Métadonnées de personnalisation dans le système SelfStar

Fatma Abdelhédi, Franck Ravat,
Olivier Teste, Gilles Zurfluh

Université de Toulouse – IRIT (UMR 5505)
118, Route de Narbonne – 31062 Toulouse cedex 9 (France)
{Fatma.Abdelhedi, Franck.Ravat, Olivier.Teste, Gilles.Zurfluh}@irit.fr

Résumé. Pour concevoir une base décisionnelle, un décideur fait généralement appel à un spécialiste : concepteur de bases de données multidimensionnelles, cogniticien ou informaticien. Celui-ci élabore un schéma multidimensionnel en termes de faits à analyser et d'axes d'analyse selon une démarche ascendante, descendante ou mixte. Ce processus, qui s'appuie sur les sources de données à analyser et le cas échéant sur les besoins des décideurs, s'avère complexe et souvent approximatif. Le projet SelfStar vise à définir des mécanismes et des outils permettant à un décideur d'élaborer lui-même un schéma multidimensionnel. Pour assister et guider le décideur, SelfStar alimente puis utilise des métadonnées personnalisées. Dans le processus d'élaboration d'une base décisionnelle, ces métadonnées vont aider l'utilisateur en guidant ses choix. Cet article présente les principes et les algorithmes qui génèrent les métadonnées associées à chaque décideur et les implantées dans un prototype.

1 Introduction

L'analyse d'une base de données par un décideur s'effectue généralement au travers d'une base décisionnelle (entrepôt, magasin) organisée selon un schéma multidimensionnel. L'élaboration d'un tel schéma repose sur une démarche ascendante, descendante ou mixte qui est mise en œuvre par un informaticien.

Dans le projet SelfStar, nous proposons que chaque décideur élabore ses propres schémas en constellation sans l'assistance d'un informaticien. Ce processus d'élaboration repose sur une démarche mixte où la base de données à analyser (la source) et les besoins d'analyse sont connus. Il est incrémental dans le sens où le décideur élabore progressivement le schéma multidimensionnel en intégrant successivement les faits, les dimensions et les hiérarchies. Il est assisté dans la mesure où des métadonnées ont été enregistrées et permettent, dans le processus d'élaboration, de limiter les choix du décideur.

Dans cet article nous nous focalisons sur la production des métadonnées qui vont permettre au système SelfStar d'assister les décideurs dans leur choix. En effet, suite à des observations faites dans des applications industrielles, on part de l'hypothèse que les décideurs effectuent des analyses proches lorsque celles-ci portent sur une seule source de données. Nous distinguons deux catégories de métadonnées : les métadonnées de base (non personnalisées) générées à partir de la source et les métadonnées de personnalisation produites à partir des schémas décisionnels.

Dans la section 2, nous présentons les objectifs et la justification du projet SelfStar. La section 3 situe nos travaux par rapport à l'état de l'art. La section 4 décrit la démarche d'élaboration d'un schéma décisionnel dans SelfStar. La section 5 présente le schéma d'une source utilisée comme exemple d'application. Les sections 6 et 7 décrivent les algorithmes de production des métadonnées de base et de personnalisation. Dans la section 8, une mise en œuvre des algorithmes est présentée. La section 9 décrit les caractéristiques du prototype que nous avons développé.

2 Le projet SelfStar

2.1 Le contexte

Les décideurs utilisent les bases décisionnelles pour analyser des données organisées selon un schéma multidimensionnel (étoile, flocon ou constellation). L'élaboration d'un tel schéma est généralement confiée à un spécialiste (informaticien, administrateur de données,...) qui utilise une démarche ascendante, descendante ou mixte (Phipps & Davis 2002), (Prat et al. 2006), (Romero & Abelló 2010). Ces démarches se distinguent par la nature des données en entrée du processus d'élaboration :

- la source pour la démarche ascendante,
- les besoins du décideur pour la démarche descendante,
- la source et les besoins du décideur pour la démarche mixte.

Les décideurs doivent faire appel à des spécialistes de la gestion des données chaque fois qu'ils souhaitent obtenir de nouvelles bases décisionnelles ou faire évoluer des schémas multidimensionnels existants. Ceci est principalement lié à la complexité :

- des principes de modélisation d'une base décisionnelle,
- des mécanismes de correspondance entre source et base décisionnelle pour assurer l'alimentation périodique de celle-ci (processus ETL) (Vassiliadis 2009).

2.2 L'objectif

Le projet SelfStar (Abdelhédi et al., 2011) propose que le décideur (par définition non informaticien) construise lui-même, à partir d'une source à analyser, la base décisionnelle dont il a besoin (nous nous limitons pour l'instant à l'élaboration du schéma multidimensionnel). Les objectifs visés par ce projet sont les suivants :

- on supprime un intermédiaire (le spécialiste en gestion des données), ce qui assouplit considérablement la démarche d'élaboration du schéma multidimensionnel,
- on évite au décideur d'exprimer (plus ou moins formellement) ses besoins d'analyse et de les communiquer au spécialiste chargé d'élaborer le schéma,
- on permet au décideur de construire de nouvelles bases décisionnelles quand il le souhaite, au gré de ses besoins.

Mais le décideur, même s'il connaît bien ses besoins, est évidemment confronté à une double complexité :

- celle de l'organisation des données de la source (schéma Relationnel, Entité-Association ou UML[1]),
- celle du processus d'élaboration du schéma multidimensionnel (fait, dimensions, hiérarchies).

Le projet SelfStar a donc pour objectif de définir une démarche complète et un environnement logiciel pour permettre à un décideur de construire un schéma en constellation. Le processus d'élaboration du schéma repose sur une démarche mixte : il part du schéma de la base source (diagramme de classes DCL) et des besoins du décideur. Ce processus est incrémental : les besoins d'analyse sont intégrés progressivement dans 3 schémas successifs (schémas intermédiaires).

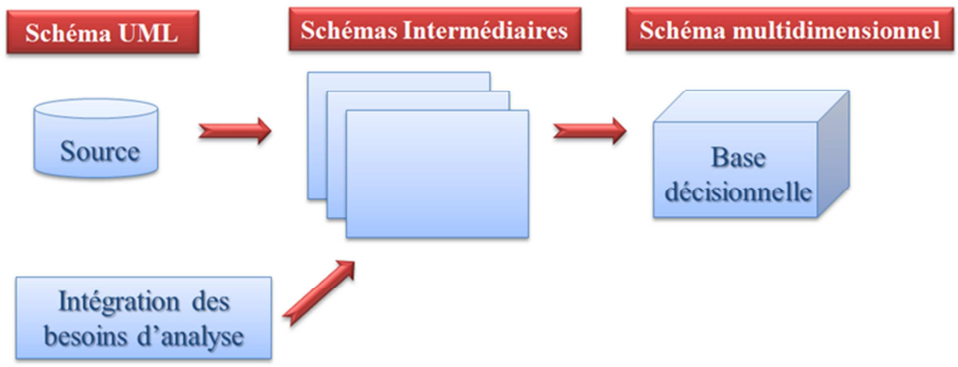

FIG. 1 – *Processus de construction d'une base décisionnelle.*

[1] Unified Modeling Language

Le choix de UML dans SelfStar est lié au fait que ce formalisme est largement reconnu pour décrire des schémas conceptuels de sources de données. De plus, le langage UML offre l'avantage d'être sémantiquement très riche des schémas de données. Dans l'hypothèse où l'on dispose de sources relationnelles, il est alors nécessaire de traduire le schéma relationnel sous la forme d'un DCL.

2.3 Justification des métadonnées

Lors d'expériences industrielles (Annoni et al. 2006), nous avons constaté que des décideurs analysant la même source de données effectuent régulièrement des analyses voisines, c'est-à-dire utilisant des faits ou des dimensions déjà utilisés. Ce constat nous a conduits à proposer au décideur, à chaque nouvelle analyse, une aide pour élaborer le schéma multidimensionnel. Cette assistance se traduit par une sélection des classes candidates à l'analyse parmi l'ensemble des classes que contient le schéma d'une source (plusieurs dizaines). Le décideur pourra ainsi rapidement localiser, parmi ce sous ensemble de classes, celles qui feront l'objet d'une analyse

À l'élaboration d'un schéma multidimensionnel, des métadonnées de personnalisation sont enregistrées par SelfStar. Lors de la construction de nouveaux schémas, ces métadonnées permettront de sélectionner un sous ensemble de classes candidates à l'analyse, en évitant au décideur une connaissance approfondie du schéma de la source. En effet, grâce à ce mécanisme, seuls les faits supposés les plus significatifs pour le décideur seront extraits de la source.

3 Travaux associés

Les bases décisionnelles (entrepôts, magasins) sont des bases de données dédiées à la prise de décisions (Inmon 1996). Les données qu'elles contiennent sont extraites de différentes sources, en particulier des bases de données internes à l'entreprise. Généralement une source est décrite par un modèle conceptuel de type entité/association ou objet alors qu'une base décisionnelle est organisée selon un modèle multidimensionnel : étoile, flocon ou constellation (Kimball 1996).Chaque décideur (ou classe de décideurs) doit disposer de bases adaptées à ses besoins de prise de décisions. Or, une particularité des besoins décisionnels est leur évolution rapide (Malinowski & Zimányi 2008) ; les analyses réalisées nécessitant d'adapter fréquemment les mesures étudiées ainsi que les axes analysés.

Plusieurs travaux ont proposé des démarches de conception pour l'élaboration des schémas multidimensionnels. Ces démarches peuvent être classées selon 3 catégories : ascendante, descendante et mixte.

La démarche ascendante, (Golfarelli et al. 1998), (Moody & Kortink 2000) et (Feki & Hachaichi 2007), utilise les sources de données pour générer des schémas multidimensionnels candidats et présentent l'inconvénient de ne pas prendre en compte les besoins des décideurs. Dans (Golfarelli et al. 1998) et (Moody & Kortink 2000), les auteurs ont défini une méthode semi-automatique pour la génération des schémas candidats à partir des sources de données opérationnelles Entité-Association. Dans un second temps, l'utilisateur peut choisir

un schéma multidimensionnel en fonction de ses besoins parmi l'ensemble des schémas générés. Dans (Song et al. 2008), les auteurs proposent de générer automatiquement des schémas candidats à partir du schéma source Entité-Association. Ils proposent une nouvelle approche intitulée *connectiontopology value* qui identifie automatiquement les faits candidats. Toutefois, ces faits peuvent ne pas satisfaire le décideur.

Dans (Pinet & Schneider 2009), les auteurs proposent un modèle *MultiDimensional-Schema* à partir d'un schéma conceptuel UML. Ce modèle représente les classes de la source sous forme d'un graphe acyclique orienté. Le décideur choisit un nœud de ce modèle pour représenter le fait. Tous les nœuds reliés au fait choisi représentent les dimensions potentielles de ce fait. Cependant, cette représentation du schéma multidimensionnel reste, à notre avis, lourde pour le décideur.

La démarche descendante selon (Trujillo et al. 2003) et (Prat et al. 2006) prend en compte les besoins des décideurs, mais les sources de données sont ignorées. Et par la suite, la correspondance entre le schéma résultat et la source de données s'avère délicate.

La démarche mixte dans (Giorgini et al. 2005) et (O. Romero & A. Abelló 2010) combine les deux processus précédents. En effet cette démarche construit d'une part des schémas candidats à partir des données (démarche ascendante) et d'autre part des schémas multidimensionnels à partir des besoins d'analyse (démarche descendante). Puis un informaticien doit confronter ces 2 types de schémas pour obtenir un schéma multidimensionnel résultat cohérent et répondant aux besoins des décideurs.

(O. Romero & A. Abelló 2010) proposent une méthode automatique *Multidimensional Design by Examples* qui suit une démarche mixte. Pour générer des schémas multidimensionnels, cette méthode prend en entrée d'une part les besoins des décideurs exprimés avec des requêtes SQL et d'autre part la source de données relationnelle. L'interrogation des sources est assurée par des requêtes SQL et une connaissance du schéma relationnel de la source. Par conséquent, la construction du schéma multidimensionnel nécessite un expert (un informaticien) pour formuler les requêtes SQL et interroger les sources de données.

Les approches que nous venons de présenter font intervenir un informaticien pour élaborer les schémas multidimensionnels. À notre connaissance, peu de travaux font intervenir uniquement le décideur dans le processus d'élaboration d'un schéma multidimensionnel. Notons cependant que l'article de (Pinet & Schneider 2009) propose au décideur de réduire le schéma multidimensionnel généré par le système, ceci afin de faire correspondre le schéma à ses besoins d'analyse. Mais les interactions avec le décideur restent limitées.

4 La démarche

Le processus proposé dans SelfStar (Abdelhédi et al., 2011) a pour point de départ le schéma de la source de données (DCL du langage UML) et les besoins (informels) du décideur. Le DCL s'avère trop complexe pour être exploité en l'état ; il va donc être transformé en un "schéma exploitable" qui contiendra uniquement les informations utiles (T1). Le processus comporte quatre phases successives dans lesquelles le décideur va interagir avec le système pour intégrer progressivement ses besoins. Chaque phase produit un nouveau sché-

ma plus complet que celui de la phase précédente. Le 4ème et dernier schéma correspond à celui de la base décisionnelle, c'est-à-dire au résultat escompté. L'élaboration du schéma décisionnel est donc incrémentale.

La *première phase* consiste à extraire de la source l'ensemble des faits candidats (susceptibles d'être choisis) et à les afficher dans le schéma intermédiaire numéro 1 (noté SI_1).

Sur le SI_1, le décideur choisit le fait qu'il souhaite analyser parmi tous les faits candidats proposés avec les mesures et leurs fonctions d'agrégations.

Dans une *deuxième phase*, le système élabore automatiquement le schéma intermédiaire numéro 2 (SI_2) ; il propose toutes les dimensions possibles associées au fait choisi.

Sur le SI_2, le décideur va pouvoir désigner les dimensions, c'est-à-dire les axes selon lesquels il souhaite analyser le fait

La *troisième phase* génère le schéma intermédiaire numéro 3 (SI_3) contenant le schéma multidimensionnel (fait (s) + dimensions) avec toutes les hiérarchies dimensionnelles possibles.

Sur le SI_3, le décideur va choisir chacune des hiérarchies correspondantes à ses besoins.

La *quatrième phase* du processus va permettre au système d'élaborer le schéma multidimensionnel qui correspond aux besoins décisionnels.

Ce processus incrémental va permettre d'enregistrer 2 catégories de métadonnées: (1) des *métadonnées de base* qui hiérarchisent les classes de la source, (2) des *métadonnées de personnalisation* qui vont assister le décideur dans l'élaboration du schéma multidimensionnel.

Les métadonnées de personnalisation sont générées en phase 4 et utilisées en phase 1 pour proposer au décideur les faits les plus significatifs.

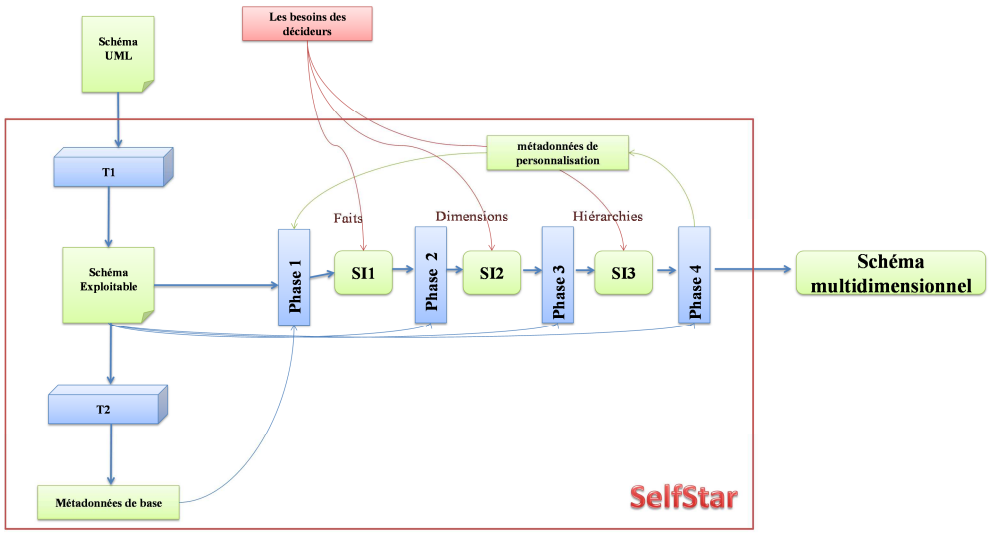

FIG. 2 – *Les 4 phases de la démarche construction d'une base décisionnelle.*

5 Exemple d'application

Pour illustrer les mécanismes proposés dans cet article, nous allons prendre l'exemple d'une source de données relative à une gestion de vente. Le DCL de cette source est présenté dans la FIG.3.

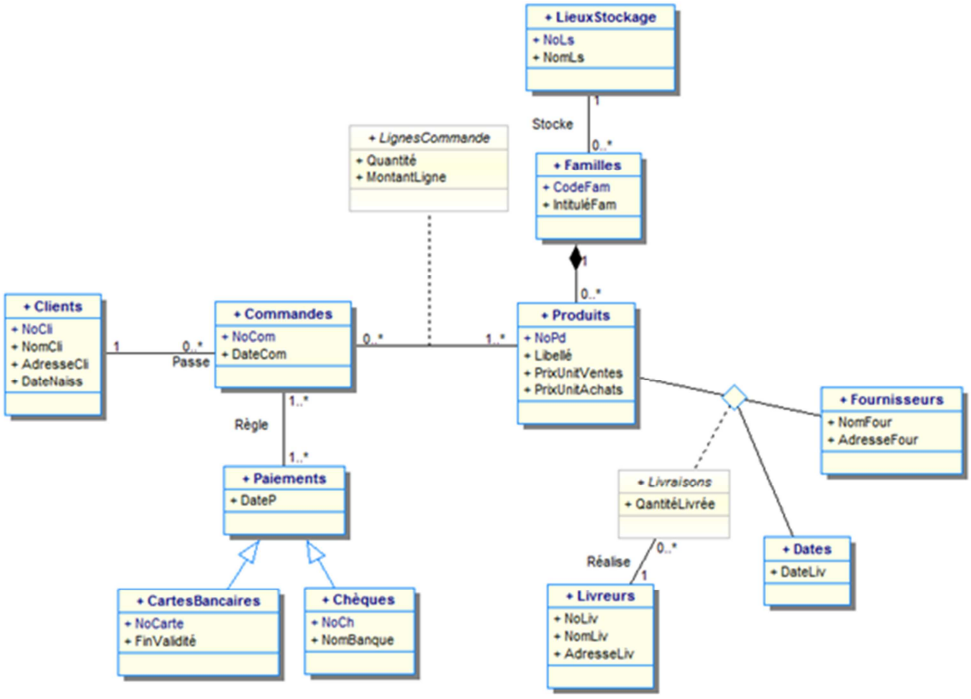

FIG. 3 – *DCL pour la gestion des vente.*

Le DCL n'étant pas aisément exploitable en l'état, il fait donc l'objet d'un prétraitement. Ce principe a déjà été proposé dans (Song et al. 2008) pour le modèle Entité-Association et dans (Pinet & Schneider 2009) pour le modèle des classes UML. Selon ce principe, le schéma conceptuel est transformé en un schéma simplifié comportant uniquement des classes d'objets et des liens binaires de type 1..N entre ces classes. La transformation d'un DCL en un schéma exploitable s'effectue comme suit :

- une classe d'objets ou d'associations de la source devient une classe dans le schéma exploitable,
- un lien d'association binaire de type 1..N est reporté en l'état,
- un lien d'association binaire de type M..N est transformé en une classe liée par 2 liens 1..N,
- un lien d'agrégation ou de composition est traité comme un lien d'association (ces types de liens ne sont pas significatifs dans un schéma multidimensionnels),

- un lien d'héritage disparaît ; la sous-classe de la source devient une classe et se retrouve au même niveau que la super-classe en héritant de ses attributs et liens (on préserve ainsi la sémantique des données).

La FIG.4 présente le schéma exploitable correspondant au DCL de la FIG.3.

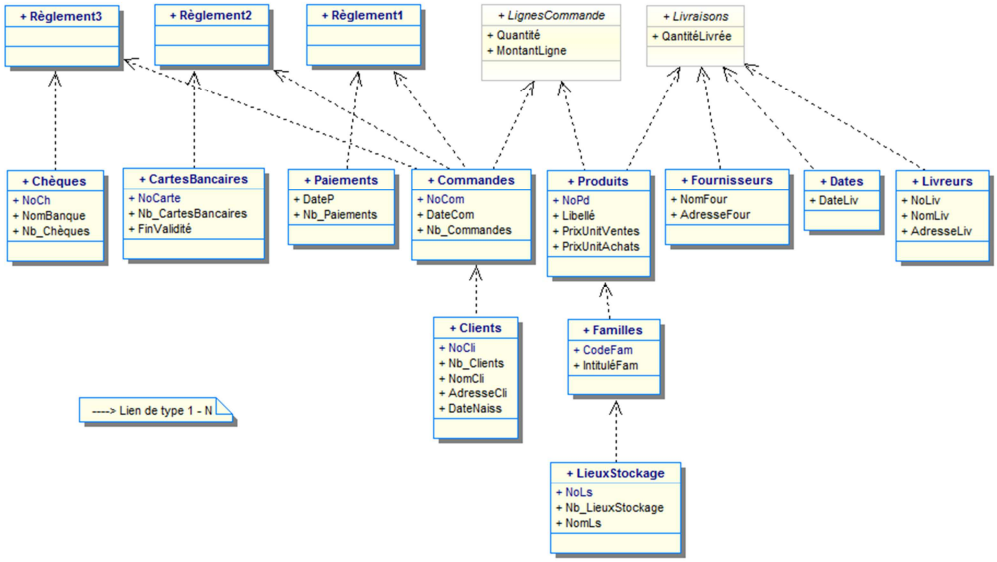

FIG.4 – Schéma exploitable du DCL.

6 Les métadonnées de base

Pour assister et guider le décideur dans la construction d'un schéma multidimensionnel, le système SelfStar alimente et utilise une métabase. Dans la démarche SelfStar (FIG 2), faits et dimensions sont extraits du schéma de la source de données selon un algorithme connu et utilisé dans les démarches ascendantes. Dans la phase1, SelfStar propose au décideur un ensemble de faits candidats qu'il visualise dans le SI1 (Schéma Intermédiaire n°1). Ces faits correspondent à des classes d'objets extraites du schéma de la source. Le décideur choisira dans le SI1 le ou les faits à analyser parmi ces faits candidats ; les autres faits non choisis disparaîtront du SI1.

Une source avec un nombre important de classes pourrait générer une grande liste de faits candidats (Song et al. 2008). Pour faciliter le choix du décideur, SelfStar n'affiche que 10 faits candidats alors que le schéma de la source peut contenir plusieurs dizaines de classes. Ces 10 faits candidats correspondent aux classes de la source qui ont la probabilité la plus grande de contenir des données à analyser. Nous allons présenter les principes et les mécanismes utilisés par SelfStar pour ordonner l'ensemble des classes du schéma source selon leurs probabilités d'être utilisées comme fait.

6.1 La relation d'ordre

Dans (Song et al. 2008), chaque classe d'un schéma source Entité-Association est associée à un poids calculé en fonction du nombre de liens 1..N émanant de cette classe. Les liens 1..N émanant d'une classe représentent des dimensions (axes d'analyse) potentielles permettant d'analyser cette classe. Par conséquent, plus le nombre de dimensions potentielles associées à une classe est important, plus cette classe est susceptible d'intéresser le décideur pour devenir un fait à analyser.

Nous reprenons ce mécanisme de comptabilisation des liens d'une classe en l'appliquant à un diagramme de classes UML (DCL). Nous complétons ce mécanisme en comptabilisant également le nombre d'attributs numériques contenus dans la classe. En effet, seuls ces attributs peuvent être agrégés dans une analyse multicritère ; par conséquent, plus une classe possède d'attributs numériques, plus forte est sa probabilité d'être un fait.

Le poids de base d'une classe (notée c) de la source est calculé par une fonction récursive qui comptabilise les attributs et les relations de cette classe.

6.2 Calcul des poids de base

Dans un schéma multidimensionnel, on distingue : (1) le fait qui contient des mesures, (2) les dimensions et leurs hiérarchies liées au fait ; elles permettent de regrouper les données contenues dans le fait. Une classe d'une source peut devenir un fait ou une dimension en fonction de ses caractéristiques : nombre d'attributs et nombre de relations.

Les attributs d'une classe

Toute classe du schéma source est un fait potentiel dont les attributs numériques peuvent être des mesures. SelfStar comptabilise le nombre d'attributs numériques d'une classe, le multiplie par un coefficient de pondération et détermine ainsi le poids provisoire de la classe.

Un attribut non numérique d'une classe est une dimension potentielle si ses valeurs ne sont pas distinctes. Par exemple, l'attribut NoPasseport, contenant des numéros distincts, n'est pas une dimension potentielle. Ces attributs non numériques sont donc comptabilisés et leur nombre est multiplié par un coefficient particulier.

PoidsAtt(c) = (NbAn(c) * CoefAn) + (NbNAn(c) * CoefNAn) où :

PoidsAtt(c) est le poids des attributs de la classe c,

NbAn(c) est le nombre d'attributs numériques,

CoefAn est le coefficient des attributs numériques,

NbNAn(c) est le nombre d'attributs non numériques,

CoefNAn est le coefficient des attributs non numériques.

La fonction Poids va être réutilisée pour cumuler l'ensemble des poids associés à chaque classe.

Les relations d'une classe

Dans un schéma multidimensionnel, la relation entre une dimension et un fait est une relation 1..N ; c'est à dire qu'une valeur de la dimension est liée à une ou plusieurs valeurs du fait. Dans la source, une classe liée directement à c (classe-fait) par une relation 1..N signifie que cette classe est une dimension potentielle pour le fait c. Dans (Song et al. 2008), il a été montré que les relations indirectes transitives entre c et les autres classes de la source permettent de constituer les hiérarchies des dimensions.

SelfStar comptabilise le nombre de relations directes et indirectes d'une classe c.

PoidsRel(c) = PoidsAtt(c) + (NbLd(c) * CoefLd) + (NbNLd(c) * CoefNLD) où :

PoidsRel(c) est le poids des relations de la classe c,

NbLd(c) est le nombre de liens directs émanant de c,

CoefLd est le coefficient des liens directs,

NbNLd(c) est le nombre de liens indirects transitifs émanant de c,

CoefNLd est le coefficient des liens indirects.

Le poids d'une classe correspond d'une classe correspond au cumul des deux poids ainsi calculés.

Poids(c) = PoidsAtt(c) + PoidsRel(c).

6.3 La métabase

La métabase utilisée par SelfStar est alimentée par le module « T2 » de la FIG.2. Ce module génère des métadonnées uniquement lors de la 1ère utilisation d'une source. Ces métadonnées seront utilisées par SelfStar uniquement lorsqu'un décideur élabore son 1er schéma multidimensionnel sur une source de données.

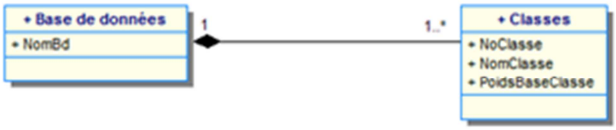

FIG.5 – Le schéma des métadonnées de base.

6.4 L'algorithme

La production des poids associés aux classes du schéma exploitable s'effectue selon l'algorithme suivant.

Notations

Le schéma exploitable, noté SE, est en entrée de l'algorithme; Il est composé d'un ensemble de n classes et de p liens de type 1..N entre ces classes : SE = ({C_1, C_2, ..., C_n}, {L_1, L_2, ..., L_p}).La sortie de l'algorithme est un ensemble de n poids correspondant aux n classe du SE : P = {P_1, P_2, ..., P_n}.La fonction **isnumeric**(A) retourne vrai si l'attribut A est numérique et la fonction **isdistinct**(A) retourne vrai si ses valeurs sont distinctes. La fonction **Liens**(Classe) retourne l'ensemble des liens directs d'une classe. La fonction récursive **PoidsLiens** est définie ci-après.

```
Algorithm PoidsDeBase
Input : SE                      -- Schéma exploitable correspondant au DCL source
Output : P                      -- poids des n classes du schéma exploitable
Constants CoefAn, CoefNAn, CoefLd, CoefNLd   -- coefficients de pondération
begin
for i←1 to n do                 -- pour chaque classe du SE
ANList ← nil                    -- liste des attributs numériques
NANList ← nil                   --liste des attributs non numérique
  for each Ak in Ci do
    if (isnumeric(Ak) ^ ¬isdistinct(Ak))  then
      ANList ← Ak
      else  if ¬distinct(Ak) then
              NANList ← Ak
            end if
    end if
  end for
  Pi ← ANList.lengh()* CoefAn           -- comptabilise les attributs numériques
  Pi ← Pi + NANList.lengh()* CoefNAn    -- comptabilise les attributs non numériques

  Pi ← Pi  + PoidsLiens (Ci,0)          -- calcule le poids d'une classe selon le nombre
                                        -- de classes liées
end
Function PoidsLiens(C, PoidsRel) return integer
-- fonction récursive de calcul du poids d'une classe C selon le nombre de liens directs et indirects
begin
  if Liens(C) <> set()  then
    if (PoidsRel == 0) then
      PoidsRel ← Liens.lengh() * CoefLd
      else
       PoidsRel ← Liens.lengh() * CoefNLd
    end if
    for each  x  in  Liens(C) do
      PoidsRel  ←  1  +   PoidsLiens(x,PoidsRel)
    end for
  end if
end
```

7 Les métadonnées de personnalisation

Dans le processus d'élaboration d'un schéma multidimensionnel à partir d'une source, SelfStar propose au décideur un ensemble de 10 faits candidats dans le SI1 (FIG.2). Comme nous venons de le voir, lorsque le décideur élabore son 1er schéma sur cette source, SelfStar utilise les métadonnées de base. Mais ensuite, pour chaque nouveau schéma élaboré, SelfStar utilisera des données personnalisées qui affinent le poids de chaque classe de la source.

7.1 Le principe

Dans la phase 4 de la FIG.2, le décideur a élaboré le schéma de la base décisionnelle qu'il souhaite analyser. Dans une étude industrielle (Annoni 2007), il a été montré qu'un décideur analyse fréquemment une source au travers d'un ensemble de bases dont les schémas sont très proches les uns des autres. Ainsi, dans tel schéma multidimensionnel, une classe est le fait analysé et dans tel autre, cette même classe est devenue une dimension ou un paramètre d'une hiérarchie. Ce constat nous a conduits à intégrer dans SelfStar des mécanismes de personnalisation.

Dans la phase 4, le schéma multidimensionnel élaboré par le décideur est analysé par SelfStar en termes de faits, dimensions et paramètres de hiérarchies. Les éléments issus de cette analyse sont comptabilisés, pondérés et cumulés aux poids de chaque classe. Mais, contrairement aux poids de base, ces poids sont personnalisés puisque associés à chaque décideur.

7.2 Calcul des poids

SelfStar analyse le schéma multidimensionnel et comptabilise les constituants suivants les mesures et les dimensiosn. Ces constituants correspondent à des classes de la source. Ils sont dénombrés par classe, affecté d'un coefficient de pondération et cumulés aux poids des classes correspondantes. La fonction récursive suivante est utilisée.

Poids(c) = Poids(c) + (NbMe(c) * CoefMe) + (NbDi(c) * CoefDi) où :

Poids(c) est le poids de la classe c dans la source,

NbMe(c) est le nombre de faits correspondants à c,

CoefMe est le coefficient des faits,

NbDi(c) est le nombre de dimensions correspondantes à c ou à ses attributs,

CoefDi est le coefficient des dimensions.

7.3 La métabase

Les poids de base (section 5) sont communs à l'ensemble des décideurs qui souhaitent analyser la même source. Par contre, les poids de personnalisation d'une classe sont associés aux différents décideurs (ceux qui ont élaboré des schémas multidimensionnels). Le schéma de la métabase de la FIG.3 doit être complété pour prendre en compte cette personnalisation des données.

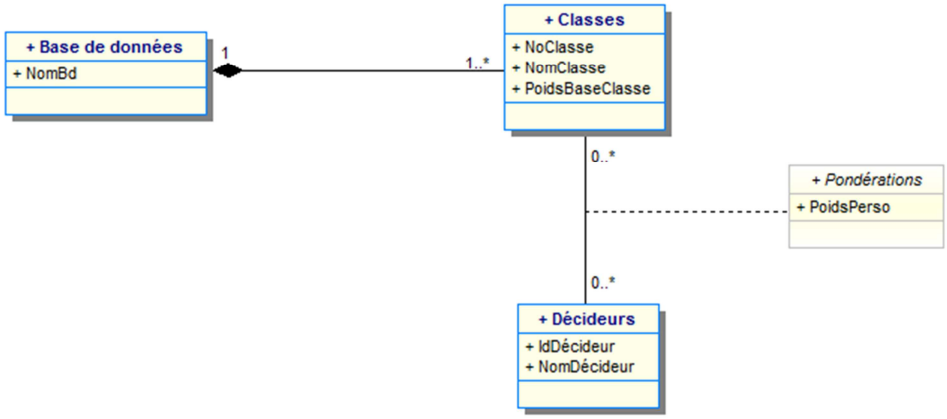

FIG.6 – Le schéma de la métabase des métadonnées de personnalisation.

7.4 Algorithme

Chaque fois qu'un décideur élabore un nouveau schéma multidimensionnel, l'algorithme suivant analyse ce schéma et génère des poids. Ces poids sont propres à chaque décideur (celui qui a conçu le schéma) et viennent majorer les poids associés aux classes de la source. Pour chaque fait, le poids de la classe-source correspondante est majoré par les coefficients CoefMe et CoefDi (respectivement coefficient de fait et de dimensions). Ces coefficients traduisent l'importance supérieure accordée à un fait par rapport à une dimension. Leurs valeurs peuvent être modifiées au cours du temps pour affiner les résultats.

Notations

Le schéma multidimensionnel, noté SM, se définit par un ensemble de p faits et un ensemble de q dimensions liées aux faits ; les hiérarchies ne sont pas prises en compte : SM = ({F_1, F_2, ..., F_p}, {D_1, D_2, ..., D_q}). Les poids personnalisés, notés PE, correspondent aux métadonnées mises à jour par l'algorithme ; il s'agit d'un ensemble de couples (P_i, U) avec Pi le poids de la classe-source i et U l'identificateur d'un décideur.

La fonction **Liens**(Classe) retourne l'ensemble des liens directs d'une classe. La fonction **source**(X) retourne l'identificateur de la classe-source dont le fait ou la dimension X est extrait.

```
Algorithm PoidsPerso
Input : SE, PE , SM, U        -- schéma exploitable, poids des classes, schéma multidimen-
sionnel et utilisateur
Output : PE                   -- nouvelles valeurs des poids
begin
for i←1 to p do               -- pour chaque fait du SM
  s ← source(Fi )
  Ps ←  Ps + Coef             -- majore le poids de la classe-source
  for  x  in  Dim(Fi) do
     s  ← source (x)
     Ps ←  Ps + 2             -- majore le poids de la classe-source
  end for
end for
```

Cet algorithme produit la variable PE. Celle-ci contient un ensemble de couples (poids, utilisateur) pour chacune des classes associées aux faits et dimensions de SM.

8 Mise en œuvre

Considérons le DCL de la FIG.3 et le schéma exploitable de la FIG.4, grâce à SelfStar, le décideur a élaboré le schéma en étoile suivant.

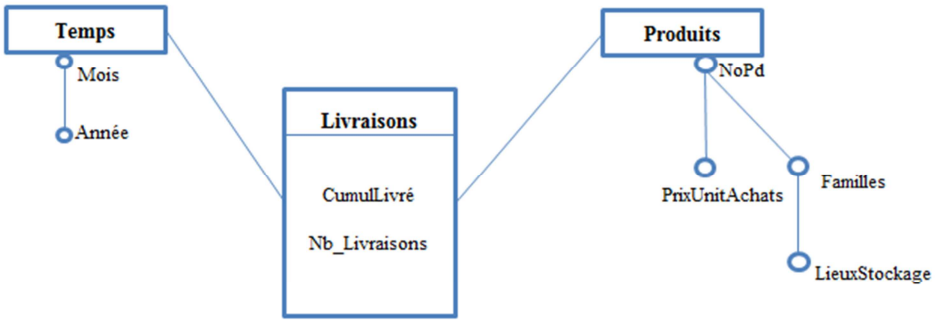

FIG.7 – Schéma multidimensionnel d'une base décisionnelle.

Nous présentons, à la suite, comment SelfStar élabore successivement les métadonnées de base et celles de personnalisation.

8.1 Calcul des métadonnées de base

Chaque fois qu'un analyse Il s'agit de calculer le poids de chaque classe appartenant au schéma exploitable. Un poids détermine la capacité de la classe à devenir un fait : plus le poids est élevé et plus la classe peut intéresser le décideur. Ces poids permettent donc d'ordonner l'ensemble des classes du DCL selon leur intérêt pour le décideur.

Nous fixons les coefficients en privilégiant les attributs numériques et les liens directs (ici du simple au double). Des ajustements de ces coefficients peuvent être réalisés après expérimentation afin d'obtenir un ordre des classes plus adapté à un décideur.

- CoefAn = 2,
- CoefNAn =1,
- CoefLd = 2,
- CoefNLd = 1.

Nous appliquons l'algorithme du § 6.4 pour calculer les métadonnées de base ; nous nous limitons ici aux classes Livraisons, Produits et Dates du schéma exploitable de la FIG.4.

- Classe Livraisons :

  Prise en compte des attributs :

  NbAn(Livraisons)****CoefAn***+NbNAn(Livraisons)****CoefNAn*** =1****2***+0****1***=2 + 0 = 2

  Prise en compte des relations :

  (NbLd(Livraisons)****CoefLd***)+(NbNLd(Livraisons)****CoefNLd***)= 4****2*** +2****1***=8+2= 10.

Le poids de la classe livraison est ainsi obtenu :

Poids (Livraisons) = 2 + 10 = 12

De la même façon, on obtient les poids des classes Produits et Dates.

- Classe Produits :

  Prise en compte des attributs :

  2 * ***2*** + 1****1*** = 5

  Prise en compte des relations :

  1 * ***2*** + 1 ****1*** = 3

Le poids de la classe livraison est ainsi obtenu :

Poids (Produits) = 5 + 3 = 8

- Classe Dates :

  Prise en compte des attributs :

  0 * ***2*** + 1****1*** = 1

  Prise en compte des relations :

  0 * ***2*** + 0 ****1*** = 0

Le poids de la classe livraison est ainsi obtenu :

Poids (Produits) = 1 + 0 = 1

8.2 Calcul des métadonnées de personnalisation

Ces données sont calculées après analyse d'un schéma multidimensionnel et sont affectées au décideur qui a élaboré le schéma. Pour le 1er schéma élaboré par un décideur, ces données sont cumulées aux métadonnées de base ; pour les schémas suivants, elles s'ajoutent aux métadonnées de personnalisation établies précédemment pour ce décideur.

Nous supposons l'utilisation des constantes de la manière suivante :

- CoefMe = 4,
- CoefDi =2,

Poids(Livraisons) =Poids(Livraisons)+(NbMe(Livraisons)*4)+(NbDi(Livraisons)*2)

= 12 + 1 * 4 + 0 * 2

= 16

Poids(Produits) =Poids(Produits)+(NbMe(Produits)*4)+(NbDi(Produits)*2)

= 8 + 0 * 4 + 1 * 2

= 10

Poids(Dates) = Poids(Dates) + (NbMe(Dates) * 4) + (NbDi(Dates) * 2)

= 1 + 0 * 4 + 1 * 2

= 3

9 Le prototype

La production de métadonnées de SelfStar a été implantée en Java dans un environnement Eclipse.

9.1 Calcul des métadonnées de base

Dès qu'une source de données est répertoriée dans SelfStar, celui-ci :

- traduit son schéma (DCL) en XML,

- élabore le schéma exploitable à partir du code XML,

- calcule les métadonnées en déterminant le poids de chaque classe grâce à l'algorithme présenté dans le §6.4.

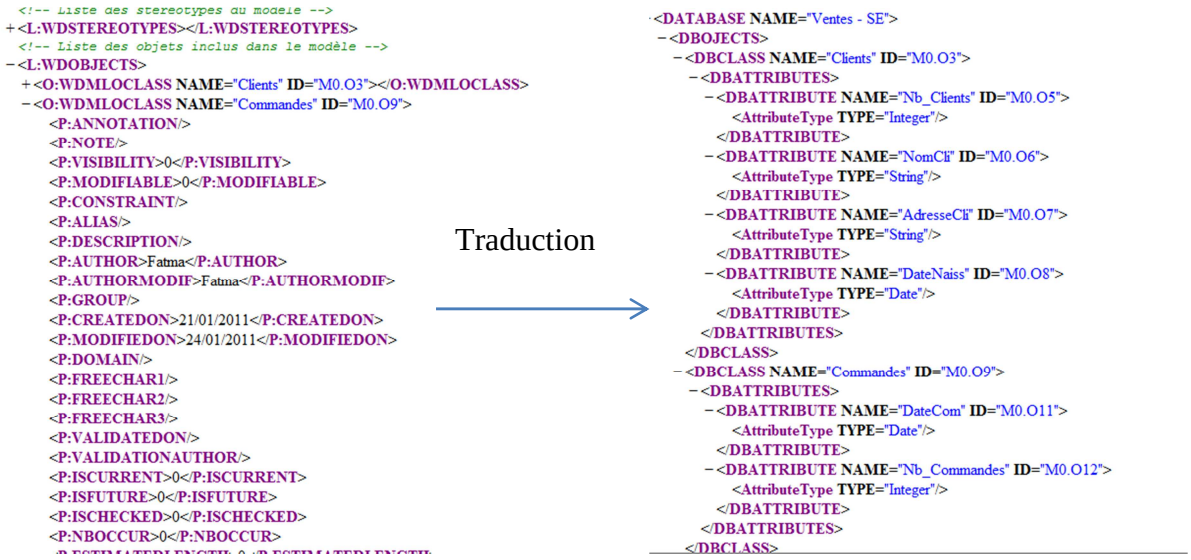

FIG.8 – Un extrait de traduction d'un DCL en un schéma exploitable.

Le calcul des poids s'effectue par parcours du graphe correspondant au schéma exploitable codé en XML. Ce parcours est effectué par l'intermédiaire de l'API DOM. Les métadonnées générées sont introduites dans un fichier XML.

9.2 Les métadonnées de personnalisation

Lors de la phase 4 (dernière étape de l'élaboration du schéma multidimensionnel par un décideur), SelfStar extrait les métadonnées de personnalisation liées à ce décideur.

Le schéma multidimensionnel sous format XML est analysé selon l'algorithme du §7.4 et les poids des classes impliquées sont augmentés. L'algorithme est, ici aussi, mis en œuvre en Java grâce à l'utilisation de l'API DOM.

10 Conclusion et perspectives

Cet article présente les mécanismes de production des métadonnées dans le système SelfStar.

Le projet SelfStar offre une assistance aux décideurs pour leur permettre d'élaborer eux-mêmes leurs schémas multidimensionnels. À partir du schéma d'une source de données à analyser, le décideur intègre progressivement ses besoins d'analyse en termes de faits, dimensions et hiérarchies.

Cette démarche est originale dans la mesure où elle permet à un décideur de construire progressivement son schéma multidimensionnel, ceci sans avoir recours à un spécialiste des

bases de données : administrateur ou informaticien. Elle se distingue nettement des autres démarches mixtes, ascendantes ou descendantes dans lesquelles l'utilisateur (le décideur) n'intervient pas directement.

La connaissance de la structure des sources par le décideur est limitée grâce à l'usage de métadonnées de personnalisation ; ces métadonnées sont enregistrées par SelfStar lors de chaque processus d'élaboration d'un nouveau schéma. Grâce à ce mécanisme, seuls les faits les plus significatifs sont extraits de la source et proposés au décideur (ceci sans réduire ses possibilités de choix).

Le prolongement de ces travaux se situe dans le processus de génération automatique d'une base décisionnelle à partir d'une source. En effet, la démarche proposée dans SelfStar permet d'élaborer un schéma multidimensionnel. Mais l'alimentation (ETL) de la base décisionnelle utilise d'autres métadonnées assurant la correspondance entre schéma multidimensionnel et schéma de la source.

Références

Abdelhédi, F., G. Pujolle, O. Teste and G. Zurfluh (2011). COMPUTER-AIDED DATA-MART DESIGN. Beijing, Chine (à apparaître).

Annoni, E. (2007). *Eléments méthodologiques pour le développement des systèmes décisionnels dans un contexte de réutilisation*. Thèse de doctorat, Université de Toulouse.

Annoni, E., O. Teste, F. Ravat, G. Zurfluh (2006). Towards Multidimensional Requirement Design. *Data Warehousing and Knowledge Discovery*, 75–84.

Feki, J. & Y. Hachaichi (2007). Conception assistée de MD: Une démarche et un outil. *Journal of decision systems*, 16(3), 303–333.

Giorgini, P., S. Rizzi, and M. Garzetti (2005). Goal-oriented requirement analysis for data warehouse design. Dans *Proceedings of the 8th ACM international workshop on Data warehousing and OLAP*. Bremen, Germany: ACM, p. 47-56.

Golfarelli, M., Maio, D. and S. Rizzi (1998). The Dimensional Fact Model: a conceptual model for data warehouses. Int. *Journal of Cooperative Information Systems*, 7(2&3), 215–247.

Inmon, W.H (1996). *Building The Data Warehouse 2 nd edition*,

Kimball, R (1996). *The data warehouse toolkit: practical techniques for building dimensional data warehouses*, Wiley New York.

Malinowski, E. and E. Zimányi (2008). Designing Conventional Data Warehouses. *Advanced data warehouse design*, 251 -- 313.

Moody, D.L. and M.A. Kortink (2000). From enterprise models to dimensional models: a

methodology for data warehouse and data mart design. *DMDW'00, Sweden*, 5.

Phipps, C. and K. Davis, (2002). Automating data warehouse conceptual schema design and evaluation. *Proc. 4th DMDW, Toronto, Canada*.

Pinet, F. and M. Schneider (2009). A Unified Object Constraint Model for Designing and Implementing Multidimensional Systems. Dans *Journal on Data Semantics XIII*. p. 37-71.

Prat, N., J. Akoka. and I. Comyn-Wattiau (2006). A UML-based data warehouse design method. *Decision Support Systems*, 42(3), 1449-1473.

Romero, O. and A. Abelló (2010). Automatic validation of requirements to support multidimensional design. *Data & Knowledge Engineering*, 69(9), 917-942.

Song, I.Y., R. Khare, Y. An, and S. Lee, S.P Kim, J. Kim, et Y.S. Moon (2008). SAMSTAR: An Automatic Tool for Generating Star Schemas from an Entity-Relationship Diagram. Dans *Conceptual Modeling - ER 2008*. p. 522-523.

Trujillo, J., S. Lujan-Mora, et I.Y. Song (2003). Applying UML for designing multidimensional databases and OLAP applications. *Advanced Topics in Database Research*, 2, 13–36.

Vassiliadis, P. (2009). A Survey of Extract–Transform–Load Technology. *International Journal of Data Warehousing and Mining*, 5(3).

Summary

To design a decisional database, decision makers require the help of a specialist: a multidimensional database designer, a knowledge engineer or a computer specialist. The specialist designs a multidimensional schema using facts and analysis axes according to a demand-driven, data-driven or hybrid approaches. This process, based on data sources to be analyzed or on decision-makers requirements, often turns out to be approximate and complex. The SelfStar project aims at defining mechanisms and tools to design a multidimensional schema by the decision-maker himself. To guide and assist the decision maker, SelfStar loads then uses personalization metadata. During the process of elaborating a decisional database, these metadata will help decision makers by guiding their choices. This paper presents principles and algorithms that generate the metadata associated to each decision maker and implemented within a prototype.

Métadonnées de personnalisation dans le système SelfStar