Personnalisation collaborative pour l'enrichissement des analyses dans les entrepôts de données complexes

Cécile Favre, Fadila Bentayeb, Omar Boussaid

Université de Lyon (ERIC Lyon 2)
5 av. Pierre Mendès-France
69676 Bron Cedex
{cfavre|bentayeb}@eric.univ-lyon2.fr, omar.boussaid@univ-lyon2.fr
http://eric.univ-lyon2.fr

Résumé. Les entrepôts de données XML constituent une bonne alternative pour la représentation, le stockage et l'analyse des données complexes. Le modèle d'un entrepôt de données est classiquement conçu à partir des sources de données disponibles et des besoins d'analyse identifiés au moment de la conception. Or, il s'avère que des besoins d'analyse peuvent émerger, dépendant souvent des connaissances des analystes. Ces connaissances concernent en particulier la manière d'agréger les données. Ainsi, pour répondre aux besoins individuels et tirer profit des connaissances des différents analystes utilisant l'entrepôt de données, nous proposons dans cet article une approche de personnalisation collaborative pour l'enrichissement des possibilités d'analyse de l'entrepôt XML. Cette approche se base sur l'expression des connaissances des analystes sur de nouvelles façons d'agréger les données, permettant à la fois de répondre aux besoins d'analyse individuels émergents et de partager les nouvelles possibilités d'analyses à travers l'enrichissement des hiérarchies de dimension qui guident la navigation dans les données de l'entrepôt XML.

1 Introduction

Les entrepôts de données ont pour vocation de permettre l'analyse de données pouvant provenir de différentes sources de données (Kimball, 1996; Inmon, 1996). Cette analyse consiste généralement en une analyse exploratoire grâce à la technologie OLAP (On-Line Analytical Processing). Pour permettre cette analyse, les données sont organisées de façon multidimensionnelle : des faits sont analysés à travers des indicateurs appelés mesures, en fonction de différents axes d'analyse appelés dimensions. Les dimensions peuvent être organisées sous forme de hiérarchies. Ces hiérarchies de dimension permettent d'obtenir différentes vues sur les données avec plusieurs niveaux de granularité, en l'occurrence des données plus ou moins résumées, grâce aux opérateurs OLAP roll-up (forage vers le haut) et drill-down (forage vers le bas).

Depuis quelques années, face au besoin d'analyser des données pouvant être qualifiées de complexes (données du Web, données multimédia, etc), on a vu émerger des entrepôts de données basés sur le langage XML (eXtensible Markup Language) capables de centraliser et d'analyser des données complexes. En effet, XML est approprié pour la structuration de données complexes provenant de différentes sources et soutenues par des formats hétérogènes (Boussaid et al., 2008). XML est un langage qui présente à la fois les données et leur structure (schéma). Leur analyse est rendue possible grâce à l'extension de langages d'interrogation tel que XQuery (Beyer et al., 2005).

XML va aussi permettre de représenter les différentes formes de hiérarchies du modèle de l'entrepôt plus ou moins complexes. Bien que souvent on se limite au cas classique des hiérarchies qualifiées de «strictes», Malinowski et Zimányi (2004) ont proposé une représentation conceptuelle de ces hiérarchies. Notons que nous ne traitons pas ici de la façon d'exploiter ces différentes formes de hiérarchies qui constitue un réel problème.

Que ce soit dans le cadre des entrepôts que l'on qualifiera de classiques, ou de ceux que l'on qualifiera de «complexes», les possibilités d'analyse dépendent finalement du modèle de l'entrepôt de données conçu initialement. Ce modèle est généralement déterminé en fonction des sources de données disponibles d'une part, des besoins d'analyse recensés au moment de la conception du modèle d'autre part. Néanmoins, des besoins d'analyse individuels peuvent émerger, dépendant souvent des propres connaissances des utilisateurs. La caractérisation de ces nouveaux besoins d'analyse au sein de l'entrepôt répond à une certaine personnalisation de l'entrepôt de données voulue par les utilisateurs. L'entrepôt de données doit donc pouvoir s'adapter en prenant en compte des nouveaux besoins utilisateurs. En effet, Y. Ioannidis et G. Koutrika définissent la personnalisation comme «*...providing an overall customized, individualized user experience by taking into account the needs, preferences and characteristics of a user or group of users*» (Ioannidis et Koutrika, 2005).

Au sein d'une organisation, on peut envisager l'utilisateur de façon individuelle. Mais on peut également considérer qu'il appartient à une communauté, la communauté des utilisateurs exploitant l'entrepôt de données de l'organisation en l'occurrence. En effet, au sein d'une organisation, différents acteurs sont amenés à prendre des décisions : à différents niveaux de responsabilité ou sur des «thématiques» différentes (des services différents dans l'entreprise). Cette communauté d'utilisateurs a donc besoin de réaliser des analyses à partir de l'entrepôt de données pour supporter la prise de décision. Ainsi, dans le contexte de cette organisation, et donc de cette communauté d'utilisateurs de l'entrepôt de données, la notion de collaboration émerge.

Il est alors intéressant de combiner les concepts de personnalisation et de collaboration. Ceci a déjà été fait dans le cadre de systèmes de personnalisation proposant des recommandations basées sur le filtrage collaboratif. Dans ce cas, il s'agit de chercher des utilisateurs qui ont les mêmes comportements, préférences, etc. avec l'utilisateur à qui l'on souhaite faire des recommandations. Ensuite, il est possible d'utiliser les informations de ces autres utilisateurs similaires pour calculer une liste de recommandations pour cet utilisateur. Ceci est valable entre autres dans les systèmes de recherche d'informations (Goldberg et al., 1992). Dans ce cas, l'aspect collaboratif est un moyen pour parvenir à une personnalisation basée sur une idée de limitation ; en effet, l'aspect collaboratif permet de s'intéresser aux informations essentielles, pertinentes pour un utilisateur.

Nous pensons qu'il est possible de combiner les concepts de personnalisation et de collaboration d'une manière différente. Plutôt que d'exploiter l'aspect collaboratif pour permettre une personnalisation, nous voulons mettre l'aspect personnalisation au service de l'aspect collaboratif. L'objectif réside alors dans le fait que l'utilisateur puisse répondre à ses propres besoins d'analyse incluant un processus de personnalisation permettant à l'utilisateur, dans ce contexte, d'exprimer ses propres connaissances. Dans ce cas, le concept de personnalisation peut être considéré comme étant étendu puisque la personnalisation n'est pas basée ici sur une opération de restriction, mais plutôt sur une opération d'extension. L'intérêt est alors de pouvoir exploiter les connaissances de cet utilisateur donné, pour que les autres utilisateurs appartenant à la même communauté (au sein d'une organisation donnée par exemple) puissent en tirer profit, dans l'esprit d'un système collaboratif dans lequel chacun apporte sa pierre à l'édifice. Ainsi, à partir d'un entrepôt initial qui constitue une base de travail, assurant l'intégrité des données et de leur chargement par rapport à leur sources, l'aspect collaboratif va se porter sur le développement, l'enrichissement incrémental de nouveaux axes d'analyse à travers la création de nouveaux niveaux de granularité définissant ou enrichissant des hiérarchies de dimension dans l'entrepôt de données complexes.

La suite de cet article est organisée de la façon suivante. Dans la section 2, nous présentons brièvement un état de l'art relatif aux différents aspects évoqués dans notre proposition, à savoir les entrepôts de données complexes, les aspects collaboratifs dans les entrepôts de données et la personnalisation dans ces derniers. Puis nous développons dans la section 3 notre proposition de système de personnalisation collaborative pour l'enrichissement des analyses dans les entrepôts de données XML. Dans la section 4, nous évoquons la mise en œuvre de notre approche avec d'une part les éléments concernant l'implémentation qui est en cours de développement et d'autre part la présentation d'une étude de cas, issu d'un projet mené dans le cadre d'une Action Concertée Incitative avec des collègues linguistes, afin d'illustrer nos propos. Enfin, nous concluons cet article et évoquons les perspectives de ce travail préliminaire dans la section 5.

2 État de l'art

Cet article aborde différents domaines. Il se situe dans le cadre des entrepôts de données complexes et propose dans ce contexte, une solution de personnalisation collaborative. C'est pourquoi, nous abordons brièvement ces trois volets dans l'état de l'art avant de positionner notre travail.

2.1 Entrepôt de données complexes

À ce jour, les travaux s'intéressant à l'entreposage de données complexes portent essentiellement sur l'exploitation du langage XML pour la structuration et le stockage des données complexes (Boussaid et al., 2008). L'entrepôt de données est finalement constitué d'une collection de documents XML représentant les faits et les dimensions. Différentes approches ont été proposées. Elles peuvent être vues comme des variantes au niveau de l'organisation des données. Pokorný (2002) a proposé un schéma en étoile XML définissant les hiérarchies de dimension comme un ensemble de collections de données XML connectées logiquement, et les faits comme des éléments XML. Golfarelli et al. (2001) proposent de stocker chaque fait dans un document XML comprenant alors les instances et les hiérarchies de dimension. Quant à eux, Hümmer et al. (2003) proposent le modèle nommé XCube qui prévoit de regrouper tous les faits dans un document, et l'ensemble des dimensions dans un autre. Park et al. proposent une plateforme nommée XML-OLAP basée sur un entrepôt de données XML où chaque fait est contenu dans un document XML et où chaque instance de la hiérarchie de dimension est elle-même stockée dans un document XML pour éviter les jointures entre les différents niveaux de la hiérarchie de dimension.

Finalement, les différents travaux diffèrent sur comment sont représentés les faits et les dimensions dans ces documents, et donc, sur le nombre de documents nécessaires au stockage des données.

Une étude de performances des différentes représentations a été conduite par Boukraa et al. (2006). Elle montre que dans le cas d'un schéma en flocon de neige (cas des dimensions hiérarchisées), les meilleures performances sont obtenues lorsque les faits sont représentés dans un seul document XML et que chacune des dimensions est contenue dans un document XML. Ce mode de représentation présente en outre l'avantage d'éviter la duplication des données sur les dimensions dans le cas d'une construction de schéma en constellation dont le principe est de présenter plusieurs faits qui partagent des dimensions. En outre, comme chaque dimension et ses hiérarchies sont représentées dans un document XML, les mises à jour des dimensions sont rendues plus facile que si les dimensions étaient regroupées avec les faits ou stockées dans un seul document. C'est ce point qui nous intéresse plus particulièrement.

2.2 Systèmes collaboratifs dans les entrepôts de données

Initialement, les utilisateurs du Web se contentaient de consulter des données mises à leur disposition sur des sites développés par des spécialistes. Par la suite, ils ont pu peu à peu accéder à des technologies pour contribuer eux-mêmes au Web (participation à des forums, création de blogs, contributions dans des sites de type wiki, etc.). Ainsi, l'évolution du Web tend aujourd'hui vers un «Web 2.0», qualifié de «social», «participatif», «collaboratif», etc.

Cet aspect collaboratif est très présent au niveau du Web de façon générale, mais a été assez peu étudié dans un contexte comme celui des entrepôts de données, alors même que c'est un domaine dans lequel cet aspect peut être très intéressant. L'aspect collaboratif doit bien sûr être introduit au niveau où l'interaction entre le système et l'utilisateur est possible. Ainsi, dans le contexte des entrepôts de données, l'analyse est une phase privilégiée. On peut citer les travaux de Cabanac et al. (2007) qui se sont intéressés à la pratique d'annotations collectives dans le contexte des bases de données décisionnelles. Cette pratique permet aux analystes de partager leurs avis sur des analyses : ils réalisent ces analyses, peuvent les commenter et aussi les partager. Citons également les travaux de Aouiche et al. (2008) proposant une visualisation des analyses basée sur les nuages de mots et un partage, une mise à disposition facilitée de ces résultats pour d'autres utilisateurs, les auteurs qualifiant alors leur approche d'OLAP collaboratif.

2.3 Personnalisation dans les entrepôts de données

La personnalisation est une thématique abordée depuis déjà assez longtemps dans les domaines de la recherche d'information et des bases de données. Dans le contexte des entrepôts de données, il s'agit d'une thématique émergente. S'inspirant des travaux des domaines de la recherche d'information ou des bases de données, les travaux prennent de plus en plus en compte les spécificités des entrepôts de données. Quels que soient ces domaines, la personnalisation consiste habituellement à exploiter les préférences des utilisateurs pour leur fournir des réponses pertinentes.

Nous pouvons citer les travaux de Bellatreche et al. (2005) qui se sont inspirés des techniques de filtrage d'information en fonction du profil utilisateur pour affiner des requêtes en y ajoutant des prédicats. L'objectif de ces travaux est de pouvoir fournir à l'utilisateur un résultat focalisé sur son centre d'intérêt, tout en prenant en compte des contraintes de visualisation.

Ravat et Teste (2008) proposent une solution pour la personnalisation de la navigation OLAP en exploitant des préférences exprimées par des poids. Dans ce cas, l'utilisateur assigne des poids aux concepts multidimensionnels afin d'obtenir directement les analyses désirées, évitant ainsi des opérations de navigation.

Giacometti et al. (2008) proposent, quant à eux, un système de recommandation d'analyses multidimensionnelles en se basant sur la navigation qu'effectue un utilisateur donné par rapport aux navigations réalisées par les autres utilisateurs.

2.4 Positionnement

Notre approche de personnalisation collaborative consiste en la possibilité d'un enrichissement des hiérarchies de dimension via une mise à jour de celles-ci. Vis-à-vis des entrepôts de données XML, dans le cadre de notre approche, l'aspect mise à jour des dimensions est donc crucial. Dans ce cas, compte-tenu des avantages à modéliser les faits dans un seul document XML et chacune des dimensions dans un document XML, nous avons choisi de baser notre approche sur un modèle présentant ces caractéristiques, en l'occurrence sur celui de Mahboubi et al. (2009) que nous détaillerons par la suite.

À travers les différents travaux faits en matière de personnalisation, nous notons un manque afin d'apporter une réponse aux besoins d'analyses individuels. Nous avons apporté une solution à ce problème en proposant une évolution de l'entrepôt de données basée sur l'intégration des connaissances des utilisateurs sur la manière d'agréger les données sous forme de règles pour créer de nouveaux axes d'analyse (Bentayeb et al., 2008). Néanmoins ce travail a été réalisé dans un contexte d'entrepôts de données «classiques».

Notre travail vise alors à étendre ce travail au cas des entrepôts de données complexes, en se focalisant également sur l'aspect collaboratif qui nous paraît tout à fait intéressant dans ce contexte, compte-tenu de la difficulté de concevoir un schéma d'entrepôt de données répondant correctement aux besoins d'analyse de leurs usagers. En effet, bien que le nombre d'usagers d'un entrepôt de données soit réduit par rapport à celui concernant une base de données par exemple, il n'en demeure pas moins qu'il peut être élevé au sein d'une organisation, en particulier dans le cas où cette organisation est structurée hiérarchiquement avec bon nombre de responsables.

Notons que vis-à-vis de cet aspect d'évolution de l'entrepôt de données, nous pouvons distinguer dans la littérature deux types d'approches : la mise à jour du modèle d'une part et la modélisation temporelle d'autre part. La première approche consiste à transformer le schéma de l'entrepôt de données (Hurtado et al., 1999; Blaschka et al., 1999). Ces travaux consistent principalement à proposer des opérateurs adaptés permettant de faire évoluer le schéma de l'entrepôt de données. Dans ce cas, un seul schéma est supporté et l'historique de l'évolution n'est pas préservé. Dans la seconde approche, l'historique des modifications est conservées en exploitant des labels de validité temporelle. Ces labels peuvent être apposés au niveau des instances des dimensions (Bliujute et al., 1998), des liens d'agrégation (Mendelzon et Vaisman, 2000), ou des versions de schéma (Bebel et al., 2004; Body et al., 2003; Morzy et Wrembel, 2004; Ravat et al., 2006). Dans ces entrepôts, chaque version décrit le schéma et les données à une certaine période. Afin de pouvoir analyser ces données, compte-tenu du modèle spécifique, une extension du language SQL est requise. L'inconvénient de ces approches résident également dans le fait qu'elle doivent être mises en œuvre dès la conception de l'entrepôt de données. Ainsi, pour permettre un point de vue collaboratif, nous adoptons une approche de mise à jour de schéma, avec une partie de l'entrepôt qui servira de base que les utilisateurs vont enrichir. La mise à jour permet alors de pouvoir implémenter le processus collaboratif, l'objectif étant un enrichissement incrémental de l'entrepôt, cela ne remet pas en cause les données de l'entrepôt, même si l'historique des modifications n'est pas conservé.

3 Personnalisation collaborative dans les entrepôts de données XML

3.1 Modèle d'entrepôt de données XML

Comme nous l'avons précisé précédemment, plusieurs modèles d'entrepôt de données XML ont été proposés dans la littérature. Nous basons nos travaux sur le modèle proposé par Mahboubi et al. (2009) qui rend plus facile et plus efficace la mise à jour des dimensions. Ce modèle propose de rassembler les faits dans un document XML, et chacune des dimensions avec ses hiérarchies sont contenues dans un document XML. Un document XML nommé $dw - model.xml$ représente le schéma de l'entrepôt (figure 1). Ensuite, les documents portant le nom $facts_f$.xml contiennent les données sur les faits (figure 2-a), c'est-à-dire les identifiants des dimensions et les mesures. Enfin, les documents $dimension_d.xml$ permettent de stocker les valeurs des attributs décrivant les dimensions et leurs hiérarchies (figure 2-b).

Les éléments auxquels nous nous intéresserons particulièrement dans le cadre de notre personnalisation collaborative sont les niveaux hiérarchiques dans les dimensions. Ainsi, notre approche aura un impact à la fois sur le document contenant la structure de l'entrepôt ($dw - model.xml$), mais également sur les documents contenant les dimensions dont les hiérarchies seront modifiées.

3.2 Processus de personnalisation collaborative proposé

À travers cet article, nous voulons poser les bases de notre proposition de personnalisation collaborative. Il s'agit d'exploiter un entrepôt de données initial. Ensuite, une couche collaborative a pour but d'enrichir incrémentalement cet entrepôt initial, au fur et à mesure que la personnalisation répond à de nouveaux besoins individuels qui seront partagés. L'originalité finalement est qu'en voulant répondre à un besoin individuel, par l'expression de ses propres connaissances, l'utilisateur va en même temps collaborer à l'enrichissement des possibilités d'analyse de l'entrepôt de données pour les autres utilisateurs de l'organisation.

Pour permettre une personnalisation collaborative des analyses dans les entrepôts de données XML, nous proposons un processus au sein duquel les utilisateurs ont bien évidemment une place centrale (figure 3). Chaque utilisateur de la communauté dans l'organisation peut avoir des besoins spécifiques en termes d'analyse, nécessitant l'ajout ou l'enrichissement des hiérarchies de dimension de l'entrepôt. Ainsi, chaque utilisateur peut exprimer ses propres connaissances pour créer un nouveau niveau de granularité. Un module permet l'acquisition des connaissances sous forme de règles de type si-alors, correspondant à une phase participative. Un module d'évolution de l'entrepôt permet ensuite de prendre en compte ces règles pour faire évoluer l'entrepôt de données, en l'occurrence les documents XML adéquats. Enfin, un module d'analyse permet à l'utilisateur qui a exprimé ses connaissances, de faire l'analyse correspondant à ses propres besoins, mais ce module permet également l'accès à ces mêmes analyses pour les autres utilisateurs de la communauté, dans un esprit collaboratif où chacun enrichit l'entrepôt pour lui et pour les autres.

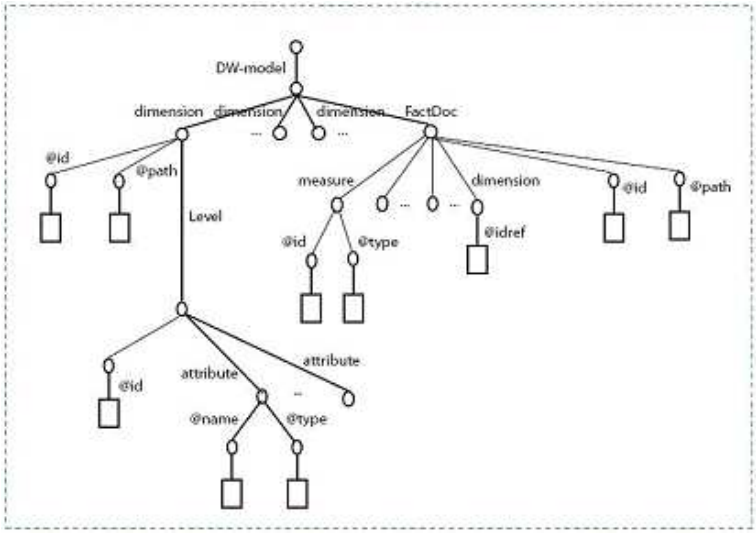

FIG. 1 – *Structure du graphe* $dw - model.xml$.

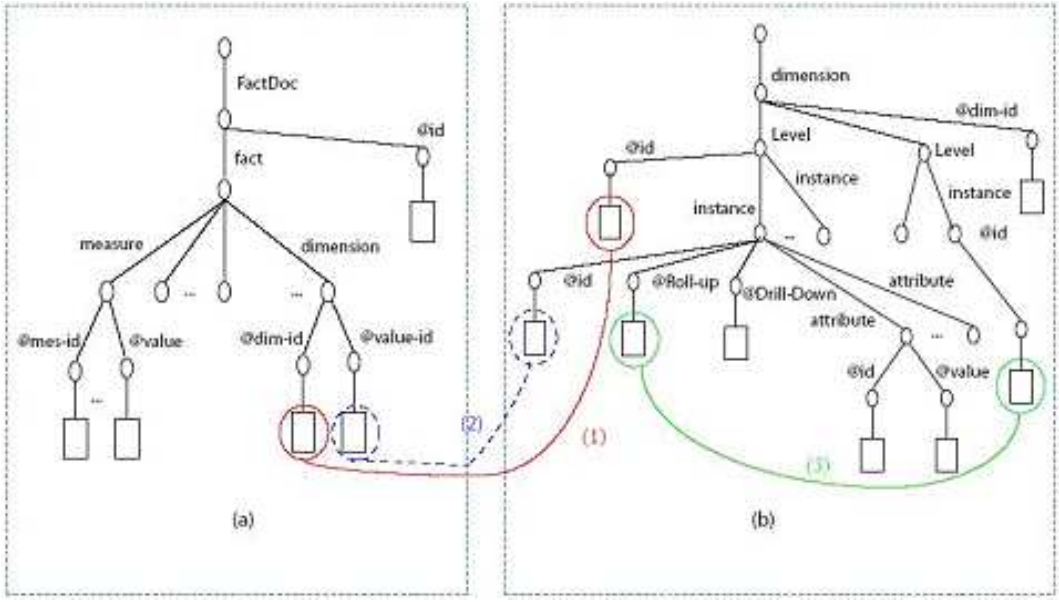

FIG. 2 – *Structure des graphes* $facts_f.xml$ *(a) et* $dimension_d.xml$ *(b).*

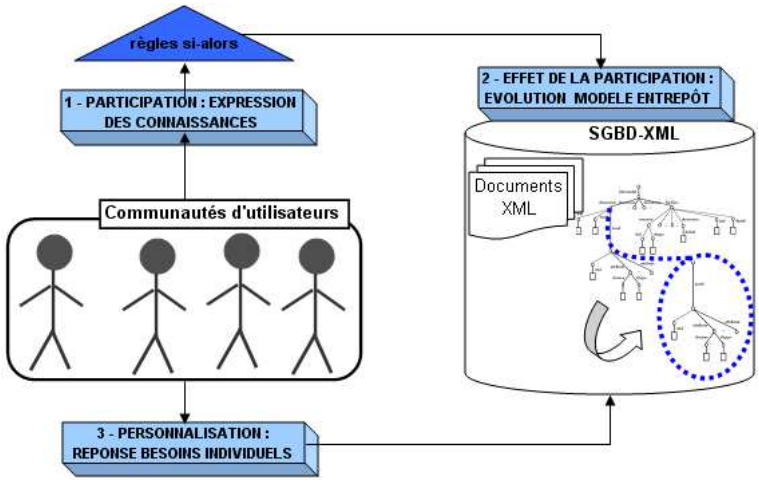

FIG. 3 – *Processus de personnalisation collaborative dans les entrepôts de données XML.*

Reprécisons que les utilisateurs disposent de l'entrepôt de données initial. Celui-ci a été conçu à partir des sources de données et d'un ensemble de besoins d'analyse globaux, correspondant à des besoins communs pour l'ensemble des utilisateurs, recensés au moment de la conception. Cet entrepôt initial permet de garantir l'intégrité des données « de base» (assurant le maintien de la cohérence de la phase de chargement des données dans l'entrepôt). Il appartient ensuite à chaque utilisateur d'ajouter les niveaux d'analyse dont il a besoin et d'en faire profiter les autres usagers.

3.3 Participation des analystes

Dans ce contexte d'entrepôt de données, la participation des analystes pour une personnalisation collaborative s'effectue dans la phase d'acquisition des connaissances qui traduit finalement l'expression d'un besoin individuel d'analyse, nécessitant un processus pouvant être qualifié de personnalisation.

La connaissance que nous considérons ici concerne la façon d'agréger les données. Ainsi, nous souhaitons représenter ces connaissances de façon simple pour les analystes. Nous avons alors choisi de représenter ces connaissances sous forme de règles dites d'agrégation qui sont des règles de type si-alors («if-then»).

Rappelons que le langage XML renferme à la fois la structure et les données elles-mêmes. Il s'agit pour les analystes d'exprimer leurs connaissances pour ajouter de nouveaux niveaux de granularité. Cet ajout a un effet à la fois sur la structure et les données elles-mêmes.

Ainsi, pour permettre l'acquisition des connaissances, nous proposons de la découper en deux étapes : expression des connaissances structurelles sur la création du nouveau niveau, expression des connaissances sur les données de ce nouveau niveau.

De ce fait, les règles d'agrégation sont de deux types : règle structure et règle données. Pour créer un niveau de granularité, il faut une règle structure et un ensemble de règles données. En effet, la règle structure permet de définir la structure des liens d'agrégation : quel niveau est créé, quels attributs le caractérisent, à quel(s) niveau(x) est-il relié, à partir de quels attributs le lien d'agrégation est défini, etc. Les règles données, quant à elles, instancient la règle structure, c'est-à-dire qu'elles définissent les liens d'agrégation au niveau des données. Cette formalisation s'inspire de celle que nous avons proposée dans (Bentayeb et al., 2008).

Soit EL le niveau de hiérarchique au dessus duquel sera construit le niveau à créer. Soit $\{EA_i, i = 1..m\}$ l'ensemble des m attributs parmi les m' attributs de EL, sur lesquels seront basées les conditions définissant les groupes d'instances. Soient GL le nouveau niveau et $GA_j, j = 1..n$ l'ensemble des n attributs du nouveau niveau GL. La règle structure notée SR est définie comme suit :

$$SR: \ if\ ConditionOn(EL,\ \{EA_i,\ i = 1..m\})\ then\ Generate(GL,\ \{GA_j,\ j = 1..n\})$$

La règle SR est instanciée par différentes règles données.

Une règle donnée est basée sur un ensemble T de z termes de règles, notés RT_x tels que :

$$T = \{RT_x, 1 \leq x \leq z\} = \{EA_x op_x \{ens|val\}_x\}$$

où EA_x est un attribut du niveau existant, op_x est un opérateur, et $\{ens|val\}_x$ est soit un ensemble de valeurs, soit une valeur (selon l'opérateur utilisé), ces valeurs appartenant au domaine de définition de EA_x.

Soient q le nombre d'instances de GL, un ensemble $R = \{r_d, d = 1..q\}$ de q règles données doivent être définies. Notons v_{dj} la valeur de l'attribut généré GA_j dans la règle données r_d.

La clause «si» (if) est basée sur une composition de conjonctions ou de disjonctions de termes de règle, la clause «alors» (then) définit les valeurs des attributs (autrement dit les instances du nouveau niveau).

Une règle données r_d est définie de la façon suivante :

$$r_d : if\ RT_1\ AND|OR\ ...\ AND|OR\ RT_x\ AND|OR\ ...\ AND|OR\ RT_z$$

$$then\ GA_1 = v_{d1}\ AND\ ...\ AND\ GA_j = v_{dj}\ AND\ ...\ AND\ GA_n = v_{dn}$$

Ainsi, la participation des utilisateurs réside dans l'expression de ces règles d'agrégation (structure et données) représentant leurs connaissances sur la façon d'agréger les données et traduisant leurs propres besoins d'analyse.

3.4 Exploitation de la participation des analystes

Une fois les connaissances exprimées, il s'agit d'exploiter cette participation, en l'occurrence, faire évoluer l'entrepôt de données en fonction des règles exprimées.

Compte-tenu des spécificités du stockage XML, l'évolution de l'entrepôt n'est pas une tâche facile. En effet, nous devons prendre en compte le fait que la structure et les données sont renfermées dans les documents, même si le schéma de l'entrepôt est représenté dans un document bien identifié $dw - model.xml$. De plus, nous devons prendre en compte l'organisation des informations sous forme d'arbre.

La création d'un nouveau niveau nécessite non seulement la création de ce niveau lui-même avec les données adéquates, mais aussi les liens avec le ou les autres niveaux (attribut @Drill-Down si le niveau est ajouté à la fin d'une hiérarchie, attributs @Drill-Down et @Roll-Up si le niveau est inséré entre deux niveaux existants).

Pour considérer cette évolution, nous devons l'envisager pour deux documents : le document représentant le modèle ($dw - model.xml$) et le document de la dimension qui est concerné par l'ajout d'un niveau ($dimension_d.xml$) puisque chaque dimension est représentée dans un document XML.

Notons que le document $facts.xml$ n'est pas modifié puisqu'il s'agit d'une partie fixe qui assure l'intégrité par rapport aux données sources et au processus d'alimentation.

Nous pouvons résumer les différentes opérations pour exploiter le résultat de la participation d'un analyste donné comme suit :

1. dans le document $dw - model.xml$:
   (a) ajouter le nœud correspond au nouveau niveau, en extrayant les informations dans la règle structure
2. dans le document $dimension_d.xml$:
   (a) ajouter l'élément représentant le niveau
   (b) exploiter les règles données pour ajouter les éléments nécessaires
   (c) mettre à jour les propriétés roll-up
   (d) si le niveau a été inséré entre deux niveaux existants, mettre à jour les propriétés drill-down du niveau supérieur

4 Mise en œuvre de notre approche

4.1 Éléments d'implémentation

Notre approche de personnalisation collaborative pour les analyses dans les entrepôts XML est actuellement en cours de développement.

Une interface web permet l'interaction avec les utilisateurs. Ainsi, cette interface va aider l'utilisateur à exprimer ses connaissances, autrement dit les règles (structure et données), de façon intuitive. L'utilisateur est en effet guidé pour choisir les éléments dans l'interface, l'aidant ainsi à exprimer ses connaissances et donc à participer à l'enrichissement des possibilités d'analyse de l'entrepôt XML. Cette interaction est développée grâce à des scripts PHP.

L'évolution de l'entrepôt de données XML nécessite des mises à jour au sein des documents XML qui sont exploités pour stocker l'entrepôt. Ainsi, nous devons développer le programme requis pour mettre à jour les documents XML.

Xupdate est un langage de requête XML dédié à la modification de données XML. Il s'agit d'une spécification de XML :DBInitiative. C'est un langage de mise à jour XML utilisé pour modifier le contenu XML en déclarant quels changements doivent être opérés sur la syntaxe XML. Différentes opérations élémentaires peuvent être combinées pour réaliser l'évolution que nous avons présentée pour apporter une personnalisation collaborative dans les entrepôts de données XML. Ces opérations élémentaires sont par exemple l'insertion d'un élément, l'insertion d'un attribut, modifier un attribut.

Une autre alternative pour notre implémentation serait de considérer le recours au DOM (Document Object Model). En effet, DOM est un modèle objet standard indépendant de toute plateforme et de tout language pour représenter des formats relatifs au HTML ou au XML. Ainsi, il serait possible d'utiliser des scripts PHP avec du DOM pour réaliser l'évolution. En particulier, mentionnons la méthode DomNode :appendChild qui permet d'insérer un nouvel élément dans un document et la méthode DomNode.setNodeValue qui met à jour un nœud et ses propriétés.

Nous sommes actuellement en train d'étudier ces deux alternatives d'implémentation.

4.2 Étude de cas

Pour illustrer notre approche, considérons le projet issu d'une Action Concertée Incitative avec des collègues linguistes. Ce projet, nommé CLAPI[1] pour «Corpus de Langues Parlées en Interaction», traite de l'intégration, du stockage, de la gestion et de l'analyse de corpus de langues parlées en interaction (Aouiche et al., 2003). Un corpus comprend des enregistrements audio et/ou vidéo d'interactions de la vie courante comme par exemple le déroulement d'un cours dans une salle de classe.

Chaque intervenant dans un enregistrement est identifié avec un pseudonyme et peut apparaître dans plusieurs interactions. Afin d'être exploité par les linguistes, les enregistrements sont reportés sous forme de texte dans des transcriptions. Ces transcriptions sont actuellement modélisées en XML : les «tokens» qui sont les formes orales d'un mots retranscrites comme «h'llo» pour «hello» et les phénomènes d'interaction tels que les pauses, les rires, les chevauchements de parole, etc. Les linguistes disposent d'une interface web dédiée pour exploiter ces transcriptions.

L'analyse multidimensionnelle sur ces transcriptions est très pertinente pour les linguistes. Citons par exemple l'intérêt pour eux d'observer les fréquences de tokens en fonction de la place de ceux-ci dans la transcription (début, milieu, fin) et l'âge de l'intervenant. Cet exemple simplifié correspond à l'instanciation du document $dw - model.xml$ présentée dans la figure 4.

```
<?xml version="1.0" encoding="utf-8">
<DW-model>
   <dimension id="time-d" path="dim-time.xml">
      <Level id="location-in-transcription">
         <attribute name="location" type="string" />
      </Level>
   </dimension>
   <dimension id="speaker-d" path="dim-speaker.xml">
      <Level id="speaker">
         <attribute name="sex" type="boolean" />
      </Level>
   </dimension>
   <dimension id="transcription-d" path="dim-transcript.xml">
      <Level id="token">
         <attribute name="term" type="string" />
      </Level>
      <Level id="transcription">
         <attribute name="transcription-name" type="string" />
      </Level>
   </dimension>
   <FactDoc id="facts" path="facts.xml">
      <measure id="frequency" type="real" />
      <dimension idref="time-d" />
      <dimension idref="speaker-d" />
      <dimension idref="transcription-d" />
   </FactDoc>
</DW-model>
```

FIG. 4 – *Document* $dw - model.xml$ *exemple.*

Ce schéma initial a été conçu à partir des données de la base CLAPI et en fonction des besoins d'analyse identifiés. Mais un linguiste peut avoir besoin d'agréger les fréquences en groupant certains emplacements des tokens. Il souhaite par exemple savoir si certains tokens apparaissent davantage à l'extrémité (début ou fin) qu'au milieu des interactions.

[1] http ://clapi.univ-lyon2.fr

Même si ce besoin n'a pas été exprimé initialement, le processus de personnalisation collaborative va non seulement permettre de répondre à ce besoin d'analyse individuel, mais également de faire profiter de cette possibilité d'analyse aux autres linguistes exploitant le système.

Le linguiste va donc formuler les règles d'agrégation traduisant son besoin d'analyse, en l'occurrence la façon d'agréger les données : une règle structure et les règles données correspondantes.

Règle structure :

$$(SR)\ if\ ConditionOn(location-in-transcription,\ \{location\})$$
$$then\ Generate(group-of-location,\ \{group-location\})$$

Règles données :

$$(r_1)\ if\ location\ in\ \{'begin',\ 'end'\}\ then\ group-location\ =\ 'extreme'$$
$$(r_2)\ if\ location\ not\ in\ \{'begin',\ 'end'\}\ then\ group-location\ =\ 'middle'$$

Grâce à ces règles, l'entrepôt de données XML peut évoluer à travers la modification du document $dw-model.xml$ d'une part, du document $dim-time.xml$ d'autre part puisqu'il correspond au document de la dimension qui va être enrichie d'un niveau hiérarchique. Le document $dw-model.xml$ est modifié pour inclure le nouveau niveau (parties en gras dans la figure 5).

```
<?xml version="1.0" encoding="utf-8">
<DW-model>
   <dimension id="time-d" path="dim-time.xml">
      <Level id="location-in-transcription">
         <attribute name="location" type="string" />
      </Level>
      <Level id="group-of-location-in-transcription">
         <attribute name="location-group" type="string" />
      </Level>
   </dimension>
   <dimension id="speaker-d" path="dim-speaker.xml">
      <Level id="speaker">
         <attribute name="sex" type="boolean" />
      </Level>
   </dimension>
   <dimension id="transcription-d" path="dim-transcript.xml">
      <Level id="token">
         <attribute name="term" type="string" />
      </Level>
      <Level id="transcription">
         <attribute name="transcription-name" type="string" />
      </Level>
   </dimension>
   <FactDoc id="facts" path="facts.xml">
      <measure id="frequency" type="real" />
      <dimension idref="time-d" />
      <dimension idref="speaker-d" />
      <dimension idref="transcription-d" />
   </FactDoc>
</DW-model>
```

FIG. 5 – *Document* $dw-model.xml$ *exemple mis à jour.*

Le document $dim-time.xml$ est également mis à jour pour représenter le nouveau niveau et ses instances, ainsi que les liens d'agrégation requis (parties en gras dans la figure 6).

Ainsi, le linguiste va pouvoir connaître les fréquences des tokens en fonction des groupes d'emplacements qu'il a défini, obtenant ainsi une réponse à ses propres besoins d'analyse. Et le niveau ainsi créé va pouvoir être exploité pour différentes analyses (détail des fréquences par groupe d'emplacements et par sexe du locuteur par exemple), par le linguiste qui en avait besoin mais également par les autres linguistes.

5 Conclusion et perspectives

Dans cet article, nous avons posé les bases d'un système de personnalisation collaborative pour l'enrichissement des analyses dans les entrepôts de données complexes XML. Nous avons explicité les principes de cette proposition,

```
<?xml version="1.0" encoding="utf-8">
<dimension dim-id="time-d">
     <Level id="location-in-transcription">
       <Instance id="begin" Roll-up="extreme">
         <attribute id="location" value="begin">
       </Instance>
       <Instance id="middle" Roll-up="middle">
         <attribute id="location" value="middle">
       </Instance>
       <Instance id="end" Roll-up="extreme">
         <attribute id="location" value="end">
       </Instance>
     </Level>
     <Level id="group-of-location-in-transcription">
       <Instance id="extreme" Drill-Down={"begin","end"}>
         <attribute id="location-group" value="extreme">
       </Instance>
       <Instance id="middle" Roll-up="middle">
         <attribute id="location-group" value="middle">
       </Instance>
     </Level>
</dimension>
```

FIG. 6 – *Document* $dim - time.xml$ *mis à jour.*

décrit le processus centré utilisateurs pour permettre la réalisation de cette proposition, nous avons détaillé les deux principaux modules pour atteindre l'objectif, à savoir la participation des utilisateurs (l'acquisition des connaissances) et l'exploitation de cette participation (l'évolution incrémentale de l'entrepôt). L'implémentation de notre approche est en cours de développement, mais nous avons indiqué quelques éléments la concernant, et nous avons illustré notre approche par une étude de cas issu d'une collaboration avec des linguistes.

Cet article présente un travail préliminaire dans le domaine de la personnalisation collaborative dans les entrepôts de données complexes ouvrant de nombreuses perspectives. Dans l'immédiat, il s'agit, à partir d'un recensement des hiérarchies complexes inspiré par la modélisation des hiérarchies établie par Malinowski et Zimányi (2004), d'aider à la saisie des règles exprimant les connaissances. Ainsi les règles seront le fondement du système, il s'agit alors de guider l'utilisateur dans l'expression de ses connaissances et d'adapter la vérification des règles saisies. Aini la formalisation devra également être enrichie par rapport aux différentes propriétés pour prendre en compte les différents types de hiérarchie.

Dans notre approche, l'apect collaboratif se traduit par un enrichissement incrémental de l'entrepôt mettant finalement à disposition des utilisateurs de la communauté les suggestions individuelles. En d'autres mots, il s'agit d'un partage des possibilités d'analyse créées individuellement. Il serait alors intéressant d'envisager l'aspect collaboratif en y introduisant l'échange de points de vue. Ceci peut être intéressant en particulier dans le cas de points de vue divergents. Nous avons d'ores et déjà proposé d'avoir recours au versionnement pour traduire des points de vue différents sur un même niveau (Bentayeb et al., 2008), mais il serait sans doute pertinent d'envisager d'autres solutions.

Par ailleurs, ce principe de personnalisation collaborative pourrait être étendu. En effet, dans notre cas, les nouveaux axes sont accessibles par les autres usagers du système. Mais nous pourrions aller au-delà en envisageant un système de recommandation. Des travaux commencent à émerger sur cet aspect de recommandation dans les entrepôts de données (Giacometti et al., 2008). Il s'agit alors de pouvoir exploiter l'aspect collaboratif que nous avons introduit. Nous pensons également que le concept de profil pourrait être pertinent. En effet, dans le contexte d'une organisation telle qu'une entreprise, la notion de métier peut être intéressante comme base pour la recommandation.

Enfin, Bentayeb (2008) a proposé l'utilisation d'une méthode de fouille de données, celle des K-means en l'occurrence, pour permettre la construction de nouveaux niveaux de granularité. Il serait sans doute pertinent de s'intéresser à des méthodes de fouille de données dans les documents XML pour faire émerger de nouveaux axes d'analyse. Ceci aurait pour objectif, entre autres, d'exploiter des connaissances dont les utilisateurs ne disposent pas.

Références

Aouiche, K., F. Bentayeb, O. Boussaid, et J. Darmont (2003). Conception informatique d'une base de données multimédia de corpus linguistiques oraux : l'exemple de clapi. In *36ème Colloque International de la Societas Linguistica Europaea, Lyon, France*, pp. 11–12.

Aouiche, K., D. Lemire, et R. Godin (2008). Collaborative OLAP with Tag Clouds - Web 2.0 OLAP Formalism and Experimental Evaluation. In *4th International Conference on Web Information Systems and Technologies (WEBIST 08), Funchal, Madeira, Portugal*, pp. 5–12.

Bebel, B., J. Eder, C. Koncilia, T. Morzy, et R. Wrembel (2004). Creation and Management of Versions in Multiversion Data Warehouse. In *19th ACM Symposium on Applied Computing (SAC 04), Nicosia, Cyprus*, pp. 717–723.

Bellatreche, L., A. Giacometti, P. Marcel, H. Mouloudi, et D. Laurent (2005). A Personalization Framework for OLAP Queries. In *8th ACM International Workshop on Data Warehousing and OLAP (DOLAP 05), Bremen, Germany*, pp. 9–18.

Bentayeb, F. (2008). K-means based Approach for OLAP Dimension Updates. In *10th International Conference on Enterprise Information Systems (ICEIS 08), Barcelona, Spain*, Volume DISI, pp. 531–534.

Bentayeb, F., C. Favre, et O. Boussaïd (2008). A User-driven Data Warehouse Evolution Approach for Concurrent Personalized Analysis Needs Integrated Computer-Aided Engineering. *Journal of Integrated Computer-Aided Engineering 15*(1), 21–36.

Beyer, K., D. Chambérlin, L. Colby, F. Özcan, H. Pirahesh, et Y. Xu (2005). Extending XQuery for Analytics. In *24th International Conference on Management of Data (SIGMOD 05), Baltimore, Maryland, USA*, pp. 503–514.

Blaschka, M., C. Sapia, et G. Höfling (1999). On Schema Evolution in Multidimensional Databases. In *1st International Conference on Data Warehousing and Knowledge Discovery (DaWaK 99), Florence, Italy*, Volume 1676 of *LNCS*, 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 Databases and Information Systems, Riga, Latvia*, pp. 27–41.

Body, M., M. Miquel, Y. Bédard, et A. Tchounikine (2003). Handling Evolutions in Multidimensional Structures. In *19th International Conference on Data Engineering (ICDE 03), Bangalore, India*, pp. 581–591.

Boukraa, D., R. B. Messaoud, et . Boussaïd (2006). Proposition d'un modèle physique pour les entrepôts XML. In *Atelier Systèmes Décisionnels (ASD 06) en conjonction avec 9th Maghrebian Conference on Information Technologies (MCSEAI 06), Agadir, Morocco*.

Boussaid, O., J. Darmont, F. Bentayeb, et S. Loudcher (2008). Warehousing Complex Data from the Web. *International Journal of Web Engineering and Technology 4*(4), 408–433.

Cabanac, G., M. Chevalier, F. Ravat, et O. Teste (2007). An Annotation Management System for Multidimensional Databases. In *9th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 07), Regensburg, Germany*, Volume 4654 of *LNCS*, pp. 89–98.

Giacometti, A., P. Marcel, et E. Negre (2008). A Framework for Recommending OLAP Queries. In *11th ACM International Workshop on Data Warehousing and OLAP (DOLAP 08), Napa Valley, California, USA*, pp. 73–80.

Goldberg, D., D. Nichols, B. M. Oki, et D. Terry (1992). Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM 35*(12), 61–70.

Golfarelli, M., S. Rizzi, et B. Vrdoljak (2001). Data Warehouse Design from XML Sources. In *4th ACM International Workshop on Data Warehousing and OLAP (DOLAP 01), Atlanta, Georgia, USA*, pp. 40–47.

Hümmer, W., A. Bauer, et G. Harde (2003). XCube : XML for data warehouses. In *6th International Workshop on Data Warehousing and OLAP (DOLAP 03), New Orleans, Louisiana, USA*, pp. 33–40.

Hurtado, C. A., A. O. Mendelzon, et A. A. Vaisman (1999). Updating OLAP Dimensions. In *2nd ACM International Workshop on Data Warehousing and OLAP (DOLAP 99), Kansas City, Missouri, USA*, pp. 60–66.

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

Ioannidis, Y. et G. Koutrika (2005). Personalized systems : models and methods from an IR and DB perspective. In *31st International Conference on Very Large Data Bases (VLDB 05), Trondheim, Norway*, pp. 1365–1365.

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

Mahboubi, H., J. C. Ralaivao, S. Loudcher, O. Boussaid, F. Bentayeb, et J. Darmont (2009). *X-WACoDa: An XML-based approach for Warehousing and Analyzing Complex Data*. Advances in Data Warehousing and Mining. IGI Publishing.

Malinowski, E. et E. Zimányi (2004). OLAP Hierarchies: A Conceptual Perspective. In *16th International Conference on Advanced Information Systems Engineering (CAiSE 04), Riga, Latvia*, Volume 3084 of *LNCS*, pp. 477–491.

Mendelzon, A. O. et A. A. Vaisman (2000). Temporal Queries in OLAP. In *26th International Conference on Very Large Data Bases (VLDB 00), Cairo, Egypt*, pp. 242–253.

Morzy, T. et R. Wrembel (2004). On Querying Versions of Multiversion Data Warehouse. In *7th ACM International Workshop on Data Warehousing and OLAP (DOLAP 04), Washington, Columbia, USA*, pp. 92–101.

Park, B.-K., H. Han, et I.-Y. Song. XML-OLAP: A Multidimensional Analysis Framework for XML Warehouses. In *7th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 05), Copenhagen, Denmark*.

Pokorný, J. (2002). XML Data Warehouse: Modelling and Querying. In *5th Baltic Conference (BalticDB&IS 06), Tallin, Estonia*, pp. 267–280.

Ravat, F. et O. Teste (2008). Personalization and OLAP Databases. *Annals of Information Systems, Numéro spécial New Trends in Data Warehousing and Data Analysis 3*.

Ravat, F., O. Teste, et G. Zurfluh (2006). A Multiversion-Based Multidimensional Model. In *8th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 06), Krakow, Poland*, Volume 4081 of *LNCS*, pp. 65–74.

Summary

The XML data warehouses are a good alternative for the representation, storage and analysis of complex data. The model of a data warehouse is classically designed from the available data sources and analysis needs identified during the conception. It turns out that the analysis needs may emerge, depending on knowledge of analysts. This knowledge may concern new ways to aggregate data. Thus, to provide an answer to individual analysis needs and take advantage of knowledge of different analysts using the data warehouse, we propose in this paper a collaborative personalization for enrichment opportunities for analysis of XML data warehouse. This approach is based on the expression of knowledge analysts on how to aggregate the data, allowing the sharing of new possibilities for analysis through the enrichment of dimension hierarchies that drive the navigation in the XML data warehouse.