modèle générique des analyses OLAP basé sur le concept de contexte d'analyse. Un contexte d'analyse traduit l'état courant d'une analyse. Il regroupe tous les éléments de la requête de l'usager ainsi que son résultat. Le contexte d'analyse est modélisé par une structure arborescente qui reflète la nature des relations entre les différentes structures multidimensionnelles (association, hiérarchie, …), entre les structures et les valeurs (instanciation) et entre les valeurs (imbrication, agrégation). Ainsi, les analyses OLAP sont représentées par un graphe traduisant leur aspect navigationnel : les nœuds représentent les contextes d'analyse et les arcs traduisent les opérations de transformation de données permettant de passer d'un contexte à un autre. Ces opérations s'appliquent à l'arbre d'un contexte d'analyse et sont indépendantes du langage utilisé.

Une approche de personnalisation globale basée sur le modèle de graphe d'analyse. La personnalisation de requête permet de personnaliser le contexte d'analyse courant et la personnalisation de la navigation permet de recommander à l'usager les prochains contextes d'analyse à visiter suivant trois scénarios de recommandation.

Un modèle de préférences OLAP portant sur le schéma ainsi que les valeurs d'une BDM. Chaque préférence est associée avec un contexte d'analyse qui précise son cadre d'application. L'ensemble des préférences contextuelles représente un profil utilisateur qui étend la BDM.

Un cadre de personnalisation des requêtes OLAP. L'objectif est d'enrichir la requête par les préférences de l'usager afin de personnaliser son résultat. La requête est d'abord traduite sous la forme du contexte d'analyse courant (sans les valeurs du résultat). Le processus de personnalisation repose sur deux étapes: a) la sélection des préférences qui sont associées au contexte d'analyse courant en gérant les conflits pouvant survenir lors de cette étape et b) la réécriture de la requête initiale en fonction des préférences. La personnalisation de requête est effectuée suivant une perspective utilisateur où toutes les préférences sont utilisées afin d'augmenter l'intérêt du résultat ou selon une perspective système où seules les K meilleures préférences qui satisfont une contrainte de personnalisation sont considérées.

Un cadre de recommandation de requêtes OLAP. Trois scénarios de recommandation sont proposés afin d'assister l'usager à différents niveaux de ses manipulations. Au cours de la formulation de requête, l'usager est guidé d'une manière interactive en lui proposant des éléments de la requête. Après l'exécution de la requête, le système recommande les requêtes suivantes en anticipant les besoins de l'usager, et des requêtes alternatives qu'il n'est pas susceptible de demander. L'algorithme de recommandation prend en entrée suivant le scénario le contexte d'analyse (partiel) induit par la requête en cours de formulation, ou le

contexte d'analyse (complet) qui résulte de la requête exécutée. L'appariement du contexte d'analyse courant avec les contextes des préférences permet de déterminer les structures et les valeurs à considérer pour générer un contexte d'analyse à recommander. Plusieurs contextes d'analyse candidats sont générées progressivement, puis sont triées afin de renvoyer les meilleurs.

Personnalisation de requêtes Vs. Recommandation de requêtes

La personnalisation et la recommandation des requêtes partagent le même objectif global qui est la personnalisation de l'analyse OLAP. Elles ont plusieurs notions communes, telles que l'appariement de contextes, la sélection de préférences, l'intégration de préférences, le tri, etc. Toutefois, ces deux fonctionnalités ont des entrées, une démarche et des sorties différentes. Nous pouvons citer les différences suivantes :

- La personnalisation des requêtes, telle que nous la définissons, agit sur une requête complète de l'usager avant son exécution, alors que la recommandation, nécessite l'exécution de la requête, puis transforme son résultat (scénarios 2 et 3), et traite des requêtes incomplètes (scénario 1).
- La personnalisation de requêtes emploie seulement les préférences sur les valeurs. L'appariement entre le contexte d'analyse courant et les contextes des préférences est total. Par contre, la recommandation considère toutes les préférences et applique un appariement partiel ou total de contextes.

Le tableau suivant présente les différences entre ces deux processus.

		Personnalisation	**Recommandation**
Timing		Après l'application de la requête et avant son exécution	Au cours de la formulation de la requête ou après son exécution
Entrée	Requête *Q*	*Q*	*Q* incomplète ou Résultat de *Q*
	Préférences	sur les valeurs	- sur les structures - sur les valeurs
Sortie (C_i' : contexte d'analyse non-évalué)		C_i', tel que Englobe(C_i',CAC_Q)	{C_i'}, tel que Englobe(C_i',CAC_Q) ou Chevauche(C_i',CAC_Q)
Options		Personnalisation naïve ou avancée	3 scénarios : (1) assistance, (2) requête anticipée, (3) requête alternative
Contexte d'analyse courant		Contexte induit par *Q* (non-évalué)	- (1) : Contexte induit par *Q* (partiel) - (2), (3) : Contexte résultat de *Q* (complet)
Appariement de contextes		Total	Total et/ou Partiel
Gestion des conflits	Politique de résolution (conflit préférence-requête)	Prédicats de requête prioritaires	Prédicats de préférence prioritaires
Intégration de préférences		enrichissement	enrichissement et/ou substitution
Nombre d'itérations (sélection-intégration)		n=1	n≥1
Tri		F_P^{RANK}: Tri de préférences	F_{CA}^{RANK} : Tri de contextes d'analyse

Tableau 6. Différences entre les processus de personnalisation et de recommandation de requêtes OLAP

Par ailleurs, la personnalisation représente une étape de la recommandation et la recommandation peut être une étape ultérieure à la personnalisation. Ainsi, les requêtes recommandées peuvent être personnalisées et réciproquement.

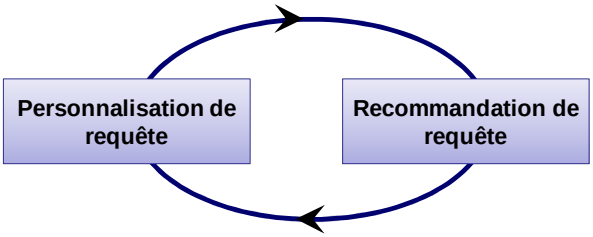


Figure 56. Complémentarité des actions de personnalisation et de recommandation de requête

2 Perspectives

Nos travaux présentent des limites que nous avons identifiées comme des perspectives à court terme de nos travaux.

Extension du profil de l'usager. Notre modèle de profil actuel intègre seulement les préférences de l'usager. Une extension de ce modèle afin d'intégrer des éléments liés au profil métier (rôle, département, …) serait judicieuse afin de concilier entre la perception de l'usager des données reflétant ses centres d'intérêt personnels et ses caractéristiques professionnelles qui reflètent le lien entre son métier et les données qui sont susceptibles de l'intéresser. Ce profil métier doit également traduire l'expertise de l'usager. Nous avons débuté récemment des travaux dans cet axe (Jerbi et al., 2011). Les réflexions de l'usager, ses interprétations et ses commentaires dans un contexte d'analyse donné sont matérialisés par le concept d'annotations contextuelles. Les annotations sont restituées lors de l'affichage du résultat de la requête offrant un cadre d'analyse personnalisé.

Modélisation du contexte. Notre modèle de préférences supporte des contextes internes. Nous envisageons étendre ce modèle afin d'inclure des contextes externes qui sont indépendants des données de la BDM et des attributs de la requête. Il faut noter que notre modèle d'arbre de contexte actuel permet d'intégrer certains éléments de contexte externe tels que la situation spatio-temporelle. Une étude exhaustive des paramètres de contextes externes dans un environnement OLAP serait nécessaire afin de généraliser cette affirmation. Cependant, l'étape de sélection des préférences restera basée sur l'appariement de contexte en identifiant le contexte externe courant de l'usager qui sera apparié aux contextes externes des préférences. L'étape d'intégration de préférences restera inchangée.

Apprentissage des préférences. Nos travaux se sont focalisés principalement sur la modélisation, puis l'exploitation de préférences afin de personnaliser les analyses de l'usager. Nous étudions actuellement une méthode d'acquisition automatique des préférences quantitatives à partir de l'historique des manipulations de l'usager (log). Certaines solutions sont d'ores et déjà explorées. Pour chaque opération de l'usager, le système ajoute une entrée au niveau du log qui distingue l'élément principal sélectionné (attribut de forage pour les opérations Rollu-up et Drill-down, dimension pour une rotation, prédicat de restriction pour les sélections, …) et le contexte de sa sélection représenté par l'autre partie de la requête. Les

éléments sélectionnés seront considérés comme des préférences avec un score qui est déduit du nombre de sélections. En raison du nombre important de contextes stockés, une étape de partionnement du log permettrait de déterminer des classes de contextes d'analyse puis de leur associer les éléments préférés.

Optimisation du processus de personnalisation. Le problème d'optimisation est commun à tous les travaux de personnalisation (Endres et Kießling, 2008 ; Koutrika et Ioannidis, 2005b). Notre objectif à court terme est de définir un cadre de personnalisation contraint où tous les algorithmes proposés seront soumis à des contraintes d'optimisation. Ces contraintes permettraient principalement de limiter le nombre d'accès à la méta-base au cours des itérations de sélection de préférences.

Plusieurs perspectives sont envisagées à long terme.

Système de mémoire de cache. La gestion de mémoire de cache est l'une des problématiques cruciales dans un système OLAP (Dittrich et al., 2005) à cause du lancement d'une succession de requêtes dans le cadre d'une analyse OLAP. Nous souhaiterions adapter le deuxième scénario de recommandation (par anticipation) afin de développer un système de mémoire de cache des résultats. Lors de l'exécution d'une requête, le système calcule le résultat et prévoie la prochaine requête en pré-chargeant son résultat. Ce résultat sera retourné à l'usager s'il demande dans l'étape suivante la requête recommandée ou sera utilisé pour calculer le résultat d'une autre requête sans interroger la BD (par exemple par calcul d'agrégats à partir du contenu du cache).

Système de personnalisation collaboratif. Notre approche de personnalisation exploite seulement le profil de l'usager courant. Ceci représente un problème lorsque le système ne dispose pas d'assez d'informations sur l'usager courant. L'une des perspectives les plus importantes est la définition d'un système collaboratif qui permet de personnaliser les analyses OLAP en fonction des profils d'un groupe d'usagers. Ceci nécessiterait la détermination des utilisateurs les plus proches de l'usager courant. La mise en œuvre d'une mesure de similarité pour comparer les profils des utilisateurs s'avère donc nécessaire. Une étape d'appariement d'utilisateurs sera ajoutée en amont des processus de personnalisation et de recommandation. Les algorithmes d'appariement de contexte, d'intégration de préférence et de génération de recommandation resteront inchangés, mais ils prendront en entrée des préférences d'usagers différents.

Personnalisation avancée de requête. Notre approche de personnalisation de requête est basée sur l'intégration de prédicats afin de ne retourner que les n-uplets pertinents. Il serait souhaitable de développer cette approche afin de :

- Ajouter si nécessaire au résultat des n-uplets qui ne sont pas retournés par la requête. Par exemple, il serait intéressant de renvoyer les publications dans un workshop connu en réponse à une requête qui affiche les publications dans des conférences. Cette extension représente une recommandation du contenu.
- Trier les n-uplets retournés en fonction des préférences. Le principal verrou scientifique qu'il est nécessaire de solutionner est la définition d'une fonction d'agrégation afin de calculer le score d'un n-uplet à partir des scores des différents attributs qui le composent. Par exemple, trouver le score du n-uplet (2010, 'SIG', 'conférences', 13) à partir de préférences portant sur les attributs Année, Equipe, et Type de manifestation, sur la mesure nombre de publications et sur les prédicats que

satisfont les valeurs du n-uplet. L'estimation du score d'attributs auxquels ne correspondent pas de préférences représentera une étape cruciale pour le tri.

Références

Dittrich, J-P, Kossmann, D., Kreutz, A. (2005). Bridging the gap between OLAP and SQL. Intl. Conf. on Very Large Data Bases (VLDB), pages 1031–1042.

Endres, M. Kießling, W. (2008). Optimization of Preference Queries with Multiple Constraints. Intl. Workshop on Personalized Access, Profile Management, and Context Awareness (PersDB), VLDB Workshops, pages 25–32.

***Jerbi, H.**, Pujolle, G., Ravat, F., Teste, O. (2011). Recommandation de requêtes dans les bases de données multidimensionnelles annotées. Ingénierie des Systèmes d'Information, Vol. 16, No. 1, pages 113–138*

Koutrika, G., Ioannidis, Y. (2005b). Constrained Optimalities in Query Personalization. ACM SIGMOD Intl. Conf.on Management of Data (SIGMOD), ACM Press, pages 73–64.

Bibliographie Générale

Abelló, A., Samos, J., Saltor, F. (2001). Understanding Facts in a Multidimensional Object-Oriented Model. Intl. Workshop on Data Warehousing and OLAP (DOLAP), ACM Press, pages 32–39.

Abelló, A., Samos, J., Saltor, F. (2003). Implementing operations to navigate semantic star schema. Intl. Workshop on Data Warehousing and OLAP (DOLAP), ACM Press, pages 56–62.

Abelló, A., Samos, J., Saltor, F. (2006). YAM2: a multidimensional conceptual model extending UML, Information Systems, Vol. 31, No. 6, pages 541–567.

Adomavicius, G., Tuzhilin, A. (2001). Multidimensional Recommender Systems: A Data Warehousing Approach. Intl. Workshop on Electronic Commerce (WELCOM), Lecture Notes in Computer Science, pages 180–192

Adomavicius, G., Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6, pages 734–749.

Agrawal, R., Gupta, A., Sarawagi, S. (1997). Modeling Multidimensional Databases. Intl. Conf. on Data Engineering (ICDE), pages 232–243.

Agrawal, R., Rantzau, R., and Terzi, E. (2006). Context-sensitive ranking. ACM SIGMOD Intl. Conf. on Management of Data (SIGMOD), pages 383–394.

Agrawal, R., & Wimmers, E. L. (2000). A Framework for Expressing and Combining Preferences. ACM SIGMOD Intl. Conf.on Management of Data (SIGMOD), ACM Press, pages 297–306.

Akbarnejad, J., Chatzopoulou, G., Eirinaki, M., Koshy, S., Mittal, S., On, D., Polyzotis, N., Varman, J.S.V. (2010). SQL QueRIE Recommendations. PVLDB, Vol. 3, No. 2, pages 1597–1600.

Aligon, J., Golfarelli, M., Marcel, P., Rizzi, S., Turricchia, E. (2011). Mining Preferences from OLAP Query Logs for Proactive Personalization. Conf. on Advances in Databases and Information Systems (ADBIS), Springer-Verlag, pages 84-97.

Balabanovic, M., Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. Communications of the ACM, Vol. 40, No. 3, pages 66–72.

Bellatreche, L., Giacometti, A., Marcel, P., Mouloudi, H., Laurent, D. (2005). A personalization framework for OLAP queries. Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 9–18.

Bellatreche, L., Giacometti, A., Marcel, P., Mouloudi, H. (2006). Personalization of MDX Queries. Journées Bases de Données Avancées (BDA).

Biondi, P., Golfarelli, M., Rizzi, S. (2011). Preference-based datacube analysis with MYOLAP. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 1328–1331.

Blaschka, M. Sapia, C. Höfling, G. (1999). On Schema Evolution in Multidimensional Databases. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer, pages 153–164.

Boutilier, C., Brafman, R. I., Domshlak, C., Hoos, H. H., Poole, D. (2004). CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. Journal of Artificial Intelligence Research, Vol. 21, pages 135–191.

Bouzeghoub, M., Kostadinov, D. (2005). Personnalisation de l'information : aperçu de l'état de l'art et définition d'un modèle flexible de profils. Conférence en recherche d'infomations et applications (CORIA), pages 201–218.

Bradley, K., Rafter, R., Smyth, B. (2000). Case-Based User Profiling for Content Personalisation. Intl. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems (AH), pages 62–72.

Brown, P., Bovey, J., Chen, X. (1997). Context-aware applications: From the laboratory to the marketplace. IEEE Personal Communications, Vol. 4, No. 5, pages 58–64.

Börzsönyi, S., kossmann, D., Stocker, K. (2001). The skyline operator. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 421-432

Bunningen, A. H., Feng, L., and Apers, P. M. G. (2006). A context-aware preference model for database querying in an ambient intelligent environment. Intl. Conf. on Database and Expert Systems Applications (DEXA), pages 33–43.

Burke, R. (2002). Hybrid Recommender Systems : Survey and Experiments. User Modeling and User-Adapted Interaction, Vol. 12, No. 4, pages 331–370.

Cabanac, G., Chevalier, M., Ravat, F., Teste, O. (2007). An Annotation Management System for Multidimensional Databases. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), pages 89–98.

Cabibbo, L., Torlone, R. (1997). Querying Multidimensional Databases. Intl. Workshop Database Programming Languages (DBPL), pages 319–335.

Cabibbo, L., Torlone, R. (1998). A Logical Approach to Multidimensional Databases. Intl Conf. on Extending Database Technology (EDBT), pages 183–197.

Cabibbo, L., Torlone, R. (2000). The Design and Development of a Logical System for OLAP. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), pages 1–10.

Chatzopoulou, G., Eirinaki, M., Polyzotis, N. (2009). Query recommendations for interactive database exploration. Intl. Conf. on Scientific and Statistical Database Management (SSDBM), IEEE Computer Society, pages 3–18.

Chaudhuri, S., Dayal, U. (1997). An Overview of Data Warehousing and OLAP Technology. ACM SIGMOD Record, Vol. 26, No. 1, ACM Press, pages 65–74.

Cauvet, C., Semmak, F. (1994). Abstraction Forms in Object-Oriented Conceptual Modeling: Localization, Aggregation and Generalization Extensions. Intl. Conf on Advanced Information Systems Engineering (CAiSE), pages 149–171.

Cherniack, M., Galvez, E.F., Franklin, M.J., Zdonik, S.B. (2003). Profile-Driven Cache Management. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 645–656.

Chirita, P.-A., Firan, C.S., Nejdl, W. (2007). Personalized query expansion for the Web. Intl. Conf. on Research and Development in Information Retrieval, pages 7–14.

Chomicki, J. (2003). Preference formulas in relational queries. ACM Trans. Database Syst. 28, 4, pages 427–466.

Codd, E.F., Codd, S.B., Salley, C.T. (1993). Providing OLAP (On Line Analytical Processing) to user analyst: an IT mandate. Rapport technique, E.F. Codd and associates.

Cuppens, F., Demolombe, R. (1991). Extending answers to neighbour entities in a cooperative answering context. Journal of Decision Support Systems, Vol. 7, No. 1, pages 1–11.

Datta, A., Thomas, H. (1999). The cube data model: A conceptual model and algebra for on-line analytical processing in data warehouses. Decision Support Systems (DSS), Vol. 27, No. 3, pages 289–301.

Dey, A. K. (2001). Understanding and using context. Personal and Ubiquitous Computing, Vol. 5, No. 1, pages 4–7.

Dinter, B., Sapia, C., Höfling, G., Blaschka, M. (1998). The OLAP Market: State of the Art and Research Issues. Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 22–27.

Dittrich, J-P, Kossmann, D., Kreutz, A. (2005). Bridging the gap between OLAP and SQL. Intl. Conf. on Very Large Data Bases (VLDB), pages 1031–1042.

Endres, M. Kießling, W. (2008). Optimization of Preference Queries with Multiple Constraints. Intl. Workshop on Personalized Access, Profile Management, and Context Awareness (PersDB), VLDB Workshops, pages 25–32.

Fan, J., Li, G., Zhou, L. (2011). Interactive SQL query suggestion: Making databases user-friendly. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 351–362.

Favre, C., Bentayeb, F., & Boussaid, O. (2007). Evolution of Data Warehouses' Optimization: a Workload Perspective. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer, pages 13–22.

Ferreira J., Silva A. (2001). MySDI: A Generic Architecture to Develop SDI Personalised Services. Intl. Conf. on Enterprise Information Systems (ICEIS), pages 262–270.

Franconi, E., Kamble, A. (2004). A Data Warehouse Conceptual Data Model. Intl. Conf. on Scientific and Statistical Database Management (SSDBM), pages 435–436.

Garrigós, I., Pardillo, J., Mazón, J., Trujillo, J. (2009). A Conceptual Modeling Approach for OLAP Personalization. Intl. Conf. on Conceptual Modeling (ER), pages 401–414.

Ghozzi, F. (2004). Conception et manipulation de bases de données dimensionnelles à contraintes, Thèse de doctorat, Université Paul Sabatier, Toulouse 3 (France), novembre 2004.

Giacometti, A, Marcel, P., Negre, E. (2009). Recommending Multidimensional Queries. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer, pages 453–466.

Giacometti, A., Marcel, P., Negre, E. (2008). A Framework for Recommending OLAP Queries. Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 73–80.

Golfarelli, M., Maio, D., et Rizzi, S. (1998). Conceptual design of data warehouses from E/R schemes, Intl. Conf. on System Sciences

Golfarelli, M. et Rizzi, S. (2009). Expressing OLAP Preferences. Intl. Conf. on Scientific and Statistical Database Management (SSDBM), IEEE Computer Society, pages 83–91.

Golfarelli, M. (2010). OLAP Query Personalization. Journées sur les Entrepôts de Données et l'Analyse en ligne (EDA).

Golfarelli, M., Rizzi, S., Biondi, P. (2011). myOLAP: An Approach to Express and Evaluate OLAP Preferences. IEEE Transactions on Knowledge and Data Engineering (TKDE), Vol. 23, No. 7, pages 1050–1064.

Gray, J., Bosworth, A., Layman, A., Pirahesh, H. (1996). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 152–159.

Gyssens, M., Lakshmanan, L.V.S. (1997). A foundation for multi-dimensional databases. Intl. Conf. on Very Large Data Bases (VLDB), pages 106–115.

Hafenrichter, B., Kießling, W. (2005). Optimization of Relational Preference Queries. Australasian Database Conference (ADC), pages 175–184.

Hamming, R. (1950). Error-detecting and error-correcting codes. Bell System Technical Journal, Vol. 26, pages 147–160.

Holland, S., Ester, M., Kießling,W. (2003). Preference mining: A novel approach on mining user preferences for personalized applications. European Conference on Principles and practice of Knowledge Discovery in Databases (PKDD), pages 204–216.

Holland, S., Kießling,W. (2004). Situated preferences and preference repositories for personalized database applications. In Intl. Conf. on Conceptual Modeling (ER), pages 511–523.

Hurtado, C. A., Mendelzon, A. O., Vaisman, A. A. (1999). Maintaining Data Cubes under Dimension Updates. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 346–355.

Ilyas, I. F., Beskales, G., and Soliman, M. A. (2008). A survey of top-k query processing techniques in relational database systems. ACM Computing Surveys, Vol. 40, No. 4.

Inmon W.H. (1996). Building the Data Warehouse, John Wiley and Sons, New York, NY, ISBN : 0764599445, 1996 (2ème ed.), 4ème ed. 2005.

Ioannidis, Y., Koutrika, G. (2005). Personalized systems: models and methods from an IR and DB perspective. Intl. Conf. on Very Large Data Bases (VLDB), pages 1365–1365.

***Jerbi, H.** (2007). Mémoire d'expertises décisionnelles à base d'annotations. Mémoire Master 2 Recherche, Université Paul Sabatier, Toulouse III, Juin 2007.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2008). Management of Context-aware Preferences in Multidimensional Databases. Intl. Conf. on Digital Information Management (ICDIM), IEEE, pages 669–675.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2009a). Modèle de Préférences Contextuelles pour les Analyses OLAP. Journées Francophones Extraction et Gestion de Connaissances (EGC), pages 253–258.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2009b). Applying Recommendation Technology in OLAP Systems. Intl. Conf. on Enterprise Information Systems (ICEIS), Springer, LNBIP 24, pages 220–233.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G (2009c). Preference-Based Recommendations for OLAP Analysis. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer-Verlag, pages 467–478.*

***Jerbi, H.**, Pujolle, G., Ravat, F., Teste, O. (2010a). Personnalisation de systèmes OLAP annotés. Congrès Informatique des Organisations et Systèmes d'Information et de Décision (INFORSID), pages 327–344.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2010b). Personnalisation du contenu des bases de données multidimensionnelles. Journées Francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA), pages 5–20.*

***Jerbi, H.**, Ravat, F., Teste, O., Zurfluh, G. (2010c). A Framework for OLAP Content Personalization. East European Conf. on Advances in Databases and Information Systems (ADBIS), Springer-Verlag, pages 262–277.*

***Jerbi, H.**, Pujolle, G., Ravat, F., Teste, O. (2011). Recommandation de requêtes dans les bases de données multidimensionnelles annotées. Revue des Sciences et Technologies de l'Information, série Ingénierie des Systèmes d'Information, Vol. 16, No. 1/2011, pages 113–138.*

Khoussainova, N., Balazinska, M., Gatterbauer, W., Kwon, Y., Suciu, D. (2009). A case for a collaborative query management system. Biennial Conference on Innovative Data Systems Research (CIDR) .

Kießling, W. (2002). Foundations of preferences in database systems. Intl. Conf. on Very Large Data Bases (VLDB), pages 311–322.

Kimball, R. (1996). The data warehouse toolkit: Practical Techniques for Building Dimensional Data Warehouses, John Wiley and Sons, ISBN : 0-471-15337-0, 1996, 2ème ed. : Ralph Kimball, Margaery Ross, The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd Edition, John Wiley & Sons, 2002.

Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J. (1997). GroupLens: Applying Collaborative Filtering to Usenet News. Communications of the ACM, Vol. 40, No. 3, pages 77–87.

Koutrika, G. Ioannidis, Y. E. (2004). Personalization of queries in database systems. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 597–608.

Koutrika, G., Ioannidis, Y. E. (2005a). Personalized queries under a generalized preference model. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 841–852.

Koutrika, G., Ioannidis, Y. (2005b). Constrained Optimalities in Query Personalization. ACM SIGMOD Intl. Conf.on Management of Data (SIGMOD), ACM Press, pages 73–64.

Kumar, N., Gangopadhyay, A., Karabatis, G., Bapna, S., Chen, Z. (2006). Navigation Rules for Exploring Large Multidimensional Data Cubes. Intl. Journal of Data Warehousing & Mining (IJDWM), Vol. 2, No. 4, pages 27–48.

Lehner, W. (1998). Modelling Large Scale OLAP Scenarios. Intl. Conf. on Extending Database Technology (EDBT), pages 153–167.

Levenshtein, V.I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. Rapport technique, 1966.

Li, C., Wang, X.S. (1996). A Data Model for Supporting On-Line Analytical Processing. Intl. Conf. on Information and Knowledge Management (CIKM), pages 81–88.

Li, C., Chang, K. C.-C., Ilyas, I. F. (2006). Supporting ad-hoc ranking aggregates. ACM SIGMOD Intl. Conf.on Management of Data (SIGMOD), ACM Press, pages 61–72.

Li, C., Wang, M., Lim, L., Wang, H., Chang, K. C.-C. (2007a). Supporting ranking and clustering as generalized order-by and group-by. ACM SIGMOD Intl. Conf.on Management of Data (SIGMOD), ACM Press, pages 127–138.

Li, H-G., Yu, H., Agrawal, D., El Abbadi, A. (2007b). Progressive ranking of range aggregates. and Knowledge Engineering (DKE), Elsevier Science Publishers, Vol. 63, No. 1, pages 4–25.

Liu, F.,Yu, C., Andmeng,W. (2004). Personalized Web search for improving retrieval effectiveness. IEEE Transactions on Knowledge and Data Engineering (TKDE), Vol. 16, No. 1, pages 28–40.

Loh, Z.X., Ling, T.W., Ang, C-H. Lee, S.Y. (2002). Adaptive Method for Range Top- k Queries in OLAP Data Cubes. Intl. Conf. on Database and Expert Systems Applications (DEXA), pages 648–657.

Maes, P. (1994). Agents that reduce work and information overload. Communications of the ACM, Vol. 37, No. 7, pages 31–40.

Mangisengi, O., Tjoa, A.M. (1998). A multidimensional modeling approach for OLAP within the framework of the relational model based on quotient relations. Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 40–46.

Marcel, P., Negre, E. (2011). A survey of query recommendation techniques for datawarehouse exploration. Journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA).

Miller, B., Albert, I., Lam, S., Konstan, J., Riedl, J. (2003). MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System. Intl. Conf. on Intelligent user interfaces (IUI), pages 263–266.

Mooney, R., Roy, L. (2000). Content-based Book Recommending using Learning for Text Categorization. Intl. Conf. on Digital Libraries (DL), ACM, pages 195–204.

Navarro, G. (2001). A guided tour to approximate string matching. ACM Comput. Surv., Vol. 33, No. 1, pages 31–88.

Okasaki, C. (2000). Breadth-first numbering: lessons from a small exercise in algorithm design. In Proceedings of the 2000 ACM SIGPLAN Intl. Conf on Functional Programming, Vol. 35, No. 9, pages 131–136.

Pavlov, D., Manavoglu, E., Pennock, D., Giles, C. (2004). Collaborative Filtering with Maximum Entropy. IEEE Intelligent Systems, Vol. 19, No. 6, pages 40–48.

Pazzani, M., Billsus, M. (2007). Content-Based Recommendation Systems. The Adaptive Web, pages 325–341.

Pedersen, T.B., Jensen, C.S., Dyreson, C. E. (2001). A foundation for capturing and querying complex multidimensional data. Information Systems (IS), Vol. 26, No. 5, Elsevier, pages 383–423.

Pitkow, J., Pirolli, P. (1999). Mining Longest Repeating Subsequences to Predict World Wide Web Surfing. Intl. Conf. on USENIX Symposium on Internet Technologies and Systems (USITS), pages 139–150.

Pitkow, J. E., Schutze, H., Cass, T. A., Cooley, R., Turnbull, D., Edmonds, A., Adar, E., Breuel, T. M. (2002). Personalized Search. Communications of the ACM, Vol. 45, No. 9, pages 50–55.

Rafanelli, M. (2003). Operators for Multidimensional Aggregate Data. Chapitre V, Multidimensional Databases: Problems and Solutions, IGI Publishing Group, ISBN 1-59140-053-8, pages 116–165.

Ravat F., Teste O., Zurfluh G. (2007a). Personnalisation de bases de données multidimensionnelles. Congrès INFormatique des ORganisations et Systèmes d'Information et de Décision (INFORSID), pages 121–136.

Ravat F., Teste O., Tournier R., Zurfluh G. (2007b). Querying Multidimensional Databases. Conf. on Advances in Databases and Information Systems (ADBIS), Springer-Verlag, pages 298–313.

Ravat F., Teste O. (2008). Personalization and OLAP Databases. Annals of Information Systems, Vol. 3, Numéro spécial “New Trends in Data Warehousing and Data Analysis”, pages 1–22.

Ravat F., Teste O., Tournier R., Zurfluh G. (2008). Algebraic and graphic languages for OLAP manipulations. Intl. Journal of Data Warehousing and Mining (IJDWM), Vol. 4, No. 1, pages 17–46.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews, ACM Conf. on Cumputer-Supported Cooperative Work, pages 175–186.

Rizzi S. (2007). OLAP preferences: a research agenda. Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 99–100.

Rizzi, S., Abelló, A., Lechtenbörger, J., Trujillo, J. (2006). Research in data warehouse modeling and design: dead or alive? Intl. Workshop on Data Warehousing and OLAP (DOLAP), pages 3–10.

Romero, O., Abelló, A. (2007). On the Need of a Reference Algebra for OLAP. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer, pages 99–110.

Sapia, C. (2000). PROMISE : Predicting Query Behavior to Enable Predictive Caching Strategies for OLAP Systems. Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), Springer, pages 224–233.

Sapia, C. (1999). On Modeling and Predicting Query Behavior in OLAP Systems. Intl. Workshop on Design and Management of Data Warehouses (DMDW), CAISE Workshops.

Sarawagi, S. (1999). Explaining differences in multidimensional aggregates. Intl. Conf. on Very Large Data Bases (VLDB), pages 42–53.

Sarawagi, S. (2000). User-adaptive exploration of multidimensional data. Intl. Conf. on Very Large Data Bases (VLDB), pages 307–316.

Sarawagi, S., Agrawal, R., Megiddo, N. (1998). Discovery-driven exploration of OLAP data cubes. Intl. Conf. on Extending Database Technology (EDBT), pages 168–182.

Sathe, G., Sarawagi, S. (2001). Intelligent rollups in multidimensional OLAP data. Intl. Conf. on Very Large Data Bases (VLDB), pages 531–540.

Satzger, B., Endres, M., Kießling, W. (2006). A Preference-Based Recommender System. Intl. Conf. on Electronic Commerce and Web Technologies (EC-Web), Springer, Heidelberg, pages 31–40.

Schmidt, A., Aidoo, A. K., Takaluoma, A., Tuomela, U., Laerhoven, K., and de Velde, M. (1999). Advanced interaction in context. Intl. Symposium on Handheld and Ubiquitous Computing, pages 89–101.

Schneider, M. (2003). Well-formed data warehouse structures. Intl. Workshop on Design and Management of Data Warehouses (DMDW), CAISE Workshops.

Stolte, C. (2003). Query, Analysis, and Visualization of Multidimensional Databases, Thèse de doctorat, Université de Stanford (Etats-Unis), Juin 2003.

Soltysiak S., Crabtree B. (1998). Automatic learning of user profiles - Towards the personalisation of agent services. BT Technology Journal, Vol 16, No 3, pages 110–117.

Stefanidis, K., Pitoura, E., Vassiliadis, P. (2007). Adding context to preferences. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 846–855.

Stefanidis, K., Drosou, M., Pitoura, E. (2009). "You May Also Like" Results in Relational Databases. Intl. Workshop on Personalized Access, Profile Management and Context Awareness in Databases (PersDB), VLDB Workshops.

Stefanidis, K., Koutrika, G., Pitoura, E. (2011). A Survey on Representation, Composition and Application of Preferences in Database Systems. ACM Transactions on Database Systems (TODS), Vol. 36, No. 3.

Sun, Y., Li, H., Councill, I.G, Huang, J., Lee, W-C., Giles, C. L. (2008). Personalized ranking for digital libraries based on log analysis. Intl. Workshop on Web Information and Data Management (WIDM), ACM, pages 133–140.

Thalhammer, T., Schrefl, M., Mohania, M. (2001). Active DataWarehouses. Complement-ing OLAP with Analysis Rules, Data and Knowledge Engineering (DKE), Elsevier Science Publishers, Vol. 39, No. 3, pages 241–269.

Torlone, R. (2003). Conceptual Multidimensional Models. Chapitre 3 de l'ouvrage Multidimensional Databases: Problems and Solutions, IGI Publishing Group(IGP), ISBN 1-59140-053-8, pages 69–90.

Vassiliadis, P., Simitsis, A., Skiadopoulos, S. (2002). Modeling ETL activities as graphs. Intl. Workshop on Design and Management of Data Warehouses (DMDW), CAISE Workshops, pages 52–61.

Xin, D., Han, J. (2008). P-cube: Answering preference queries in multidimensional space. Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, pages 1092–1100.

Zhang, Y., Callan, J., Minka, T. (2002). Novelty and Redundancy Detection in Adaptive Filtering. Intl. Conf. on Research and development in information retrieval (SIGIR), ACM, pages 81–88.

Annexes

Annexe 1. Algèbre de manipulation OLAP

Définition. Une table multidimensionnelle T est définie par (S, L, C, R) où

- S = (F, {$f_1(m_1)$, $f_2(m_2)$,…}) représente le sujet d'analyse relatif au fait F et ses mesures observées {m_1, m_2,…} agrégées à l'aide de fonctions d'agrégation f_1, f_2,…
- L = (D^L, H^L, P^L) représente l'axe d'analyse en ligne de T au travers d'une dimension courante D^L, d'une hiérarchie courante H^L et d'une liste ordonnée de paramètres affichés P^L = <All, p^{HL}_{max},…p^{HL}_{min}>, $H^L \in H^{DL}$ et $D^L \in Star^{CS}(F)$
- C = (D^C, H^C, P^C) représente l'axe d'analyse en colonne de la table T au travers d'une dimension courante D^C, d'une hiérarchie courante H^C et d'une liste ordonnée de paramètres affichés P^C = <All, p^{HC}max,…p^{HC}min>, $H^C \in H^{DC}$ et $D^C \in Star^{CS}(F)$
- R = $pred_1 \wedge \ldots \wedge pred_v$ est le prédicat de restriction composé d'une conjonction de prédicats normalisés portant sur les dimensions et/ou sur le fait F.

Constructeur.

Le constructeur est essentiel car il permet de définir à partir d'une constellation une première TM initiant le processus exploratoire OLAP. Une TM est obtenue à partir d'un constructeur défini comme suit.

$$\mathbf{DISPLAY}(F_c, \{f_1(m_1),\ldots,f_t(m_t)\}, DL, HL, DC, HC) = T_{RES}$$

- F_c est le sujet de l'analyse représenté par le fait,
- $\{f_1(m_1),\ldots,f_t(m_t)\}$ est l'ensemble des mesures $\forall i \in [1..t]$, $m_i \in M_c \wedge f_i \in$ {SUM, AVG, MIN, MAX, COUNT,…},
- $DL \in Star(F_c)$ est l'axe d'analyse en ligne,
- $HL \in H_{DL}$ est la hiérarchie courante utilisée pour graduer les lignes,
- $DC \in Star(F_c)$ est l'axe d'analyse en colonne,
- $HC \in H_{DC}$ est la hiérarchie courante utilisée pour graduer les colonnes,
- $T_{RES} = (S_{RES} ; L_{RES}; C_{RES} ; R_{RES})$ est la TM résultat tel que
  - $S_{RES} = (F_c, \{f_1(m_1),\ldots,f_t(m_t)\})$,
  - $L_{RES} = (DL, HL, <All, p_{Lmax}>)$,
  - $C_{RES} = (DC, HC, <All, p_{Cmax}>$ *et*
  - $R_{RES} = \bigwedge_{\forall D_i \in Star(F_c)} D_i.All = 'all'$.

Opérations de manipulation

Chaque opérateur :

- porte en entrée sur une TM source notée $T_{SRC} = (S_{SRC} ; L_{SRC}; C_{SRC} ; R_{SRC})$, et
- produit en sortie une TM résultat notée $T_{RES} = (S_{RES} ; L_{RES}; C_{RES} ; R_{RES})$.

On pose :

- $S_{SRC} = (F_c, M_c\})$, $M_c = \{f_1(m_1),\ldots,f_t(m_t)\}$
- $L_{SRC} = (DL, HL, PL)$, $PL = <All, p_{Lmax},\ldots, p_{Lmin}>$
- $C_{SRC} = (DC, HC, PC)$, $PC = <All, p_{Cmax},\ldots, p_{Cmin}>$
- $R_{SRC} = \bigwedge_{\forall D_i \in Star(F_c)} D_i.p_j \theta v_k$

Dans une TM, chaque paramètre a son domaine de définition ordonné ; on notera dom(p_i) = <v_1, v_2, …>.

DRILLDOWN(T_{SRC}, D, p_{inf}) = T_{RES}

Conditions : $D \in \{DL; DC\}$;

$D = DL \Rightarrow p_{inf} \in HL \wedge \nexists p_k \in PL \mid \text{level}_{HL}(p_k) < \text{level}_{HL}(p_{inf})$ (1)

$D = DC \Rightarrow p_{inf} \in HC \wedge \nexists p_k \in PC \mid \text{level}_{HC}(p_k) < \text{level}_{HC}(p_{inf})$

Résultat : $T_{RES} = (S_{SRC}; L_{RES}; C_{RES}; R_{SRC})$

$D = DL \Rightarrow L_{RES} = (DL, HL, <All, p_{Lmax}, \ldots, p_{Lmin}, p_{inf}>) \wedge C_{RES} = C_{SRC}$

$D = DC \Rightarrow L_{RES} = L_{SRC} \wedge C_{RES} = (DC, HC, <All, p_{Cmax}, \ldots, p_{Cmin}, p_{inf}>)$

ROLLUP(T_{SRC}, D, p_{sup}) = T_{RES}

Conditions : $D \in \{DL; DC\}$;

$D = DL \Rightarrow p_{sup} \in HL \mid \text{level}_{HL}(p_{Lmin}) < \text{level}_{HL}(p_{sup})$

$D = DC \Rightarrow p_{sup} \in HC \mid \text{level}_{HC}(p_{Cmin}) < \text{level}_{HC}(p_{sup})$

Résultat : $T_{RES} = (S_{SRC}; L_{RES}; C_{RES}; R_{SRC})$

$D = DL \Rightarrow L_{RES} = (DL, HL, <All, p_{Lmax}, \ldots, p_{sup}>) \wedge C_{RES} = C_{SRC}$

$D = DC \Rightarrow L_{RES} = L_{SRC} \wedge C_{RES} = (DC, HC, <All, p_{Cmax}, \ldots, p_{sup}>)$

ROTATE(T_{SRC}, D_{old}, D_{new}, H_{new}) = T_{RES}

Conditions : $D_{old} \in \{DL; DC\}$; $D_{new} \in Star(F_c)$; $H_{new} \in H_{new}$

Résultat : $T_{RES} = (S_{SRC}; L_{RES}; C_{RES}; R_{SRC})$

$D_{old} = DL \Rightarrow L_{RES} = (D_{new}, H_{new}, <All>) \wedge C_{RES} = C_{SRC}$

$D_{old} = DC \Rightarrow L_{RES} = L_{SRC} \wedge C_{RES} = (D_{new}, H_{new}, <All>)$

NEST(T_{SRC}, D, p_i, D_{new}, p_{new}) = T_{RES}

Conditions : $D \in \{DL; DC\}$; $D_{new} \in Star(F_c)$; $p_{new} \in A_{new}$

$D = DL \Rightarrow p_i \in PL$

$D = DC \Rightarrow p_i \in PC$

Résultat : $T_{RES} = (S_{SRC}; L_{RES}; C_{RES}; R_{SRC})$

$D = DL \Rightarrow L_{RES} = (DL, HL, <All, p_{Lmax}, \ldots, p_i, p_{new}, \ldots, p_{Lmin}>) \wedge C_{RES} = C_{SRC}$

$D = DC \Rightarrow L_{RES} = L_{SRC} \wedge C_{RES} = (DC, HC, <All, p_{Cmax}, \ldots, p_i, p_{new}, \ldots, p_{Cmin}>)$

AGREGATE(T_{SRC}, D, $f(p_i)$) = T_{RES}

Conditions : $D \in \{DL; DC\}$; $f \in \{\text{SUM, COUNT, MAX, MIN,…}\}$

$D = DL \Rightarrow p_i \in PL$

$D = DC \Rightarrow p_i \in PC$

Résultat : $T_{RES} = (S_{SRC}; L_{RES}; C_{RES} ; R_{SRC})$

$D = DL \Rightarrow L_{RES} = L_{SRC}$ où $\text{dom}(p_i) = <v_1, f(v_1), v_2, f(v_2), \ldots> \wedge C_{RES} = C_{SRC}$

$D = DC \Rightarrow L_{RES} = L_{SRC} \wedge C_{RES} = C_{SRC}$ où $\text{dom}(p_i) = <v_1, f(v_1), v_2, f(v_2), \ldots>$

SWITCH(T_{SRC}, D, p_i, v_1, v_2) = T_{RES}

Conditions : $D \in \{DL; DC\}$; $(v_1, v_2) \in \text{dom}(p_i)^2 | \text{dom}(p_i) = <\ldots, v_1, \ldots, v_2, \ldots>$

Résultat : $T_{RES} = (S_{SRC}; L_{RES}; C_{RES} ; R_{SRC})$

$D = DL \Rightarrow L_{RES} = L_{SRC}$ où $\text{dom}(p_i) = <\ldots, v_2, \ldots, v_1, \ldots> \wedge C_{RES} = C_{SRC}$

$D = DC \Rightarrow L_{RES} = L_{SRC} \wedge C_{RES} = C_{SRC}$ où $\text{dom}(p_i) = <\ldots, v_2, \ldots, v_1, \ldots>$

PULL(T_{SRC}, D, $f(m_i)$) = T_{RES}

Conditions : $D \in \{DL; DC\}$; $f(m_i) \in M_c$

Résultat : $T_{RES} = (S_{RES}; L_{RES}; C_{RES} ; R_{SRC})$

$S_{RES} = (F_c, M_c \backslash \{f(m_i)\})$ (2)

$D = DL \Rightarrow L_{RES} = (DL, HL, PL \oplus <f(m_i)>)$ (3) $\wedge C_{RES} = C_{SRC}$

$D = DC \Rightarrow L_{RES} = L_{SRC} \wedge C_{RES} = (DC, HC, PC \oplus <f(m_i)>)$

PUSH(T_{SRC}, D, p_i) = T_{RES} (2)

Conditions : $D \in \{DL; DC\}$;

$D = DL \Rightarrow p_i \in PL$

$D = DC \Rightarrow p_i \in PC$

Résultat : $T_{RES} = (S_{RES}; L_{RES}; C_{RES} ; R_{SRC})$

$S_{RES} = (F_c, M_c \cup \{p_i\})$

$D = DL \Rightarrow L_{RES} = (DL, HL, PL \backslash \{p_i\}) \wedge C_{RES} = C_{SRC}$

$D = DC \Rightarrow L_{RES} = L_{SRC} \wedge C_{RES} = (DC, HC, PC \backslash \{p_i\})$

SELECT(T_{SRC}, *pred*) = T_{RES}

Conditions : $pred = pred_1 \wedge \ldots \wedge pred_t$ est un prédicat en forme conjonctive normale ;

$\forall s \in [1..t]$, $pred_s = E_i.A_j \ \theta \ v_k$ où $E_i \in \{F_c\} \cup Star(F_c)$, $A_j \in M \cup A$, $\theta \in \{= ; < ; \leq ; > ; \geq ; \neq\}$, $v_k \in \text{dom}(A_j)$

Résultat : $T_{RES} = (S_{SRC}; L_{SRC}; C_{SRC} ; R_{RES})$

$R_{RES} = \text{pred}$

ADDM(T_{SRC}, $f(m_i)$) = T_{RES}

Conditions : $m_i \in M_c$; $f \in \{\text{SUM, COUNT, MAX, MIN,…}\}$; $f(m_i) \notin M_c$

Résultat : $T_{RES} = (S_{RES}; L_{SRC}; C_{SRC} ; R_{SRC})$

$S_{RES} = (F_c, M_c \cup \{f(m_i)\})$

DELM(T_{SRC}, $f(m_i)$) = T_{RES}

Conditions : $f(m_i) \in M_c$

Résultat : $T_{RES} = (S_{RES}; L_{SRC}; C_{SRC} ; R_{SRC})$

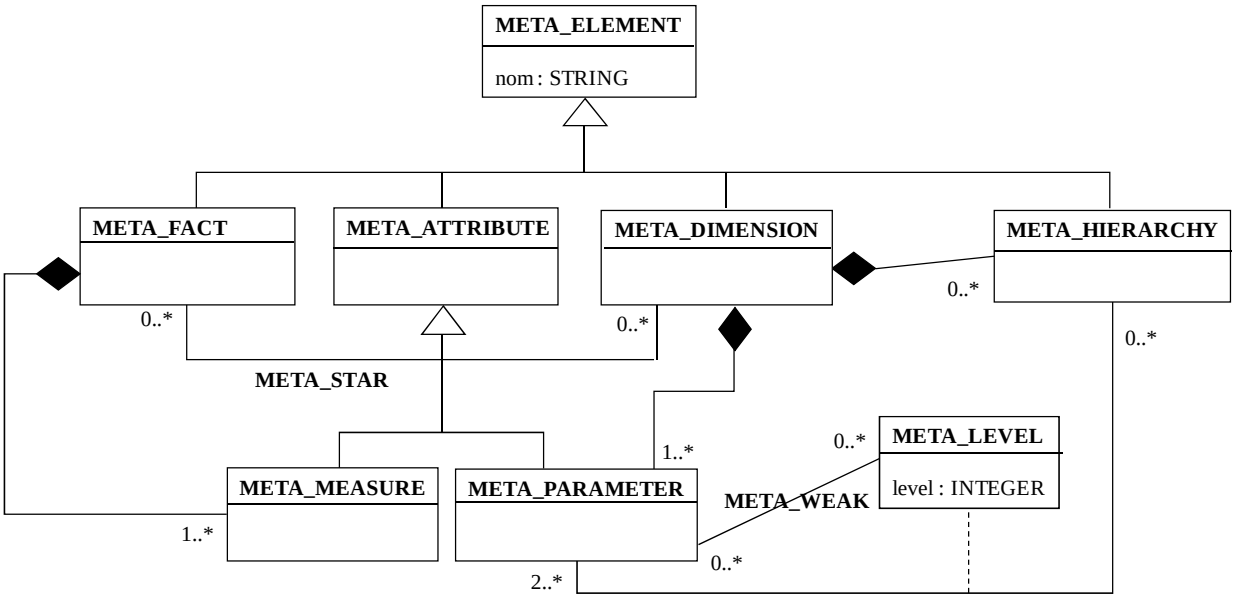
$S_{RES} = (F_c, M_c \setminus \{f(m_i)\})$

(1) $level_H(p)$ est une fonction retournant l'indice de position (entier) du paramètre p dans la hiérarchie H.

(2) $\setminus$ est l'opération de différence d'ensembles : $\{e_1, e_2, e_3\} \setminus \{e_2, e_4\} = \{e_1, e_3\}$

(3) $\oplus$ est l'opération de concaténation de listes : $\langle e_1, e_2, e_3 \rangle \oplus \langle e_4 \rangle = \langle e_1, e_2, e_3, e_4 \rangle$

Annexe 2. Schéma conceptuel (simplifié) de la méta-base (représenté au format UML)



Résumé (court)

Le travail présenté dans cette thèse aborde la problématique de la personnalisation des analyses OLAP au sein des bases de données multidimensionnelles.

Une analyse OLAP est modélisée par un graphe dont les nœuds représentent les contextes d'analyse et les arcs traduisent les opérations de l'utilisateur. Le contexte d'analyse regroupe la requête et le résultat. Il est décrit par un arbre spécifique qui est indépendant des structures de visualisation des données et des langages de requête. Par ailleurs, nous proposons un modèle de préférences utilisateur exprimées sur le schéma multidimensionnel et sur les valeurs. Chaque préférence est associée à un contexte d'analyse particulier. En nous basant sur ces modèles, nous proposons un cadre générique comportant deux mécanismes de personnalisation. Le premier mécanisme est la personnalisation de requête. Il permet d'enrichir la requête utilisateur à l'aide des préférences correspondantes afin de générer un résultat qui satisfait au mieux aux besoins de l'usager. Le deuxième mécanisme de personnalisation est la recommandation de requêtes qui permet d'assister l'utilisateur tout au long de son exploration des données OLAP. Trois scénarios de recommandation sont définis : l'assistance à la formulation de requête, la proposition de la prochaine requête et la suggestion de requêtes alternatives. Ces recommandations sont construites progressivement à l'aide des préférences de l'utilisateur. Afin valider nos différentes contributions, nous avons développé un prototype qui intègre les mécanismes de personnalisation et de recommandation de requête proposés. Nous présentons les résultats d'expérimentations montrant la performance et l'efficacité de nos approches.

Mots-clés: OLAP, analyse décisionnelle, personnalisation de requête, système de recommandation, préférence utilisateur, contexte d'analyse, appariement d'arbres de contexte.

Abstract

This thesis investigates OLAP analysis personalization within multidimensional databases. OLAP analyse is modeled through a graph where nodes represent the analysis contexts and graph edges represent the user operations. The analysis context regroups the user query as well as result. It is well described by a specific tree structure that is independent on the visualization structures of data and query languages. We provided a model for user preferences on the multidimensional schema and values. Each preference is associated with a specific analysis context. Based on previous models, we proposed a generic framework that includes two personalization processes. First process, denoted query personalization, aims to enhancing user query with related preferences in order to produce a new one that generates a personalized result. Second personalization process is query recommendation that allows helping user throughout the OLAP data exploration phase. Our recommendation framework supports three recommendation scenarios, *i.e.*, assisting user in query composition, suggesting the forthcoming query, and suggesting alternative queries. Recommendations are built progressively basing on user preferences.

In order to implement our framework, we developed a prototype system that supports query personalization and query recommendation processes. We present experimental results showing the efficiency and the effectiveness of our approaches.

Keywords: OLAP, decision-support analysis, query personalization, recommender system, user preference, analysis context, context trees matching.