Chapitre 3

État de l'art

3.1 Introduction

Dans le cadre de notre proposition de personnalisation des analyses dans les entrepôts de données basée sur une évolution du schéma de ce dernier, nous présentons dans ce chapitre un état de l'art qui porte sur deux thématiques : il s'agit de l'évolution de modèle dans les entrepôts de données d'une part et de la personnalisation d'autre part. Ces deux thématiques sont présentées dans ce chapitre de façon indépendante. Elles seront liées par la suite, lors de la présentation de nos contributions.

Ainsi, nous présentons dans la section 3.2 un état de l'art sur les travaux traitant de la problématique de l'évolution de modèle. Cet état de l'art est précédé d'un positionnement de cette problématique. Puis, dans la section 3.3, après avoir dressé un panorama des travaux en matière de personnalisation dans différents domaines que sont la recherche d'informations (RI), les bases de données (BD) et l'interaction homme-machine (IHM) nous évoquons les quelques travaux émergents réalisés en matière de personnalisation dans les entrepôts de données. Enfin, nous concluons ce chapitre dans la section 3.4.

3.2 Évolution de modèle dans les entrepôts de données

3.2.1 Introduction

Les modèles multidimensionnels classiques [CT98, Kim96, Leh98] considèrent les faits comme la partie dynamique des entrepôts de données et les dimensions comme des entités statiques. L'historisation des données est assurée par la dimension *Temps*.

Les autres dimensions sont supposées temporellement invariantes, compte tenu de l'hypothèse selon laquelle les dimensions sont supposées être orthogonales les unes par rapport aux autres et donc orthogonales par rapport à la dimension *Temps*. Cependant, en pratique, des changements peuvent se produire dans le schéma des dimensions et plus généralement sur l'ensemble du schéma de l'entrepôt. En effet, comme nous l'avons souligné précédemment, le schéma peut être amené à évoluer suite à l'évolution des sources de données ou des besoins d'analyse.

La technologie d'entreposage de données s'est inspirée et s'inspire encore aujourd'hui des travaux réalisés dans le domaine des bases de données. Par exemple, les travaux sur les vues [Han87], sur les index [Val87], etc. ont été adaptés pour être appliqués aux spécificités des entrepôts. En ce qui concerne le domaine d'intérêt de ce chapitre, la mise à jour des bases de données [Rod92], les bases de données temporelles [SA86] et les bases de données multiversions [TG89] ont nourri les travaux sur l'évolution des entrepôts de données.

Ainsi, on retrouve aujourd'hui, dans la littérature, différents travaux sur la mise à jour de schéma dans les entrepôts de données, le versionnement de ces derniers pour prendre en compte l'évolution des dimensions, etc. Comme nous l'avons présenté dans [FBB07f], nous proposons ici de classer les différents travaux selon deux familles que nous baptisons respectivement : «modélisation temporelle» et «mise à jour de schéma». Ces deux familles se distinguent respectivement par la conservation ou non de la trace des évolutions subies par le schéma. Chacune de ces familles présente différentes approches que nous nous proposons d'étudier et de comparer.

La suite de cette section est organisée de la façon suivante. Tout d'abord, nous évoquons, sur un exemple issu du cas LCL, les évolutions que peuvent subir un modèle et l'impact qu'elles ont en terme de cohérence des analyses. Ensuite, nous présentons les travaux existants s'intéressant à cette problématique. Puis, nous nous proposons de discuter ces travaux.

3.2.2 Évolution de modèle dans les entrepôts : un exemple illustratif

Dans cette section, nous nous attachons à décrire les évolutions possibles d'un modèle d'entrepôt de données. Nous classons ces évolutions selon deux types : les évolutions sur le schéma d'une part et les évolutions sur les données d'autre part. Pour illustrer ces évolutions et leurs impacts, nous nous proposons de baser notre discours sur un modèle implémenté en relationnel, issu du cas bancaire LCL. Pour illustrer notre propos, nous parlons de table, de clé, etc. En particulier, nous parlerons de table de faits par opposition aux autres tables : les tables de dimension. Ces

dernières représentent donc à la fois les dimensions elles-mêmes, qui sont caractérisées par les tables directement liées à la table de faits et les niveaux de granularité qui composent leurs hiérarchies.

Le schéma multidimensionnel de la figure 3.1 permet d'analyser la mesure `NBI` (Net Banking Income). Le NBI correspond à ce que rapporte un client à l'établissement bancaire. Cette mesure est analysée selon les dimensions `CUSTOMER` (client), `AGENCY` (agence) et `YEAR` (année). La dimension `AGENCY` présente une hiérarchie. Il est ainsi possible d'agréger les données selon le niveau de granularité `COMMERCIAL_UNIT` (unité commerciale) qui est un regroupement d'agences par rapport à leur localisation géographique. Ces unités commerciales sont elles-mêmes regroupées selon un deuxième niveau de granularité : le niveau `DIRECTION`. Ce schéma constitue notre schéma initial. À chaque fois que nous explicitons une évolution, elle est réalisée sur ce schéma de base.

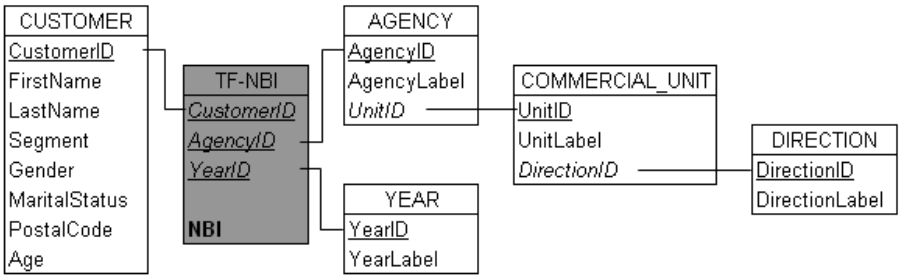

Fig. 3.1 – Schéma multidimensionnel pour observer le NBI

3.2.2.1 Évolution du schéma multidimensionnel

Un schéma multidimensionnel peut subir des évolutions qui peuvent remettre en cause le schéma existant en ayant des impacts d'importance variable sur les données. Par exemple, les évolutions de schéma impactant la table des faits ont en général des conséquences importantes sur les données entreposées. En effet, la volumétrie de l'entrepôt dépend généralement de celle de la table des faits. Ainsi l'impact sur les données de la table des faits peut être considérable. Nous évoquons les évolutions de schéma dans l'ordre décroissant selon l'importance de l'impact sur l'entrepôt.

Tout d'abord, une évolution possible est l'ajout d'une dimension. Cela équivaut à augmenter le niveau de détail de la table des faits (les faits y seront plus détaillés), puisque les mesures présentes dans la table des faits seront décrites par une dimension supplémentaire et présenteront ainsi davantage de descripteurs. Par exemple, il s'agirait d'ajouter une dimension `PRODUCT`, identifiant ainsi un NBI pour une année,

un client, une agence et un produit donnés. L'impact sur les données est considérable puisqu'il s'agit non seulement d'ajouter une table de dimension mais également de recalculer l'ensemble des données de la table des faits lorsque les données sources nous permettent de recalculer les mesures pour les anciens faits.

Une autre évolution possible est la suppression d'une dimension, qui permet de diminuer le niveau de détail de la table des faits (les faits y seront moins détaillées). Par exemple, il est possible de supprimer la dimension agence. Ainsi le NBI serait identifié pour une année et un client donnés. Là encore il faut recalculer les agrégats de la table de faits `TF_NBI` puisque l'identifiant `idAgency` serait supprimé.

Une autre modification qui touche la table des faits est l'ajout d'une mesure. Cette mesure peut être dérivée à partir d'une mesure existante. L'impact est moindre que lorsque l'on touche à une dimension, puisque cela ne remet pas en cause les données existantes de la table des faits. Cependant, cela nécessite de calculer pour chaque fait cette nouvelle mesure. Par exemple, on peut vouloir ajouter dans la table des faits `TF_NBI` une mesure telle que `OVERHEADS` correspondant aux frais de gestion d'un client, par agence et par année. Dans ce cas, cette mesure doit être calculée (à partir des sources de données) pour chacune des lignes de la table des faits. Cette dernière doit partager exactement les mêmes dimensions que les autres mesures de la table des faits. Si ce n'est pas le cas, il faut envisager la création d'une autre table des faits qui pourra partager une partie des dimensions existantes.

La suppression d'une mesure, quant à elle, touche également la table des faits. Néanmoins, aucun recalcul de la table des faits n'est nécessaire, puisqu'il s'agit seulement de «supprimer une colonne». Notons, que cette modification structurelle n'est possible que s'il existait plusieurs mesures pour analyser les faits.

Ensuite, viennent les modifications touchant aux hiérarchies de dimension, telles que l'ajout de nouveaux niveaux de granularité enrichissant une hiérarchie existante ou définissant une nouvelle hiérarchie. Par exemple, il est possible d'ajouter un niveau de granularité `PCS`, représentant les catégories socio-professionnelles, pour créer une hiérarchie sur la dimension `CUSTOMER`. La table `PCS` doit être créée et alimentée, la table de dimension `CUSTOMER` doit être enrichie par un attribut la reliant à `PCS`.

Dans le cas de la suppression de niveau de hiérarchie, l'importance de l'impact dépend de la localisation de ce niveau dans la hiérarchie. En effet, si le niveau est dans une position intermédiaire dans la hiérarchie, il faut assurer la cohérence en maintenant les liens nécessaires dans la hiérarchie. Par exemple, si le niveau `COMMERCIAL_UNIT` est supprimé, il faut assurer le lien entre les niveaux `AGENCY` et

DIRECTION. Si, par contre, c'est le niveau DIRECTION qui est supprimé, il n'y a pas de maintenance particulière à assurer, si ce n'est la suppression elle-même du niveau et celle du lien entre les niveaux COMMERCIAL_UNIT et DIRECTION.

Ces différentes modifications d'ordre structurel enrichissent (ajout de dimension, de mesure, de niveau de hiérarchie) ou appauvrissent (suppression de dimension, de mesure, de niveau de granularité) les analyses. Néanmoins, ces modifications ne remettent pas en cause la véracité, la cohérence des analyses. Le problème de cohérence des analyses se pose lors de l'évolution des données, comme nous allons le montrer dans ce qui suit.

3.2.2.2 Évolution des données

Quelle que soit l'évolution opérée sur le schéma, il est nécessaire de répercuter cette évolution au niveau des données elles-mêmes. Cette évolution nécessite parfois de disposer de données pour alimenter l'entrepôt, en particulier dans le cas d'ajout de mesure, d'ajout de niveau de hiérarchie, etc. Cela modifie entre autres le processus de chargement des données.

Il est plus difficile de définir de façon exhaustive les évolutions possibles des données que celles du schéma. De façon générale, les évolutions de données peuvent être de l'ordre de l'insertion, la suppression et la modification. Dans le contexte des entrepôts de données, ces trois opérations de base ont des conséquences différentes selon le concept sur lequel elles sont appliquées.

Envisageons tout d'abord le cas de la table des faits. L'insertion de données (de faits) correspond à la phase classique d'alimentation. Compte tenu de la non-volatilité et de l'historisation des données [Inm02], les faits ne sont pas amenés à être supprimés. En d'autres termes, il ne s'agit pas de supprimer de la table de faits un enregistrement complet. De la même façon, les données de la table des faits ne devraient pas être modifiées. Néanmoins, [RG06] évoquent la nécessité de mettre à jour les valeurs des mesures, lorsqu'elles ont fait l'objet d'erreurs ou lorsque les événements qu'elles traduisent évoluent et ce, contrairement au principe de non volatilité des données auquel répondent les entrepôts de données.

Concernant les tables de dimension, l'insertion de données (instances de dimension) correspond également à la phase classique d'alimentation. La suppression dans une table de dimension ne peut avoir lieu que si les données précédemment récoltées n'y font plus référence. Ainsi, l'ensemble du problème de cohérence des données, et donc des analyses, peut être ramené la plupart du temps au problème de la modification des données sur les instances de dimension. Nous revenons sur ce point par

la suite.

Kimball a évoqué ce problème en introduisant trois types de «Slowly Changing Dimensions» ou «dimensions changeantes à évolution lente» qui constituent en fait trois possibilités pour gérer les changements dans les structures multidimensionnelles [Kim96]. L'hypothèse de départ est de dire que l'identifiant de la dimension ne change pas, ce sont ses descripteurs qui évoluent. Par exemple, l'identifiant d'un client ne change pas, mais il peut changer d'adresse. La première solution est d'écraser l'enregistrement avec la nouvelle valeur. Cette solution engendre la perte de l'historique. Ainsi, cette solution est intéressante lorsque l'ancienne valeur de l'attribut n'a plus de sens ou qu'elle peut disparaître. La deuxième solution consiste à créer un enregistrement supplémentaire. Chaque enregistrement correspond alors à une description unique valide pendant une période donnée. Il s'agit en effet de conserver toutes les versions des membres de la dimension. Cependant, dans une telle représentation, les comparaisons des données le long des évolutions sont rendues difficiles, puisque les liens entre elles ne sont pas conservés, bien que les évolutions le soient. La troisième solution est de créer un champ conservant l'ancienne valeur de l'attribut dans le même enregistrement. Cependant des limitations existent pour cette solution, si par exemple il y a une succession de changements à prendre en compte, puisque des recouvrements entre les versions peuvent apparaître mais ne peuvent être traités.

Kimball a ainsi défini les bases de solutions permettant de gérer l'évolution des dimensions, en insistant sur le fait qu'il est important de conserver l'historisation des données telle que Inmon l'évoque dans sa définition d'un entrepôt de données [Inm02]. Mais dans quelle mesure cette historisation de données garantit une cohérence des analyses ? C'est ce que nous abordons dans ce qui suit.

3.2.2.3 Cohérence des analyses

Au-delà de pouvoir réaliser des analyses en concevant un entrepôt de données, l'objectif réel est de disposer d'un entrepôt de données qui assure la cohérence des analyses. L'atteinte de cet objectif est conditionnée largement par le fait que l'entrepôt de données soit un miroir de la réalité. De notre point de vue, le problème de l'évolution du modèle dans les entrepôts de données ne doit pas être dissocié du problème de cohérence des analyses. Ainsi, il faut savoir reconnaître les cas où la cohérence des analyses n'est pas mise en danger, même si l'historisation des données n'est pas assurée. Nous parlons de problème de cohérence des analyses lorsque l'évolution subie impacte les analyses en modifiant leurs résultats. Il s'agit d'un problème considérable puisque, par définition, les résultats des analyses sont utilisés

pour prendre des décisions.

Comme nous avons pu le remarquer précédemment, le problème de cohérence des analyses se pose essentiellement dans l'évolution des dimensions et de leurs hiérarchies. En effet, l'hypothèse classique d'indépendance (on parle aussi d'orthogonalité) des dimensions entre elles sous-entend l'indépendance des dimensions avec la dimension temps. Ceci implique que ces dernières sont temporellement invariantes. Or ce n'est pas le cas dans la réalité.

Si un attribut référence un autre niveau de la hiérarchie, nous parlons dans ce cas de «descripteur hiérarchique», la perte de l'historisation des données sur cet attribut induit forcément une incohérence des analyses, dans la mesure où le lien d'agrégation est modifié. Par exemple, dans le schéma de la figure 3.1, la modification de la valeur de l'attribut `UnitID` de la table `AGENCY` entraîne des changements considérables du point de vue de l'analyse [BMBT02]. En effet, on peut considérer que l'on veut obtenir une analyse en prenant en compte un «temps consistant», correspondant au fait que l'on considère les faits selon la période où ils sont valides : avant une certaine date ce NBI a été réalisé par une certaine unité commerciale, puis il est rattaché ensuite à une autre unité commerciale. Il est également possible de considérer que l'agence appartient encore à l'ancienne unité commerciale. Enfin, il peut être intéressant de considérer que l'agence a toujours appartenu à l'unité commerciale dans laquelle elle a été affectée nouvellement. On observe la même problématique de cohérence d'analyse lorsqu'un attribut d'un niveau est un «descripteur direct», c'est-à-dire un descripteur du niveau lui-même, tel que l'attribut `MaritalStatus` qui représente la situation familiale (marié, célibataire, etc.) et qui peut intervenir dans l'analyse pour réaliser un regroupement. Par contre, lorsqu'un descripteur direct n'intervient pas dans l'analyse, tel que l'attribut `FirstName`, l'historisation n'est pas nécessaire et le problème de cohérence des analyses ne se pose pas.

Après avoir présenté les évolutions que peut subir un modèle et avoir montré l'interaction entre l'historisation des données et la cohérence des analyses, nous nous attachons, par la suite, à évoquer les différents travaux qui permettent de gérer ces évolutions.

3.2.3 Évolution de modèle dans les entrepôts : l'existant

3.2.3.1 Mise à jour de modèle dans les entrepôts de données

Les travaux proposant la mise à jour de modèle sont caractérisés par le fait qu'ils ne présentent qu'un modèle. Les évolutions sont donc appliquées pour constituer

un nouveau modèle. Ainsi la traçabilité des différentes évolutions n'est pas assurée. Nous avons classé ces travaux en trois courants.

Un premier courant est la proposition d'opérateurs pour faire évoluer le modèle. Dans [HMV99], les auteurs proposent un modèle formel pour la mise à jour des dimensions et de leur hiérarchie, en définissant des opérateurs qui répondent non seulement à une évolution des instances des dimensions, mais également à une évolution structurelle des dimensions, telle que l'ajout d'un niveau de granularité en fin de hiérarchie. Ils étudient également l'effet de ces mises à jour sur les vues matérialisées et proposent également un algorithme pour réaliser leur maintenance de façon efficace.

Dans [BSH99], les auteurs proposent, non seulement des évolutions au niveau des dimensions, mais également au niveau des faits. L'évolution qu'ils proposent est réalisée à un niveau conceptuel, indépendant de l'implémentation. Ils proposent ainsi une algèbre comprenant quatorze opérateurs d'évolution qui peuvent être combinés pour réaliser des opérations d'évolution complexes. Par exemple, il est proposé d'ajouter un niveau et ce, à n'importe quel endroit dans la hiérarchie de dimension, contrairement à ce qui est possible dans l'approche proposée par [HMV99]. Ce travail est étendu dans [Bla00] en proposant également la propagation de ces changements du niveau conceptuel vers le niveau logique.

Ces travaux ont été exploités dans [BG02] afin de proposer un gestionnaire d'entrepôts qui permet de gérer la création et l'évolution du schéma de l'entrepôt et ce, de façon indépendante du mode de stockage des données (relationnel, etc.).

Un deuxième courant s'est s'inspiré des travaux sur les opérateurs d'évolutions en se focalisant sur la création de nouveaux niveaux dans les hiérarchies de dimension. L'objectif est de s'intéresser à comment créer ces niveaux, non pas à comment représenter cette opération. Ainsi, il s'agit d'une mise à jour du modèle de l'entrepôt qui ne remet pas en cause la cohérence de l'analyse des données existantes puisqu'il s'agit d'un enrichissement du modèle. On peut citer le travail proposé dans [MT06] qui permet d'enrichir des hiérarchies de dimension à la fois au niveau de la structure et des données et ce, de façon automatique. En partant du principe qu'une hiérarchie de dimension représente des relations sémantiques entre des valeurs, ils proposent d'exploiter les relations d'hypéronymie («*is-a-kind-of*») et de méronymie («*is-a-part-of*») de WordNet[1]. Les niveaux de granularité sont créés en fin de hiérarchie.

Le troisième courant se base sur l'hypothèse qu'un entrepôt de données est un

[1] http ://wordnet.princeton.edu/

ensemble de vues matérialisées construites à partir des sources de données [Bel02]. Dans [HMV99], les auteurs se sont intéressés à la maintenance de vues pour propager l'évolution du modèle sur les cubes de données, représentés par des vues. Dans [Bel02], il s'agit de s'intéresser à la maintenance de vues matérialisées induite directement par une évolution des sources de données. Ainsi, il s'agit de ramener le problème de l'évolution des sources de données à celui de la maintenance des vues. La prise en compte d'évolution suite à des besoins est proposée à travers l'ajout d'attributs dans les vues et la modification de domaine de définition des attributs, tous deux réalisés par l'administrateur. Nous renvoyons le lecteur, pour de plus amples détails sur la maintenance de vues matérialisées dans ce contexte, vers l'état de l'art proposé par [BBDG05] sur la maintenance de vues matérialisées issues de sources de données hétérogènes.

Nous avons présenté trois courants s'inscrivant dans une mise à jour du modèle de l'entrepôt. Dans le premier courant, les opérateurs permettent de faire évoluer le modèle ; l'évolution des données est succinctement évoquée dans ces travaux. Le deuxième courant s'est intéressé précisément à comment réaliser une évolution telle que la création de nouveaux niveaux de granularité dans les hiérarchies de dimension. Enfin le troisième courant permet une évolution du modèle induite directement par l'évolution des sources de données, en posant l'hypothèse qu'un entrepôt est un ensemble de vues matérialisées. Ces trois courants répondent à des problématiques différentes. Le premier permet de proposer une évolution du modèle de l'entrepôt pour répondre à un besoin d'évolution traité par l'administrateur. Le deuxième propose une solution pour trouver les données nécessaires afin de réaliser une évolution spécifique dans le but de répondre à des besoins d'évolution liés davantage à l'analyse. Enfin, le troisième courant répond plus particulièrement à un besoin d'évolution en réponse à l'évolution des sources de données.

3.2.3.2 Modélisation temporelle des entrepôts de données

Les travaux proposant une modélisation temporelle de l'entrepôt s'opposent à ceux présentant une mise à jour de modèle sur le plan de l'historisation des changements. En effet, les approches se distinguent sur la traçabilité des évolutions subies par le modèle. Pour assurer cette traçabilité, des extensions temporelles sont nécessaires pour enrichir le modèle. Nous distinguons alors trois courants qui utilisent des étiquettes temporelles à différents niveaux. En effet, ces étiquettes sont apposées soit au niveau des instances, soit au niveau des liens d'agrégation, ou encore au niveau des versions du schéma. Nous détaillons ces différentes approches dans ce qui suit.

Le premier courant propose ainsi de gérer la temporalité des instances de dimensions [BSSJ98]. Inspiré des travaux sur les bases de données temporelles [SA86], un schéma en étoile temporel est proposé pour représenter le fait que les informations dans un entrepôt de données sont valides sur une durée donnée. Il s'agit donc de représenter les données en «temps consistant». Le principe est d'omettre la dimension temps qui permet habituellement l'historisation des données et d'ajouter une étiquette temporelle au niveau de chacune des instances des tables de dimension et des faits de l'entrepôt.

Le deuxième courant propose, quant à lui, de gérer la temporalité des liens d'agrégation [MV00]. Il s'agit de pouvoir gérer des dimensions temporelles pour lesquelles les hiérarchies ne sont pas fixes au niveau des instances. Ainsi le chemin d'agrégation défini pour une instance le long d'une hiérarchie peut évoluer au cours du temps. Pour interroger ce modèle, les auteurs proposent un langage de requêtes nommé TOLAP.

Le dernier courant et non le moindre, est la gestion de la temporalité au niveau de versions du modèle. En effet, la gestion des versions constitue une voie de recherche très explorée et prometteuse. Cela consiste à gérer différentes versions du modèle de l'entrepôt, chaque version étant valide pendant une durée donnée. De nombreux travaux s'inscrivent dans cette alternative. Nous en présentons ici un échantillon représentatif.

Le modèle proposé dans [EK00] présente des fonctions de mise en correspondance qui permettent la conversion entre des versions de structures. Ces fonctions sont basées sur la connaissance des évolutions opérées. Dans [BMBT02, BMBT03], les auteurs proposent une approche qui permet à l'utilisateur d'obtenir des analyses en fonction de différentes situations. En effet, le modèle proposé permet de choisir dans quelle version analyser les données (en temps consistant, dans une version antérieure, ou dans une nouvelle version). Dans [RTZ06], les auteurs proposent un modèle multidimensionnel en temps consistant se caractérisant par le fait qu'il permet des évolutions sur un modèle en constellation. Le versionnement permet également de répondre à des «*what-if analysis*», en créant des versions alternatives, en plus des versions temporelles, pour simuler des changements de la réalité [BEK+04]. Différents travaux se sont ensuite focalisés sur la possibilité de réaliser des analyses en prenant en compte différentes versions [MW04, GLRV06].

Ainsi, la modélisation temporelle constitue aujourd'hui une alternative en pleine expansion qui suscite de nouveaux problèmes qu'il faut résoudre. Dans ces différents courants, les évolutions du modèle sont donc bien conservées et assurent la cohérence des analyses. Ce type de solutions implique une ré-implémentation des outils de

chargement de données, d'analyse, avec la nécessité d'étendre les langages de requêtes afin de gérer les particularités de ces modèles. Il est donc nécessaire, dans ce cas, de prévoir au moment de la conception comment vont être gérées les évolutions à venir.

3.2.4 Discussion

Dans cette section, nous présentons un ensemble de critères que nous avons proposés dans [FBB07b] et que nous avons jugé pertinents pour évaluer les travaux sur l'évolution de modèle dans les entrepôts de données. Nous les comparons ensuite selon les critères sélectionnés.

3.2.4.1 Critères de comparaison

Nous avons déterminé trois groupes de critères que nous avons jugés pertinents compte tenu des objectifs des entrepôts de données. Ils concernent d'une part les caractéristiques des approches, ensuite la mise en place de ces approches et enfin, leur performance.

Les critères sur les caractéristiques des approches sont :
- historisation des dimensions ;
- cohérence des analyses ;
- approche orientée utilisateurs.

Tout d'abord, il s'agit de savoir si l'historisation des dimensions est assurée. En effet, ce critère permet de déterminer si oui ou non les dimensions sont considérées comme temporellement invariantes. Ensuite, l'idée est de mesurer la cohérence des analyses lors de l'application de l'approche. Enfin, il s'agit de déterminer si l'approche se focalise sur le besoin utilisateur qui doit être au centre du processus décisionnel.

Les critères sur la mise en place des approches sont :
- nécessité d'implémenter la solution lors de la conception ;
- complexité de la mise en œuvre (analyse, chargement).

Il est ainsi intéressant d'étudier comment sont mises en œuvre les approches : d'une part si elles doivent être choisies dès le moment de la conception de l'entrepôt, d'autre part si elles sont complexes à mettre en œuvre (par exemple, en mesurant la nécessité d'adapter des outils).

Enfin, les critères sur les performances liées aux approches sont :
- stockage ;
- temps de réponse aux analyses.

Compte tenu de l'objectif lié aux entrepôts qui est l'analyse «en ligne», donc souhaitée rapide, les performances constituent un aspect crucial, non seulement au niveau des analyses, mais également au niveau de la capacité de stockage, étant donné que par définition, la volumétrie des entrepôts de données est d'emblée importante.

3.2.4.2 Comparaison des travaux

La comparaison des travaux selon les deux principales familles (mise à jour de modèle et modélisation temporelle) est récapitulée dans le Tableau 3.1, où un + (resp. -) signifie que l'approche a une influence positive (resp. négative) sur le critère précisé en en-tête de ligne.

		Mise à jour de modèles			Modélisation temporelle		
		Opérateurs d'évolution	Enrichissement hiérarchies	Maintenance de vues	Instances	Liens d'agrégation	Versions
Caractéristiques	historisation des dimensions	-	-	-	+	+	+
	cohérence des analyses	-	+	-	+	+	+
	approche orientée utilisateurs	-	+	-	-	-	-/+
Mise en place	mise en œuvre dès la conception	+	+	+	-	-	-
	complexité	+	+	+	-	-	-
Performances	stockage	+	+	+	-	-	-
	temps de réponse aux analyses	+	+	+	-	-	-

TAB. 3.1 – Comparatif des travaux sur l'évolution de modèle.

Les approches s'inscrivant dans une modélisation temporelle permettent d'assurer l'historisation des dimensions. Concernant les approches de mise à jour de modèle,

cette historisation n'est pas assurée. Néanmoins, des mises à jour telles que l'ajout de niveaux de granularité ne remettent pas en cause la cohérence des analyses, même si l'historisation des modifications subies par le modèle ne sont pas conservées. De ce fait, les travaux sur l'enrichissement de hiérarchies de dimension ne posent pas de problème de cohérence des analyses, tout comme les approches suivant une modélisation temporelle.

Concernant la place des utilisateurs dans le processus de gestion des évolutions, cette dernière est variable selon les approches. Pour répondre à l'évolution des besoins d'analyse, permettant une implication des utilisateurs, il s'avère qu'on peut imaginer qu'elle peut être indirecte. Il s'agit de récolter au fur et à mesure ces besoins et de mettre en œuvre les solutions pour faire évoluer le modèle de l'entrepôt en fonction de ces besoins. Les approches temporelles qui permettent un choix de la version dans laquelle les utilisateurs veulent réaliser leur analyse est positive également de ce point de vue. Cet aspect sur la place des utilisateurs est important, d'autant plus que la personnalisation dans les entrepôts de données devient un enjeu crucial, ce que nous montrons dans ce qui suit.

Concernant la mise en place des approches, les modélisations temporelles nécessitent d'être prévues dès la conception de l'entrepôt et nécessitent la conception d'outils spécifiques pour l'alimentation et l'analyse de l'entrepôt de données. Ces approches peuvent donc être complexes à mettre en œuvre. En effet, la lourdeur de la mise en œuvre d'un entrepôt de données «classique» est reconnue, on imagine donc aisément la difficulté accrue lorsqu'il s'agit d'une modélisation temporelle.

Enfin, concernant la performance, il faut savoir que la modélisation temporelle nécessite de plus grands espaces de stockage, au niveau du stockage d'étiquettes temporelles, de versions, de méta-données, etc. Par ailleurs, les temps de réponse dans les approches temporelles sont également plus longs pour prendre en compte les spécificités du modèle. La réécriture des requêtes est souvent nécessaire pour prendre en compte par exemple les différentes versions.

Pour conclure cette discussion, nous souhaitons mettre en avant que même si la modélisation temporelle fait l'objet de nombreux travaux de recherche (sur le versionnement en particulier), son utilisation n'est pas encore généralisée dans la pratique. Par exemple, SAP-BW permet à l'utilisateur de choisir quelle version des hiérarchies il souhaite utiliser pour l'analyse [ASA00]. Cependant, le versionnement de schéma n'a pas été complètement exploré et aucun outil commercial dédié n'est disponible, à notre connaissance, pour la conception et l'administration.

Par ailleurs, étant donné que ces approches nécessitent d'être prises en compte

dès la conception de l'entrepôt, celles-ci ne pourront être mises en œuvre facilement pour les entreprises qui utilisent d'ores et déjà une architecture décisionnelle basée sur un entrepôt de données «classique». Les entreprises exploitent des entrepôts dont les données sont mises à jour. C'est le cas de l'entreprise avec laquelle nous collaborons. L'ensemble de la structure commerciale de LCL a changé. Il n'y aura plus de trace de l'ancienne structure, les analyses se feront comme si la structure actuelle avait toujours été. Il s'agit finalement d'un arbitrage entre complexité (pour assurer l'exactitude des analyses) et simplicité (en ayant des analyses pouvant être erronées). Bien entendu, le coût (de conception, de maintenance, etc.) est proportionnel à la complexité de l'approche.

3.3 Personnalisation

Dans cette section, nous traitons des travaux consacrés à la personnalisation dans des domaines aussi variés que l'interaction homme-machine (IHM), les bases de données (BD), la recherche d'informations (RI), mais également des travaux plus récents dans le contexte des entrepôts de données.

3.3.1 Personnalisation en IHM, BD et RI

Dès lors que l'on souhaite répondre à des besoins utilisateurs peut se poser la question de la personnalisation vis-à-vis de ces derniers. Ainsi, l'idée même de permettre une personnalisation de l'information n'est pas nouvelle et a été abordée par différentes communautés scientifiques telles que celle de l'interaction homme-machine, des bases de données et de la recherche d'informations.

Ce besoin de personnalisation est en partie dû à la profusion des données parmi lesquelles chaque utilisateur cherche des réponses particulières (que ce soit dans une base de données ou grâce à un moteur de recherche sur Internet). Cette profusion s'explique par différentes raisons parmi lesquelles : l'augmentation des capacités de stockage, la baisse de leur coût, les progrès faits en matière de partage et de distribution des données et l'avènement d'Internet. L'accès à une information pertinente devient alors un enjeu crucial pour l'utilisateur. Il est alors nécessaire d'éviter une surcharge d'informations.

La personnalisation de l'information peut être définie comme étant un ensemble de préférences individuelles, pouvant être représentées de différentes manières, qui vont être utilisées pour fournir des réponses à l'utilisateur les plus pertinentes possibles. Généralement, la personnalisation est basée sur la notion de profil utilisateur

[BK05]. Le contenu de ce profil varie selon les approches.

Dans le domaine de l'IHM, le profil va contenir des informations qui vont permettre au système d'adapter l'affichage des résultats selon les préférences de l'utilisateur. C'est le cas de l'environnement Yahoo ! qui recueille dans le profil un certain nombre d'informations personnelles et adapte la page d'accueil en fonction des centres d'intérêt de l'internaute. Dans le domaine de la RI, le profil utilisateur peut être représenté de différentes manières dont nous évoquons ici quelques exemples. Dans certains cas, le profil utilisateur peut être confondu avec la requête elle-même de l'utilisateur. Dans ce cas, le profil est alors défini par un vecteur de mots-clés, avec éventuellement un poids associé à chaque mot-clé [PG99]. Un profil utilisateur peut également contenir les statistiques d'actions avec le système (nombre de clicks, temps de lecture, etc.) [BRS00]. Ceci permet par la suite d'inférer sur les préférences en connaissant davantage son comportement. Une autre alternative consiste à stocker dans le profil utilisateur des fonctions d'utilités sur un domaine d'intérêt, qui permettent d'exprimer l'importance relative des sujets de ce domaine, les uns par rapport aux autres [CGFZ03]. Dans le domaine des BD, le profil utilisateur peut contenir par exemple les habitudes d'interrogation de celui-ci, en l'occurrence les prédicats souvent utilisés dans ses requêtes ou des ordres dans ces prédicats [KI04].

La notion de profil utilisateur apparaît comme étant à la base de la personnalisation, mais elle est loin d'être définie de façon standard. Ainsi, dans [BK05], les auteurs tentent de classifier les différents types d'informations pouvant être contenus dans un profil et de définir un modèle de profil générique et flexible pouvant s'adapter à différents scénarios de personnalisation.

Ces profils sont ensuite utilisés dans le processus de traitement du système. Le contenu du profil peut être utilisé de différentes façons. Il peut remplacer la requête, permettre de l'enrichir (ajout de critères de sélection, de nouveaux mots-clés) ou être utilisé pour adapter les résultats, dans leur contenu (filtrage) ou dans leur forme de présentation.

3.3.2 Personnalisation dans les entrepôts de données

Si la personnalisation n'est pas une idée nouvelle dans les domaines précédemment évoqués, elle constitue un axe de recherche émergent dans le domaine des entrepôts de données. L'intérêt de cet axe de recherche peut être motivé à la fois vis-à-vis de la volumétrie des données connue pour être importante dans les entrepôts de données et du rôle central que joue l'utilisateur dans le processus décisionnel. En effet, il est en interaction directe avec le système au niveau de l'analyse des données,

en particulier dans le contexte de la navigation. Différentes pistes ont d'ores et déjà été initiées.

La première proposition s'inspire largement du domaine de la recherche d'informations (en RI ou en BD). En effet, il s'agit d'affiner la requête de l'utilisateur pour mieux répondre à ses besoins [BGMM06, BGM+05]. Dans ce cas, le concept de profil est utilisé. Il s'agit d'exprimer des préférences et de satisfaire des contraintes de visualisation. Ce travail trouve un intérêt particulier dans la mesure où l'aspect visualisation est primordial dans le contexte de l'analyse en ligne.

La seconde voie se focalise davantage sur l'utilisation du système et se rapproche davantage de ce qui se fait en IHM. En effet, dans [RTZ07], la personnalisation s'effectue au niveau de la navigation. Il s'agit de représenter les habitudes d'analyse de l'utilisateur, sous forme de coefficient de préférences, pour faciliter sa navigation.

Un autre type de travail est abordé dans [CGL+07]. Il s'agit de considérer une analyse en ligne comme une session interactive durant laquelle l'utilisateur lance des requêtes. Ainsi, il est intéressant que chaque utilisateur dispose de son propre espace de travail. L'objectif étant d'organiser les requêtes, de faciliter leur réutilisation et voire de partager ces requêtes avec d'autres utilisateurs.

Nous pouvons également citer des travaux qui se sont intéressés à rendre les entrepôts de données «actifs». Il s'agit de les munir de règles d'analyse devant être définies par les décideurs. Si la motivation mise en avant dans ces travaux n'est pas la personnalisation, il nous semble intéressant d'évoquer ces travaux qui permettent tout de même de placer les utilisateurs (autrement dit les analystes) au cœur du système. En effet, ces travaux ont pour objectif de reproduire le travail de l'analyste afin d'automatiser certaines tâches d'analyse récurrentes et éventuellement impacter les données sources de l'entrepôt en fonction des résultats de ces analyses [TSM01, TS02]. Par exemple, il s'agit de diminuer le prix de vente d'un produit dans la base de production, à la suite de l'exécution d'une règle d'analyse sur l'entrepôt. Ainsi, d'une certaine façon, ces travaux permettent de personnaliser l'utilisation de l'entrepôt de données. Ces travaux sont basés sur l'utilisation de règles «événement, condition, action» (ECA). Ainsi, l'exécution d'analyses dans les entrepôts de données est rendue plus flexible.

Enfin, mentionnons qu'afin de pouvoir rendre l'analyse plus flexible, un langage à base de règles a été développé dans [EV01] pour la gestion des exceptions dans le processus d'agrégation. Le langage IRAH (Intensional Redefinition of Aggregation Hierarchies) permet de redéfinir des chemins d'agrégation pour exprimer des exceptions dans les hiérarchies de dimensions prévues par le modèle. Tout comme pour

les entrepôts de données actifs, la motivation évoquée n'est pas la personnalisation. Néanmoins, ce travail permet aux utilisateurs d'exprimer eux-mêmes les exceptions dans le processus d'agrégation. En effet, afin de prendre en compte ces exceptions, les utilisateurs définissent et exécutent un programme IRAH, produisant ainsi une révision des chemins d'agrégation. L'exemple considéré est l'étude des prêts d'une compagnie de crédit en fonction de la dimension emprunteur qui est hiérarchisée. La catégorie de l'emprunteur est définie en fonction de son revenu. Mais les auteurs expliquent qu'il est possible que l'analyste veuille ré-affecter un emprunteur dans une autre catégorie en voulant tenir compte d'autres paramètres que le revenu. Dans ce cas, le processus d'agrégation doit tenir compte de cette «exception». Dans ces travaux les auteurs proposent alors un langage à base de règles qui permet de définir des analyses révisées qui tiennent compte de ce type d'exception. Ces travaux ont été élargis afin de proposer la maintenance des cubes de données dans ce contexte d'analyse «révisée» dans [EVT02]. L'objectif est d'éviter la reconstruction totale du cube, en ne recalculant que les cellules sujettes à une modification induite par la révision du cube. Si ce langage constitue une alternative à la rigidité du schéma multidimensionnel dans le processus d'agrégation pour les utilisateurs, il ne fait qu'en modifier les chemins, sans pour autant permettre la création de nouveaux axes d'analyse.

La motivation avancée par les auteurs des deux derniers travaux (entrepôts de données actifs et gestion d'exceptions dans le processus d'agrégation) n'est pas la personnalisation elle-même. Néanmoins, il nous a paru intéressant de les évoquer dans la mesure où les solutions proposées placent l'analyste au cœur du système en lui offrant la possibilité de transcrire ses propres règles d'analyse (entrepôts de données actifs), ou d'exprimer sa propre manière d'agréger les données au niveau des instances (gestion d'exception dans le processus d'agrégation)

3.4 Conclusion

Dans ce chapitre, nous avons tenté de fournir une vision globale de la problématique de l'évolution du modèle (schéma et données) dans les entrepôts de données et des solutions qui ont pu être proposées depuis quelques années. Ces solutions sont nécessaires pour faire face à la fois à l'évolution des besoins d'analyse et des sources de données, même si les spécificités liées à l'évolution des besoins d'analyse telles que la place de l'utilisateur dans ce processus d'évolution n'ont pas été pris en compte. Nous avons mené une étude comparative de ces travaux selon différents critères. Nous avons montré que la modélisation temporelle permet d'assurer une cohérence des analyses, mais que cette solution a un coût.

Nous tenons à exprimer l'idée que, pour nous, la problématique de l'évolution de modèle est bien différente de celle du rafraîchissement de l'entrepôt. Le rafraîchissement correspond à un processus représenté par la phase ETL qui consiste essentiellement à l'ajout de données provenant des sources, sans remettre en cause les données présentes dans l'entrepôt. Dans le cadre de l'évolution de modèle, on sous-entend des évolutions de schéma et des données qui traduisent une réalité mais pouvant aller à l'encontre du principe de non-volatilité des données.

Nous avons dressé une étude comparative des travaux portant sur l'évolution de modèle dans les entrepôts de données. Suite à cette étude, il apparaît qu'il y a un manque de lien entre l'évolution de l'entrepôt et l'origine de cette évolution. Comme nous l'avons évoqué précédemment, pour concevoir un modèle d'entrepôt de données, il est nécessaire de prendre en compte non seulement les sources de données, mais également les besoins d'analyse. Ainsi, lorsque les sources de données ou les besoins d'analyse évoluent, une évolution du modèle de l'entrepôt est peut-être nécessaire. Les différents travaux que nous avons présentés apportent des solutions à certains aspects de cette problématique. Par exemple, la maintenance de vues matérialisées permet d'assurer une certaine propagation de l'évolution des sources de données. Malheureusement, aucune solution ne prend réellement en considération l'émergence de besoins d'analyse.

Ainsi, l'enjeu réside dans le fait d'accorder à l'utilisateur une réelle place dans le processus décisionnel, au-delà de l'exploration des analyses possibles de l'entrepôt. Il s'agit ainsi de pouvoir répondre à des besoins de façon personnalisée, faisant face ainsi à l'émergence de besoins d'analyse qui n'est que trop peu considérée bien qu'elle soit réelle.

En effet, nous avons également montré l'importance de la personnalisation, qui a émergé dans le domaine de la recherche d'information (que ce soit dans les bases de données ou plus généralement sur Internet) et qui émerge de nos jours dans le contexte des entrepôts de données. Cette personnalisation peut s'apparenter à apporter au système davantage de flexibilité. Cette flexibilité peut être assurée en ayant recours à l'expression de règles, comme c'est le cas dans les entrepôts de données actifs. C'est ainsi que nous avons basé notre approche de personnalisation des analyses sur un modèle à base de règles. C'est ce que nous présentons dans le chapitre suivant.