

MA-004-127-1

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD DAHLAB DE BLIDA



FACULTE DES SCIENCES

DEPARTEMENT D'INFORMATIQUE

MEMOIRE DE FIN D'ETUDE POUR L'OBTENTION Du Diplôme de master en
INFORMATIQUE

THEME

INTÉGRATION DE LA CONTEXTUALISATION DANS LE
PROCESSUS DE PERSONNALISATION DES SYSTÈMES OLAP

Présenté par :

Mme Zedek Anissa

Promotrice :

Mme N.Rezoug

Soutenue le : 22/09/2012

devant le jury composé de :

- Mme M. Fareh
- Mme D. Boumehdi
- Mme FZ. Zahra

Président
Examineur
Examineur

MA-004-127-1

Promotion 2011/2012

Remerciements

Je tiens dans un premier temps à remercier ma promotrice Mme Nachida Rezoug pour m'avoir confié ce travail de recherche, sa confiance qu'elle a mise en ma personne ainsi que son aide précieuse tout au long du déroulement de ce mémoire.

Je n'oublie pas tous mes professeurs sans qui nous ne serions pas arrivés là où nous sommes aujourd'hui grâce à leur dévouement leur pédagogie et leur bienveillance.

Je dis merci aussi au membre de notre administration pour leur travail qui n'est pas toujours estimé à sa juste valeur.

Un grand merci à vous tous pour toute contribution qui m'a permis d'arriver à présenter aujourd'hui ce travail à votre jugement et être évalué et je l'espère obtenir mon diplôme de master en informatique en spécialité de l'ingénierie du logiciel.

Dédicace

Je dédie ce travail d'abord à mon cher époux Mhamed qui a été à mes côtés tout au long de cette période, qui m'aura énormément fait rire, m'aura soutenu dès le début et m'aura aidé dans les périodes de doute.

Je dédie aussi ce mémoire à ma mère Hafedha, à la femme courageuse qu'elle est, à tous les moments où elle s'est retrouvée seul à nous élever mais où elle a continué à nous apporter tout l'amour et le soutien dont on avait besoins pour continuer notre chemin. Que dieu nous la garde encore longtemps.

A ma sœur, ou dirai je à ma meilleur amie Nesrine, au petit bout de femme qu'elle est devenue et sur qui je peux compter à n'importe quelle épreuve.

Ainsi qu'à mon père Mahfoudh paix a son âme mais qui j'espère me vois tout la haut et est fier de moi.

A mes beaux-parents, ou plutôt : grands-parents : mamaZhor et papa Hassin, parents : mamaHassiba et tonton Rochdi, frère : Saad Eddine, Nabil, Riad et sœurs : Djawida, lila, pour leurs appels quotidien pour demander de mes nouvelles et leurs encouragements précieux. Ainsi qu'à mes neveux et nièces qui ont eux aussi participer par leurs attentions envers leur tata nissa.

Spécialement à ImeneTouari sans qui je crois que je n'aurais pas eu la chance d'être en face de vous aujourd'hui.

A tous mes cousins, cousines, tentes et oncles : Amel, Sabiha, faiza, Djaafar, tata Lila, chehrazade, Tarik, Samira, Maha, Raouf, et j'enoublie surement milles excuse ; pour une présence incomparable.

A mon amie Yousra qui j'espère, tout se passera bien pour elle et je lui souhaite tout le bonheur du monde.

Merci à vous tous et j'espère ne pas vous décevoir.

Résumé

Les systèmes OLAP (Online Analytical Processing) sont des applications informatiques orientées vers l'analyse de manière interactive selon différents axes d'analyse. Aujourd'hui OLAP permet aux décideurs de solliciter des traitements d'analyse qui facilitent leurs tâches quotidiennes de prise de décision. Mais ces derniers se trouvent incapables de distinguer l'information qui leur est pertinente de celle qui ne l'est pas. Et les systèmes actuels s'avèrent inadaptés à ces exigences.

Dans ce mémoire nous nous intéressons à la personnalisation des analyses OLAP et plus précisément à la conception et réalisation d'un système recommandé personnalisé ; ainsi d'une part on aura un système qui va présenter un ensemble de requêtes adapté au besoin analytique individuel et d'une autre part à recommander le résultat jugé pertinent pour chaque utilisateur. Une fois encore la prise en compte des préférences de l'utilisateur s'avère insuffisante c'est pourquoi l'intégration de l'information contextuelle dans le processus d'analyse de la requête est indispensable. Ce qui nous mène à la mise en place d'un système recommandé personnalisé contextualisé.

Nous proposons un cadre générique pour la prise en compte de l'utilisateur dans le processus d'analyse OLAP qui va considérer cette analyse comme une exploration interactive des données basées sur une succession de requêtes, où les préférences de l'utilisateur sont déduites de manière implicite et le contexte sera défini par l'utilisateur lui-même (de manière explicite).

Une étape de matching va précéder l'étape de recommandation, elle va permettre d'effectuer des combinaisons entre le contexte et les préférences utilisateurs pour générer des préférences contextualisées, qui vont être la base de l'étape suivante. Pour cela nous avons utilisé un cadre de références de requêtes MDX pour leur facilité de manipulation par rapport aux autres types de requêtes. La recommandation va être construite progressivement par l'appariement du contexte d'analyse courant avec les préférences contextualisées déjà obtenues précédemment.

Mots clés : analyse, personnalisation de la requête, recommandation de la requête, préférences utilisateurs, contexte d'analyse.

Summary

The OLAP systems (Online Analytical Processing) are computer applications oriented towards an interactive way of analysis and following different axes of analysis.

However due to big data mass that OLAP processes and consequently due to the important number of responses to the multiple requests submitted as well as the users' interactions, the latter cannot really distinguish which answer is relevant and which is not. And the current systems are not suitable for these requirements.

In this paper we are interested in the personalization of the OLAP analyses and more precisely the design and realization of a recommended and personalised system. Once more, taking into consideration the user's preferences has revealed to be insufficient and this is why the integration of the contextual information (of the context) in the analytical process is indispensable. This leads to the implementation of a recommended, personalised and contextualised system.

We suggest a generic frame when considering the user in the OLAP analysis process. This analysis is considered as an interactive exploration of data based on the succession of requests where the preferences are deduced in an implicit way then the context which will be defined by the user (in an explicit way). A matching stage will precede the recommendation stage; this will allow the creation of a set of combinations between the context and the users' preferences and then will generate a set of contextualised preferences that will be the next stage basis, for this we used a field of references of requests MDX since they are easier to use in comparison to the other type of requests. The recommendation will be built progressively by matching the context of the current analysis with the contextualized preferences already obtained before.

Key words: Analysis, personalization of request, recommendation of request, users' preferences, context of analysis.

Table des matières :

Chapitre1 : généralité sur les entrepôts de données et les systèmes OLAP

I.	Introduction.....	13
II.	Définition d'un entrepôt de données.....	14
1.	Caractéristique d'un entrepôt de données.....	14
2.	Architecture conceptuelle d'un data warehouse.....	15
3.	En amont et en aval du datawarehouse.....	16
4.	Les types de données utilisées dans l'entrepôt de données.....	18
4.1.	Données détaillées.....	18
4.2.	Données agrégées.....	18
4.3.	Les métadonnées.....	19
4.4.	Les datamarts ou magasin de données.....	19
4.5.	Les cubes de données.....	20
5.	L'approche multidimensionnelle dans le datawarehouse.....	20
III.	Définition des systèmes OLAP.....	21
1.	Les outils de navigation OLAP.....	22
2.	Les différentes implémentations OLAP.....	23
2.1.	ROLAP.....	23
2.2.	MOLAP.....	24
2.3.	HOLAP.....	24
2.4.	DOLAP.....	24
3.	Les différents modèles de données.....	25
3.1.	Le modèle en étoile.....	25
3.2.	Le modèle en flocons.....	25
4.	La requête OLAP.....	25
IV.	Conclusion.....	26

Chapitre2 : La personnalisation dans les systèmes OLAP

I.	Introduction.....	27
II.	Définition de la personnalisation.....	27

1.	Les principes mis en jeu dans la personnalisation.....	27
1.1.	Profil utilisateur.....	27
1.2.	Le contexte utilisateur.....	28
1.3.	Les préférences utilisateurs.....	28
2.	Etat de l'art.....	29
III.	Synthèse et discussion.....	31
IV.	Conclusion.....	32

Chapitre3 : Les systèmes recommandés

I.	Introduction.....	33
II.	Définition des systèmes recommandés.....	33
1.	Principe de la recommandation.....	34
1.1.	La recherche d'information.....	34
1.2.	Le filtrage d'information.....	34
2.	Les types de recommandations.....	34
2.1.	Le filtrage basé sur le contenu.....	34
2.2.	Le filtrage collaboratif.....	35
2.3.	Le filtrage hybride.....	35
3.	Etat de l'art.....	36
III.	Conclusion.....	37

Chapitre 4 : La contextualisation

I.	Introduction.....	38
II.	Définition du contexte.....	38
III.	Les systèmes recommandés VS les systèmes recommandés contextualisés	39
IV.	Les méthodes d'intégration de l'information contextuelle.....	40
1.	Le pré-filtrage contextuel.....	41
2.	Post –filtrage contextuel.....	41
3.	La modélisation contextuelle.....	41
V.	Caractéristique des systèmes recommandés contextualisés CARS.....	42
1.	Complexité.....	42
2.	L'interactivité.....	42
3.	La flexibilité.....	42
VI.	Etat de l'art.....	43

VII. Conclusion	43
-----------------------	----

Chapitre 5 : Présentation du système

I. Introduction.....	44
II. Architecture du système	44
✚ Les fonctions offline.....	45
✚ Les fonctions online.....	46
1. Le matching.....	46
2. La contextualisation.....	47
3. La recommandation.....	48
4. Le profil et le contexte.....	49
4.1. Le méta modèle de profil.....	49
4.2. Le méta modèle de contexte.....	50
5. Le scénario d'analyse de la requête.....	50
6. Qu'est-ce qu'une requête MDX.....	52
6.1. Type de données MDX.....	52
III. Conclusion	53

Chapitre 6 :L'implémentation

I. Introduction.....	55
II. Présentation de SQL Serveur 2008.....	55
III. Présentation de Visuel Studio 2008 Expresse.....	56
IV. Modules et type de données utilisées.....	57
1. .L'utilisation du Business Intelligence Development Studio.....	57
2. La base de données AdventureWorksDW2008.....	57
3. Création d'une base Analysis Services.....	58
4. La création du cube de données.....	59
5. La partie lancement de la requête.....	60
6. La détermination du contexte.....	61
7. La recommandation.....	61
V. Conclusion.....	61
Conclusion et Perspectives.....	62
Bibliographie	64

Table des figures :

1. Figure(1) : caractéristique d'un Datawarehouse.
2. Figure(2) : cycle d'un Datawarehouse actif.
3. Figure(3) : le concept de Datamart.
4. Figure(4) : exemple de cube de données.
5. Figure(5) : principe de la personnalisation.
6. Figure(6) : tableaux comparatif des travaux de personnalisation.
7. Figure(7) : classification de recherches effectuées dans le domaine des systèmes recommandés.
8. Figure(8) : méthodes d'intégration du contexte dans les systèmes recommandés.
9. Figure(9) : l'architecture globale du système.
10. Figure (10): méta modèle de profil.
11. Figure (11) : méta modèle de contexte.
12. Figure (12) : schéma d'organisation des tables.
13. Figure (13) : création du cube.
14. Figure (14) : lancement de la requête.
15. Figure (15) : choix du contexte.

Introduction général :

Les systèmes décisionnels actuels sont un outil indispensable aux décideurs et utilisateurs dans différents domaines tels que le domaine commercial, financier, ou autre. Car il facilite le processus de prise de décision et permet des solutions précises, fiables et adéquates aux exigences et besoins des utilisateurs finaux.

Ce processus de prise de décision, est en perpétuel développement car face à la concurrence il faut toujours plus de rapidité et de précision.

Ce qui est le cas avec la mise en place de la personnalisation de l'information, qui a fait ses débuts dans la recherche d'information (RI), celle-ci intègre l'utilisateur dans le processus de personnalisation, ce qui permet un apprentissage à partir des préférences déterminées par l'utilisateur même ou déduites implicitement par le système. Ensuite vient le tour des bases de données (BD), où il n'est pas courant de mêler l'utilisateur au processus de recherche. Donc la personnalisation est souvent faite par l'extension des langages de requêtes pour pouvoir supporter les expressions de préférences.

C'est pourquoi les systèmes décisionnels proposent d'enregistrer les préférences qui se répètent au cours des interactions répétées de l'utilisateur avec le système, et les utiliser par la suite dans l'analyse de la requête, cela pour obtenir des résultats qui soient en adéquation avec les préférences des utilisateurs.

Mais cependant, la prise en considération de ces préférences n'est pas suffisante pour délivrer des résultats pertinents. Le contexte dans lequel se trouve l'utilisateur au moment de l'interaction est tout aussi important.

C'est pourquoi la contextualisation est un domaine émergent qui sollicite beaucoup d'intérêt et de réflexions sur la méthode d'intégration de l'information contextuelle dans le processus de personnalisation.

Dans notre travail nous allons justement nous intéresser à l'intégration de l'information contextuelle, car notre but est de proposer un système personnalisé, recommandé, contextualisé.

Ce travail consiste d'une part à personnaliser, recommander ce qui veut dire un système qui présente un ensemble de requêtes conçus selon les besoins analytique individuelles de chaque utilisateur et d'une autre part à intégrer l'information contextuelle dans le processus de personnalisation de la requête.

Notre plan de présentation commencera par le premier chapitre qui sera une présentation de généralités sur les entrepôts de données et les systèmes OLAP ensuite un deuxième chapitre qui va s'intéresser à la personnalisation dans les entrepôts de données, un troisième sur les systèmes recommandés, un quatrième qui abordera la contextualisation, puis le chapitre cinq qui va présenter l'architecture du système choisis et enfin le dernier celui qui va aborder la réalisation.

Chapitre1 : Généralités sur les entrepôts de données et les systèmes OLAP

I. Introduction :

Un datawarehouse (ou entrepôt de données) est un serveur informatique dans laquelle est centralisé un volume important de données qui proviennent de différentes sources de renseignements d'une entreprise (notamment les bases de données internes). L'organisation des données est conçue pour que les personnes intéressées aient accès rapidement et sous forme synthétique à l'information stratégique dont elles ont besoin pour la prise de décision.

Si un entrepôt de données utilise le principe des bases de données relationnelles, il s'en distingue par de nombreux points. Tout d'abord, il n'applique pas un modèle relationnel précis, car les tables n'ont pas toujours une structure commune. Les entrepôts de données servent justement à croiser des informations a priori non liées directement (exemple : rattacher les informations des systèmes de production avec celles du support client pour en tirer des informations qui ont un sens).

Une fois l'information structurée, reste à mettre en place les outils de reporting qui représente ces informations. C'est le rôle des logiciels décisionnels qui se chargent aussi bien du datamining que de la présentation de ces données. Les systèmes OLAP correspondent à une technique d'analyse particulière organisant les données sous forme de dimensions.

Le multidimensionnel est au cœur même des entrepôts de données. Il est soit présent directement au sein des bases de données, soit au sein d'outils décisionnels qui émulent alors la technique d'analyse OLAP.

Donc l'analyse OLAP devient une tâche assez périlleuse vue que la représentation des données est sous forme multidimensionnelle, car cela va demander énormément de ressources matériels et logiciel ainsi que beaucoup de temps de traitement, ceci en plus de la difficulté de manipulation des opérations OLAP.

Dans ces grandes masses de données, l'analyse OLAP offre aux utilisateurs un accès de base à leurs entrepôt de données plutôt que des fonctions d'analyse avancés réservé aux analystes.

Chapitre1 : Généralités sur les entrepôts de données et les systèmes OLAP

La plupart des fournisseurs de systèmes OLAP proposent des solutions OLAP multidimensionnelles afin de réaliser ce type d'analyse, malheureusement la capacité limitée des cubes contraint de nombreux administrateurs à déployer et gérer des centaines de bases de données redondantes afin de suivre le rythme des demandes organisationnelle toujours plus nombreuses ; pour tirer le meilleur des applications. Cependant l'analyse OLAP doit dépasser le stade des cubes standards et proposer une totale interactivité en temps réelle avec l'ensemble de l'entrepôt de données.

II. Définition d'un entrepôt de données :

Le Data Warehouse ou Entrepôt de Données : désigne une sorte de base de données qui est utilisée pour collecter et stocker les informations volatiles qui viennent de sources hétérogènes et où chaque information se voit affecter une date et un numéro de version pour éviter une information déjà présente et permettre le suivi de son évolution.

Selon **Ralph Kimball**¹ : un entrepôt de données est : "une copie des données de transactions spécifiquement structurées pour la requête et l'analyse". [14]

Selon **Bill Inmon** : l'entrepôt de données est une base de données au niveau détail consistant en un référentiel global centralisé de l'entreprise. [5]

Selon **Barry Devlin** : il décrit un entrepôt de données comme étant "une mémoire unique, complète et cohérente de données provenant de sources diverses et mises à la disposition des utilisateurs finals sous une forme compréhensible et utilisable dans un contexte commercial. [4]

1. Caractéristique d'un entrepôt de données :

Selon **Bill Inmon** : le père de ce concept :

- Orienté sujet
- Intégré
- Non volatile
- La collecte de données est variée dans le temps selon les décisions de gestion dans son ouvrage: "Building the Data Warehouse" en 1996.

¹**Ralph Kimball** est le spécialiste incontesté du data warehouse. Depuis 25 ans, il anime des conférences, des formations et des séminaires sur le data warehouse dans le monde entier et tient une rubrique régulière dans le magazine américain *Intelligent Enterprise*.

Chapitre1 : Généralités sur les entrepôts de données et les systèmes OLAP

✚ Définition des caractéristiques:

✓ *Orienté sujet :*

les données sont organisées par thème, les données sont propres a un seul processus métier.

✓ *Intégré :*

Les données proviennent de sources hétérogènes utilisant chacune un type de format. Elles sont intégrées avant d'être proposées à utilisation

✓ *Non volatile :*

Les données ne disparaissent pas et ne changent pas au fil des traitements, au fil du temps (Read-Only).

✓ *Historié :*

Les données non volatiles sont aussi datées. On peut ainsi visualiser l'évolution dans le temps d'une valeur donnée.

Le degré de détail de l'archivage est bien entendu relatif à la nature des données.

Toutes les données ne méritent pas d'être archivées.

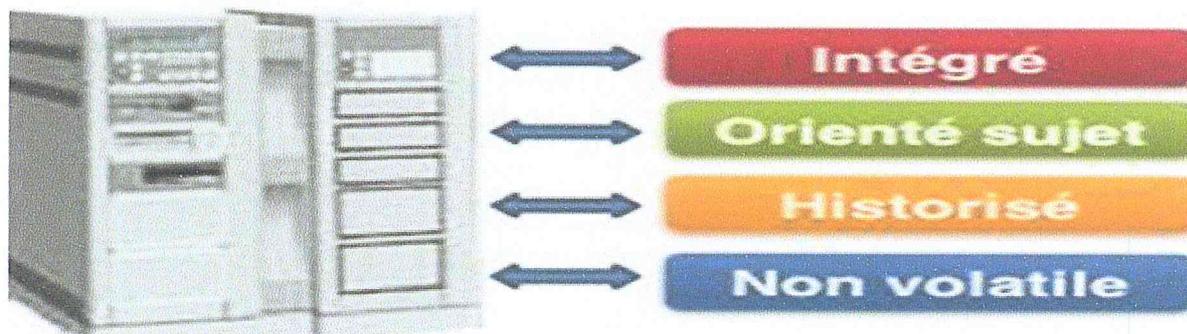


Figure 1: caractéristiques d'un Data warehouse.[16]

2. L'architecture conceptuelle d'un data warehouse :

✓ Concept de Data Warehouse actif:

Un Data Warehouse est à la base une entité passive où les tâches d'analyses de données et de prise de décision se font manuellement par les analystes.

Mais la difficulté de réalisation de ces tâches et la perte de temps ont mené à penser à introduire la fonctionnalité d'automatisation dans un entrepôt de données

Chapitre1 : Généralités sur les entrepôts de données et les systèmes OLAP

qui est apporté par l'extension de ces entrepôts passifs avec des règles d'analyses qui vont imiter l'analyste dans son analyse et sa prise de décision.

Les règles d'analyses étendent les règles de base ECA (événement, condition, action) avec des mécanismes d'analyse de données multidimensionnelles.

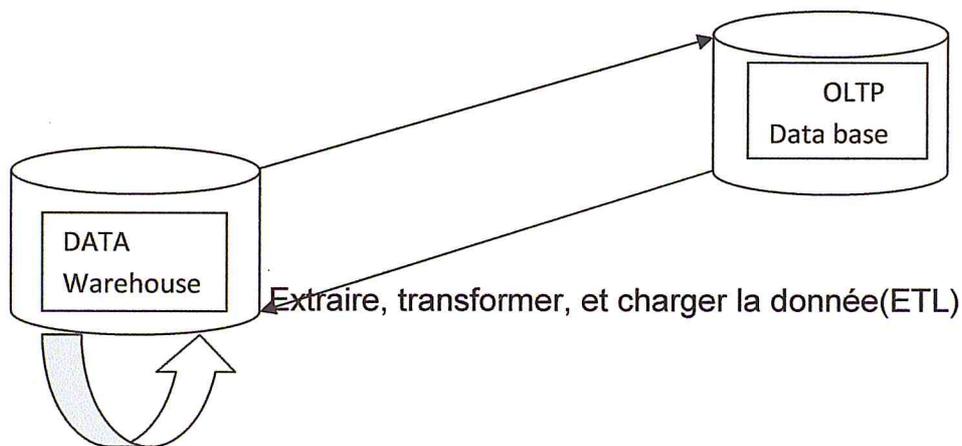
L'architecture résultante de cette extension est appelée: Data Warehouse Actif.

Un entrepôt de données actif peut exporter ou retourner les décisions aux systèmes OLTP (Online Transaction Processing) directement ou après confirmation de l'utilisateur. Ce en réalisant une approche décisionnelle en boucle fermée, que nous appelons le cycle de l'entrepôt de données (ou data warehouse) actif ou cycle de ADW pour faire court.

✓ Le cycle d'un Data Warehouse actif :

La figure ci-dessous montre le déroulement, les étapes, et le lieu de chaque processus que comporte ce cycle.

Exporter les décisions



Processus d'analyse et de prise de décision

Figure 2: cycle d'un data warehouse actif [19].

3. En amont et en aval du Data Warehouse :

3.1. **En amont :** en amont du data warehouse se place la logistique d'alimentation des données de l'entrepôt :

Chapitre1 : Généralités sur les entrepôts de données et les systèmes OLAP

- extraction des données de production, transformations éventuelles et chargement de l'entrepôt (c'est l'ETL ou Extract, Transform and Load ou encore datapumping).
- au passage les données sont épurées ou transformées par les phases suivantes :
 - ✓ un filtrage et une validation des données (les valeurs incohérentes doivent être rejetées).
 - ✓ un codage (une donnée représentée différemment d'un système de production à un autre impose le choix d'une représentation unique pour les futures analyses).
 - ✓ une synchronisation (s'il y a nécessité d'intégrer en même temps ou à la même «date deux valeurs » des événements reçus ou constatés de manière décalée).
 - ✓ une certification (pour rapprocher les données de l'entrepôt des autres systèmes

« Légaux » de l'entreprise comme la comptabilité

3.2. En aval :

On trouve Un Datamart (ou magasin de données) qui est une vue partielle du Data warehouse mais orientée métier. C'est un sous-ensemble du Data warehouse contenant des informations se rapportant à un secteur d'activité particulier de l'entreprise ou à un métier qui y est exercé.

Il se situe en aval du Data warehouse et est alimenté par celui-ci. On peut donc créer plusieurs datamarts correspondant au différent besoin des utilisateurs.

Cela permet de réduire le nombre d'opération sur les bases de production.

De plus cela permet d'offrir aux utilisateurs un outil spécifiquement adapté à leurs besoins. Cet outil sera plus petit et permettra donc un accès plus rapide à l'information.

La figure suivante montre le concept de Datamart.

Chapitre1 : Généralités sur les entrepôts de données et les systèmes OLAP

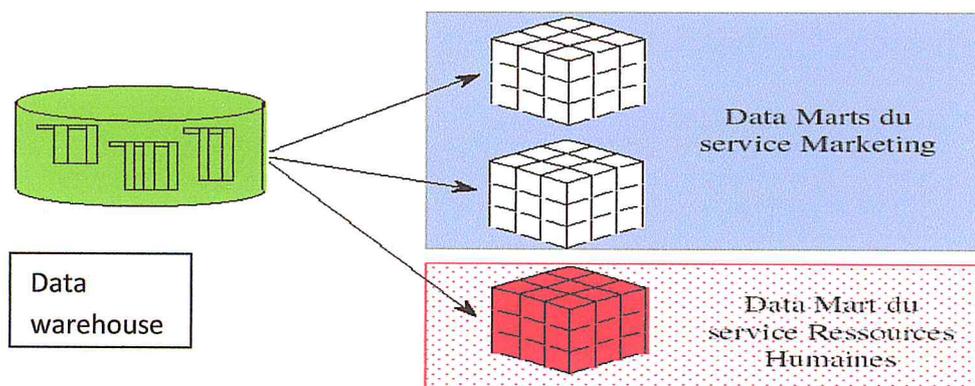


Figure 3 : le concept de Datamart [17].

4. Les types de données utilisées dans l'entrepôt de données :

4.1. Données détaillées :

Elles constituent le socle de l'entrepôt de données et sont directement issues des systèmes de production.

Elles reflètent les événements les plus récents. Même si elles sont détaillées, les données ne sont pas forcément identiques à celles des bases sources. En effet, L'entrepôt n'a pas besoin d'avoir un niveau de détail élevé, c'est pourquoi il faudra définir des règles d'agrégation. Par exemple en grande distribution, les bases de production raisonnent en référence produit alors qu'en décisionnel, la référence de base est le ticket de caisse.

4.2. Données agrégée :

Elles correspondent à des éléments d'analyse représentatifs des besoins utilisateurs. Elles constituent déjà un résultat d'analyse et une synthèse de l'information contenue dans le système décisionnel, et doivent être facilement accessibles et compréhensibles.

Cette forme agrégée facilite l'accès et la navigation au sein des données. La définition complète de l'information doit être mise à la disposition de l'utilisateur pour une bonne compréhension.

Chapitre1 : Généralités sur les entrepôts de données et les systèmes OLAP

4.3. Les Métadonnées :

Un suivi administratif est nécessaire pour contrôler le contenu du Data Warehouse lors de son alimentation, sa mise à jour et son utilisation.

En effet, l'alimentation est réalisée depuis des sources de nature diverses et les processus de création des informations décisionnelles sont complexes.

Pour pallier à ces problèmes, il est nécessaire de se doter d'un dictionnaire des données unique composé des métadonnées.

Les métadonnées sont des « données sur les données » et définissent les informations relatives à l'entrepôt et aux processus associés.

Le dictionnaire des données ou référentiel est au cœur du Data Warehouse et décrit les informations nécessaires à l'administration et à la gestion de l'entrepôt de données.

Les métadonnées gèrent le contrôle de l'information en assurant sa fiabilité, sa cohérence, sa réplication, sa distribution, leur historisation et la détermination du périmètre de calcul des données.

4.4. Les Datamarts ou Magasin de données :

Le risque d'échec du Data Warehouse est dans sa difficulté d'utilisation.

En revanche, le Datamart minimise la complexité informatique et se concentre sur les besoins des utilisateurs. Le Datamart est une base de données qui se greffe au Data Warehouse et qui reprend les données qu'il contient.

Son approche est plus simple car elle est centrée sur un objectif commun à un groupe relativement limité d'utilisateurs.

Les données du Data Warehouse peuvent être dupliquées dans plusieurs Datamarts, ce qui représente une différence fondamentale avec l'entrepôt. Le magasin de données ne contient que les données dont les utilisateurs ont besoin, il devient alors très simple d'exploiter son contenu.

Donc l'entrepôt de données sert à intégrer les données tandis que les magasins de données servent à répondre aux requêtes analytiques.

Chapitre1 : Généralités sur les entrepôts de données et les systèmes OLAP

4.5. Les cubes de données :

Le cube est une modélisation d'une base de données multidimensionnelle. Cela permet aux utilisateurs de l'entrepôt de données de pouvoir trouver, extraire et évaluer les données dont ils ont besoin sans l'aide des informaticiens. Chaque côté du cube dispose d'une dimension et les caractéristiques correspondantes.

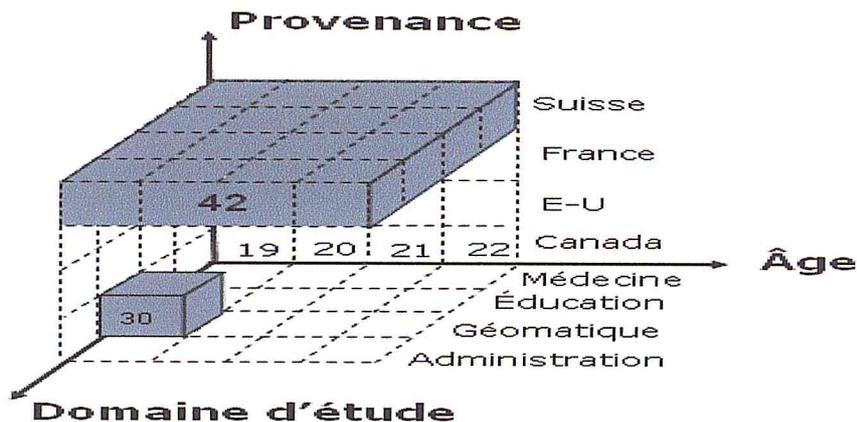


Figure 4 : exemple de cube de données.[17]

5. L'approche multidimensionnelle dans le data warehouse :

Cette approche repose sur de nouveaux concepts tels que les dimensions, les membres, les mesures, les faits et les cubes de données [Kimball et Ross, 2002; Thomsen, 2002].

C'est une manière pour agréger les données et les rendre plus accessible et facilement exploitable au moment de l'analyse.

✓ Les dimensions : Une **dimension** est un axe ou un thème d'analyse. L'âge, le domaine d'étude et la provenance sont des exemples de dimensions (dans la figure 4).

Une dimension est composée de **membres** organisés selon une hiérarchie, elle-même basée sur le niveau de détail des membres. Pour une dimension « provenance », les continents, les pays et les provinces sont des exemples de niveaux hiérarchiques et le Canada est un exemple de membre appartenant au niveau pays.

✓ Les mesures : Une **mesure** est un attribut souvent numérique (ex. nombre d'étudiants) qui est analysé en fonction des différentes dimensions (ex.

Chapitre1 : Généralités sur les entrepôts de données et les systèmes OLAP

âge, domaine d'étude, provenance). Les mesures peuvent être agrégées selon les hiérarchies des dimensions (ex. par pays et par continent), formant ainsi une vue plus globale de l'information [Rivest et al, 2001]. Un ensemble des mesures agrégées selon un ensemble de dimensions et les membres de ces dimensions forment un **cube (vue dans les cubes de données)** ou un **hyper cube** [16].

- ✓ Les faits : À l'intérieur de ce cube, les différentes combinaisons de membres de dimensions et de mesures forment les **faits**. Un fait représente ainsi la valeur d'une mesure et les membres de chacune des dimensions ayant servi à son calcul. Les faits peuvent être de types détaillés ou agrégés [14]. Les faits détaillés forment le niveau de détail le plus fin d'analyse, ils sont composés des mesures calculées et des membres les plus fins de chacune des dimensions ayant servi à calculer les mesures.

III. Définition des systèmes OLAP :

Les outils OLAP se définissent comme «une nouvelle catégorie de logiciels axés sur l'exploration et l'analyse rapide des données selon une approche multidimensionnelle à plusieurs niveaux d'agrégation » [Mattieu Grimaud, 2008].

La technologie OLAP est basée sur l'**approche multidimensionnelle**. Donc le mot OLAP, acronyme de « On-Line AnalyticalProcessing », désigne l'ensemble des technologies permettant la prise de décision stratégique rapide et fiable sur des données modélisées en multidimensionnel.

Ces technologies, pour mériter le label OLAP, doivent obéir aux règles de **CODD²[18]** :

²Edgar Frank Codd (23 out 1923-18 avril 2003) fut un informaticien britannique. Il est considéré comme l'inventeur du modèle relationnel des SGBDR.

Chapitre1 : Généralités sur les entrepôts de données et les systèmes OLAP

- Conception et vue multidimensionnelle : un outil OLAP doit se baser sur un modèle multidimensionnel pour faire de l'analyse.
- Transparence : la technologie utilisée, la conception ainsi que toutes les spécifications techniques doivent être invisibles à l'utilisateur.
- Accessibilité : les outils OLAP doivent permettre d'accéder aux données de manière rapide. Une information pertinente et on-time doit être fournie en tout temps.
- Rapidité : le chargement des données ne doit pas freiner l'analyse.
- L'outil doit pouvoir supporter de grosses requêtes (c'est la caractéristique la plus difficile à satisfaire).
- Architecture Client Serveur : pour un accès uniforme et des traitements plus rapides et plus sophistiqués.
- Dimensions génériques : essayer, autant que possible, d'avoir une unicité dans la définition des dimensions. Ne pas avoir deux dimensions client.
- Gestion des matrices creuses : en mathématiques, les matrices creuses sont des matrices qui contiennent beaucoup de zéros. En informatique, il existe des algorithmes qui utilisent cette spécificité pour optimiser le stockage de ce type de matrices. Les performances sont, en général, au rendez vous. Les outils OLAP doivent avoir cette capacité d'optimisation d'espace de stockage par la gestion des matrices creuses.
- Multi-utilisateurs : les outils OLAP sont, par définition, destinés à un accès concurrent.
- Croisement inter dimensions illimité : l'utilisateur ne doit avoir aucune restriction quand au nombre de croisements qu'il fait entre les dimensions.
- Intuitifs : les utilisateurs d'outils OLAP ne sont pas forcément informaticiens. Il est donc nécessaire d'offrir des solutions adaptées à leur style cognitif.
- Affichage flexible : l'utilisateur doit pouvoir aisément " arranger " son résultat au format désiré.
- Nombre illimité de dimensions et de niveaux d'agrégation.

1. Les outils de navigation OLAP :

- ✓ Drill up /down : Cela permet d'aller vers les informations détaillées dans une hiérarchie ou au contraire de remonter d'un niveau de

Chapitre1 : Généralités sur les entrepôts de données et les systèmes OLAP

granularité. Il s'agit donc de « zoomer ou de dé-zoomer » sur une dimension.

- ✓ Rotate : Consiste à effectuer une rotation de l'hyper cube afin de présenter une face différente. Il s'agit donc de modifier une dimension de lecture.
- ✓ Slice : Consiste à ne travailler que sur une tranche de l'hyper cube. Une des dimensions est alors réduite à une seule valeur.
- ✓ Scoping : Consiste à ne travailler que sur un sous-cube. On s'intéressera alors seulement à une partie des données.
- ✓ Dice : Extraction d'un bloc de données : S'élection de deux ou plusieurs dimensions.
- ✓ Drill-across : Exécution de requêtes impliquant plus d'un cube ayant une dimension commune.[09]

2. Les différentes implémentations OLAP :

2.1. Rolap :

Relational OLAP. Comme son nom l'indique, il utilise le concept relationnel pour stocker des données modélisées dans le format multi dimensionnel.

Les analyses (drill-down, pivot, ajout de dimensions, etc.) sont transformées en requêtes SQL classiques qui sont exécutées sur les tables.

R-OLAP utilise aussi la notion de tables d'agrégats, c'est-à-dire créer des tables contenant des données sommaires et les stocker en mémoire en cas d'utilisation.

Les outils modernes permettent aussi la gestion du cache, l'optimisation des requêtes et la création de tables d'agrégats à la demande.

La technologie R-OLAP perd beaucoup de terrain face à ces concurrents (qui suivent) car elle implique beaucoup de lourdeur et d'émulation pour son implémentation. On simule des opérations sur des matrices avec du SQL, et le fait de simuler deux conceptions apparemment différentes apporte son lot de gestion lourde et de manque de performances. R-OLAP reste la solution de choix dans le cas de gros volumes de données avec un accès restreint.

Chapitre1 : Généralités sur les entrepôts de données et les systèmes OLAP

2.2. Molap :

Multi dimensional OLAP. Contrairement à R-OLAP, M-OLAP permet de stocker les données directement en un format permettant des opérations matricielles.

Selon le constructeur, on trouvera un mode de stockage à base de tableau de données, de technologies propriétaires et même à base de fichiers plats ! L'avantage de ce mode de stockage est la capacité à effectuer des calculs très poussés en un temps record vu que tous les calculs sont pré-compilés ! Le mode de stockage permet de pré calculer les résultats afin d'avoir accès directement à toute donnée, quel que soit le niveau de détail.

M-OLAP reste la meilleure solution du moment en termes de performances et d'efficacité.

Reste que cette solution, à double tranchant, montre très rapidement ses limites quand on commence à jouer avec de gros volumes de données. En effet, le " pré-calcul " des résultats devient très pénible quand il s'agit de gros volumes de données.

2.3. Holap :

HybridOlap. C'est la solution " en vogue " du moment, car elle permet de minimiser les défaillances des technologies R-OLAP et M-OLAP. Il s'agit en fait d'un mix des deux solutions.

On utilisera un mode de stockage propriétaire pour les tables d'agrégat et les tables intermédiaires (permettant de ne pas avoir les points faibles du R-OLAP). On conservera un mode relationnel pour les tables de bas niveau.

2.4. Dolap :

Desktop OLAP. Le cas spécial, Il ne s'agit en fait pas d'une technologie particulière mais plutôt d'un mode de fonctionnement.

D-OLAP permet à l'utilisateur d'enregistrer une partie de la base de données multi dimensionnelle en local.

On voit très vite l'utilité d'une telle solution pour les commerciaux et les " nomades " de l'entreprise. Cela permettrait à un commercial, par exemple, de faire des analyses

Chapitre1 : Généralités sur les entrepôts de données et les systèmes OLAP

sur les ventes, conserver ses résultats, et vérifier l'évolution de ses analyses, une fois revenu de son voyage d'affaire.

3. Les différents modèles de données :

3.1. Le modèle en étoile : le modèle de données « en étoile » est typique des structures multidimensionnelles stockant des données atomiques ou agrégées.[07]

Le modèle en étoile est souvent considéré (à tort) comme un modèle dé normalisé, ce qui permet une économie de jointures à l'interrogation. Il est ainsi optimisé pour les requêtes d'analyse. Le modèle en étoile est implémenté sur un SGBD relationnel classique, tel que : Oracle, SQL Server. La table située au centre de l'étoile est la table de **faits**: ce sont les éléments mesurés dans l'analyse comme les montants, les quantités, les taux, etc. Les tables situées aux extrémités de l'étoile sont les tables de **dimensions** (ou encore axes d'analyse) ou niveaux de suivi. Ce sont les **dimensions** explorées dans l'analyse comme par exemple le temps (jour, mois, période, ...), etc.

3.2. Le modèle en flocons : le modèle de données dit « en flocon » est une variante du modèle en étoile : chaque table de dimension est re-normalisée pour faire apparaître la hiérarchie sous-jacente. La normalisation n'est pas indispensable car ni mises à jour ni suppressions ne sont effectuées directement sur l'entrepôt de données. L'intérêt principal du modèle en flocon réside dans le gain en espace de stockage qui est de l'ordre de 5 à 10%.[06]

4. La requête OLAP :

Une requête OLAP consiste à extraire des données du cube en appliquant des critères sur les tables de dimension et en choisissant un ou plusieurs indicateurs contenus dans la table de faits.

Les requêtes OLAP sont complexes et nécessitent des calculs d'agrégats et de multiples jointures, (Exemple : on souhaite connaître le CA par magasin, par mois et par produit pour l'année 2001 sur la région parisienne).

Chapitre1 : Généralités sur les entrepôts de données et les systèmes OLAP

Afin d'augmenter la performance de ces requêtes on peut stocker ou matérialiser tout, ou une partie du cube.

Cela veut dire que chaque niveau d'agrégation est pré-calculé en combinaison avec les autres dimensions.[09]

IV. Conclusion :

Dans ce chapitre nous avons introduit les généralités qui vont introduire notre environnement de travail à savoir les entrepôts de données (ou Data warehouse) qui sont une manière pour stocker les données qui ont été nettoyées, validées et historiés. Les données sont généralement extraites de l'opérationnel à l'aide d'un ETL (mais pas obligatoirement).

Un entrepôt est souvent modélisé en étoile, c'est à dire en table de faits (les chiffres qu'on veut analyser) et en dimensions (les axes d'analyse par exemple temps, clients, produits,...).

Ensuite vient la notion d'outils ou de systèmes OLAP qui sont apparus dans le but d'organiser les données de ces entrepôts par domaine/thème et d'en ressortir des résultats pertinents pour les décideurs.

Dans le cas idéal, une analyse OLAP s'appuierait sur un entrepôt de données dont les tables de faits et les dimensions correspondent mais ce n'est pas toujours possible.

Mais vu que chaque entreprise ou chaque domaine traité nécessite un traitement particulier la notion de personnalisation s'est imposée dans le domaine du décisionnel, et c'est ce que nous allons voir dans le chapitre qui suit.

I. Introduction :

Dans le langage courant moderne la personnalisation est l'appropriation d'un médium électronique, écrit ou d'un produit de consommation effectuée selon des données personnelles fournies par un usager, ou par la volonté de l'utilisateur lui-même.

Dans le web, la personnalisation implique la création automatique de pages web qui s'adaptent aux intérêts de l'internaute. La personnalisation s'organise selon des données implicites ou explicites. L'expression « sur mesure » est parfois utilisée lorsqu'il est question de données explicites.[04]

Ensuite avec l'arrivée des entrepôts de données et de la représentation multidimensionnelle l'intérêt de la personnalisation des données pour une analyse précise n'a cessé d'augmenter.

Mais dans ce chapitre nous allons nous intéresser à la personnalisation dans les systèmes OLAP.

II. Définition de la personnalisation :

L'approche de personnalisation fournit une vue globale, personnalisée, individualisée, qui prend en compte les besoins, les préférences, les caractéristiques et l'expérience d'un utilisateur ou groupe d'utilisateurs. [02]

La personnalisation d'un système consiste à définir, puis à exploiter un profil utilisateur qui regroupe un ensemble de caractéristiques servant à configurer ou à adapter le système à l'utilisateur, afin de fournir des réponses plus pertinentes à l'utilisateur [17].

1. Les principes mis en jeu dans la personnalisation :

1.1. Profil utilisateur :

Le profil utilisateur est une mise en facteur de la partie invariante des préférences de l'utilisateur. [11]

Le profil contient la description du centre d'intérêt de l'utilisateur mais ne se limite pas uniquement aux préférences de ce dernier, il peut également contenir des données personnelles, des préférences sur la qualité des données

Chapitre 2 : la Personnalisation dans les systèmes OLAP

et sur la sécurité ou encore des préférences sur le contenant et sur le processus de la production et/ou de la livraison des données .

La classification, l'organisation et la structuration des données du profil est un élément clé du processus de personnalisation.

1.2. Le contexte utilisateur :

L'informatique sensible au contexte est apparue au milieu des années quatre-vingt dix inspirée par les travaux de.[18]

Ce terme fait référence à des systèmes capables de percevoir un ensemble de conditions d'utilisation – le *contexte* – afin d'adapter en conséquence leur comportement en termes de délivrance d'informations et de services.

Mais dans le contexte de la personnalisation, se sont les contraintes que doit satisfaire ce processus de personnalisation afin d'obtenir un résultat cohérent avec les besoins de l'utilisateur.

1.3. Les préférences utilisateurs :

C'est le processus d'évaluation des différents critères ou paramètres de préférence d'un profil utilisateur afin de les classer par ordre d'importance par rapport au contexte et/ou à l'utilisateur.

Implication de l'utilisateur

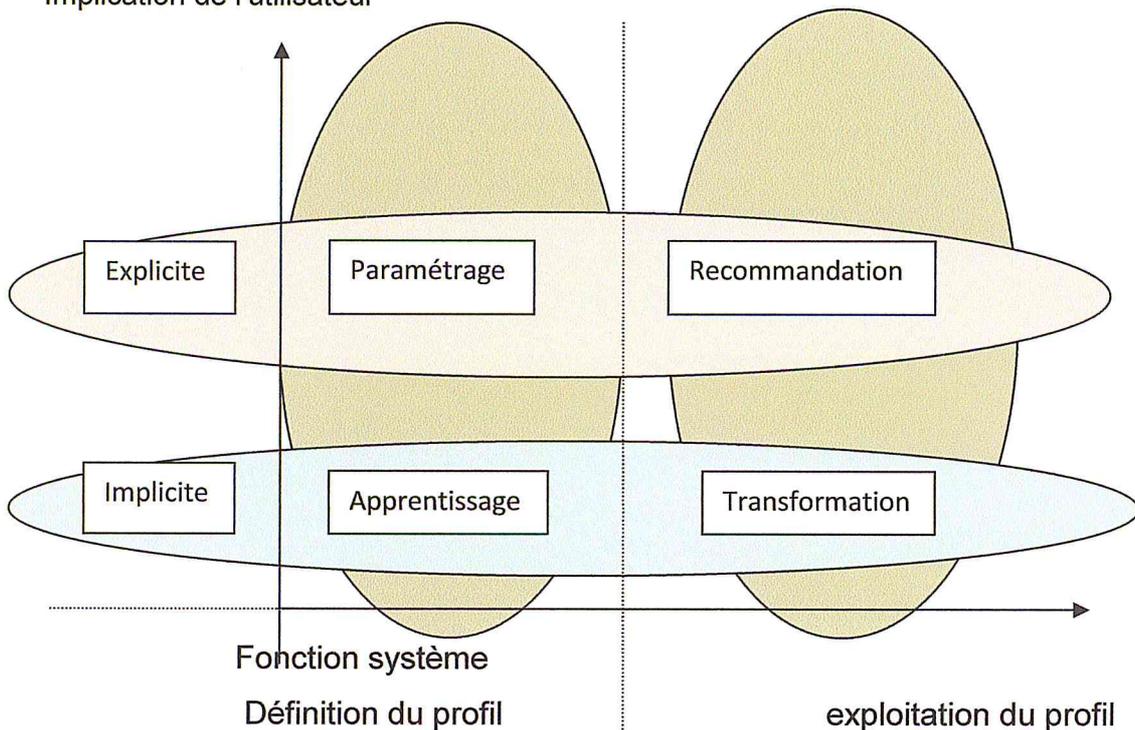


Figure 5 : principe de la personnalisation [08].

2. Etat de l'art :

Dans la conception des systèmes décisionnels, la dimension utilisateur a vocation à être considérée avec de plus en plus d'attention, sous-estimer son importance risque de mettre en péril les résultats souhaités.

Cette prise en compte peut se faire de manières multiples comme :

De définir, puis exploiter le profil utilisateur qui ont débuté avec les travaux de :

- Travaux de [Bellatrache, Giacometti, Marcel, Mouloudi, et Laurent 2005] :

Ce dernier consiste à adapter les données affichées en un cube de données en se basant sur les contraintes et les préférences des utilisateurs.

Ces dernières sont exprimées explicitement par l'utilisateur.

Cela permet de calculer la partie de la requête qui satisfait les contraintes et les préférences des utilisateurs. En plus une structure de visualisation est proposée.

- Ensuite les travaux de [Ravat, et Test 2009] :

C'est une approche quantitative qui est basée sur les concepts de base du multidimensionnel et les règles de personnalisation. Les règles sont basées sur le formalisme ECA et affectent des poids de priorité aux attributs du schéma multidimensionnel, de nouveaux opérateurs OLAP spécifiques à un système OLAP personnalisé ont été également définis.

- Et aussi ceux basés sur le profil :

Ce type d'approche n'est pas nouveau dans le monde de la recherche on citera à cet effet le modèle générique de profil de Bouzeghoub dans la RI.[08] Mais la nouveauté apportée par les travaux de Kozmin, et Niedrit [02] est la prise en charge de l'aspect spécifique des entrepôts de données.

En effet il a été proposé un ensemble d'attributs décrivant le profil en se basant sur le Framework de Zackman[08] ; la description du profil ainsi faite répond aux questions (quoi, qui, quand,), ensuite une collecte des attributs depuis plusieurs sources est effectuée.

Une autre approche est celle conceptuelle qui va se baser sur le schéma :

- ✓ La conceptualisation : Garrigos et al[03] ont conçu un modèle conceptuel pour la personnalisation en se basant sur le mécanisme ECA : (Événement- Condition-Action) de personnalisation. Cette

Chapitre 2 : la Personnalisation dans les systèmes OLAP

approche permet de créer des cubes de données dans les deux types de personnalisation :

- 1) statique : créer plusieurs cubes d'OLAP pour plusieurs utilisateurs au design-time.
- 2) dynamique : créer un cube de donnée est pour chaque utilisateur durant le run-time en prenant en compte les besoins et les actions réalisées par l'utilisateur il utilise aussi le principe d'ECA.

Ou bien exprimer les préférences utilisateur dans les requêtes OLAP :

- ✓ exprimer les préférences des utilisateurs dans les requêtesOlap :
[Golfarelli, S. Rizzi 2009] ont présentés un algèbre de préférences, qui peut être exprimées avec des données numériques ou catégoriques ou encore au niveau des agrégations de table de fait.

Cet algèbre intègre une notion de préférences des attributs, des mesures et des hiérarchies.

Il existe aussi un autre algèbre composé de plusieurs constructeurs appelé **Paréto**.

L'originalité de cet algèbre est la déclaration des préférences dans la clause group-by de la requête.

La personnalisation a touché également la représentation visuelle :

- ✓ La visualisation : les travaux entrepris dans cet axe la sont les travaux de Mansmann,etsholl [06]qui sont imminent dans ce domaine car il présente une technique de visualisation hiérarchique ou les résultats sont représentés sous forme d'arbre de décomposition.

Mais la première représentation de ce genre a été PROCLARITY ,Proclarityanalytic 6[],et ensuite le modèle REPPORT PORTAL (XMLA consulting 2006), toutefois ces deux représentations ne permettent pas un développement de valeur par interaction ni d'avoir des résultats utiles en terme d'affichage , alors que le modèle de Mansmann et sholl [06] permet de générer l'arbre de décomposition tout en prenant compte des modèles de données sur lesquelles il se base et les préférences de ces utilisateurs.

Enfin la plus basic et la plus ancienne de toutes est l'approche qui intervient au niveau de la modélisation :

- ✓ La modélisation : Cette approche utilise des tableaux de bords prédéfinis et vise à modéliser des scénarios préétablis en utilisant des

Chapitre 2 : la Personnalisation dans les systèmes OLAP

mécanismes automatiques, comme les travaux de Thalhammer, Schrel, [11] avec la création des data warehouses actifs.

Dans le cycle d'analyse de notre système proposé l'utilisateur va être pris en compte de manière implicite dans l'analyse de la requête ce qui va permettre une déduction de ces préférences sans l'intervention de ce dernier.

III. Synthèse et discussion :

La problématique émergente de ce chapitre est le besoin de personnalisation des systèmes OLAP en fonction des besoins analytiques individuels [04].

Car nous proposons d'utiliser la personnalisation afin de rendre possible aux décideurs une manipulation plus rapide et efficace des outils OLAP.

Ce tableau ci-dessous montre une synthèse des travaux réalisés dans la personnalisation :

Travaux	Définition du profil	Exploitation du profil
Sapia , 2000	Explicite	Transformation
Espil et al, 2001	Explicite	Transformation
Thalhammer et al, 2001	Explicite	Transformation
Bellatreche et al, 2005	Explicite	Transformation
Favre et al, 2007	Explicite	Transformation
Giacometti et al, 2008 ;2009	Implicite	recommandation

Figure6 : Tableau comparatif des travaux de personnalisation[11]

On remarque que la quasi-totalité des travaux effectués définit le profil utilisateur de manière explicite ensuite on l'exploite par transformation, seul très peu d'entre eux s'intéresse à la recommandation.

IV. Conclusion :

Dans cette partie nous nous sommes intéressés à la personnalisation dans les systèmes OLAP, son origine, un aperçu sur son état de l'art ainsi que les principes mis en jeu dans ce processus de personnalisation.

On a mis l'accent sur le concept de profil utilisateur, ses préférences, et son implication dans le système. Et on a terminé par une petite synthèse qui a mis en avant les travaux les plus importants.

Penser à un système personnalisé est une étape très importante mais insuffisante car, on aspire à des systèmes qui anticipent les résultats et propose plusieurs alternatives à celle-ci.

Donc dans le prochain chapitre nous allons nous intéresser aux systèmes recommandés qui mettent en avant les aspects voulus.

I. Introduction :

Les systèmes d'aide à la décision permettent aux utilisateurs de prendre des décisions stratégiques, et vu la pression concurrentielle la vitesse de prise de décision doit augmenter et doit offrir un accès plus personnalisé à l'information, les systèmes de recommandations sont une façon de palier à ce besoin.

Ils sont utilisés dans le e-Commerce sur le web tel que : Amazon, Movie Lensetc. qui conseil des films, voyages,.....etc. aux utilisateurs.

En outre les outils OLAP permettent une infinie possibilité de résultats l'ors quelles sont appliquées à des données multidimensionnelles, mais pour rendre cette tâche facile d'utilisation il faut les étendre avec des techniques de recommandation.

II. Définition des systèmes recommandés :

Les systèmes de recommandation sont une forme spécifique de filtrage de l'information (SI) visant à présenter les éléments d'information (films, musique, livres, news, images, pages Web, etc) qui sont susceptibles d'intéresser l'utilisateur [12]. Généralement, un système de recommandation permet de comparer le profil d'un utilisateur à certaines caractéristiques de référence, et cherche à prédire l'« avis » que donnerait un utilisateur. Ces caractéristiques peuvent provenir de□:

- l'objet lui-même, on parle d'approche basée sur le contenu ou *content-based approach*,
- l'utilisateur,
- l'environnement social, on parle « d'approche de filtrage collaboratif ou *collaborative filtering*.

Les systèmes de recommandation peuvent être définis de plusieurs façons, qui peuvent se rapporter à différents types de données ou approches spécifiques. La définition que nous utiliserons est une définition générale de **Robin Burke**[04], que nous avons traduite :

Système de recommandation : Système capable de fournir des recommandations personnalisées ou permettant de guider l'utilisateur vers

des ressources intéressantes ou utiles au sein d'un espace de données important.

1. Le principe de la recommandation :

1.1. **La recherche d'information** : Rechercher une information consiste à trouver les documents pertinents adaptés au besoin de l'utilisateur. Ce besoin est représenté par une requête qui peut être un document, une description ou des mots clés.

1.2. **Le filtrage d'information** :

Au contraire de la recherche d'information, le filtrage d'informations effectue une sélection des documents les plus similaires à la requête. C'est par exemple le cas d'une recommandation spontanée du système de filtrage.

La requête peut être représentée par un comportement, comme le fait de cliquer sur un item, ou être basée sur un profil d'intérêt qui peut aussi se baser sur le comportement de l'utilisateur ou sur d'autres informations.

Le filtrage commence donc après la définition du besoin de l'utilisateur, il permet d'éliminer les documents qui peuvent ne pas intéresser l'utilisateur.

Le filtrage offre à l'utilisateur un gain d'effort et de temps.

2. Les types de recommandation :

On distingue trois types de recommandation le premier qui s'intéresse au contenu, le deuxième qui s'intéresse à l'avis des autres personnes ayant le même centre d'intérêt et enfin le troisième qui est une symbiose des deux précédents.

2.1. **Le filtrage basé sur le contenu** :

Un système qui utilise le filtrage basé contenu exploite seulement les représentations des documents et les informations qui peuvent être dérivées de ces documents.

Un tel type de filtrage pourrait par exemple utiliser la similarité des documents dans une matrice termes-documents pour déterminer la pertinence d'un document.

Si un utilisateur exprime un intérêt pour un document, les documents similaires seront jugés potentiellement pertinents aussi.

L'exemple du système PRES (acronym for Personalized Recommender System) [07] qui est un système de recommandation basé-contenu.

✓ Limite des systèmes basés sur le contenu :

Quelques limites des systèmes de filtrage basé contenu sont présentées par **Berrutet Denos** [10].

La première difficulté pour ce type des systèmes c'est qu'il ne traite que des documents textes, nous ne pouvons pas indexer des documents multi media.

De plus les systèmes basés contenu ne considèrent que le facteur thématique comme critère de pertinence malgré qu'il existe d'autres facteurs tels que : la fiabilité de la source d'information, la qualité scientifique et le degré de précision des faits présentés. Les auteurs évoquent aussi l'effet « entonnoir » qui restreint le champ de vision de l'utilisateur: à force de ne recommander que des items reliés à une seule et même thématique, et du fait que l'intérêt de l'utilisateur est déterminé à partir des items choisis, on crée un effet d'auto-entraînement qui restreint toujours plus la définition des intérêts de l'utilisateur et empêche l'exploration de thématiques différentes.

2.2. Le filtrage collaboratif :

Le filtrage collaboratif est un parmi les technologies les plus populaires dans le domaine des systèmes de recommandation [10].

Le filtrage collaboratif se base sur l'idée que les personnes à la recherche d'information devraient se servir de ce que d'autres ont déjà trouvé et évalué.

Dans la vie quotidienne, si quelqu'un a besoin d'une information, il essaye de s'informer généralement auprès de ses amis, ses collègues, qui vont à leurs tours lui recommander des articles, des films, des livres, etc. Cette collaboration entre les gens permet d'améliorer l'échange de connaissances, mais cela prend beaucoup de temps vu que cette ressource d'information ne peut pas toujours à notre disposition. C'est à partir d'ici que l'idée de filtrage collaboratif été née, le besoin être d'automatiser et de rendre l'échange des expériences et des avis personnels de certains personnes utilisables par d'autres. Selon **Golberg** [03] le filtrage collaboratif est l'automatisation des processus sociaux.

2.3. Le filtrage hybride :

Le filtrage hybride combine les deux types de filtrage basé sur le contenu et le filtrage collaboratif.

C'est une approche proposée pour essayer de remédier aux lacunes de chaque méthode.

3. Etat de l'art :

Sur quelle page web me rendre pour trouver l'information que je cherche ?

Quels sont les films que je n'ai pas encore vus et qui pourrai me plaire ?

Dans de nombreux domaines le foisonnement de ressources rend difficile leur exploitation et leur exploration.

L'exemple le plus souvent mis en avant pour illustrer ce foisonnement est l'explosion du web, impliquant des milliards de page et de ressources.

Si le web est le domaine où cette subordonnance s'observe le plus nettement, d'autre domaine renferme des quantités importantes de données, en particulier les bases de données cinématographiques et musicales.....

Un autre domaine prolifique enfin celui de la recherche scientifique.

Un autre aussi relatif à la problématique de la subordonnance de données est celui des systèmes recommandés, mais vu les résultats concluants lors de son évaluation dans le domaine du web [Srivastava et al. 2000]. Et de la recherche scientifique ; son application à OLAP sera très intéressante et bénéfique.

Donc voyons un peu ce qui s'est fait au niveau de OLAP :

D'après les recherches sur le domaine, nous constatons que les premiers à traiter le problème de la recommandation de requêtes OLAP (et particulièrement de requêtes MDX [02] dans [05].

L'idée d'employer ce que les autres utilisateurs ont fait pour produire des recommandations est très populaire dans le domaine de la recherche d'information, Adomavicius [12], l'adaptation de ces techniques existantes à OLAP se voit dans les travaux de [11], et Yang [13] qui ont un point commun : prévoir la prochaine requête OLAP.

Néanmoins, la principale préoccupation de Sapia [10] est de prétraiter les données mais pas de recommander une requête, Sapia [10] utilise un modèle de Markov pour prévoir la prochaine requête, tandis que Yang [13] ne tiennent pas compte du séquençement des requêtes.

Jerbi et al, et Bentayeb et al,[04] proposent des méthodes de personnalisation vues comme des méthodes de recommandation qui prennent en compte les

préférences de l'utilisateur mais ne s'intéressent pas aux autres analystes qui peuvent avoir des ressemblances avec l'utilisateur courant, et encore moins au séquence ment de requêtes lors de la session d'analyse puisqu'elles se limitent à la dernière requête posée. Notons cependant que Bentayeb et al,[04]utilisent l'algorithme de classification non supervisée (clustering) des K-means dans leur phase de recommandation.

III. Conclusion :

Dans ce chapitre, on a vu un aperçu sur les systèmes recommandés, leurs principes et les types de recommandation qui sont le filtrage basé sur le contenu qui va être axé sur le contenu lui-même pour déterminer la pertinence du document a présenter a l'utilisateur.

Contrairement le filtrage collaboratif va s'intéresser aux avis d'autre personne sur le même domaine de recherche ou le même document et enfin l'approche hybride qui va combiner les deux précédentes de plusieurs manières et qui va palier au manque rencontré dans ces derniers. Ensuite on termine avec une petite synthèse sur les travaux qui ont touché au domaine des systèmes recommandés selon les trois types de filtrages, mais la chose qui résulte de cette étude est que l'axe de l'analyse est toujours l'utilisateur mais en tant qu'entité statique c'est-à-dire que les résultats destinés a cet utilisateurs seront toujours les même en toutes circonstance, ce qui n'est pas en adéquation avec la réalité.

Donc dans le chapitre qui suit ; on va s'intéresser au contexte de l'utilisateur ou a la contextualisation.

I. Introduction :

Avec les résultats étudiés en matière de personnalisation de systèmes OLAP et plus précisément, la recommandation d'informations aux utilisateurs, on a pu constater que les résultats n'étaient pas très pertinents par rapport aux exigences de l'utilisateur car le système ne prend en compte que le profil de ce dernier qui est déjà établi ou intégré explicitement par lui-même sans considération de l'environnement de cet individu ni les raisons qui l'ont poussé à émettre cette requête.

Les systèmes de recommandation qui ont déjà été développés depuis ces dernières années et les nombreuses recherches sur l'amélioration des méthodes traditionnelles.

La majorité des recherches axées sur l'amélioration de l'algorithme de recommandation et des applications combinées des systèmes de recommandation traditionnelles, mais ils ont tous un inconvénient similaire. La majorité des systèmes de recommandation ont utilisé les données recueillies en vertu de l'environnement similaire pour formuler des recommandations. Il a été découvert, dans une expérience réelle, que si les comportements passés des clients ont été examinés et l'information contextuelle a été ignorée, il est souvent cause de soupçons dans les résultats de recommandation. Dans les dernières années, il y avait aussi de nombreuses études prouvant que, pour améliorer un système de recommandation, il convient de différencier la collecte des données de l'établissement du profil. Dans le processus d'analyse, les effets de l'information contextuelle devraient également être envisagés et être utilisés comme critère de recommandation et fournir des résultats de recommandation appropriés.

II. Définition du contexte :

Selon **Basil et Brezillon [04]**: il est difficile de trouver une définition pertinente et qui satisfait toutes les disciplines. On remarque que pour chaque discipline ou le contexte est admis, on a une définition différente. Celle qui va nous intéresser est celle du domaine des bases de données contextuelles.

- **Définition du contexte:** Les Capacités contextuelles ont été ajoutées à une partie de la gestion des systèmes de base de données, en intégrant les préférences des utilisateurs et le retour des différentes réponses aux requêtes de bases de données en fonction du contexte dans lequel ces requêtes ont été exprimées. Et les préférences de l'utilisateur particulières correspondent à des contextes spécifiques. Plus spécifiquement, dans les travaux de Stephanidis [05] un ensemble de paramètres contextuels est introduit et les préférences sont définies pour chaque combinaison d'attributs relationnels et ces paramètres contextuels.

Donc **Le contexte** peut être perçut comme une extension du profil utilisateur. Il contient des informations complémentaires permettant une meilleure adaptation du résultat de la requête à son environnement local.

III. Les systèmes recommandés VS les systèmes recommandés contextualisés :

Pour intégrer l'information contextuelle dans les systèmes de recommandation, une nouvelle idée, appelé CARS (Context-Aware Systèmes de recommandation) ce qui veut dire les systèmes recommandés sensible au contexte, a récemment émergé.

Les systèmes (CARS), qui traitent de la modélisation, de prévision et des préférences utilisateur en intégrant des informations contextuelles disponibles dans la recommandation.

Ces préférences à long terme sont généralement exprimées en notes et sont modélisés en une fonction R , mais en prenant compte le contexte. En d'autres termes, les notes sont définies avec la fonction $Note$ qui est défini comme ceci: $R: L'utilisateur \times l'attribut \times contexte \rightarrow Note$, où $L'utilisateur$ et $l'attribut$ sont les domaines utilisateur et objets, respectivement, $Note$ est le domaine de votes, et le contexte précise l'information contextuelle associée à l'application. [01]

La prise en compte de l'information contextuelle :

L'information contextuelle peut être intégrée de trois manières différentes dans le processus de recommandation : [05]

- a) **De manière explicite** : explicitement, c'est à dire par approche directe des personnes concernées et la collecte de ces informations contextuelles se fait directement soit en demandant des questions ou par obtention d'informations par d'autres moyens. Par exemple, un site Web peut obtenir des informations contextuelles en demandant à une personne de remplir un formulaire Web, ou à répondre à un questionnaire spécifique avant de fournir l'accès à certaines pages Web.
- b) **De manière implicite** : à partir des données de l'environnement, comme un changement d'emplacement de l'utilisateur détecté par une compagnie de téléphonie mobile. Alternativement, des informations contextuelles temporelles qui peuvent être obtenues à partir de l'horodatage d'une transaction.
- Rien ne doit être fait dans ces cas en termes d'interaction avec l'utilisateur ou d'autres sources d'informations contextuelles: la source de l'information contextuelle implicite est directement accessible et les données sont extraites automatiquement.
- c) **La déduction du contexte** : en utilisant des méthodes statistiques ou de data mining. Par exemple, l'identité des personnes d'un ménage feuilletant les chaînes de télévision (mari, femme, fils, fille, etc) ne peut pas être explicitement appelée par une société de télévision par câble, mais elle peut être déduite avec une précision raisonnable en observant les programmes de télévision regardée.

IV. Les méthodes d'intégration de l'information contextuelle :

Il existe trois manières de procéder : selon l'étape du processus de recommandation ou l'on introduit l'information contextuelle donc :

1 **Pré-filtrage contextuelle**: l'information contextuelle est utilisée pour filtrer les informations pertinentes avant qu'elle ne soit introduite dans le processus de recommandation classique.

Chapitre4 : la Contextualisation

C'est-à-dire Filtrer les votes avant d'appliquer les algorithmes de recommandation en sélectionnant les votes qui ne concernent que le contexte donné.

Pour les types du Pré-Filtrage, on en distingue deux :

- Exact Pre-Filtering (EPF) : Le contexte est bien précisé, exemple : Hiver 2008vacances.
- GeneralizedPre-Filtering : Le contexte est généralisé, exemple : 2008 vacances.

2 **Post-filtrageContextuelle:** l'information contextuelle est utilisée après le processus classique de recommandation (2D) qui est appliquée aux données standards (non-contextuels). C'est-à-dire qu'il utilise les approches de recommandations puis introduit le contexte, exemple: Soit un utilisateur « u ». Trouver les utilisateurs similaires à « u » en analysant la donnée choisie par l'utilisateur dans un contexte donné pour calculer la probabilité qu'il choisisse la même donnée dans ce contexte. Les recommandations seront contextualisées en leurs introduisant cette probabilité.

Il existe deux méthodes du Post-Filtering : Weight et Filter (Calcule les probabilités citées).

- Weight : Classe les recommandations résultantes de l'approche appliquée (basée sur le contenu, filtrage collaboratif ou recommandations hybrides) selon leur ordre de pertinence dans un contexte donné.
- Filter : Garde que les recommandations pertinentes.

Comparaison des deux méthodes :

Post-Filtering (filter) est meilleure que Pre-Filetring (EPF) et ce dernier est meilleur que Post-F (weight) : Le choix de la méthode du Post-F détermine qui de Pre-Filetring ou Post-Filetring est meilleure que l'autre.[01]

3 **La modélisationcontextuelle:** les informations contextuelles sont utilisées à l'intérieur de l'algorithme de recommandation. Cette méthode n'est pas très utilisée, car on aura à faire à plusieurs dimensions, ce qui n'est pas facile à appliquer dans les systèmes de recommandation.

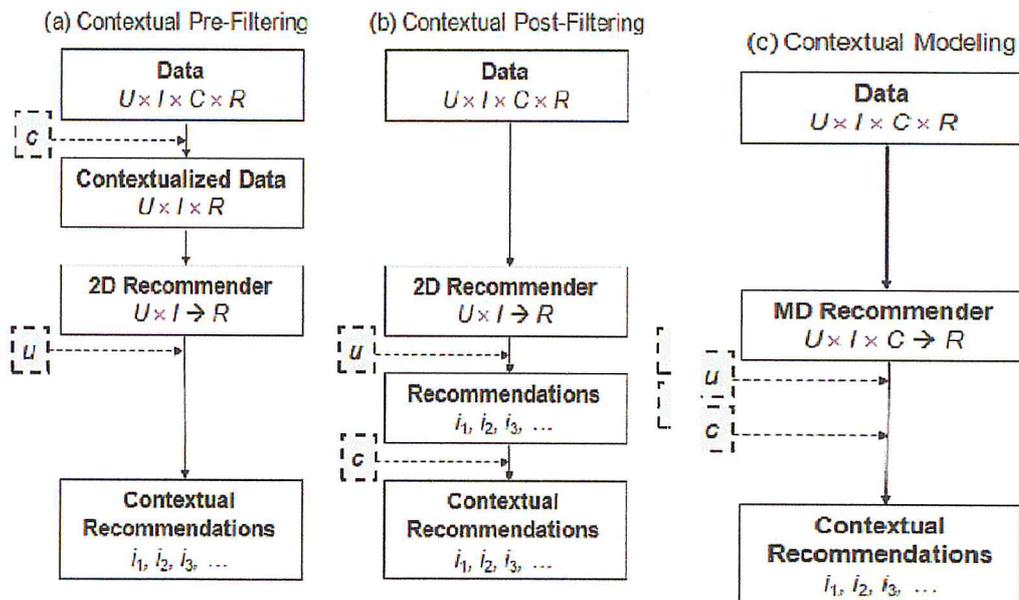


Figure 8 : méthode d'intégration du contexte dans les systèmes recommandés [05].

V. Caractéristiques des systèmes recommandés contextualisés CARS :

1. **Complexité** : Depuis que les systèmes CARS ont impliqué les utilisateurs et les attributs dans le processus de recommandation, mais aussi différents types d'informations contextuelles, les recommandations peuvent être beaucoup plus complexes par rapport aux recommandations non-contextuelles.
2. **L'interactivité** : L'information contextuelle doit généralement être obtenue à partir de la participation de l'utilisateur dans le système CARS. Par exemple, pour utiliser les informations contextuelles, un système CARS peut avoir besoin d'obtenir auprès de l'utilisateur certaines spécifications avant d'émettre des recommandations spécifiques au contexte.
3. **La flexibilité** : Plus les domaines d'utilisation du système, sont nombreux mieux c'est.

VI. Etat de l'art :

Dans le domaine des systèmes recommandés sensible au contexte les travaux sont pratiquement inexistant sauf deux qui ont été des idées proposés et se sont celles de :

Bellatreche et al [05] : dans cette approche une proposition de personnalisation des réponses de requêtes OLAP a été proposée et plus précisément un calcul du sous-ensemble le plus pertinent de faits pour la réduction du cout de traitement ainsi qu'une structure de visualisation pour représenter les résultats obtenu, ou les profils utilisateurs et les contextes sont exprimer sous formes d'ensemble recommandés sur les membres du cube et des contraintes de visualisation.

Et celle de **Jerbi et al [04]** : ici l'idée de recommandation est incluse dans le processus de personnalisation et elle n'est pas totalement basé sur le profil de l'utilisateur car elle est calculé selon des fonctions heuristiques ou par des fonctions d'utilités qui considères indirectement des propriétés du profil ; ainsi la personnalisation peut être faite après la génération du résultat en filtrant la liste de ses éléments.

VII. Conclusion :

Cette partie vient de mettre en avant un domaine de recherche émergent qui vient se positionner dans un environnement de systèmes décisionnel, de personnalisation, de recommandation, qui sont eux aussi des domaines ou les challenges ne cesse d'augmenter avec l'apparition de nouvelles exigences et de nouvelles concurrences.

Donc dans notre humble travail nous allons nous aussi essayer de nous positionner face à ces défis, donc la partie qui va suivre va consister à présenter notre système, son architecture ainsi que les algorithmes choisis pour ce dernier.

I. Introduction :

Nous avons basé notre approche sur les systèmes recommandés contextualisés, précédemment défini, et on va introduire le contexte de manière explicite par l'utilisateur. Nous présenterons l'approche d'intégration du contexte dans les applications de personnalisation et l'exploitation des données contextuelles pour effectuer les recommandations. Pour chaque partie du système, nous donnerons sa définition, ses algorithmes et son déploiement dans les différents niveaux de l'application.

II. Architecture du système :

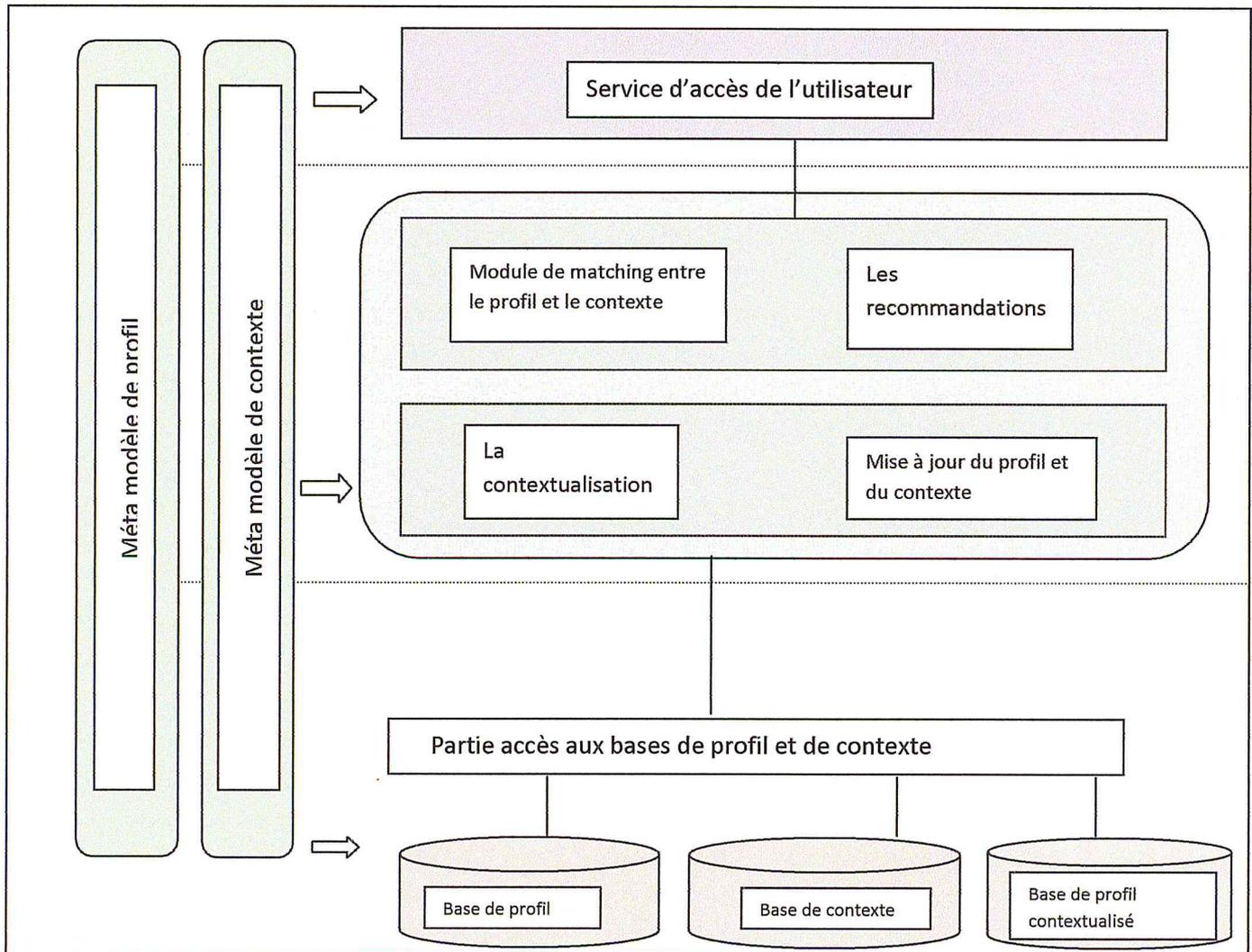


Figure 9 : l'architecture globale du système.

A partir de ce schéma nous allons détailler chaque module qui le compose et nous intéresser au rôle qu'il a dans le processus ainsi qu'à sa principale fonction. Sans oublier les algorithmes et méthodes utiliser afin de veiller au bon fonctionnement de ces derniers.

Nous remarquons que l'architecture proposée est présentée sous forme d'une décomposition en trois niveaux le premier qui est le plus bas est la partie stockage, la dixième qui va être la partie de traitement et la dernière consiste en l'interface d'interaction de l'utilisateur avec le système.

- ✓ Pour la partie interface : l'utilisateur va disposer d'une interface d'accès pour pouvoir lancer sa requête et définir le contexte d'interaction de son choix parmi la liste de contexte prédéfinis.
- ✓ Pour la partie traitement : nous allons détailler chaque partie ; pour mieux comprendre la fonctionnalité de chacune. Mais avant cela remarquons qu'il y a deux types de traitements, ceux effectués offline et les autres online.

✚ Les fonctions offline :

Ce sont les fonctions qui permettent de générer les préférences utilisateur à partir du log de l'utilisateur.

L'algorithme choisit pour cet effet est inspiré des travaux de [SofineAbbar et al. Dans son document : contexte AwareRecommenderSystems : A service oriented approach] qui a proposer un algorithme pour l'extraction automatique des contextes, mais nous allons l'utiliser pour l'extraction des préférences utilisateurs à partir du log.

L'algorithme est comme suit :[04]

Entrée: le fichier log de l'utilisateur et qu'on va noter 'L'.

Sortie : l'ensemble des préférences utilisateur noté : $Pr = \{pr_1, \dots, pr_i\}$, α est le seuil de préférence choisis par l'utilisateur (de manière explicite) ;

1: $Pr = \varnothing, \xi := \infty$;

2 : $PrE \leftarrow \text{Préférences}(L)$, (extraction des instances de préférence) ;

3 : $Pr \leftarrow$ appliquer le clustering hiérarchique ;

4 : pour tout cluster $pi \in Pr$ faire :

5: calculer le Centroïde du cluster;

6: répéter l'opération;

7: $\xi_{ant} := \xi$;

8: pour toute instance $t_j \in PrE$ faire :

9: affecter chaque t_j au cluster qui lui est le plus proche ; ($pr_i \in Pr \vee pr_j \in Pr$) ;

10: pour tout cluster $pr_i \in Pr$ faire :

11: mise à jour du centroïde, $\text{centre} := \text{new centre}(pr_i)$;

12: calculer $\xi =$;

13: jusqu'à ce que $\xi \geq \xi_{ant}$

14: retourner Pr .

 Les fonctions onlines :

En premier intéressons-nous au module de matching :

1. Le matching :

Cette partie nous allons la considérer comme principal car elle inclut la partie de notre contribution au système que nous allons proposer.

Cette partie va en final essayer de trouver pour toutes les combinaisons d'une instance de la base de profile et du contexte choisis, les préférences qui sont importantes pour l'utilisateur ou pas.

Pour savoir les quelles prendre en considération lors de la recommandation.

1.1. L'algorithme choisis [04]:

Il est présenté comme suit :

Input: ensemble des préférences $Pr=\{pr_1, \dots, pr_i\}$, le seuil de tolérance β .

Output: ensemble de profile contextualisé PC.

```
1:  $PC \leftarrow \emptyset$  ;  $cf := 0$  ;  $sp := 0$  ;
2:  $C \leftarrow$  Contexte ;
3: for all  $pi \in Pu$  do
4: for all  $cj \in C$  do
5: calculer  $cf :=$  confiance( $pr_i ; c_j ; \beta$ ) ;
6: calculer  $sp :=$  support( $pr_i ; c_j ; \beta$ ) ;
7: calculer  $s_{ij} :=$  fusionner ( $cf ; sp$ ) ;
8: if  $s_{ij} \geq \beta$  then
9:  $PC = PC \cup (pi ; c_j ; s_{ij})$ ;
10: return PC.
```

Maintenant intéressons-nous au module de contextualisation,

2. La contextualisation :

Dans cette partie également, nous avons été confrontés à un choix entre trois approches : le pré-filtrage, le post-filtrage et la modélisation contextuelle.

Mais notre choix s'est posé sur le pré-filtrage ; car le pré-filtrage associé au modèle multidimensionnel donnera une application flexible à n'importe quel type d'évaluation et elle est moins coûteuse que post-filtrage.

Cette approche consiste à : [05]

Filtrer les votes avant d'appliquer l'algorithme de recommandation en sélectionnant les votes qui ne concernent que le contexte donné, et l'approche de pré-filtrage sera de type : Exact Pre-Filtering (EPF) car

l'information contextuelle sera exacte et définit explicitement par l'utilisateur.

Quant au module de recommandation il est proposé comme suit :

3. La recommandation :

Après le matching, il résultera un ensemble de tuples correspondant à un ensemble de requêtes candidates, pour pouvoir recommander ces requêtes à l'utilisateur il faudra les ordonner selon les préférences préférées de cet utilisateur même ;[05], on va les ordonner selon l'algorithme suivant :

1. Pour chaque requête q_i de l'ensemble QR des recommandations, trier ces références en fonction de l'ordre \leq .
 2. Pour chaque ensemble trié de références, prendre le premier élément, c'est-à-dire la référence préférée que nous appelons le maximum.
 3. Trier ces maximums en fonction de l'ordre \leq .
 4. Retourner l'ensemble ordonné QR des requêtes tel que chaque maximum a été remplacé par la requête qui lui correspond.
- Finalement, l'ensemble QR des requêtes candidates ordonnées en fonction du profil utilisateur est recommandé à l'utilisateur.

Remarque : la requête recommandée par défaut à l'utilisateur est la première de l'ensemble des requêtes candidates à recommander, mais si l'utilisateur refuse cette proposition il se verra proposer celle d'après dans la liste et ainsi de suite.

Ensuite voyons plutôt les éléments qui entrent en jeu dans ces processus :

- ✚ En fait faisons remarquer que ces éléments sont en relations avec toutes les couches du système mais leur apparition au niveau traitement est la plus

visible, car dans la partie stockage ce n'est que la partie ou ils sont stockés pour une question de nécessité, et la partie interface ou ils sont sollicité par la requête de l'utilisateur.

4. **Les deux éléments principaux** : sont le profil et le contexte : ils sont définis sur une base de méta modèle pour permettre l'expression des caractéristiques liés aux profils utilisateur et à leurs contextes.

4.1. Le méta modèle de profil :

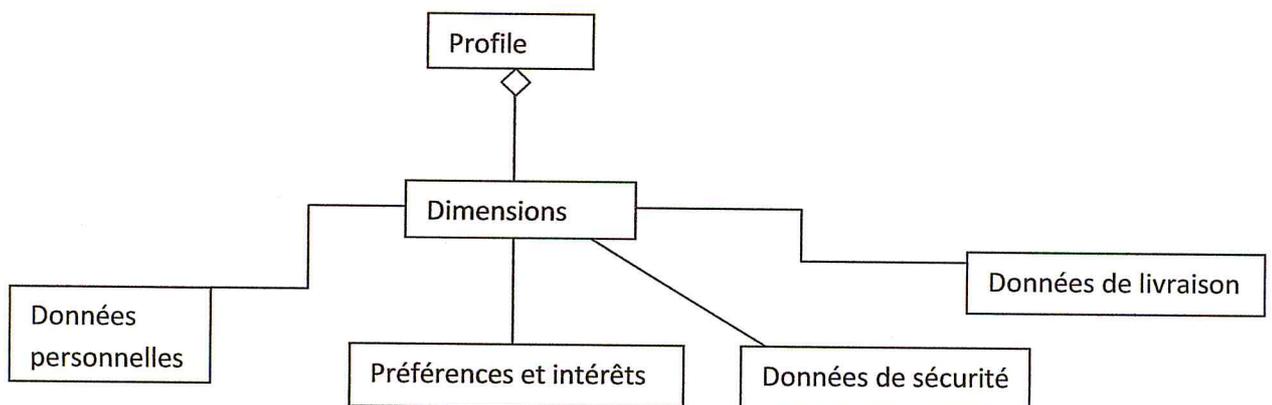


Figure 10 : méta modèle de profil.[03]

- 4.1.1. **Données personnelles** : se seront : l'âge, l'adresse et le numéro de carte bancaire en cas d'achat.
- 4.1.2. **Données de livraisons** : se seront toutes les informations comme : le délai de livraison, le mode de livraison ainsi que les frais de livraison.
- 4.1.3. **Les données de sécurité** : ce sont les informations pour garantir un bon déroulement de la transaction : la confidentialité des numéros de carte bancaire, des formulaires qui précisent les clauses du contrat effectué ainsi que les montants détaillé de chaque type de transaction.
- 4.1.4. **Préférences et intérêts** : cette partie va faire appel à un algorithme de Préférences Manning pour permettre une

acquisition implicite des préférences de l'utilisateur. (vue dans la partie antérieure)

- ✚ Notons que dans notre mémoire les informations de profils nécessaires seront uniquement les préférences et intérêts qui sont déduites à partir du log utilisateur.

4.2. Le méta modèle de contexte :

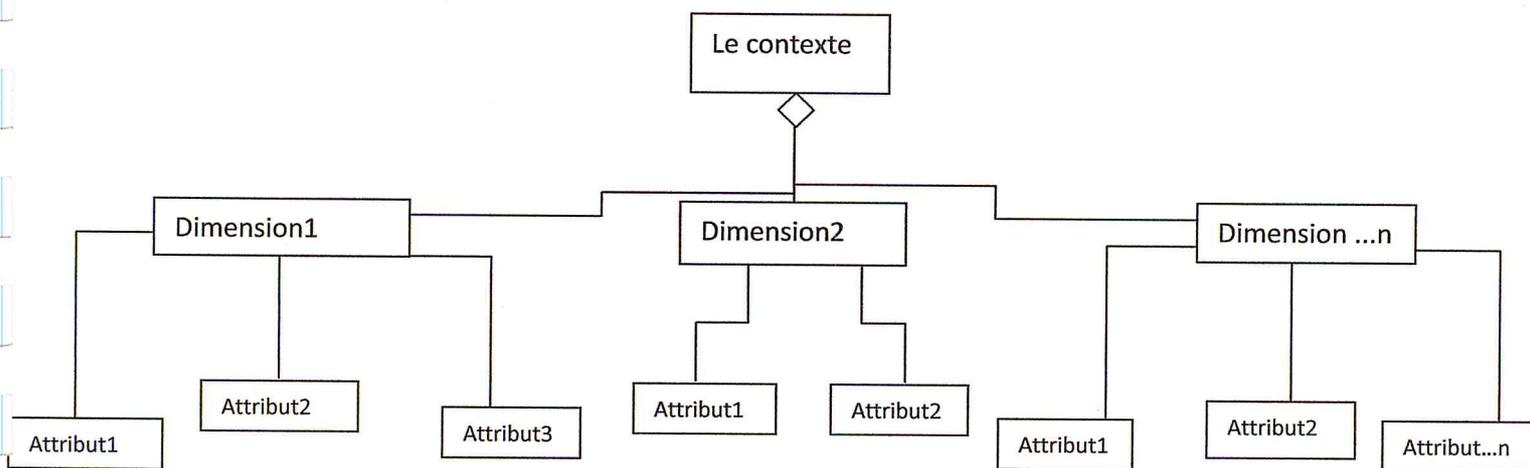


Figure11 : méta modèle de contexte.[03]

- ✚ Mais pour l'implémentation de notre système nous avons choisis un modèle de contexte avec une seule dimension pour une facilité d'implémentation et de testes.

Le contexte sera défini explicitement par l'utilisateur au moment de son accès à l'interface de communication avec le système.

5. Scénario d'analyse de la requête :

- 1) Mining préférences : à l'arrivée d'une nouvelle requête MDX, la première étape sera d'extraire les préférences de l'utilisateur et prendre en compte celle qui vérifie la condition du seuil imposé. Donc en sortie on obtient les préférences de l'utilisateur Pr (qui est un ensemble ordonnés de préférences).
- 2) Le contexte est fixé par l'utilisateur de manière explicite.

- 3) Le matching : il va consister à trouver la bonne combinaison entre le profil et le contexte afin d'obtenir un profil contextualisé qui va être par la suite intégré à la requête initiale. Une fois, le profil contextuelle acquis nous allons commencer la recommandation.
 - 4) La recommandation : elle va consister en la reformulation de la requête de recommandation tout en respectant le formalisme de recommandation choisis préalablement (approche hybride) ; donc :
 - i. Le traitement de cette requête bien qu'il va prendre en compte le profil et le contexte mais il va être similaire au traitement qui va prendre en compte un seul et unique paramètre car le fait de contextualiser le profil avant cette étape réduit le nombre de paramètres et revient à traiter une simple requête MDX.
 - ii. Les profils contextualisés qu'on va prendre en considération au moment de la recommandation obéiront également à un seuil de tolérance défini préalablement.
 - iii. Les recommandations jugées pertinentes seront classées et agrégées ensuite les Top K les plus pertinentes seront, elles seulement présentées à l'utilisateur.
 4. La mise à jour du profil et du contexte : cette opération se fait au fur et à mesure qu'un changement s'opère sur le profil ou le contexte.
- ✓ La partie stockage : ce sera une forme de base de données multidimensionnelle qui abritera trois bases de données, une pour les profils utilisateur, une pour les contextes et une autre pour les profils contextualisés qui résulte de l'étape de matching.

Remarque :

Les données utilisées seront un jeu de données sous forme de bases de données multidimensionnelles, ce dernier ayant été récupéré et intégré pour

permettre la création du cube et par la suite l'interrogation par requêtes MDX bien évidemment.

La base de données s'appelle :EchoPilote, elle sera définie en détail dans le chapitre suivant.

6. Qu'Est-ce qu'une requête MDX :

Le MDX (de l'anglais Multidimensional Expressions, expressions multidimensionnelles) est un langage de requête pour les bases de données OLAP, analogue au rôle de SQL pour les Bases de données relationnelles. C'est aussi un langage de calcul avec une syntaxe similaire à celle des tableurs.[04]

Le langage des expressions multidimensionnelles propose une syntaxe spécialisée pour interroger et manipuler les données multidimensionnelles mémorisées dans un cube OLAP. Bien qu'il soit possible de traduire certaines expressions dans le langage SQL traditionnel, cela nécessite une syntaxe SQL souvent maladroite même pour des expressions MDX très simples. MDX a été adopté par une large majorité de fournisseur de la technologie OLAP et est devenu un standard de facto pour les systèmes OLAP.

6.1. Types de données MDX :

MDX possède six types de données primaires, qui sont le scalaire, la dimension et la hiérarchie, le niveau, le tuple, le membre, et l'ensemble.

Le scalaire : est soit un nombre, soit une chaîne de caractères. Il peut être explicitement défini, par exemple le nombre 5 ou la chaîne de caractères "OLAP". Il peut aussi être retourné par une fonction MDX comme par exemples 'Aggregate' qui retourne un nombre,'UniqueName' retourne une chaîne de caractères, 'Value' renvoie un nombre ou chaîne, etc...

La dimension : est une dimension d'un cube. Les dimensions organisent les informations dans un cube, ce sont les axes d'analyse des indicateurs. MDX considère que les dimensions sont mutuellement indépendantes. Une dimension peut contenir quelques membres (voir ci-dessous) organisés en

hiérarchies. La hiérarchie d'un cube olap peut être spécifiée par son nom unique tel que [Time].[Fiscal] ou retourné par une fonction MDX comme 'Hierarchy' Les hiérarchies sont contenues dans des dimensions, les spécifications MDX dans OLEDB for OLAP ne font pas la distinction entre les types de données dimension et hiérarchie. Quelques implémentations, telle que Microsoft Analysis Services les traitent différemment.

Le niveau dans une hiérarchie de dimensions peut être défini par son nom unique, par exemple [Time]. [Fiscal]. [Month] ou retourné par une fonction MDX 'Level'.

Un membre :est un membre d'une hiérarchie de dimensions. Il peut être défini par son nom unique (exemple : [Time]. [Fiscal]. [Month]. [August 2006]), par un nom dérivé (exemple : [Temps]. [Fiscal]. [2006]. [Q2]. [August]) , ou être retourné par une fonction MDX : 'PrevMember', 'Parent', 'FirstChild'.

Un tuple :est une collection ordonnée d'un ou plusieurs membre de différentes hiérarchies. Les tuples peuvent être définis explicitement comme une liste de membres, par exemple ([Time].[Fiscal].[Month].[August 2006], [Customer].[By Geography].[All Customers].[USA], [Measures].[Sales]) ou être retournés par une fonction MDX : 'Item'.

Un ensemble (set) : est une collection ordonnée de tuples partageant les mêmes hiérarchies (dimensions). Il peut être défini explicitement comme une liste de tuples, par exemple {([Measures].[Sales], [Time].[Fiscal].[2006]), ([Measures].[Sales], [Time].[Fiscal].[2007])}, ou retourné par une fonction MDX : 'Crossjoin', 'Filter', 'Order', ou 'Descendants'.

III. Conclusion :

Dans le chapitre que nous venons de présenter est apparu l'architecture du système que nous avons mis en place.

On vient de montrer ses aspects sur un plan théorique avec les modules de traitement et de communication le contenant ainsi que les modèles de données sur lesquels il doit se basé.

Chapitre 5 : Présentation du système.

Une fois, notre vue sur la conception du système établit, place maintenant au chapitre suivant qui est : l'implémentation. Il va nous permettre de montrer notre proposition dans un cadre applicatif, en passant par la mise en œuvre, les logiciels utilisés ainsi que les étapes de réalisation et d'exécution.

I. Introduction :

Ce chapitre va aborder la partie implémentation comme énoncé préalablement, nous allons exploiter le langage C# et différents modules composant respectivement SQL Serveur 2008 et Visuel Studio 2008 Expresse.

L'intérêt porter à ce type de logicielle et de langage est motiver par les caractéristique et les fonctionnalités que chacun d'eux apporte au niveau de notre application en terme de fonctionnalités et de portabilités.

II. Présentation de SQL Serveur 2008 :

Microsoft SQL Server est un système de gestion de base de données (abrégié en SGBD ou SGBDR pour : Système de gestion de base de données relationnelles) développé et commercialisé par la société Microsoft.

Bien qu'il ait été initialement Co-développé par Sybase et Microsoft, Ashton-Tate a également été associé à sa première version, sortie en 1989. Cette version est sortie sur les plates-formes Unix et OS/2. Depuis, Microsoft a porté ce système de base de données sous Windows et il est désormais uniquement pris en charge par ce système.

En 1994, le partenariat entre les deux sociétés ayant été rompu, Microsoft a sorti la version 6.0 puis 6.5 seul, sur la plate-forme Windows NT.

- Microsoft a continué de commercialiser le moteur de base de données sous le nom de SQL Server
- Tandis que Sybase, pour éviter toute confusion, a renommé Sybase SQL Server en Sybase Adaptive Server Enterprise.

Microsoft SQL Server fait désormais partie de la stratégie technique de Microsoft en matière de base de données. Le moteur MSDE, qui est la base de SQL Server, doit à terme remplacer le moteur Jet (celui qui gère les bases Access) dans les applications telles que Exchange et Active Directory.

La version 2005 de SQL Server est sortie le 3 novembre 2005 en même temps que Visual Studio 2005. La prise en charge de Windows Vista et de Windows Server 2008 n'a été ajoutée qu'à partir du Service Pack 2 (SP2). Actuellement le Service Pack 3 est disponible.

La version 2008 de SQL Server (nom de code Katmai) est disponible depuis août 2008. Elle est actuellement au niveau de service pack 2. Elle est disponible en 9 langues, dont le français.

III. Présentation de Visual Studio 2008 Express :

Microsoft Visual Studio Express est un ensemble d'environnements de développement intégrés gratuits développé par Microsoft. Il s'agit d'une version allégée de Microsoft Visual Studio.

Visual Studio Express est composé de ces différents produits, chacun supportant un seul langage : Visual Basic, C#, C++, J#, Web Developer et SQL Server.

L'installation passe par le téléchargement d'un logiciel de moins de 3 Mo qui télécharge et installe la version de Visual studio choisie (356 Mo pour C#). Le choix d'installation du répertoire par défaut est défini lors de la première installation d'une des versions choisie (version, langage, internationalisation).

- ✚ Visual C# Express Edition est la partie de Visual Studio Express Edition permettant la programmation en C#. (cette partie permet d'implémenter les différents algorithmes proposés dans le chapitre précédent).
- ✚ La partie SQL Serveur permet d'assister SQL Serveur dans la création du cube

Comme on va le voir dans la section qui suit.

Remarque : nous remarquons qu'il est difficile et parfois même impossible de dissocier SQL Serveur de Visual Studio surtout dans le domaine de développement des applications décisionnelles. Comme c'est le cas dans notre travail les deux parties fusionnent et donnent naissance à une application homogène.

IV. Modules et type de données utilisées :

1. L'utilisation du Business Intelligence Development Studio:

Business Intelligence Development Studio est un environnement intégré dans SQL Serveur 2008 qui permet de développer des constructions Business Intelligence, telles que des cubes, des sources de données, des rapports et des lots ' Integration Services'.

Business Intelligence Development Studio contient des modèles de projets qui fournissent un contexte pour développer des constructions spécifiques. Par exemple, de choisir un projet 'Analysis Services' si notre objectif est de créer une base de données 'Analysis Services' qui contient des cubes, des dimensions ou des modèles d'exploration de données (et c'est le cas dans notre application).

Dans Business Intelligence DevelopmentStudio,on peut aussi développer des projets dans une solution indépendante de tout serveur. Par exemple, inclure un projet 'Analysis Services', un projet' Integration Services' et un projet' Reporting Services' dans la même solution.

On peut également déployer les objets sur un serveur de test à des fins de test lors du développement, puis déployer ultérieurement le résultat des projets sur un ou plusieurs serveurs intermédiaires ou de production.

2. La base de données : Adventure Works DW 2008 :

Il existe plusieurs versions des bases de données AdventureWorks . Il existe des versions de SQL Server 2000, 2005 et 2008, mais notre travail est concentré sur les versions 2008. Pour SQL Server 2008, il existe 3 bases de données.

OLTP AdventureWorks : est une norme de données en ligne contenant le traitement des transactions par an de données pour un fabricant de bicyclettes fictive (Adventure Works Cycles). Cette base de données est utile pour les scénarios de démonstration, et comprend des données pour la fabrication, ventes, achats, gestion des produits, gestion des contacts et des ressources humaines.

Chapitre 6 : l'implémentation

AdventureWorksDW : est un entrepôt de données contenant une année de données dans un schéma en étoile, contenant effet multiple et tables de dimensions. Cette base de données est excellente pour la construction de SQL Server Analysis Services (SSAS) des cubes OLAP et la démonstration de la Business Intelligence (BI) les capacités de SQL Server et technologies clientes BI tels que MOSS 2007, Reporting Services et Excel.

Le LT AdventureWorks (Light) : est une base de données exemple très simplifiée et plus petit qui est utile pour ceux qui sont nouveaux à la technologie de base de données relationnelle.

Pour nos testes la base de données : Adventure Works DW était la plus adaptées à nos besoins vue sa schématisation en étoile.

3. Création d'une base Analysis Services :

Tout d'abord pour pouvoir passer à la création de notre indispensable cube nous devons passer par quelques étapes importantes ; la première étant la création d'une nouvelle source de données qui dans notre cas sera tout simplement l'intégration de la source de données AdventureWorks2008 pour ainsi pouvoir utiliser ses tables pour créer un schéma en étoile qui est le suivant :

Une fois la vue de la source de données acquise place à la prochaine étape :

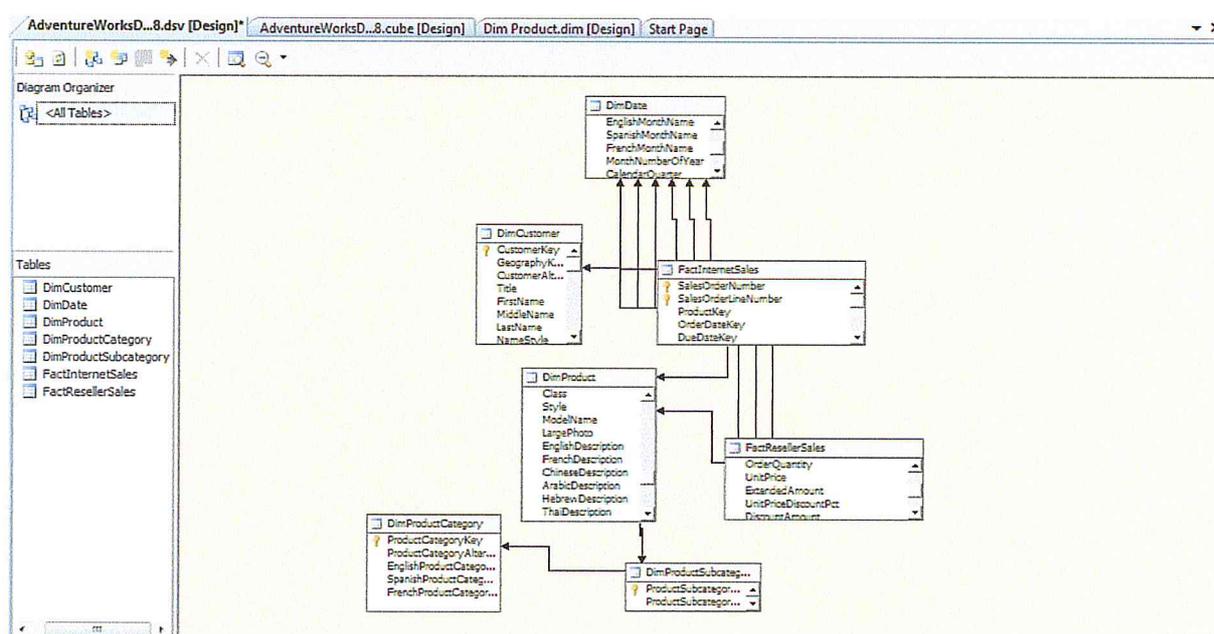


Figure12 : schéma d'organisation des tables.

4. La création du cube de données :

On utilise pour ce fait le module cube Wizard qui va permettre à partir de table existante de configurer les mesures et dimensions du cube.

Dans notre cas la mesure sera : la vente par internet 'Fact Internet Sales'.

Quant aux dimensions nous avons choisis : produit 'Dim Product', Client 'Dim Customer', Promotion 'Dim Promotion', et la dimension qui va être prise en considération en tant que dimension du contexte est la période ou le temps qui est représentée par 'Dim Date'.

La dimension produit contient une hiérarchie définie comme suit : Dim Product → Dim Subcatégorie

→ Dim Catégorie.

Il faut spécifier que les attributs de dimensions sont choisis selon le besoin.

Une vue du cube est disponible grâce à l'onglet cube structure du module Visual Studio de SQL Serveur2008.

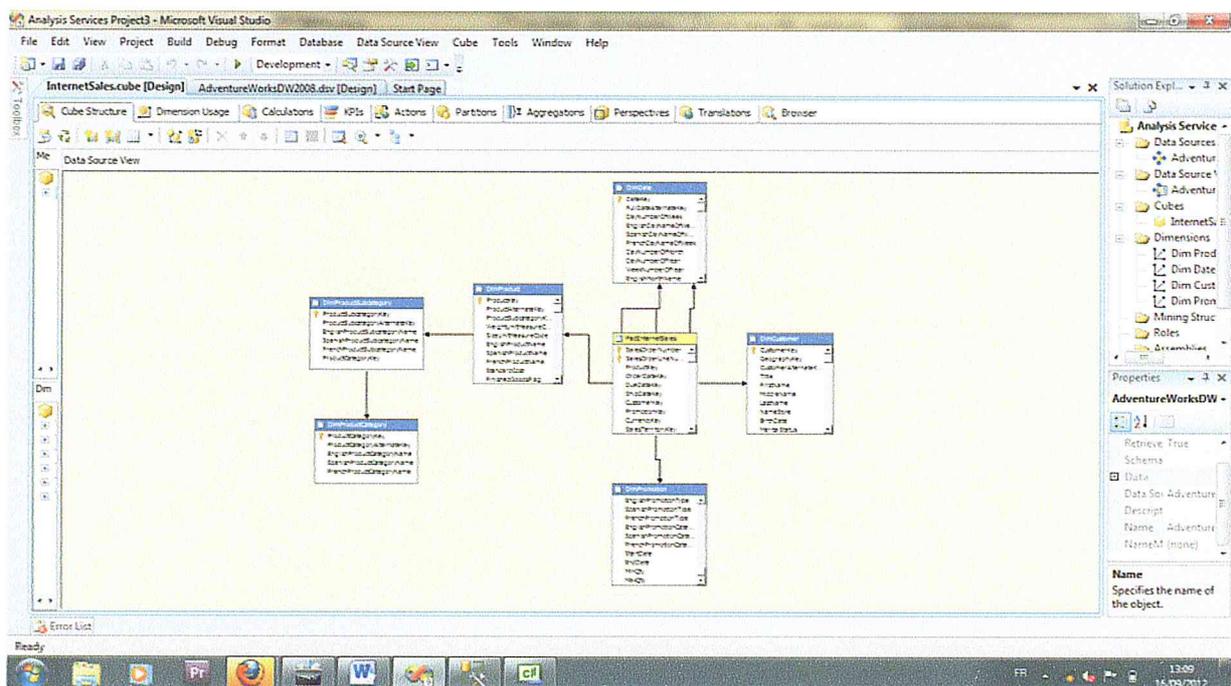


Figure 13 : création du cube.

Chapitre 6 : l'implémentation

Une fois le cube créé la partie interrogation peut commencer.

5. La partie lancement de requêtes :

On a interrogé notre cube de données à l'aide de requêtes MDX ; ces dernières vont être lancées à partir de C# ; donc vu que c'est un code et son écriture prend beaucoup plus de temps que de lancer une requête via une interface dédiée à ceci, nous avons choisis de prédéfinir trois requêtes à l'avance pour permettre un essai ; ces requêtes seront accessibles via une interface graphique comme ceci :

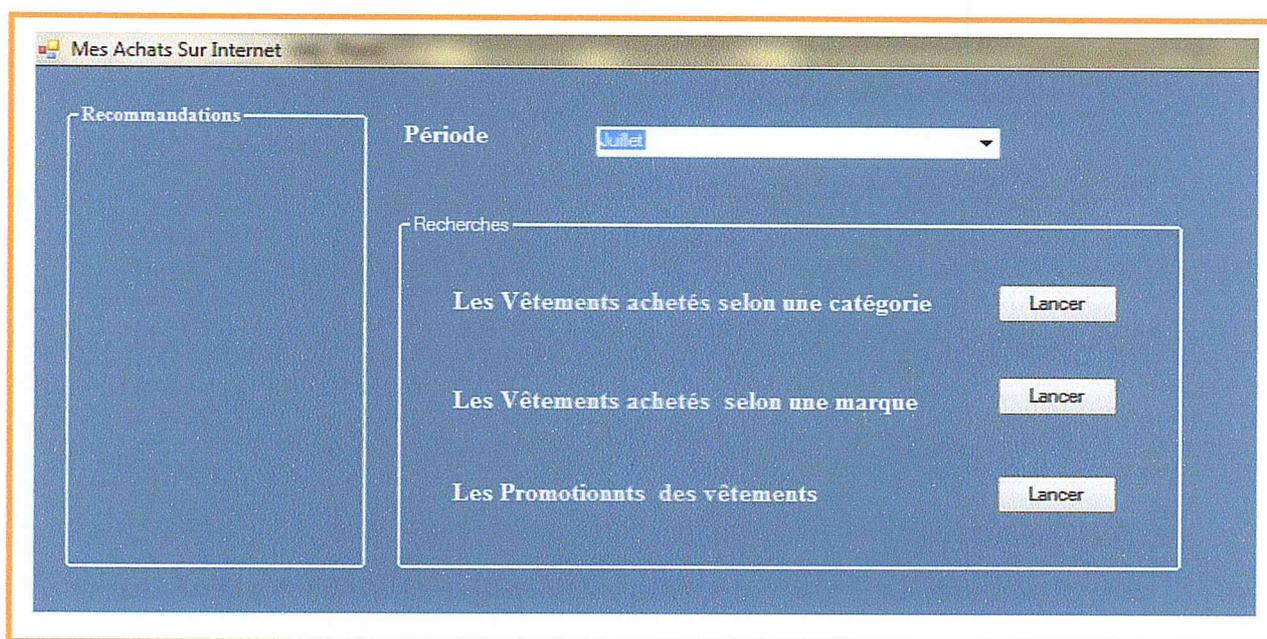


Figure14 : lancement de la requête.

6. La détermination du contexte :

Elle se fera de manière explicite à partir d'une liste déroulante sur notre interface contenant tous les mois de l'année. Ainsi l'utilisateur pourra définir la période dans laquelle il se trouve au moment du lancement de la requête. L'interface sera présentée comme ceci :

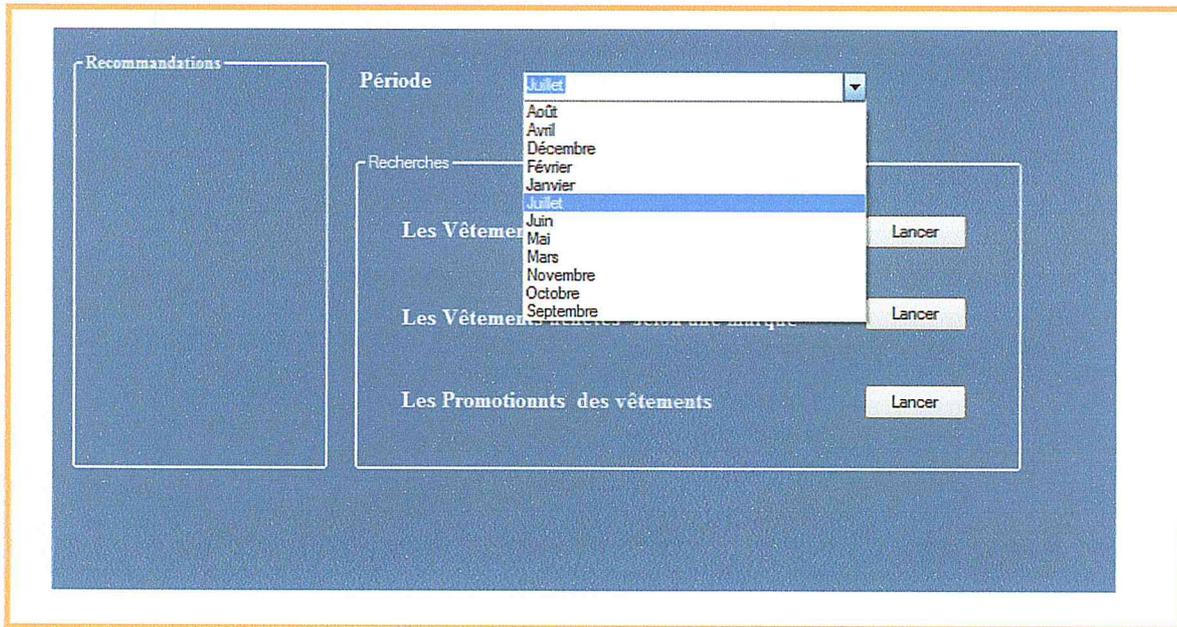


Figure 15 : choix du contexte.

7. La recommandation :

Quant au résultat il sera recommandé instantanément au moment du lancement de la requête et du choix du contexte, si l'utilisateur aime et valide la recommandation, le résultat de la requête recommandé lui sera fournis par contre si l'utilisateur n'est pas satisfait il aura la possibilité de choisir une requête recommandé alternative à celle proposé ou bien continuer la navigation.

V. Conclusion :

Le chapitre qu'on vient d'aborder nous nous met dans le cadre du déroulement de l'application à travers les vues obtenues ainsi que les interfaces graphique qui vont synchroniser les étapes de notre traitement.

L'interface graphique propose un environnement familier à l'utilisateur.

En montrant notre cadre applicatif nous venons de conclure notre chapitre ainsi que notre mise en place du système personnalisé recommandé contextualisé, mais ce travail consiste en un exemple de démonstration et non pas une application entreprise, mais elle peut être adapté à tout besoin de chaque entreprise.

Conclusion

Dans ce mémoire, nous présentons un système de recommandation de requêtes MDX qui est un système qui permet l'analyse de la requête en l'adaptant au besoin et contexte de l'utilisateur.

Notre système tient compte des précédentes navigations réalisées dans le cube, utilisant un fichier log qui va contenir les requêtes MDX lancées précédemment par l'utilisateur prenant ainsi en compte une caractéristique essentielle en OLAP : les hiérarchies, car cela va permettre à l'aide d'un algorithme de Mining Préférences, d'extraire les préférences de l'utilisateur.

Ce qui permet par la suite de proposer des recommandations en adéquation avec le profil de l'utilisateur.

Quant au contexte il est défini par le décideur au moment du lancement de sa requête.

Notre approche est implémentée à l'aide de deux principaux logiciels ; SQL Serveur 2008 et Visual C# 2008 Express Edition. Les expérimentations que nous avons menées se sont effectuées sur des données synthétiques (une base de données multidimensionnelle qui a permis la création du cube à l'aide du module Analysis Services du Framework de SQL Serveur : SQL Serveur Business Intelligence Development Studio) montrent que des recommandations peuvent être calculées efficacement et que notre système peut être réglé afin d'obtenir des recommandations de bonne qualité et une meilleure prise en compte des préférences et du contexte de l'utilisateur ce qui donne une meilleure précision dans les résultats retournés.

Perspectives

Nos travaux futurs devraient nous permettre de :

- mener des expériences sur des données réelles afin d'améliorer la qualité des requêtes recommandées.

– proposer d’autres instanciations de notre cadre comme en personnalisant les requêtes au plus tôt dans la démarche...

– étendre notre définition de requête pour prendre en compte une plus grande partie du langage MDX.

A plus long terme, nous voudrions contribuer à un système collaboratif de management de requêtes comme et concevoir une plate-forme de génération de recommandations de requêtes MDX en donnant aux utilisateurs et aux administrateurs OLAP la possibilité d’adapter l’approche à leurs besoins en proposant diverses méthodes de calcul des sessions et/ou requêtes candidates.

Cette plate-forme intégrera notre cadre générique et devrait également inclure une nouvelle méthode, prenant en compte un plus grand nombre de mesures ainsi que des techniques qui permettront à l’utilisateur de rester en dehors du processus de traitement de la requête (c’est-à-dire que tout se fera implicitement).

Bibliographie

- [1] Bouzeghoub, M., Kostadinov, D. Personnalisation de l'information : aperçu de l'état de l'art et définition d'un modèle flexible de profils, CORIA'05, pp. 201–218.
- [2] Koutrika, G., Ioannidis, Y.: Answering queries based on preference hierarchies. In: Proc. VLDB, Auckland, New Zealand 2008.
- [3] Bentayeb, F., Boussaid, O., Favre, C., Ravat, F., Teste, Personnalisation dans les entrepôts de données : bilan et perspectives, 5ème journées sur les Entrepôts de Données et l'Analyse en ligne (EDA'09), Revue des Nouvelles Technologies de l'Information, RNTI-B-5, Cepadues Editions.
- [4] Jerbi, C., Ravat, F., Teste, O., Zurfluh, G. Applying Recommendation Technology in Olap Systems. 11th International Conference on Enterprise Information Systems (ICEISE 09).
- [5] Bellatreche, L. Giacometti, A. Marcel, P. Mouloudi, H. and Laurent, D. A personalization framework for OLAP queries. In DOLAP '05: Proceedings of the 8th ACM international workshop on Data warehousing and OLAP, pages 9–18, New York, NY, USA, 2005. ACM.
- [6] Mansmann, S., Scholl, M. H.: Visual OLAP: A New Paradigm for Exploring Multidimensional Aggregates. In Proceedings of IADIS International Conference on Computer Graphics and Visualization (MCCSIS'08), Amsterdam, The Netherlands, 24 - 26 July, 2008, pp. 59-66.
- [7] Golfarelli, S. Rizzi: Expressing OLAP Preferences. Berlin Heidelberg, Scientific and Statistical Database Management, LNCS, vol. 5566/2009, pp. 83-91.
- [8] Nachida Rezoug, Omar Boussaid, Fahima Nader: Profilisation et contextualisation des systèmes OLAP : Aperçu de l'état de l'art. In: CIIA, Vol. 825 CEUR-WS.org (2011).

Bibliographie

- [9] Rizzi, S, « OLAP Preferences: a Research Agenda », 10th International Workshop on Data Warehousing and OLAP (DOLAP'07), ACM, pp.99-100, Lisbon (Portugal), Novembre 2007.
- [10] [Sapia, et al., 1998] C. Sapia, M. Blaschka, G. Höfling, B. Dinter, V, « Extending the E/R Model for the Multidimensional Paradigm », ER Workshops, pp. 105-116, 1998.
- [11] Thalhammer, et al., 2001 T. Thalhammer, M. Schrefl, M. Mohania, « Active Data Warehouses: Complementing OLAP with Analysis Rules », Data and Knowledge Engineering 39(3), 241–269, 2001.
- [12] Vassiliadis, et al., 1999 P. Vassiliadis, T.K. Sellis, « A Survey of Logical Models for OLAP Databases », SIGMOD Record, ACM, 28(4), pp.64-69, 1999.
- [13] Wang, et al., 2005 Wang, H., Li, J., He, Z., Gao, H., « OLAP for XML Data », 5th Intl. Conf. on Computer and Information Technology (CIT), IEEE Computer Society, pp. 233–237, 2005.
- [14] Kimball, 1996, KIMBALL Ralph. The Data Warehouse Toolkit. New York, NY, USA : John Wiley & Sons, 1996, 464 p. ISBN 0-47115-337-0
- [15] Lee et al., 2006, LEE Ki Yong, KIM Myoung-Ho. Efficient Incremental Maintenance of Data Cubes. In : DAYAL U., WHANG K., LOMET D.B., ALONSO G., LOHMAN G.M., KERSTEN M.L., CHA S.K., KIM Y. (Eds.). Proceedings of the 32nd International Conference on Very Large Data Bases, September 12-15, 2006, Seoul, Korea. New York, NY, USA : ACM Press, 2006, pp. 823-833. ISBN 1-59593-385-9.
- [16] Carmel Domshlak, Avigdor Gal, and Haggai Roitman, Rank Aggregation for Automatic Schema Matching In: IEEE Trans. Knowl. Data Eng., Vol. 19, Nr. 4 2007, p. 538-553. James Lin, Michael.
- [17] Thomsen, and James A. Landay, A visual language for sketching large and complex interactive designs. In: CHI ACM 2002, p. 307-314.

Bibliographie

[18] B.N. Schilit, D.M. Hilbert, J. Trevor, IEEE Wireless Communications, Volume: 9 Issue: 5, Oct 2002, pp. 46 -54.

[19] Google image : image DW. Caractéristiques.

[20] Olapdéfinitions: [altaplana .com/oalp/glossary.html](http://altaplana.com/oalp/glossary.html).

[21] DW actif/ndiki.fr/topic3/index.html.