Evolution de modèle dans les entrepôts de données : existant et perspectives

Cécile Favre, Fadila Bentayeb, Omar Boussaid

Université de Lyon
Laboratoire ERIC - Lyon 2
5 av. Pierre Mendès-France, 69676 Bron Cedex
{cfavre,bentayeb}@eric.univ-lyon2.fr
omar.boussaid@univ-lyon2.fr

Résumé. Un entrepôt de données permet d'intégrer des sources de données hétérogènes à des fins d'analyse. Un des points clés de la réussite du processus d'entreposage de données réside dans la définition du modèle de l'entrepôt en fonction des sources de données et des besoins d'analyse. Une fois l'entrepôt conçu, le contenu et la structure des sources de données, tout comme les besoins d'analyse sont amenés à évoluer et nécessitent ainsi une évolution du modèle de l'entrepôt (schéma et données). Dans cet article, nous présentons un panorama de différents travaux portant sur l'évolution du modèle dans les entrepôts de données. Nous comparons et discutons ces travaux selon les critères qui nous semblent pertinents pour cette problématique. Nous dressons également les perspectives de recherche qui en découlent.

1 Introduction

Les données proviennent aujourd'hui de sources hétérogènes et présentent une volumétrie de plus en plus conséquente. La structuration et le stockage de ces données dans un entrepôt constituent alors un support efficace pour permettre des analyses en vue de prises de décision. Cette structuration est représentée grâce à un schéma caractérisant une organisation dite multidimensionnelle des données. Une organisation multidimensionnelle répond à l'objectif d'analyser des activités au travers de *faits* composés d'indicateurs, appelés *mesures*, ainsi que de *dimensions* qui constituent les différents axes d'observation des mesures. Ces dimensions peuvent présenter des *hiérarchies* qui offrent la possibilité de réaliser des analyses à différents *niveaux de granularité*.

La réussite du processus d'entreposage repose entre autres sur une bonne conception du schéma, puisque c'est ce dernier qui va déterminer les possibilités d'analyse de l'entrepôt. Ainsi, de nombreux travaux de recherche ont été menés sur la conception de schéma des entrepôts de données. Ces travaux témoignent aujourd'hui de la nécessité de prendre en compte à la fois les sources de données et les besoins d'analyse (Nabli et al., 2005), plutôt que d'avoir recours, par exemple, à une approche uniquement guidée par les données telle que celle proposée par Golfarelli et al. (1998).

Une fois l'entrepôt de données construit, les sources de données et les besoins d'analyse peuvent subir des changements. Ainsi, lorsque les sources de données ou les besoins évoluent, le schéma de l'entrepôt peut être amené à évoluer, tout comme les données qu'il contient, on parle alors d'évolution du modèle de l'entrepôt.

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 (Hanson, 1987), sur les index (Valduriez, 1987), etc. ont été adaptés pour être appliqués aux spécificités des entrepôts. En ce qui concerne le domaine d'intérêt de cet article, la mise à jour des bases de données (Roddick, 1992), les bases de données temporelles (Snodgrass et Ahn, 1986) et les bases de données multiversions (Tansel et Garnett, 1989) 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 modèle dans les entrepôts de données, le versionnement de ces derniers pour prendre en compte la temporalité dans les dimensions, etc. Nous proposons ici de classer les différents travaux selon deux familles : celle que nous dénommons «modélisation temporelle» et celle que nous appelons «mise à jour de modèle». Ces deux familles se distinguent respectivement par la conservation ou non de la trace des évolutions subies par le modèle. Chacune de ces familles présentent différentes approches que nous nous proposons d'étudier et de comparer.

L'objectif de cet article est alors multiple. Tout d'abord, compte-tenu de l'objectif d'analyses qui anime tout processus d'entreposage de données, nous voulons poser le problème de l'évolution de modèle en terme de cohérence de ces analyses. Ensuite, nous voulons présenter un panorama des différents travaux traitant de l'évolution de modèle, en dressant les caractéristiques de ceux-ci. Nous souhaitons également discuter ces travaux et proposer une étude comparative selon différents critères que nous avons jugé pertinents, tels que la conservation de l'historisation des données, la cohérence des analyses, la complexité de la mise en œuvre, etc. Cette discussion permettra d'évoquer par la suite les perspectives que nous envisageons.

La suite de cet article est organisée de la façon suivante. Tout d'abord, dans la Section 2, nous évoquons, sur un exemple issu d'un cas d'étude réel, 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 dans la Section 3, en détaillant chacune des deux familles de travaux pour gérer l'évolution des entrepôts : la mise à jour et la modélisation temporelle. Par la suite, nous nous proposons de discuter ces travaux dans la Section 4 en présentant une sélection de critères et en comparant les travaux selon ces derniers. Nous dressons alors dans la Section 5 les enjeux du domaine en termes de perspectives de recherche, avant de conclure dans la Section 6.

2 Evolution 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 classifions ces évolutions selon deux types : les évolutions sur le schéma d'une part, et d'autre part les évolutions sur les données. Pour illustrer ces évolutions

et les impacts qu'elles ont, nous nous proposons de baser notre discours sur un modèle implémenté en relationnel, issu du cas réel de l'établissement bancaire LCL-Le Crédit Lyonnais[1].

Le schéma multidimensionnel de la Figure 1 permet d'analyser la mesure PNB (Produit Net Bancaire). Le PNB correspond à ce que rapporte un client à l'établissement bancaire. Cette mesure est analysée selon les dimensions CLIENT, AGENCE et ANNEE. La dimension AGENCE présente une hiérarchie. Il est ainsi possible d'agréger les données selon le niveau de granularité UNITE_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 de base. A chaque fois que nous explicitons une évolution, elle est réalisée sur ce schéma de base. Pour illustrer notre propos, nous nous plaçons dans un contexte relationnel et 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.

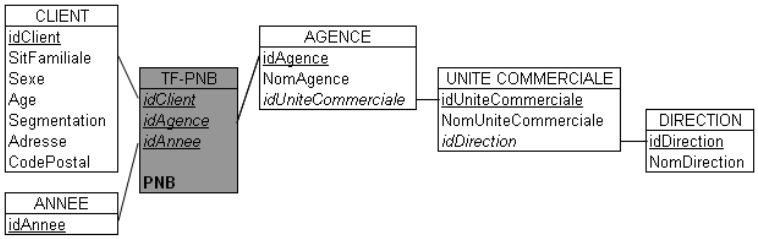

FIG. 1 – *Schéma multidimensionnel pour observer le PNB.*

2.1 Evolution du schéma multidimensionnel

Un schéma multidimensionnel peut subir des évolutions qui peuvent remettre plus ou moins en cause le modèle existant en ayant des impacts plus ou moins importants sur les données. Par exemple, les évolutions de schéma impactant la table des faits ont en général des conséquences importantes sur les données. En effet, la volumétrie de l'entrepôt dépend généralement de la volumétrie 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 les données.

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 données y seront plus détaillées), 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 PRODUIT_BANCAIRE, identifiant ainsi un PNB 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

[1]Collaboration avec la Direction d'Exploitation Rhône-Alpes Auvergne de LCL–Le Crédit Lyonnais dans le cadre d'une Convention Industrielle de Formation par la Recherche (CIFRE)

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 baisser le niveau de détail de la table des faits (les données y seront moins détaillées), puisque les mesures présentes dans la table des faits seront décrites par une dimension de moins. Par exemple, il est possible de supprimer la dimension agence. Ainsi le PNB serait identifié pour une année et un client donnés. Là encore il faut recalculer les agrégats de la table de faits TF_PNB puisque l'identifiant idAgence 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_PNB une mesure telle que FRAIS_GESTION 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 est nécessaire, puisqu'il s'agit seulement de «supprimer une colonne». Notons, que cette modication structurelle n'est possible que s'il existait plusieurs mesures pour analyser les faits. Dans notre exemple, la suppression de la mesure PNB n'est pas envisageable, étant donné que c'est la seule mesure de la table des faits TF_PNB. En la supprimant, le schéma multidimensionnel n'a plus lieu d'exister.

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

Dans le cas de la suppression de niveau de granularité, l'impact est plus ou moins important selon 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 UNITE_COMMERCIALE est supprimé, il faut assurer le lien entre les niveaux AGENCE 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 UNITE_COMMERCIALE et DIRECTION.

Ces différentes modifications d'ordre structurel enrichissent (ajout de dimension, de mesure, de niveau de granularité) 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 posent lors de l'évolution des données, comme nous allons le montrer dans ce qui suit.

2.2 Evolution 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 granularité, 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 un contexte spécifique qu'est l'entrepôt de données, ces trois opérations de base ont des conséquences différentes selon l'endroit où elles sont appliquées.

Envisageons tout d'abord le contexte 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 (Inmon, 2002), 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, Rizzi et Golfarelli (2006) évoquent la nécessité de mettre à jour les valeurs des mesures, lorsqu'elles ont fait l'objet d'erreurs ou lorsque les évènements qu'elles traduisent évoluent.

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.

Kimball (1996) évoquait 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. 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 (1996) 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 (2002) l'évoque dans sa définition d'un entrepôt de données. Mais dans quelle mesure cette historisation de données garantit une cohérence des analyses ?

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 1, la modification de la valeur de l'attribut idUniteCommerciale de la table AGENCE entraîne des changements considérables du point de vue de l'analyse (Body et al., 2002). 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 PNB 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 appartenue 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 SitFamiliale 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 Adresse, 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, dans la suite de cet article, à évoquer les différents travaux qui permettent de gérer ces évolutions.

3 Evolution de modèle dans les entrepôts : l'existant

3.1 Mise à jour de modèle dans les entrepôts de données

La famille reconnue sous le terme de mise à jour de modèle est caractérisée par le fait que ses approches 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. Hurtado et al. (1999) proposent un modèle formel pour la mise à jour des dimensions et de leur hiérarchie, en proposant 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 (les cubes) et proposent également un algorithme pour réaliser leur maintenance de façon efficace.

Blaschka et al. (1999) proposent, non seulement des évolutions au niveau des dimensions, mais également l'évolution des faits. L'évolution qu'ils proposent est réalisée à un niveau conceptuel, indépendant de l'implémentation. Ils proposent ainsi une algèbre comprenant quatorze opérations 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 Hurtado et al. (1999). Blaschka (2000) étend ces travaux en proposant également la propagation de ces changements du niveau conceptuel vers le niveau logique.

Benitez-Guerrero (2002) s'inspire des travaux précédents 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 (relationel, 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, et non pas comment représenter cette opération. Ainsi, dans les deux approches présentées, 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.

Mazón et Trujillo (2006) proposent 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[2]. Les niveaux de granularité sont créés en fin de hiérarchie.

De notre côté, nous avons proposé d'enrichir les hiérarchies de dimension en exploitant les connaissances des utilisateurs eux-mêmes, pour répondre au besoin de personnalisation des analyses (Favre et al., 2007). Ces connaissances sont représentées d'une part par une méta-règle d'agrégation qui représente la structure du lien d'agrégation entre deux niveaux de granularité, puis par des règles «si-alors» qui représentent ce lien au niveau des instances elles-mêmes. Les niveaux créés peuvent être insérés dans une hiérarchie ou créés à la fin de celle-ci.

Le troisième courant se base sur l'hypothèse qu'un entrepôt de données est un ensemble de vues matérialisées construites à partir des sources de données (Bellahsene, 2002). Alors que Hurtado et al. (1999) se sont intéressés à la maintenance de vues pour propager l'évolution du modèle sur les cubes de données, représentés par des vues, Bellahsene (2002) s'intéresse 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

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

sur la maintenance de vues matérialisées dans ce contexte, vers l'état de l'art proposé par Badri et al. (2005) sur la maintenance de vues matérialisées issues de sources de données hétérogènes.

Nous avons présenté trois courants s'incrivant dans une mise à jour du modèle de l'entrepôt. Les opérateurs permettent de faire évoluer le modèle ; l'évolution des données est succintement évoquée dans ces travaux. Le deuxième courant s'est intéressé précisément à la provenance des données pour 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 en répercussion directe de 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 : l'un permettant de proposer une évolution du modèle de l'entrepôt pour répondre à un besoin d'évolution traité par l'administrateur ; l'autre de trouver les données nécessaires pour réaliser une évolution particulière pour répondre à des besoins d'évolution liés à l'analyse, le troisième de répondre plus particulièrement à un besoin d'évolution en réponse à l'évolution des sources de données.

3.2 Modélisation temporelle des entrepôts de données

La famille reconnue sous le terme de modélisation temporelle s'oppose à celle des travaux sur la mise à jour de modèle vis-à-vis de 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 (Bliujute et al., 1998). Inspiré des travaux sur les bases de données temporelles (Snodgrass et Ahn, 1986), un schéma en étoile temporel est proposé pour représenter le fait que les informations dans un entrepôt de données sont valables 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 (Mendelzon et Vaisman, 2000). Les auteurs proposent un modèle multidimensionnel. 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 troisième 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 actuellement. 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. Pour des raisons de place, nous en présentons ici seulement un extrait.

Le modèle proposé par Eder et Koncilia (2000) propose 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. Body et al. (2002, 2003) proposent une approche qui permet à l'utilisateur d'obtenir des analyses en fonction de ses besoins de comparaisons des données. 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). Ravat et al. (2006) 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 (Bebel et al., 2004). Différents travaux se sont ensuite focalisés sur la possibilité de réaliser des analyses en prenant en compte différentes versions (Morzy et Wrembel, 2004; Golfarelli et al., 2006).

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 réimplémentation 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.

4 Discussion

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

4.1 Critères de comparaison

Nous avons déterminé trois groupes de critères : ils concernent d'une part les caractéritiques des approches, ensuite la mise en place de ces approches, et enfin la 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 oeuvre (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.

4.2 Comparaison des travaux

La comparaison des travaux est récapitulée dans le Tableau 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. 1 – *Tableau comparatif des travaux sur l'évolution de modèle.*

Concernant l'historisation des dimensions, il s'avère que les approches s'inscrivant dans une modélisation temporelle permettent d'assurer cette historisation. 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 hiérarchie de dimension ne posent pas de problème de cohérence des analyses, tout comme les approches suivant une modélisation temporelle.

Concernant la place des utilisateurs, 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. Il est donc intéressant d'avoir une approche comme celle que nous avons proposée (Favre et al., 2007) afin de répondre à des besoins d'analyse personnalisés. 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. D'autant plus que la personnalisation dans les entrepôts de données devient un enjeu crucial.

En effet, la personnalisation dans les entrepôts de données constitue une nouvelle voie de recherche. Les travaux dans ce domaine se focalisent en particulier sur la visualisation de données basée sur la modélisation des préférences utilisateurs et sur l'exploitation du concept de profil. Par exemple, dans Bellatreche et al. (2005), les auteurs proposent d'affiner les requêtes pour montrer une partie des données qui répond aux préférences utilisateurs.

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 si la modélisation temporelle fait l'objet de nombreux travaux de recherche, entre autres à travers les travaux sur le versionnement, son utilisation n'est pas encore généralisée dans la pratique. A notre connaissance, aucun outil commercial ne permet cette gestion des entrepôts. Par ailleurs, étant donné que ces approches nécessitent d'être prises en compte dès la conception originale 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-Le Crédit Lyonnais a changé. Il n'y aura plus de trace de l'ancienne structure, et 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.

5 Evolution de modèle dans les entrepôts : perspectives

Nous avons dressé une étude comparative des travaux portant sur l'évolution de modèle dans les entrepôts de données. Suite à cette étude, différentes perspectives de recherche nous semblent être pertinentes.

La première concerne le manque de lien entre l'évolution de l'entrepôt et l'origine de cette évolution. Comme nous l'avons évoqué en introduction, 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, l'enrichissement de hiérarchies de dimension prend en compte l'émergence de besoins d'analyse ; la maintenance de vues matérialisées permet d'assurer une certaine propagation de l'évolution des sources de données.

L'enjeu est alors de disposer d'un cadre général qui permette non seulement de gérer l'évolution de l'entrepôt, mais qui prenne également en compte l'origine de cette évolution. Il s'agit donc, dans un premier temps, de concevoir un cadre formel qui permette de représenter les liens entre sources de données, besoins d'analyse et entrepôt de données. L'objectif est, en quelque sorte, de définir une fonction prenant en paramètres le schéma des sources de données, celui des besoins, le résultat de cette fonction étant un schéma multidimensionnel.

Notons S un ensemble de sources de données, B un ensemble de besoins d'analyse et M une structure multidimensionnelle (telle qu'un entrepôt de données en l'occurence). On a alors $M = \mathcal{F}(S, B)$, où $\mathcal{F}$ représente notre cadre formel.

Grâce à ce cadre formel, l'objectif est de pouvoir exprimer une évolution des sources de données et/ou des besoins d'analyse et d'être ainsi en mesure de déterminer l'évolution de l'entrepôt. Notons Δ pour symboliser une évolution. Notre objectif est de pouvoir déterminer ΔM lorsque ΔS et/ou ΔB se produisent. Pour cela, les travaux de Bernstein et Rahm (2000) qui envisagent des scénarios tels que l'ajout d'une source de données pour la gestion de modèle nous paraissent intéressants.

Ce cadre formel doit être indépendant de tout choix d'implémentation (relationnel, multidimensionnel, etc.) et être placé dans un premier temps au niveau conceptuel. Ainsi, l'évolution exprimée au niveau conceptuel doit ensuite être propagée au niveau logique, puis physique. Néanmoins, il ne s'agit pas forcément d'une propagation automatique. Nous considérons cette démarche comme une aide à l'administration de l'entrepôt. En effet, il s'agit plutôt d'une approche semi-automatique qui guide l'administrateur. L'évolution subie par une source de données peut-être techniquement prise en compte dans l'évolution de l'entrepôt, mais cela constitue une possibilité, pas un automatisme. Ceci peut dépendre de critères de performance, de pertinence, etc. En outre, les évolutions peuvent se situer non seulement au niveau de la structure, mais également au niveau des données elles-mêmes. Les deux types d'évolution doivent être pris en compte dans le cadre formel.

La deuxième perspective va au-delà de l'évolution même du modèle de l'entrepôt. En effet, l'évolution du modèle évoquée jusque là ne doit pas nous faire oublier les autres composants de l'entrepôt et de l'architecture décisionnelle a fortiori. Ainsi, une fois cette évolution réalisée, il s'agit de propager cette-ci à l'ensemble des outils liés à l'entrepôt. Il s'agit par exemple de réaliser la maintenance des structures multidimensionnelles liées à l'entrepôt, telles que les cubes de données, les magasins de données (dans la configuration où un magasin de données est une extraction de l'entrepôt lui-même), d'assurer la maintenance des structures d'optimisation, telles que les index, etc. L'objectif est ainsi d'assurer la parfaite cohérence de l'entrepôt de données. Pour ce faire, nous préconisons d'étendre le paradigme de Bouzeghoub et al. (1999)

qui consiste à considérer le processus de rafraîchissement d'un entrepôt de données comme un «*workflow*» (flux de travail).

Notre troisième et dernière perspective se focalise sur l'optimisation. En effet, l'optimisation de performances dans les entrepôts de données constitue un enjeu réel du domaine, a fortiori quand on va vers des modèles temporels plus complexes, donc plus coûteux pour l'analyse et le stockage. Comme nous l'avons évoqué dans la perspective précédente, la maintenance de la cohérence dans l'entrepôt et donc a fortiori dans les structures d'optimisation est cruciale. Mais dans un contexte d'évolution du modèle, nous pensons qu'une maintenance de cohérence des structures d'optimisation existantes n'est pas forcément suffisante. En effet, l'évolution du modèle de l'entrepôt peut nécessiter une évolution des choix d'optimisation. Il s'agit alors de réaliser un réel redéploiement de la stratégie d'optimisation. Pour cela, la réalisation de tests de performances grâce à une charge de requêtes significatives semble appropriée. Afin qu'une étude de performances soit la plus exploitable possible, il est pertinent de la réaliser sur l'entrepôt en question avec une charge représentative de l'utilisation de ce dernier. Ainsi, lorsqu'une charge de requêtes pertinentes a été définie précédemment, il nous semble important de pouvoir maintenir sa cohérence en fonction de l'évolution qu'a subi l'entrepôt, sans pour autant avoir à reconstruire une nouvelle charge de requêtes. Un outil qui permettrait de maintenir la cohérence d'une charge de requêtes en prenant en compte l'évolution de schéma (ajout d'attribut dans la clause SELECT d'une requête) et celle des données (mise à jour de valeur d'un attribut dans la clause WHERE d'une requête) paraît alors très utile.

6 Conclusion

Dans cet article, 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. Nous avons mené une étude comparative de ces travaux selon différents critères. Nous avons ainsi montré que la modélisation temporelle permet d'assurer une cohérence des analyses, mais que cette solution a un coût réel.

Par la suite, il nous a semblé pertinent d'évoquer les perspectives concernant cette problématique d'évolution de modèle en justifiant le besoin d'un cadre général pour la gérer, en n'omettant pas le lien avec l'origine de ces évolutions, en l'occurence l'évolution des sources de données et des besoins d'analyse. Par ailleurs, il nous paraît important de répercuter ces évolutions au niveau des composants de l'entrepôt (index, etc.) et de ceux de l'architecture décisionnelle (magasins de données, etc.). Enfin, la maintenance de la cohérence de charges de requêtes pour tester la performance d'un entrepôt est intéressante pour redéployer une stratégie d'optimisation qui peut être pertinente suite à l'évolution du modèle de l'entrepôt.

Références

Badri, M., F. Boufares, C. F. Ducateau, et F. Gargouri (2005). Etat de l'art de la maintenance des entrepôts de données issus de systèmes d'information hétérogènes. In *Vèmes Journées Scientifiques Génie Electrique et Informatique (GEI 05), Sousse, Tunisie*, pp. 13–18.

Bebel, B., J. Eder, C. Koncilia, T. Morzy, et R. Wrembel (2004). Creation and Management of Versions in Multiversion Data Warehouse. In *XIXth ACM Symposium on Applied Computing (SAC 04), Nicosia, Cyprus*, pp. 717–723. ACM Press.

Bellahsene, Z. (2002). Schema Evolution in Data Warehouses. *Knowledge and Information Systems 4*(3), 283–304.

Bellatreche, L., A. Giacometti, P. Marcel, H. Mouloudi, et D. Laurent (2005). A Personalization Framework for OLAP Queries. In *VIIIth ACM International Workshop on Data Warehousing and OLAP (DOLAP 05), Bremen, Germany*, pp. 9–18. ACM Press.

Benitez-Guerrero, E. I. (2002). *Infrastructure adaptable pour l'évolution des entrepôts de données*. Ph. D. thesis, Université Joseph Fourier - Grenoble 1.

Bernstein, P. A. et E. Rahm (2000). Data Warehouse Scenarios for Model Management. In *XIXth International Conference on Conceptual Modeling (ER 00), Salt Lake City, Utah, USA*, Volume 1920 of *LNCS*, pp. 1–15. Springer.

Blaschka, M. (2000). *FIESTA : A Framework for Schema Evolution in Multidimensional Databases*. Ph. D. thesis, Institut für Informatik des Technischen Universität München.

Blaschka, M., C. Sapia, et G. Höfling (1999). On Schema Evolution in Multidimensional Databases. In *Ist International Conference on Data Warehousing and Knowledge Discovery (DaWaK 99), Florence, Italy*, Volume 1676 of *LNCS*, pp. 153–164. Springer.

Bliujute, R., S. Saltenis, G. Slivinskas, et C. Jensen (1998). Systematic Change Management in Dimensional Data Warehousing. In *IIIrd International Baltic Workshop on Databases and Information Systems, Riga, Latvia*, pp. 27–41.

Body, M., M. Miquel, Y. Bédard, et A. Tchounikine (2002). A Multidimensional and Multiversion Structure for OLAP Applications. In *Vth ACM International Workshop on Data Warehousing and OLAP (DOLAP 02), McLean, Virginia, USA*, pp. 1–6. ACM Press.

Body, M., M. Miquel, Y. Bédard, et A. Tchounikine (2003). Handling Evolutions in Multidimensional Structures. In *XIXth International Conference on Data Engineering (ICDE 03), Bangalore, India*, pp. 581–591. IEEE Computer Society.

Bouzeghoub, M., F. Fabret, et M. Matulovic-Broqué (1999). Modeling the Data Warehouse Refreshment Process as a Workflow Application. In *International Workshop on Design and Management of Data Warehouses (DMDW 99), Heidelberg, Germany*, Volume 19 of *CEUR Workshop Proceedings*, pp. 6. CEUR-WS.org.

Eder, J. et C. Koncilia (2000). Evolution of Dimension Data in Temporal Data Warehouses. Technical report, University of Klagenfurt.

Favre, C., F. Bentayeb, et O. Boussaïd (2007). Dimension Hierarchies Updates in Data Warehouses : a User-driven Approach. In *IXth International Conference on Enterprise Information Systems (ICEIS 07), Funchal, Madeira, Portugal*.

Golfarelli, M., J. Lechtenborger, S. Rizzi, et G. Vossen (2006). Schema Versioning in Data Warehouses : Enabling Cross-Version Querying via Schema Augmentation. *Data and Knowledge Engineering 59*(2), 435–459.

Golfarelli, M., D. Maio, et S. Rizzi (1998). Conceptual Design of Data Warehouses from E/R Schemes. In *XXXIst Annual Hawaii International Conference on System Sciences (HICSS 98), Big Island, Hawaii, USA*, Volume 7, pp. 334–343.

Hanson, E. N. (1987). A Performance Analysis of View Materialization Strategies. In *ACM SIGMOD International Conference on Management of Data (SIGMOD 87), San Francisco, California, USA*, pp. 440–453. ACM Press.

Hurtado, C. A., A. O. Mendelzon, et A. A. Vaisman (1999). Maintaining Data Cubes under Dimension Updates. In *XVth International Conference on Data Engineering (ICDE 99), Sydney, Australia*, pp. 346–355. IEEE Computer Society.

Inmon, W. H. (2002). *Building the Data Warehouse* (Third ed.). John Wiley & Sons.

Kimball, R. (1996). *The Data Warehouse Toolkit*. John Wiley & Sons.

Mazón, J.-N. et J. Trujillo (2006). Enriching Data Warehouse Dimension Hierarchies by Using Semantic Relations. In *XXIIIrd British National Conference on Databases (BNCOD 2006), Belfast, Northern Ireland*, Volume 4042 of *LNCS*, pp. 278–281. Springer.

Mendelzon, A. O. et A. A. Vaisman (2000). Temporal Queries in OLAP. In *XXVIth International Conference on Very Large Data Bases (VLDB 00), Cairo, Egypt*, pp. 242–253. Morgan Kaufmann.

Morzy, T. et R. Wrembel (2004). On Querying Versions of Multiversion Data Warehouse. In *VIIth ACM International Workshop on Data Warehousing and OLAP (DOLAP 04), Washington, Columbia, USA*, pp. 92–101. ACM Press.

Nabli, A., A. Soussi, J. Feki, H. Ben-Abdallah, et F. Gargouri (2005). Towards an Automatic Data Mart Design. In *VIIth International Conference on Enterprise Information Systems (ICEIS 05), Miami, Florida, USA*, pp. 226–231.

Ravat, F., O. Teste, et G. Zurfluh (2006). A Multiversion-Based Multidimensional Model. In *VIIIth International Conference on Data Warehousing and Knowledge Discovery (DaWaK06), Krakow, Poland*, Volume 4081 of *LNCS*, pp. 65–74. Springer.

Rizzi, S. et M. Golfarelli (2006). What Time Is It in the Data Warehouse ? In *VIIIth International Conference on Data Warehousing and Knowledge Discovery (DaWaK 06), Krakow, Poland*, Volume 4081 of *LNCS*, pp. 134–144. Springer.

Roddick, J. F. (1992). Schema Evolution in Database Systems : an Annotated Bibliography. *SIGMOD Rec. 21*(4), 35–40.

Snodgrass, R. et I. Ahn (1986). Temporal databases. *Computer 19*(9), 35–41.

Tansel, A. U. et L. Garnett (1989). Nested Historical Relations. In *ACM SIGMOD International Conference on Management of Data (SIGMOD 89), Portland, Oregon, USA*, pp. 284–294. ACM Press.

Valduriez, P. (1987). Join Indices. *ACM Transactions on Database Systems (TODS) 12*(2), 218–246.

Summary

A data warehouse allows integrating heterogeneous data sources for analysis purposes. One of the key points of the data warehousing process success is the design of the model according to the available data sources and the analysis needs. However, once the data warehouse is built, several changes on contents and structures can usually happen on these sources. And analysis needs could also evolve. Therefore, these evolutions have to be reflected in the data warehouse model. In this paper, we provide an overall view of the state of the art in data warehouse model evolution. We compare the different works according to different criteria that seem to be important for this problematic and we discuss the advantages and the disadvantages of the existing solutions. Moreover we define the next research directions in the domain.