DataGuide : une approche pour l'implantation de schémas NoSQL

Faten Atigui*, Asma Mokrani*, Nicolas Travers**,*

*Laboratoire CEDRIC, CNAM, Paris France - prenom.nom@cnam.fr
**Léonard de Vinci Pôle Universitaire, Research Center, Paris La Défense, France
prenom.nom@devinci.fr

Depuis plusieurs décennies, le stockage et l'exploitation de données reposent principalement sur les bases de données relationnelles. Avec le *big data*, le volume des données a explosé, l'hétérogénéité s'est décuplée, posant des problèmes de transformation des SGBDR à de nouveaux stockages sur le *Cloud*, que ce soit en matière de stockage, d'interrogation, de coût ou de performance. Avec plus de 225 solutions NoSQL différentes, il est difficile de déterminer la solution la plus adaptée à ses besoins. Les conséquences d'un mauvais choix peuvent engendrer des problèmes de passage à l'échelle, de cohérence de données ou d'évolutivité.

Les solutions pour la transformation digitale d'une base de données restent essentiellement cloisonnées dans leur démarche. En effet, les méthodes reposent soit sur une transposition du modèle physique posant des problèmes de passage à l'échelle (Hamouda et Zainol, 2017; Boussahoua et al., 2017), soit un modèle purement logique proposant des transformations respectant la logique métier (Abdelhédi et al., 2017; Chebotko et al., 2015). Toutefois, elles ne prennent pas en compte les problématiques d'optimisation à l'origine de cette transformation.

DataGuide. Notre approche génère des *schémas logiques* pour chaque famille NoSQL et le relationnel, reposant sur le *schéma conceptuel*, puis en en passant d'un *schéma logique* à l'autre par raffinement. Afin de formaliser et d'automatiser l'implantation de schémas, nous avons adopté une architecture dirigée par les modèles (MDA) comme illustrée dans la figure 1, avec : 1) le **PIM1** (*Modèle Indépendant de la Plateforme*) de premier niveau intègre les besoins fonctionnels du SI (données et requêtes) qui sert de base de réflexion pour la modélisation du système cible, 2) le **PIM2** est le modèle indépendant de second niveau, commun aux cinq familles de modèles (NoSQL & relationnel) qui permet d'effectuer des *raffinements* du modèle en générant l'ensemble des modèles dénormalisés possibles grâce à des règles de transformation (exprimées en `QVT`) guidées par une heuristique de génération, 3) Les **PSMx** (*Modèles Spécifiques à la Plateforme*) obtenus par transformation des PIM2 compatibles avec la famille de données cible (*ex.* agrégation pour les documents), les choix de stratégies de sharding et d'indexation sont obtenus grâce à un modèle de coût générique aux 5 familles.

La particularité de notre approche est de reposer sur un méta-modèle commun (PIM2) intégrant les 5 familles de modèles de données capable d'intégrer toutes les possibilités de *raffinement* de schémas (fusion et éclatement). La figure 2 montre le méta-modèle du PIM2 intégrant tous les concepts : *rows* (instances), clés-valeurs (valeurs simples ou complexes). Des concepts peuvent également être reliés par des liens pour l'intégration d'une base de données

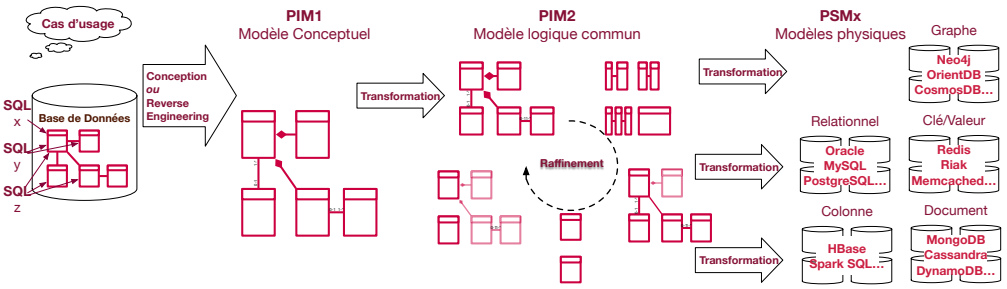

FIG. 1 – *DataGuide : une approche pour l'implantation de schémas NoSQL*

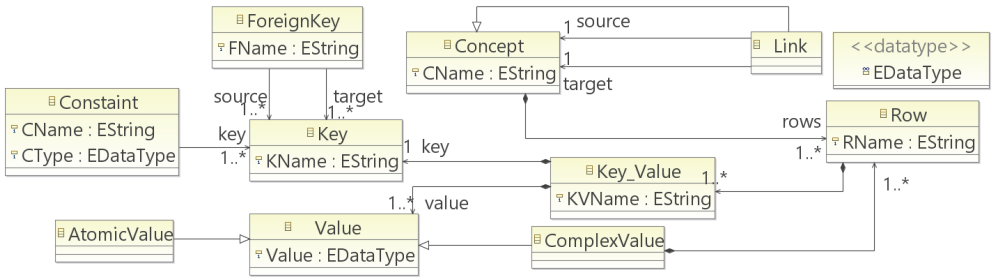

FIG. 2 – *PIM2 - Méta-modèle commun pour les 5 familles de modèles de données*

graphes. Les valeurs complexes sont représentées par la composition de *rows*, permettant à une valeur d'être multiple avec un schéma structurellement identique à un concept. Le but est de faciliter l'application de règles de raffinement sans limites d'itérations pour la production de schémas. Les contraintes et clés étrangères sont associées aux clés, appartenant à des concepts distincts afin de favoriser la transformation des contraintes en concepts.

Une heuristique applique les règles de raffinement afin de générer un arbre de possibilités dont les premiers noeuds s'orientent sur les éclatements, puis vers les fusions, et réduit l'espace grâce au cas d'usage. Le schéma le plus efficace donné par le modèle de coût sera implanté.

Références

Abdelhédi, F., A. A. Brahim, F. Atigui, et G. Zurfluh (2017). MDA-Based Approach for NoSQL Databases Modelling. In *DaWaK'17*, pp. 88–102. LNCS.

Boussahoua, M., O. Boussaid, et F. Bentayeb (2017). Logical Schema for Data Warehouse on Column-Oriented NoSQL Databases. In D. Benslimane, E. Damiani, W. I. Grosky, A. Hameurlain, A. Sheth, et R. R. Wagner (Eds.), *DEXA*, pp. 247–256. Springer.

Chebotko, A., A. Kashlev, et S. Lu (2015). A Big Data Modeling Methodology for Apache Cassandra. In *International Congress on Big Data*, New York, USA, pp. 238–245. IEEE.

Hamouda, S. et Z. Zainol (2017). Document-Oriented Data Schema for Relational Database Migration to NoSQL. In *Innovate-Data*, pp. 43–50. IEEE.