BI4people : le décisionnel pour tous

Oksana Grabova*, Somayeh Sobati Moghadam*, Samaneh Chagheri*, Jérôme Darmont*

*Université de Lyon (ERIC Lyon 2), 5 avenue Pierre Mendès-France, 69676 Bron Cedex
nom.prenom@univ-lyon2.fr — http://eric.univ-lyon2.fr

Résumé. Cette démonstration a pour objet un système décisionnel en mode *Software as a Service* destiné aux très petites entreprises, associations et individus. L'objectif est de permettre une prise en main simplifiée du processus décisionnel, en masquant les phases d'intégration de données et de conception d'un entrepôt et en permettant une navigation simple dans les données étudiées.

1 Introduction

Les technologies décisionnelles telles que les entrepôts de données et l'analyse en ligne (OLAP) ont longtemps nécessité des investissements lourds, uniquement possibles dans les grands organismes et entreprises. Avec l'apparition de l'infonuagique (*cloud computing*) et du paiement de ressources (espace de stockage, temps de calcul) à la demande, il est devenu possible de démarrer un projet décisionnel à coût réduit. De nombreuses solutions décisionnelles dans le nuage ont donc vu le jour. Pour autant, ces outils restent encore réservés aux spécialistes et hors de portée des très petites entreprises ou associations, voire du simple citoyen. Pour ce type de public, des solutions existent, mais plutôt de type tableur pour les données et production de graphiques pour le rendu, comme Google fusion tables (Gonzalez et al., 2010).

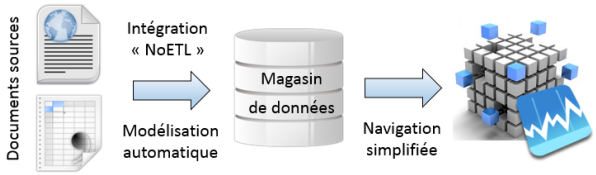

FIG. 1 – *Schéma global de fonctionnement de BI4people*

Le prototype BI4people présenté dans cet article a pour objectif d'aller plus loin que ces solutions, en appliquant au sein d'une application web (en mode *Software as a Service*) le processus de conception d'un magasin de données (Figure 1), depuis l'intégration des données « sans ETL » (transparente pour l'utilisateur) à la navigation OLAP au sein d'une interface conviviale et simple, en passant par la modélisation multidimensionnelle automatique du magasin de données. L'objet de la démonstration est d'illustrer le fonctionnement de BI4people : à

partir d'un ou deux fichiers d'entrée de type CSV (qui sont pour l'instant les seuls types de documents possibles en entrée de BI4people, mais qui seront complétés à terme par des données issues du Web via une recherche par mots clés, comme des pages web, des documents XML ou des ensembles de triplets RDF), il s'agit d'afficher en temps réel à l'utilisateur les données analysables (mesures) et les axes d'analyse possibles (dimensions), ainsi que de lui permettre de naviguer dans les données dans une interface ergonomique basée sur des graphiques et implémentant les principaux opérateurs OLAP.

Les méthodes utilisées dans BI4people sont présentées dans la section 2. Nous discutons nos choix technologiques dans la section 3. Finalement, nous faisons le bilan du travail effectué jusqu'ici et listons les nombreuses perspectives ouvertes dans la section 4.

2 Principes de fonctionnement de BI4people

2.1 Intégration des données

Afin que le système reste extrêmement simple à utiliser, nous nous inscrivons dans la tendance *NoETL* prônée par Middelfart (2013). Il s'agit en fait d'effectuer les tâches d'extraction, transformation et chargement de manière automatique et transparente pour l'utilisateur, qui ne doit fournir que les données sources en sa possession.

Nous ne traitons actuellement que des données tabulaires au format CSV, avec une ligne d'entête définissant des attributs. Chaque fichier source est transcrit dans une table d'un petit ODS. Pour cela, nous procédons en premier lieu à une analyse syntaxique qui permet d'en déterminer la structure : noms des attributs et types de données de chaque attribut. Nous analysons ensuite l'unicité des données de chaque attribut, afin d'identifier les clés primaires candidates et de dédoublonner, le cas échéant, les valeurs de la clé primaire sélectionnée. S'il existe plusieurs clés candidates pour une table, celle possédant le type de données le plus court est choisie. La présence de l'attribut dans un autre fichier source, et donc dans une autre table de l'ODS (clé étrangère), peut également être un facteur de choix. En cas d'absence de clé primaire candidate, nous en créons une de type entier que nous incrémentons nous-même.

En sortie du module responsable de cette phase d'ETL, nous obtenons une série de requêtes SQL permettant de : 1) créer les tables (contraintes de clés primaires comprises) ; 2) y ajouter les contraintes de clés étrangères ; 3) y insérer les données. Cet ODS alimente ensuite le processus de modélisation automatique du magasin de données.

2.2 Modélisation automatique du magasin de données

Cette étape consiste à construire un schéma multidimensionnel physique à partir des tables de l'ODS, de nouveau sans implication de l'utilisateur non-expert, que nous souhaitons interroger le moins possible et, le cas échéant, uniquement sur des problématiques métier qu'il est susceptible de maîtriser. Cela exclut donc une modélisation basée sur les besoins (qui s'exprimeront implicitement, à terme, par l'intégration de données issues du Web en plus des données privées de l'utilisateur), au profit d'une modélisation basée sur les données.

Nous adaptons pour cela l'approche de modélisation automatique de schéma d'entrepôt de données de Phipps et Davis (2002) notamment, compte-tenu des objectifs de simplicité que nous nous sommes fixés, en ne proposant pas à l'utilisateur de choisir entre différents

schémas. Nous avons choisi cette méthode car elle ne nécessite ni l'intervention d'un expert, ni de l'utilisateur (dans notre variante) et ne nécessite pas la conception préalable d'une ontologie de domaine.

Cette approche est constituée des étapes suivantes : 1) identification des objets d'analyse (tables de faits) ; ce sont des tables contenant des mesures (attributs numériques non clés) et des références à des dimensions (attributs, majoritairement textuels, qui ne sont ni des clés ni des mesures) ; 2) détection des associations entre tables par identification de couples clé primaire-clé étrangère ; 3) création des hiérarchies des dimensions par analyse des cardinalités des associations entre tables.

2.3 Analyse en ligne simplifiée

Une fois que l'utilisateur a indiqué les données brutes qu'il souhaite analyser, qu'un magasin de données a été automatiquement créé et alimenté, l'utilisateur peut choisir l'objet de son analyse et naviguer dans les données sans avoir besoin de maîtriser les opérateurs OLAP, à l'instar de M. Jourdain qui dit de la prose sans le savoir.

Pour parvenir à cet objectif, nous avons évalué les outils OLAP libres et leurs possibilités d'intégration dans BI4people. Compte-tenu du cadre technologique que nous avons choisi (Section 3), seul le serveur ROLAP Mondrian et ses surcouches web telles que JPivot et ses successeurs sont appropriés. Toufefois, ces outils nous semblent encore complexes et destinés à des spécialistes. Nous avons donc décidé de développer une interface graphique ad-hoc, simple et intuitive pour naviguer dans les données, qui implémente les opérateurs OLAP principaux (*roll-up*, *drill-down*, *slice*, *dice*, *rotate*).

3 Choix technologiques

Afin de réaliser notre application en ligne, nous avons privilégié des outils libres. Le choix du langage de programmation PHP, qui permet la création d'interfaces AJAX et dispose de bibliothèques graphiques très efficaces, s'est donc rapidement imposé.

Pour l'exploitation du magasin de données, nous nous sommes orientés vers une approche ROLAP en mémoire vive, c'est-à-dire l'utilisation d'opérateurs OLAP au sein d'un SGBD relationnel opérant en mémoire. C'est en effet une architecture qui presente les avantages, notamment au niveau des performances, des approches MOLAP, mais sans leur lourdeur (Grabova et al., 2010). Cela évite également de recourir à une « boîte noire » utilisant une base de données vectorielle telle que QlikView.

Finalement, parmi les SGBD en mémoire vive libres et compatibles avec PHP et SQL, nous avons retenu Infobright. Les autres systèmes candidats présentent en effet des inconvénients majeurs dans notre contexte. SQLite ne supporte pas les opérations d'altération de table dont nous avons besoin lors de la phase d'ETL. De plus, SQLite est connu pour être lent lors de l'exécution de requêtes de regroupement. FastDB, malgré d'excellentes performances, autorise l'insertion de doublons pour une clé primaire. CSQL, outre des difficultés d'installation importantes, ne supporte que SQL92, et donc pas les opérateurs OLAP. Enfin, PosgreSQL était une possibilité intéressante, mais nous lui avons préféré le stockage orienté colonnes d'Infobright, bien adapté au stockage et au traitement d'une base de données multidimensionnelle.

4 Conclusion et perspectives

BI4people est actuellement à un stade de développement très préliminaire, aussi de nombreuses améliorations sont prévues, notamment au niveau de l'intégration des données. Il est en effet souhaitable qu'un utilisateur puisse non seulement téléverser ses données privées dans le système, mais aussi bénéficier, comme c'est le cas dans Google fusion tables, d'une fonctionnalité de recherche sur le Web pour intégrer des données externes complémentaires. Des ontologies seront alors nécessaire pour gérer les conflits sémantiques entre les sources de données, en les représentant dans un formalisme pivot qui permette d'importer et de stocker l'ensemble des données de manière sémantique et automatique et de construire un entrepôt conceptuel (Romero et Abelló, 2007).

Par ailleurs, la plateforme BI4people est destinée à servir de terrain d'expérimentation pour des travaux de recherche. Au laboratoire ERIC, deux projets doctoraux l'exploitent. Le premier est consacré à la performance du système, de manière à garantir une réponse instantanée à l'utilisateur. Le second s'intéresse à la confidentialité des données, notamment dans un cadre collaboratif que nous envisageons pour le futur. Toute participation extérieure à notre laboratoire au développement de BI4people est la bienvenue.

Références

Gonzalez, H., A. Y. Halevy, C. S. Jensen, A. Langen, J. Madhavan, R. Shapley, et W. Shen (2010). Google fusion tables : data management, integration and collaboration in the cloud. In *1st ACM Symposium on Cloud Computing (SoCC 2010), Indianapolis, Indiana, USA*, pp. 175–180.

Grabova, O., J. Darmont, J.-H. Chauchat, et I. Zolotaryova (2010). Business intelligence for small and middle-sized entreprises. *SIGMOD Record 39*(2), 39–50.

Middelfart, M. (2013). The Inverted Data Warehouse based on TARGIT Xbone – How the biggest of data can be mined by "the little guy". In *7th International Workshop on Business Intelligence for the Real Time Enterprise (BIRTE 2013), Riva del Garda, Italy*. Invited industrial talk.

Phipps, C. et K. C. Davis (2002). Automating data warehouse conceptual schema design and evaluation. In *4th International Workshop on Design and Management of Data Warehouses (DMDW 2002), Toronto, Canada*, Volume 58 of *CEUR Workshop Proceedings*, pp. 23–32.

Romero, O. et A. Abelló (2007). Automating multidimensional design from ontologies. In *10th ACM International Workshop on Data Warehousing and OLAP (DOLAP 2007), Lisbon, Portugal*, pp. 1–8.

Summary

This demonstration presents a so-called personal business intelligence system in Software as a Service mode. This system targets very small enterprises, organizations and individuals. We aim at providing a simple approach to the decision support process, by masking data integration and warehouse modeling phases, and by allowing a simple navigation through data.