Modèle d'entrepôt de données à base de règles

Cécile Favre, Fadila Bentayeb, Omar Boussaïd

Laboratoire ERIC, Université Lumière Lyon2
5, avenue Pierre Mendès-France
69676, Bron Cedex, France
{cfavre, bentayeb}@eric.univ-lyon2.fr
boussaid@univ-lyon2.fr
http://eric.univ-lyon2.fr

Résumé. Les entrepôts de données constituent une réponse aux besoins d'analyse des entreprises. Mais, cette solution est lourde à mettre en œuvre et à maintenir, en particulier au niveau de l'évolution des besoins d'analyse. Pour répondre à ce problème, nous proposons un nouveau modèle d'entrepôt de données à base de règles *R-DW*. Les règles permettent d'intégrer les connaissances de l'utilisateur dans l'entrepôt. Ce modèle est composé de deux parties : une partie fixe, définie en extension, comprenant une table des faits et les dimensions de premier niveau ; une partie évolutive, définie en intension par des règles. Grâce à ces règles, notre modèle *R-DW* permet de créer des hiérarchies de dimension de façon dynamique, rendant ainsi possible l'évolution des contextes d'analyse et renforçant l'interaction entre l'utilisateur et le système d'aide à la décision.

1 Introduction

Pour gérer une masse de données de plus en plus conséquente, provenant de sources hétérogènes, la mise en place d'un processus décisionnel est devenue nécessaire. Le stockage et la centralisation de ces données dans un entrepôt constituent un support efficace pour l'analyse. Un entrepôt de données est défini par une collection de données orientées sujet, intégrées, non volatiles et historisées, organisées pour le support d'aide à la décision (Inmon, 1996). À partir d'un entrepôt de données, des contextes d'analyse multidimensionnels ciblés, appelés communément cubes de données, sont construits. Ces cubes répondent à des besoins d'analyse prédéfinis en amont, l'objectif étant d'observer des faits, à travers une ou plusieurs mesures, en fonction de différentes dimensions. Il s'agit par exemple d'observer les niveaux de ventes en fonction des produits, des magasins et de la période d'achat. L'analyse en ligne OLAP (*On Line Analytical Processing*) est un outil basé sur la visualisation permettant aux décideurs d'une entreprise la navigation et l'exploration de ces cubes de données.

Mais si l'entreposage de données a connu un essor important grâce aux possibilités qu'il offre, sa mise en place et sa maintenance ne sont pas des tâches faciles. D'une part, la mise en œuvre d'un entrepôt de données nécessite un important travail préalable à la fois d'étude de l'existant et des besoins d'analyse, mais aussi de modélisation. D'autre part, la maintenance de l'entrepôt nécessite, non seulement un rafraîchissement périodique des données, mais également une évolution du schéma pour répondre à de nouveaux besoins d'analyse.

Les modèles multidimensionnels (Cabibbo et Torlone, 1998; Kimball, 1996) 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. Cependant, en pratique, des changements peuvent se produire dans le schéma des dimensions. Pour les prendre en compte, deux alternatives existent : la première propose la mise à jour du schéma (Blaschka et al., 1999; Hurtado et al., 1999) et la seconde consiste à gérer différents schémas en les historisant (Bliujute et al., 1998; Body et al., 2002; Chamoni et Stock, 1999; Eder et Koncilia, 2001). Ces deux approches constituent une réponse au problème de l'évolution des dimensions, lorsque cette dernière est orientée par l'évolution des données elles-mêmes. En revanche, elles n'apportent pas de solution à l'émergence de nouveaux besoins d'analyse qui sont orientés, non pas par l'évolution des données, mais par l'expression de nouvelles connaissances. En effet, une fois l'entrepôt constitué, l'utilisateur ne peut qu'effectuer des analyses prévues par le modèle.

Le travail de recherche exposé ici est réalisé en collaboration avec LCL - Le Crédit Lyonnais[1]. Pour compléter la politique marketing nationale, les responsables commerciaux sont amenés à faire, au niveau local, des demandes marketing. Une demande marketing est la formulation d'une demande de ciblage pour une action marketing ponctuelle (opération spécifique à un produit ou à un événement). Des données hétérogènes sont amenées à enrichir nos connaissances sur les demandes marketing. La demande marketing constitue alors un objet d'étude complexe. Pour analyser ces données complexes, il est nécessaire de les intégrer. Dans (Favre et al., 2005), nous avons proposé une architecture d'entreposage virtuel combinant la médiation et l'entreposage. Du fait que les applications considérées génèrent des informations de façon fréquente, la médiation nous dispense des tâches de rafraîchissement. Cependant, ces données sont sous un format non approprié à l'analyse. L'approche d'entreposage contribue à remédier à ce problème. L'analyse permet de générer des connaissances qu'il faut réintroduire comme source d'informations dans notre dispositif d'entreposage virtuel. Par ailleurs, établir de façon exhaustive les besoins de l'ensemble des utilisateurs est une tâche complexe. Parfois, les utilisateurs disposent de connaissances qui ne sont pas représentées dans l'entrepôt et qui sont susceptibles d'orienter l'analyse des données. Il est donc intéressant que l'utilisateur puisse exprimer ses connaissances pour définir de nouveaux axes d'analyse. Le modèle de l'entrepôt doit donc permettre l'évolution des contextes d'analyse grâce à des connaissances du domaine que l'utilisateur pourra intégrer dans l'entrepôt de données. Mais ce type d'évolution n'est pas facile à réaliser dans les modèles classiques des entrepôts qui sont peu flexibles.

Les travaux ayant porté sur l'augmentation de la flexibilité dans les entrepôts de données font généralement recours à des langages à base de règles. Certains auteurs se sont attachés à rendre la définition du schéma évolutive (Kim et al., 2003; Peralta et al., 2003). D'autres ont proposé des modèles supportant différents types de contraintes (Carpani et Ruggia, 2001; Hurtado et Mendelzon, 2002; Ghozzi et al., 2003) pour résoudre le problème de la cohérence des données. D'autres encore ont apporté une réponse au traitement des exceptions dans le processus d'agrégation, rendant ce dernier plus souple (Espil et Vaisman, 2001). Les langages à base de règles ont ainsi permis de rendre plus flexible l'entrepôt de données, mais les contextes d'analyse fournis par le modèle n'en demeurent pas moins figés.

[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)

Dans cet article, nous proposons un nouveau modèle d'entrepôt de données à base de règles baptisé *R-DW* (*Rule-based Data Warehouse*). Il s'agit d'une solution conceptuelle prometteuse qui pose différents problèmes tels que la performance et l'optimisation, l'évolution du modèle. Dans cet article, nous nous attachons à présenter notre modèle et son apport. Notre modèle *R-DW* est composé de deux parties : une partie "fixe", définie en extension, et une partie "évolutive", définie en intension grâce à des règles. La partie fixe comprend une table de faits et les dimensions de premier niveau. La partie évolutive comprend un ensemble de règles qui définissent de nouvelles hiérarchies de dimension basées sur les dimensions existantes et des nouvelles connaissances. Les règles permettent alors d'établir les liens sémantiques entre les données, en définissant le passage entre deux niveaux de granularité.

Notre modèle *R-DW* présente plusieurs avantages par rapport aux modèles d'entrepôt existants. Il permet de :

- créer des hiérarchies de dimension de façon dynamique ;
- faire évoluer les contextes d'analyse ;
- renforcer l'interaction entre l'utilisateur et le système d'aide à la décision en permettant à celui-ci d'intégrer ses propres connaissances.

Cet article est organisé de la façon suivante. Nous introduisons tout d'abord un exemple motivant notre approche dans la section 2. Puis, nous présentons dans la section 3 un état de l'art sur l'évolution de schéma et la flexibilité apportée par l'utilisation des langages à base de règles dans les entrepôts de données. Nous présentons ensuite notre modèle *R-DW* dans la section 4 et définissons un cadre formel pour celui-ci dans la section 5. Nous exposons ensuite la mise en œuvre de notre modèle *R-DW* et l'application de celui-ci aux données bancaires dans la section 6, avant de conclure et d'indiquer les perspectives dans la section 7.

2 Exemple introductif

Pour illustrer notre approche de modélisation d'entrepôts de données à base de règles, nous utilisons, tout au long de cet article, le cas réel de LCL-Le Crédit Lyonnais. Le PNB annuel (Produit Net Bancaire) correspond à ce que rapporte un client à l'établissement. C'est donc une mesure intéressante qu'il convient d'étudier selon différents axes que peuvent être les caractéristiques de la clientèle (situation familiale, âge...), la structure commerciale de l'établissement, l'année... Le modèle multidimensionnel présenté dans la Figure 1 répond à ce besoin d'analyse.

Prenons le cas de la personne en charge de la clientèle étudiante au Crédit Lyonnais. Cette personne sait que certaines agences ouvertes récemment ne regroupent que des étudiants. Mais cette connaissance n'est pas visible dans le modèle et ne peut donc pas être utilisée a priori pour réaliser une analyse prenant en compte le type d'agence (agence dédiée ou non aux étudiants).

Notre modèle *R-DW* permet d'apporter une réponse à ce besoin d'analyse. La partie fixe du modèle est composée de la table des faits *TF_PNB* et des tables de dimension *CLIENT*, *ANNEE* et *AGENCE* (Figure 1). Nous ajoutons à cette partie fixe une partie évolutive contenant un ensemble de règles qui traduisent la connaissance de l'utilisateur. La connaissance sur les agences étudiantes peut être présentée par les règles suivantes :
$(R1) si\ idAgence \in$ {'01903','01905','02256'} $alors\ dim\_type\_agence =$ 'étudiant'
$(R2) si\ idAgence \notin$ {'01903','01905','02256'} $alors\ dim\_type\_agence =$ 'non étudiant'

Le modèle conceptuel induit (Figure 2) permet alors d'effectuer de nouvelles analyses générées par les connaissances de l'utilisateur. Grâce à notre modèle *R-DW*, il est donc possible

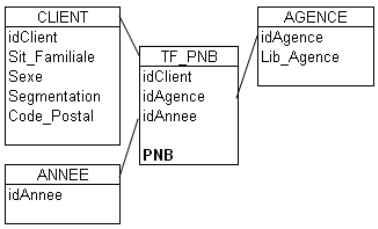

FIG. 1 – *Modèle conceptuel d'entrepôt de données pour l'analyse du PNB.*

de construire des agrégats, en considérant que les faits à agréger relèvent d'une agence étudiante $(R1)$, ou au contraire d'une agence non dédiée aux étudiants $(R2)$.

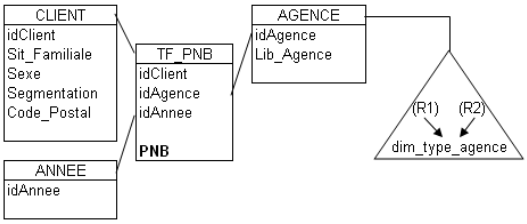

FIG. 2 – *Modèle conceptuel d'entrepôt de données à base de règles pour l'analyse du PNB.*

3 Etat de l'art

La prise en compte de nouveaux besoins d'analyse nécessite l'évolution de l'entrepôt de données. Deux alternatives peuvent être suivies. La première alternative propose la mise à jour du schéma, la seconde consiste à gérer différents schémas, en les historisant. Les travaux qui s'inscrivent dans la première alternative (Blaschka et al., 1999; Hurtado et al., 1999) consistent à migrer les données vers le schéma le plus récent. Dans ce cas, un seul schéma est supporté. L'avantage de cette approche est qu'elle fournit une comparaison des données dans le temps. Cependant, les données peuvent perdre de leur sens, voire disparaître (dans le cas d'une suppression d'un niveau de granularité dans une dimension). De plus, travailler avec la version la plus récente du schéma masque l'existence d'évolutions. Une analyse peut alors fournir de fausses conclusions. La deuxième alternative consiste à permettre une historisation des dimensions (Bliujute et al., 1998; Body et al., 2002; Chamoni et Stock, 1999; Eder et Koncilia, 2001) qui correspond à un versionnement de schémas. En effet, il s'agit de conserver chacune des versions du schéma. Le problème posé est alors de ne pas pouvoir avoir des comparaisons des données dans le temps, transversales aux différents schémas. Ces deux alternatives apportent une réponse à l'évolution des données. Mais elles n'apportent pas de solution à l'émergence de nouveaux besoins d'analyse qui sont orientés, non pas par l'évolution des données, mais par l'expression de nouvelles connaissances. Ces deux types de modélisation, malgré leurs inconvénients, permettent néanmoins d'apporter une certaine flexibilité temporelle au modèle.

Cette notion de flexibilité concerne l'ensemble de l'entrepôt. Différents travaux ont porté sur l'utilisation de langages à base de règles dans les entrepôts de données, pour apporter une flexibilité, que ce soit pour la définition de leur schéma, leur administration ou leur utilisation.

Différents travaux se sont intéressés à l'utilisation de langages à base de règles pour rendre la définition du schéma de l'entrepôt plus flexible. Deux alternatives sont possibles : utiliser les règles pour exprimer soit les besoins d'analyse, soit les connaissances sur la construction des entrepôts. Les travaux présentés dans (Kim et al., 2003) s'inscrivent dans la première alternative. Dans le contexte de la CRM (*Customer Relationship Management*), des règles de type "*si-alors*" permettent de représenter les campagnes marketing à analyser. La clause "*si*" présente les caractéristiques déterminant la population cible de la campagne marketing, et la clause "*alors*" comprend les caractéristiques de celle-ci. Le schéma de l'entrepôt est généré grâce à un algorithme qui extrait les mesures et les dimensions dans les clauses de la règle. Les travaux présentés dans (Peralta et al., 2003) s'inscrivent dans la deuxième alternative. Chacune des règles, qui représente une connaissance sur la construction de l'entrepôt de données, est spécifiée par une description, des structures cibles, un état de départ, des conditions d'application et un état final. Un algorithme gère l'ordre d'exécution des règles qui vont permettre une succession de transformations sur le schéma source pour obtenir le modèle logique final de l'entrepôt. Si cet ensemble de travaux propose de générer de façon automatique un schéma d'entrepôt en utilisant des règles, le problème de l'évolution n'est pas pris en compte.

Une des tâches importantes liées à l'administration de l'entrepôt est d'assurer la cohérence des données. En effet, si l'analyse est l'objectif primordial du processus décisionnel, celle-ci doit être faite avec cohérence. Ainsi, il est capital que l'administrateur puisse définir des contraintes d'intégrité pour assurer la cohérence à la fois de l'alimentation de l'entrepôt et de l'analyse des données. Dans (Carpani et Ruggia, 2001), les auteurs présentent un modèle conceptuel de données qui supporte un langage de contraintes, permettant d'assurer la cohérence des données. Les contraintes considérées peuvent porter sur des instances d'une dimension, ou sur différents niveaux d'une dimension, ou encore sur différentes dimensions simultanément. Dans (Hurtado et Mendelzon, 2002), les auteurs proposent des contraintes de dimension qui portent sur les chemins d'agrégation. Il s'agit d'exprimer, par exemple, que si le pays de vente est le Canada, alors les ventes seront agrégées par ville, puis par province. Le schéma enrichi de ces contraintes constitue alors un bon modèle pour inférer sur l'additivité, facilitant ainsi le processus d'analyse. Pour compléter ces approches, dans (Ghozzi et al., 2003), les auteurs proposent un modèle à contraintes pour les bases multidimensionnelles présentant un schéma en constellation. Ils définissent une typologie des contraintes sémantiques pouvant apparaître, non seulement au sein d'une même dimension, mais également entre les dimensions. Ils étendent les opérateurs de manipulation des données, en tenant compte de ces contraintes. Dans ces travaux, l'expression des contraintes d'intégrité permet d'utiliser la sémantique des données pour la gestion des incohérences dans les analyses. La sémantique des données n'est cependant pas exploitée pour l'analyse elle-même. Et si les approches proposées permettent des analyses cohérentes, l'évolution de ces dernières n'est pas évoquée.

Afin de pouvoir rendre l'analyse plus flexible, un langage à base de règles a été développé dans (Espil et Vaisman, 2001) 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 dimension du modèle. Ce langage constitue une alternative à la rigidité du schéma multidimensionnel lors

du processus d'agrégation, mais il ne fait qu'en modifier les chemins.

L'ensemble des travaux utilisant des langages à base de règles apporte une flexibilité, que ce soit dans la définition du schéma de l'entrepôt, dans son administration ou son utilisation. C'est précisément cette flexibilité que nous recherchons au niveau de l'analyse. L'évolution de l'analyse est conditionnée par celle des dimensions. La mise à jour du schéma de l'entrepôt ou le versionnement de schémas constituent une réponse au problème de l'évolution des dimensions, lorsque cette dernière est orientée par l'évolution des données elles-mêmes. En revanche, elles n'apportent pas de solution à l'émergence de nouveaux besoins d'analyse qui sont orientés par l'expression de nouvelles connaissances. Ainsi, pour répondre à notre objectif de faire évoluer les possibilités d'analyse de l'entrepôt en intégrant les connaissances de l'utilisateur, nous proposons un modèle d'entrepôt de données à base de règles.

4 Présentation du modèle *R-DW*

Un entrepôt de données présente une modélisation dite "dimensionnelle", qui répond à l'objectif d'observer les faits, à travers des mesures, en fonction des dimensions. Le schéma en étoile, qui constitue le schéma conceptuel de base pour un entrepôt de données, se compose classiquement d'une table des faits centrale et d'un ensemble de tables de dimension. D'un point de vue conceptuel, pour répondre à des besoins de performance et aux spécificités des données, le modèle en étoile a été étendu. D'une part, il a donné lieu à un schéma en constellation pour prendre en compte la nécessité de faire coexister plusieurs tables de fait qui partagent des dimensions. D'autre part, il a évolué vers un schéma en flocon de neige, dans lequel les dimensions ont été hiérarchisées. Cependant, ces modélisations ne constituent pas une réponse suffisamment flexible face à l'émergence de nouveaux besoins d'analyse. Dans ce travail, nous proposons alors un nouveau modèle d'entrepôt de données basé sur les règles (*R-DW*).

Le modèle *R-DW* est un modèle d'entrepôt de données composé de deux parties : une partie fixe, définie en extension, et une partie évolutive, définie en intension par des règles (Figure 3). La partie fixe peut être vue comme un schéma en étoile puisqu'elle comprend une table de faits et les dimensions de premier niveau (dimensions ayant un lien direct avec la table des faits). La partie évolutive comprend un ensemble de règles qui génèrent de nouvelles hiérarchies de dimension qui se basent sur la connaissance de l'utilisateur et sur les dimensions existantes.

Le métamodèle présenté dans la Figure 4 permet de généraliser le modèle *R-DW*. Il reprend en effet les classes de *Table de faits*, *Dimension* pour définir la partie fixe du modèle. Il comprend également la représentation de la partie évolutive, avec, en particulier, les classes *Règle définie en extension* et *Règle définie en intension* qui héritent de la classe *Règle*. En effet, il existe deux méthodes pour exprimer les règles qui génèrent les hiérarchies de dimension. Les règles sont exprimées en extension lorsqu'elles sont basées sur des valeurs connues qu'il est possible d'énumérer ; dans le cas contraire, elles sont définies en intension grâce à des fonctions.

Les règles exprimées en extension sont des règles de type "*si-alors*". Dans la clause "*alors*" figure la définition du niveau de granularité supérieur, en fonction de conditions exprimées dans la clause "*si*" qui portent sur les niveaux de granularité inférieurs et les connaissances de l'utilisateur. La règle suivante définit le niveau hiérarchique type_agence à travers l'attribut *dim_type_agence* en extension :
$si\ idAgence \in \{$'01903','01905','02256'$\}\ alors\ dim\_type\_agence =$ 'étudiant'
Les règles exprimées en intension sont des règles de calcul qui permettent d'inférer sur le ni-

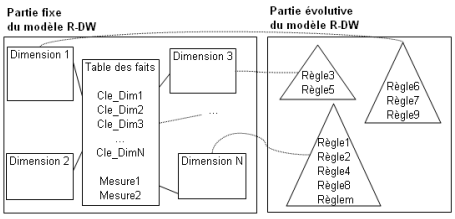

FIG. 3 – *Modèle R-DW.*

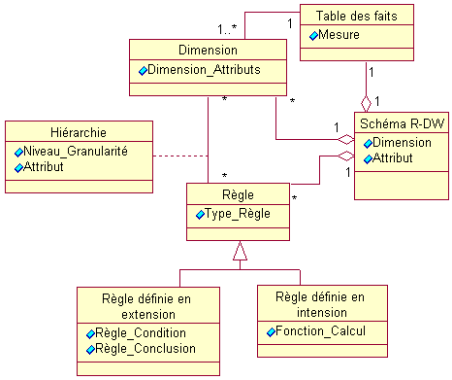

FIG. 4 – *Métamodèle de l'entrepôt de données R-DW.*

veau de granularité en fonction des niveaux inférieurs. Cette règle de calcul peut correspondre à une fonction de scoring, une extraction de caractères... Considérons par exemple le code postal du client dont on observe le PNB. En extrayant les deux premiers caractères du code postal, on obtient l'indicatif du département. À partir du code postal, on peut donc inférer sur le niveau hiérarchique département. Un autre exemple concerne les dates : il est possible d'extraire facilement les informations concernant les jour, mois, et année à partir de leur décomposition. Ce type d'extraction peut également être pertinent pour les données qui répondent à des normes : l'identification de produits, de magasins... C'est le cas par exemple au Crédit Lyonnais où la segmentation des clients comprend deux caractères. Le premier est un chiffre qui correspond au segment du client, le second est une lettre qui représente son potentiel. Il est alors facile, à partir de la segmentation du client, de définir son segment.

L'intérêt de ce modèle est donc d'offrir à l'utilisateur la possibilité de définir ses propres règles pour déterminer de nouvelles hiérarchies de dimension. L'utilisateur peut ensuite effectuer une analyse selon les niveaux de granularité définis par les règles.Le modèle devient plus flexible. La modélisation sous forme de règles offre de réelles possibilités en terme de contenu des niveaux de granularité. En effet, pour générer un niveau de granularité, il est possible :

- d'intégrer les connaissances de l'utilisateur (Exemple : cas des agences étudiantes) ;
- d'introduire les informations disponibles dans d'autres sources de données (Exemple : si nous avons une source où figure le nombre d'employés par agence, il est possible de faire l'analyse du PNB selon le nombre d'employés de l'agence) ;

– de combiner les attributs d'une même dimension. (Exemple : analyse du PNB en fonction de l'appartenance du client aux groupes 'femmes mariées' ou 'hommes mariés', constitués à partir de la combinaison des attributs *Sit_Familiale* et *Sexe*).

5 Cadre formel de l'approche

5.1 Modèle *R-DW*

Nous représentons le modèle d'entrepôt à base de règles *R-DW* par le triplet suivant :

$$\textit{R-DW} = (\mathcal{F}, \mathcal{E}, \mathcal{U})$$

où $\mathcal{F}$ est la partie fixe de *R-DW*, $\mathcal{E}$ la partie évolutive et $\mathcal{U}$ l'univers de l'entrepôt *R-DW*.

Définition 1. *Univers de l'entrepôt*

L'*univers de l'entrepôt* $\mathcal{U}$ est un ensemble d'attributs, tel que :

$\mathcal{U} = \{B_1, B_2, \ldots, B_z, C_1, C_2, \ldots\}$, où $B_\alpha, 1 \leq \alpha \leq z$, est un attribut prédéfini et $C_\beta, \beta \geq 1$, est un attribut généré.

Définition 2. *Partie fixe de R-DW*

La *partie fixe de R-DW* est représentée par $\mathcal{F} = (F, \mathcal{D})$, où F est une table de faits, et $\mathcal{D} = \{D_s, 1 \leq s \leq t\}$ est l'ensemble des dimensions de premier niveau qui ont un lien direct avec la table des faits F. Nous supposons que ces tables de dimension sont indépendantes.

Une table de dimension D_s contient une clé primaire $D_s.PK$ et un ensemble de n_s attributs $\{D_s.Z_n, 1 \leq n \leq n_s\}$.

La table de faits F contient une clé composée d'un ensemble de t ($t \geq 1$) clés étrangères $\{F.K_s, 1 \leq s \leq t\}$ où $F.K_s = D_s.PK$, et un ensemble de m mesures notées $\{F.M_q, 1 \leq q \leq m\}$.

Exemple 2. Dans la Figure 1, (TF_PNB,$\{AGENCE, ANNEE, CLIENT\}$) constitue la partie fixe de l'entrepôt *R-DW* pour l'analyse du PNB.

L'expression de nouveaux besoins d'analyse se traduit par la définition de niveaux de granularité dans les hiérarchies de dimension.

Définition 3 *Hiérarchie de dimension et niveau de granularité*

Soit *R-DW* $= (< F, \mathcal{D} >, \mathcal{E}, \mathcal{U})$ un entrepôt de données.

Soit $D_s.H_k, k \geq 1$ une *hiérarchie de la dimension* $D_s \in \mathcal{D}$.

$\{L_1, L_2, \ldots, L_i, \ldots, L_w, w \geq 1\}$ forme la hiérarchie de dimension $D_s.H_k$, avec $L_1 \prec L_2 \prec \ldots \prec L_i \prec \ldots \prec L_w$.

$L_i, 1 \leq i \leq w$ est appelé *niveau de granularité* de la hiérarchie H_k de la dimension D_s et est noté $D_s.H_k.L_i$ ou L_i^{sk}. Les niveaux de granularité sont définis par des attributs générés.

Définition 4 *Attribut généré*

Nous appelons *attribut généré* un attribut qui caractérise un niveau de granularité dans une hiérarchie de dimension. Les modalités de cet attribut sont définies grâce à des règles qui seront présentées par la suite.

$\{D_s.H_k.L_i.A_b, 1 \leq b \leq d\}$ est l'ensemble des d attributs générés, qui caractérisent le niveau de granularité L_i de la hiérarchie H_k de la dimension D_s.

Pour des raisons de simplification, nous supposons que chaque niveau d'une hiérarchie de dimension ne contient qu'un seul attribut généré, même s'il est possible de générer plusieurs attributs par niveau de granularité. Ainsi, l'attribut généré caractérisant le niveau de granularité L_i de la hiérarchie H_k de la dimension D_s est noté $L_i^{sk}.A$. Ces attributs générés sont définis grâce à la partie évolutive de *R-DW*.

Définition 5. *Partie évolutive de R-DW*

La *partie évolutive de R-DW* est représentée par $\mathcal{E} =< \mathcal{R}, \mathcal{L} >$ avec $< \mathcal{R}, \mathcal{L} >= \{< \mathcal{R}_i, L_i^{sk}.A >\}$ où $\mathcal{R}_i = \{r_{ij}, 1 \leq j \leq v, 1 \leq i \leq w\}$ est un ensemble de v règles définissant les modalités de l'attribut généré $L_i^{sk}.A$ qui représente le niveau de granularité Li de la hiérarchie H_k de la dimension D_s.

En fonction des connaissances dont nous disposons, nous exprimons les règles soit en extension, soit en intension. Nous allons définir les règles exprimées en extension (resp. intension) qui vont permettre de créer le lien entre les différents niveaux de granularité dans les dimensions grâce à l'expression de conditions (resp. de fonctions de calcul).

Définition 6. *Règle définie en extension*

Une *règle définie en extension* est une règle de type "*si-alors*", qui permet de marquer le lien sémantique qui existe entre deux niveaux de granularité dans une hiérarchie de dimension.

Elle est basée sur des termes de règle, notés RT_p tel que $RT_p = U_r\ op\ \{ens|val\}, 1 \leq p \leq n$ où $U_r \in \mathcal{U}$ l'univers de l'entrepôt ; *op* est un opérateur soit relationnel (=, <, >, ≤, ≥, ≠, ...), soit ensembliste (∈, ∉, ...) ; ens est un ensemble de valeurs et val est une valeur finie.

Exemple 6a. Les expressions $idAgence \in$ {'01903','01905','02256'}, $idAnnee < 2001$ et $Sexe =$ 'F' constituent des termes de règle.

Une règle définie en extension est basée sur une composition de *(conjonctions|disjonctions)* de ces termes de règles : $r_{ij} : si\ RT_1\ (and|or)\ RT_2 \ldots (and|or)\ RT_n\ alors\ L_i^{sk}.A = val$

Exemple 6b. Les règles suivantes définissent les modalités de l'attribut dim_type_agence en extension :

r_{11} : $si\ idAgence \in$ {'01903','01905','02256'} $alors\ dim\_type\_agence =$ 'étudiant'

r_{12} : $si\ idAgence \notin$ {'01903','01905','02256'} $alors\ dim\_type\_agence =$ 'non étudiant'

Les règles définies en extension permettent donc de créer ou d'enrichir une hiérarchie de dimension en définissant les modalités de l'attribut en fonction d'une condition ou de composition de conditions basées sur les attributs des niveaux inférieurs de la dimension.

Exemple 6c. La règle suivante permet de définir la valeur *'femmes_mariées'* de l'attribut *dim_groupe_personnes* à partir des attributs *Sit_Familiale* et *Sexe* :

$si\ Sit\_Familiale =$ 'Marié' $and\ Sexe =$ 'F' $alors\ dim\_groupe\_personne =$ 'femmes mariées'

Définition 7. *Règle définie en intension*

Une *règle définie en intension* est une règle qui permet de calculer l'attribut qui caractérise le niveau de granularité ajouté dans la hiérarchie de dimension, à partir d'attributs de niveaux inférieurs : $r_{ij} : L_i^{sk}.A = f(\mathcal{U})$

où $f(\mathcal{U})$ désigne une fonction quelconque *(extraction de caractères|fonction de scoring| ...)* pouvant s'appliquer sur un ou plusieurs attributs de l'univers de l'entrepôt $\mathcal{U}$.

Exemple 7. La règle suivante définit l'attribut $dim\_departement$ en intension :

$dim\_departement = gauche(Code\_Postal, 0, 2)$

où $gauche(chr, x, y)$ est une fonction qui permet d'extraire y caractères de la chaîne de caractères chr à partir de la position x.

5.2 Processus d'agrégation

$\mathcal{RE}$ désigne l'ensemble des règles définies en extension.Dans la règles r_{ij} définie en extension, la condition dans la clause "*si*" est notée $body(r_{ij})$, et la conclusion dans la clause "*alors*" est notée $head(r_{ij})$.

Pour réaliser une analyse à partir de l'entrepôt *R-DW*, il est nécessaire de prendre en compte les règles. Nous avons donc défini un algorithme qui permet le calcul d'agrégats (Figure 5).

Pour des raisons de clarté, nous nous restreignons à une agrégation selon un attribut déterminé par un ensemble de règles définies en extension . Cet algorithme permet de construire une table d'agrégats $TAgreg$ à partir de l'entrepôt et des caractéristiques de l'analyse (attribut A selon lequel est faite l'agrégation, mesure M_q, opérateur d'agrégation op).

Cet algorithme peut être utilisé pour répondre à la requête décisionnelle "Quel est le PNB moyen par type d'agence ?". Dans ce cas, $TAgreg$ est une table contenant deux tuples où figurent le PNB moyen pour les agences de type 'étudiant' d'une part, et de type 'non étudiant' d'autre part. Les valeurs correspondantes ont été obtenues par l'exécution des requêtes suivantes :

(1) SELECT MOY(PNB) FROM TF_PNB WHERE idAgence IN ('01903','01905','02256') ;
(2) SELECT MOY(PNB) FROM TF_PNB WHERE idAgence NOT IN ('01903','01905','02256') ;

```
Algorithme Calcul_Agreg
Input :  table des faits F,
             ensemble des règles définies en extension RE,
             attribut A,
             mesure F.M_q,
             opérateur d'agrégat op,
Output : table des agrégats TAgreg
Début
    Pour chaque r_ij ∈ RE
       Si A ∈ head(r_ij) Alors
          TAgreg =' SELECT op(F.M_q) FROM F WHERE body(r_ij)'
       Fin Si
    Fin pour
Fin
```

FIG. 5 – *Algorithme de calcul d'agrégats*

6 Mise en œuvre et application aux données bancaires

Pour valider notre approche, nous avons réalisé une implémentation du modèle *R-DW*. Nous avons développé une plateforme Web (HTML/PHP), qui interface le SGBD Oracle. La table de faits et les tables de dimension sont définies dans Oracle. Deux tables permettent de regrouper respectivement les règles définies en extension et les règles définies en intension. La plateforme Web permet à l'utilisateur de visualiser et de définir les règles qui génèrent des axes d'analyse. Elle permet également la visualisation des résultats d'analyse. Concernant l'analyse, nous nous sommes restreints, dans un premier temps, à une requête décisionnelle mettant en jeu une agrégation selon un niveau hiérarchique d'une dimension donnée. Cette agrégation est implémentée sous la forme d'une procédure stockée en PL/SQL du SGBD.

Nous avons appliqué notre modélisation aux données bancaires qui concernent l'analyse du PNB. La partie fixe du schéma est représentée dans la Figure 1. À partir de ces dimensions, et des connaissances de l'utilisateur, différentes hiérarchies de dimension ont été représentées par l'ensemble de la Figure 6, qui constitue la partie évolutive du schéma *R-DW*. Ainsi, à partir de ce nouveau modèle, l'utilisateur pourra effectuer des analyses sur le PNB, non seulement en utilisant les dimensions de premier niveau, mais également en faisant intervenir des niveaux de granularité comme le type d'agence, la période, le département, les classes d'âge...

FIG. 6 – *Partie évolutive de l'entrepôt R-DW pour l'analyse du PNB.*

7 Conclusion

Dans cet article, nous avons proposé un nouveau modèle d'entrepôt de données à base de règles nommé *R-DW*. Les règles permettent d'intégrer de nouvelles connaissances de l'utilisateur dans l'entrepôt. Notre modèle *R-DW* est composé de deux parties : une partie fixe, définie en extension, comprenant une table des faits et les dimensions de premier niveau ; une partie évolutive, définie en intension par des règles, qui détermine les niveaux de granularité dans les hiérarchies de dimension. Notre modèle *R-DW* présente plusieurs avantages. Il permet de créer des hiérarchies de dimension de façon dynamique en renforçant l'interaction entre l'utilisateur et le système d'aide à la décision. En effet, les hiérarchies de dimension sont générées grâce aux propres connaissances de l'utilisateur. Cette génération de hiérarchies de dimension "à la demande" permet de faire évoluer les contextes d'analyse. Par ailleurs, nous avons proposé un métamodèle qui décrit le modèle *R-DW*. L'implémentation que nous avons réalisée a permis d'appliquer notre modèle aux données bancaires réelles de LCL-Le Crédit Lyonnais.

Ce travail ouvre différentes perspectives. Tout d'abord, nous voulons mesurer la performance de notre approche en termes d'espace de stockage et de temps de réponse. En terme de performance, il s'agit également d'étudier le problème de la matérialisation des vues et des structures d'index. Ensuite, concernant la définition des règles, l'atout de notre approche est de pouvoir faire intervenir l'utilisateur en lui laissant la possibilité d'introduire ses connaissances dans le système. Mais il nous semble intéressant, en parallèle, de pouvoir l'aider à découvrir de nouveaux axes d'analyse. Pour ce faire, nous pensons que des méthodes d'apprentissage non supervisé peuvent être utilisées. Il s'agit également de définir un langage qui permette de valider les règles utilisées, que ce soit pour la gestion des conflits entre les règles, ou pour la vérification de contraintes sur celles-ci.

Références

Blaschka, M., C. Sapia, et G. Höfling (1999). On Schema Evolution in Multidimensional Databases. In *DaWaK'99 : 1st International Conference on Data Warehousing and Knowledge Discovery*, pp. 153–164.

Bliujute, R., S. Saltenis, G. Slivinskas, et C. Jensen (1998). Systematic Change Management in Dimensional Data Warehousing. In *3rd International Baltic Workshop on DB and IS*.

Body, M., M. Miquel, Y. Bédard, et A. Tchounikine (2002). A Multidimensional and Multiversion Structure for OLAP Applications. In *DOLAP'02 : 5th ACM International Workshop on Data Warehousing and OLAP*.

Cabibbo, L. et R. Torlone (1998). A Logical Approach to Multidimensional Databases. In *EDBT'98 : 6th International Conference on Extending Database Technology*, pp. 183–197.

Carpani, F. et R. Ruggia (2001). An Integrity Constraints Language for a Conceptual Multidimensional Data Model. In *SEKE'01 : XIII International Conference on Software Engineering Knowledge Engineering*.

Chamoni, P. et S. Stock (1999). Temporal Structures in Data Warehousing. In *DaWaK'99 : 1st International Conference on Data Warehousing and Knowledge Discovery*, pp. 353–358.

Eder, J. et C. Koncilia (2001). Changes of dimension data in temporal data warehouses. In *DaWaK'01 : 3rd International Conference on Data Warehousing and Knowledge Discovery*.

Espil, M. M. et A. A. Vaisman (2001). Efficient Intensional Redefinition of Aggregation Hierarchies in Multidimensional Databases. In *DOLAP'01 : 4th ACM International Workshop on Data Warehousing and OLAP*.

Favre, C., F. Bentayeb, O. Boussaid, et N. Nicoloyannis (2005). Entreposage virtuel de demandes marketing : de l'acquisition des objets complexes à la capitalisation des connaissances. In *2ème atelier FDC de EGC05, Paris*, pp. 65–68.

Ghozzi, F., F. Ravat, O. Teste, et G. Zurfluh (2003). Constraints and Multidimensional Databases. In *ICEIS'03 : 5th International Conference on Enterprise Information Systems*.

Hurtado, C. A. et A. O. Mendelzon (2002). OLAP Dimension Constraints. In *PODS'02 : 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*.

Hurtado, C. A., A. O. Mendelzon, et A. A. Vaisman (1999). Updating OLAP Dimensions. In *DOLAP'99 : 2nd ACM International Workshop on Data Warehousing and OLAP*.

Inmon, W. H. (1996). *Building the Data Warehouse*. John Wiley & Sons.

Kim, H. J., T. H. Lee, S. G. Lee, et J. Chun (2003). Automated Data Warehousing for Rule-Based CRM Systems. In *14th Australasian Database Conference on Database Technologies*, pp. 67–73.

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

Peralta, V., A. Illarze, et R. Ruggia (2003). On the Applicability of Rules to Automate Data Warehouse Logical Design. In *CAiSE Workshops*.

Summary

Data warehouses are an answer to enterprises'analysis needs. However, such a system is hard to build and maintain, particularly when these analysis needs evolve. To solve this problem, we propose a new data warehouse model based on rules called Rule-based Data Warehouse (*R-DW*). The rules are used to integrate user's knowledge in the data warehouse. This model is composed of two parts : one fixed part, defined extensionally, composed of a fact table and dimensions of the first level ; a second evolving part, defined intentionally with rules. Having these rules we are able to dynamically create dimension hierarchies. It makes thus possible the contexts of analysis evolution, and it increases the interaction between the user and the decision support system.