Vers l'intégration multidimensionnelle d'Open Data dans les entrepôts de données

Alain Berro, Imen Megdiche, Olivier Teste

Vers l'intégration multidimensionnelle d'Open Data dans les entrepôts de données

Alain Berro *, Imen Megdiche *, Olivier Teste *

*IRIT-Université de Toulouse
118 Route de Narbonne, 31062 Toulouse
{Alain.Berro, Imen.Megdiche, Olivier.Teste}@irit.fr

Résumé. L'émergence de nombreuses sources d'Open Data poussent plusieurs communautés de recherche ainsi que des entreprises à développer des outils permettant leur exploitation. En particulier, les données statistiques présentes dans les Open Data peuvent constituer des informations utiles aux analyses décisionnelles. Toutefois les Open Data très hétérogènes et disséminés en plusieurs morceaux de données sur le web, rendent difficile leur intégration au sein d'un entrepôt de données. Les travaux actuels sur l'intégration des Open Data proposent des processus d'intégration basés sur des Linked Open Data, dont la mise en place n'est pas automatisée. Dans cet article, nous proposons un processus visant à automatiser l'entreposage multidimensionnel des Open Data. Notre démarche repose sur la transformation des Open Data en un graphe générique et enrichi favorisant leur intégration. Ce graphe sert de support pour la définition semi-automatique et incrémentale du schéma multidimensionnel d'entreposage.

1 Introduction

Les données issues du web posent plusieurs défis pour l'informatique décisionnelle : ces données sont très riches en informations mais d'une complexité qui rend difficile leur intégration automatique dans les entrepôts de données (Ravat et al., 2010). Dans ce cadre, nous nous intéressons plus particulièrement à l'entreposage des open data qui sont un type de données du web en pleine croissance et qui ont la spécificité de contenir un nombre important d'informations utiles pour les décideurs notamment des données statistiques.

L'Open Data (ou données ouvertes) sont définies comme étant des données disponibles sous licence libre destinées à la réutilisation et à la redistribution par n'importe quelle personne (Coletta et al., 2012) (Mazón et al., 2012) (Eberius et al., 2012). Cependant elles ont d'autres propriétés telles qu'une importante hétérogénéité en format (XSL, CSV, RDF, TXT, PDF,...), en structure et en sémantique et couvrent plusieurs domaines (politique, santé, commerce ...).

Nous illustrons quelques problèmes des open data à travers les deux sources de données (datasets) de la Figure 1. Ces sources en format excel sont extraites du site data.gouv.fr. Elles montrent des statistiques sur les accidents de travail en France. Le premier dataset [1] décrit l'évolution des accidents de travail et de trajet pour le personnel civil y compris la gendarmerie

1. http ://www.data.gouv.fr/DataSet/30382535.

entre 2000 et 2010 alors que le deuxième dataset [2] concerne l'évolution des accidents de service et de trajet et les maladies professionnelles pour le personnel militaire entre 2001 et 2011.

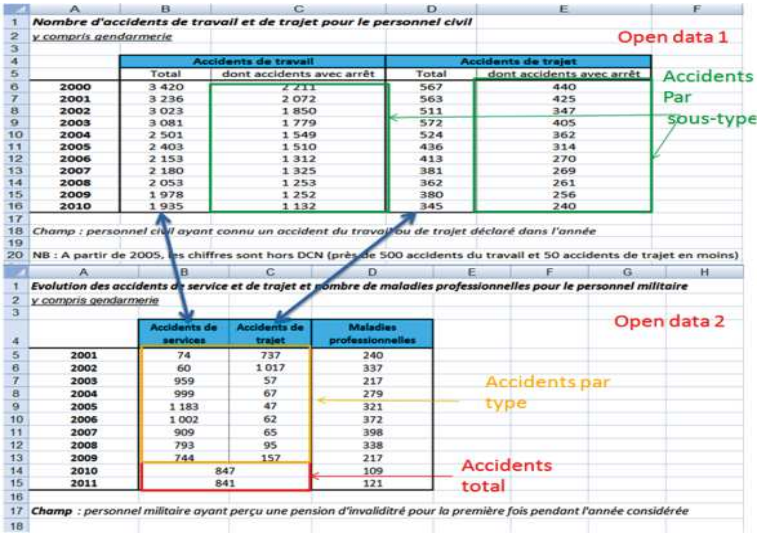

FIG. 1: Exemples d'open data.

Nous avons le problème d'hétérogénéité sémantique entre les deux termes "accidents de travail" et "accidents de services". Par ailleurs, nous remarquons la présence de plusieurs niveaux d'agrégation des données. En effet, dans le premier dataset l'évolution des accidents est comptée pour chaque type d'accident sur deux niveaux (total et sous-total). En revanche, pour le deuxième dataset nous avons le nombre total pour chaque type d'accident de 2001 à 2009 et pour 2010 et 2011 nous n'avons que la somme des accidents de service et de trajet. Nous remarquons aussi, pour le deuxième dataset, un problème de qualité de données. En fait le nombre d'accidents de 2001 et 2002 semblent avoir été inversés entre les deux colonnes accidents de service et de trajet.

Notre problématique consiste à définir une approche d'intégration suffisamment flexible pour prendre en compte les problèmes évoqués et pour pouvoir découvrir un schéma multidimensionnel des open data, et ceci, en automatisant autant que possible ce processus.

Le reste de l'article est organisé comme suit. Un état de l'art sur l'intégration des open data sera présenté dans la section 2. La section 3 illustre notre processus d'entreposage automatique d'open data basé sur les graphes. Nous expliquons, en s'appuyant sur des exemples, la construction des graphes des open data tout au long de la section 4, l'enrichissement des graphes dans la section 5 , l'intégration des graphes dans la section 6 et la définition du schéma multidimensionnel dans la section 7. La section 8 conclura avec les perspectives de ce travail.

2. http ://www.data.gouv.fr/DataSet/30382596.

2 Etat de l'art

Dans la littérature, les travaux ont porté essentiellement sur l'intégration des open data dans le nuage des Linked Open Data (LOD) (Bizer et al., 2009). La plupart de ces travaux s'inscrit dans un processus d'intégration en quatre phases : (1) sélection ou création d'ontologie, (2) conversion des données en RDF (RDF, 2013) en considérant l'ontologie choisie, (3) publication des données RDF en Linked Data (Bizer et al., 2009), (4) liaisons avec des sources externes en RDF et publication dans le LOD.

(Bohm et al., 2012) ont proposé un processus d'intégration en LOD des open data gouvernementales en se basant sur un schéma d'ontologie d'intégration. (Zapilko et Harth, 2011) ont enrichi et intégré dans le LOD des open data statistiques exprimés en RDF Data Cube Vocabulary (vocabulary, 2012). Néanmoins, ils supposent que les sources ont le même niveau d'agrégation de données ce qui restreint les possibilités d'intégration d'un grand nombre d'open data, notamment ceux de l'exemple présenté à la Figure 1. (Sabou et al., 2012) ont amélioré la sémantique du système TourMIS par transformation des données en open RDF en se guidant par une ontologie et en ajoutant des liens vers DBpedia. (Plu et Scharffe, 2012) ont proposé un workflow pour la publication et la liaison des open data du domaine de transport en LOD. Le problème commun entre ces travaux est que la phase d'intégration n'est pas automatisée et nécessite une intervention humaine importante. En effet, lors de la conversion des open data en RDF, il faut passer par des règles de mapping entre la source de données et la cible en RDF. Par ailleurs, l'intervention humaine est un passage obligatoire lors de la création ou le choix de l'ontologie servant de support pour l'intégration.

D'autre part, dans le cadre de ré-conciliation des données matricielles provenant du web nous citons les travaux de (Pivk et al., 2004) et (Sais, 2007). (Pivk et al., 2004) a proposé des heuristiques pour connaître la structure logique des données dans des tableaux HTML. Il génère des frames logiques lui permettant de peupler des ontologies sous-jacentes aux données. (Sais, 2007) a proposé de découvrir le schéma des tableaux XML en se guidant par un schéma global d'ontologie. Ces deux approches intéressantes permettent la découverte des schémas d'ontologie des données matricielles, toutefois notre focus est de découvrir un schéma multidimensionnel à travers la ré-conciliation des données.

Par ailleurs, la conception d'un entrepôt de données a été classée par (Rizzi et al., 2006) en trois approches : (a) approches "descendantes" (Prat et al., 2006) permettant de construire un schéma multidimensionnel en partant des besoins des utilisateurs, (b) approches "ascendantes" (Husemann et al., 2000) partant d'une analyse des sources de données pour la conception du schéma, (c) approches "mixtes" (Romero et Abello, 2010) (Zepda et al., 2008) permettant de concevoir un schéma d'entrepôt en partant à la fois des besoins des utilisateurs et des données.

Nous proposons, dans cet article, d'établir une démarche incrémentale mixte d'intégration automatique des open data qui vise à rapprocher le plus possible les open data ré-conciliés en vues multidimensionnelles. Notre démarche repose sur la visualisation et l'enrichissement des graphes d'open data. Ces derniers permettent :

- d'intégrer les différentes structures des open data suivant une description générique commune et simple ;
- de faciliter la maintenance et d'assurer une évolutivité de l'entrepôt de données puisque la structure en graphe est flexible pour les mises à jour et la maintenance ;

– d'aider le concepteur à la découverte et à la conception mixte de son schéma multidimensionnel par des détections automatiques d'éléments du graphe pouvant être transformés en sujets ou en axes d'analyse.

3 Processus d'entreposage automatique d'Open Data

Nous proposons, en Figure 2, l'architecture fonctionnelle de notre démarche incrémentale d'intégration d'open data en quatre phases :

1. Phase de construction automatique des graphes à partir des open data de structure matricielle. Cette phase a pour objectif de différencier automatiquement la structure de la zone de données. Les éléments structurels peuvent former les axes et les sujets d'analyse ;
2. Phase d'enrichissement automatique des graphes. Nous avons choisi la technique des Treillis de Galois (Birkho, 1967) pour ajouter des éléments de généralisation afin de se rapprocher d'une organisation multidimensionnelle des données. Les ontologies ou wordnet (Fellbaum, 1998) sont des techniques envisageables pour cette phase ;
3. Phase d'intégration automatique des graphes. Nous fusionnons de manière automatique les différents graphes enrichis en un graphe unique intégré ;
4. Phase de définition incrémentale et semi-automatique des composants multidimensionnels. L'objectif de cette phase est de construire un schéma multidimensionnel à partir duquel nous pouvons constituer les cubes multidimensionnels. Elle repose sur des interactions du décideur ou l'utilisateur avec le graphe intégré.

Dans la suite, nous expliquons plus en détails les différentes phases de l'architecture.

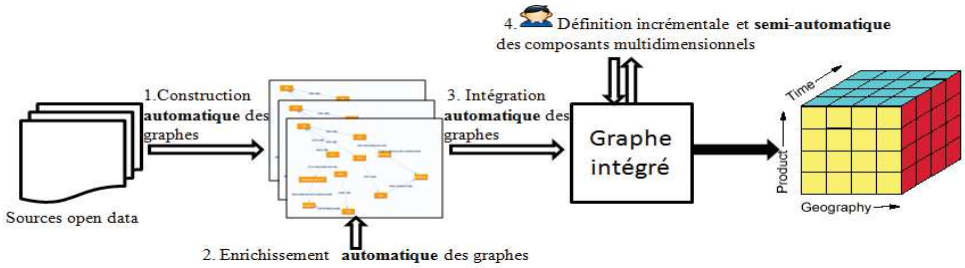

FIG. 2: Architecture fonctionnelle globale.

4 Construction automatique du graphe d'Open Data

Nous proposons de construire un graphe $G(V, E)$ à partir des open data de structure matricielle M. Le graphe contient la structure et les données afin d'aider les décideurs dans le choix des concepts pouvant former le schéma multidimensionnel.

Définition 1. Une source de données est représentée par une matrice $M = m_{(i,j)} \in [1, l] \times [1, c]$ tel que $m_{(i,j)} \in \{LABEL \cup NUMBER\}$.

Définition 2. Un graphe $G(V, E)$ décrit les relations entre les données numériques et les données structurelles ainsi que les éventuelles relations entre les données structurelles.

L'ensemble V des sommets est partitionné en deux sous-ensembles :

- $V_{LAB_{i,j}}, (i,j) \in [1,l] \times [1,c]$: l'ensemble des sommets représentant les données de type $LABEL \cup NUMBER$. Ils peuvent former des dimensions d'un cube ;
  - $V_{LAB_{LastLine,j}}, j \in [1,c]$: l'ensemble des sommets des labels qui délimitent la zone de données en ligne $LastLine$,
  - $V_{LAB_{i,LastCol}}, i \in [1,l]$: l'ensemble des sommets des labels qui délimitent la zone de données en colonne $LastCol$,
  - $V_{LAB_{i,j}} = V_{LAB} \backslash \{V_{LAB_{LastLine,m}} \cup V_{LAB_{n,LastCol}}\}$ *et* $(i,j) \in [1,l] \backslash LastLine \times [1,c] \backslash LastCol$.
- $V_{NBR_{i,j}}, (i,j) \in [1,l] \times [1,c]$: l'ensemble des sommets des données numériques de type $NUMBER$. Ces sommets peuvent former des cellules d'un cube.

L'ensemble E des arcs est partitionné en deux sous-ensembles :

- E_{Dim} : l'ensemble des arcs (u,v) représentant les liens entre les labels où $u \in V_{LAB_{i,j}}$ et $v \in V_{LAB_{i,j}}$;
- E_{Fact} : l'ensemble des arcs (u,v) représentant les liens entre la donnée numérique et les labels auxquels elle appartient où $u \in V_{NBR_{i,j}}$ et $v \in V_{LAB_{k,l}}$.

Description du processus. Les phases de transformation de la matrice M en graphe G :

1. Phase de détection de la zone de données numériques. Nous partitionnons les données de la matrice sur deux collections $Collection_{Label}$ et $Collection_{Number}$ selon leur type et leur emplacement par rapport à la zone de données numériques (ne contient que des données de type $NUMBER$ qui seront insérées dans la collection $Collection_{Number}$). Nous insérons les données de la ligne et de la colonne délimitant cette zone et le reste des données de la matrice dans la collection $Collection_{Label}$;
2. Phase de construction du graphe.
   (a) insérer dans $V_{LAB_{LastLine,j}}$,$V_{LAB_{i,LastCol}}$ et $V_{LAB_{i,j}}$ les sommets correspondants de la $Collection_{Label}$,
   (b) insérer dans E_{Dim} les arcs liants les sommets des sous-ensembles de $V_{LAB_{i,j}}$ si nous détectons des cellules fusionnées sur plusieurs lignes ou colonnes,
   (c) insérer dans $V_{NBR_{i,j}}$ les sommets de la $Collection_{Number}$,
   (d) insérer dans E_{Fact} les arcs liants les sommets de $V_{NBR_{i,j}}$ aux sommets labels $V_{LAB_{LastLine,j}}$ et $V_{LAB_{i,LastCol}}$ correspondants à la colonne i et à la ligne j.

Exemple. Le processus appliqué sur le premier dataset de la Figure 1 donne :

- Une zone de données numériques tel que $LastLine = 5$ et $LastCol = 1$;
- $Collection_{Label} = \{m(4,2), m(4,4), m(5,2), m(5,3), m(5,4), m(5,5), m(1,1), m(2,1), m(6,1), ..., m(16,1), m(18,1), m(20,1)\}$;
- $Collection_{Number} = \{m(6,2), ..., m(16,5)\}$;
- Les éléments du graphe correspondent à :
  - $V_{LAB_{LastLine,j}} = \{V_{LAB_{5,2}}, V_{LAB_{5,3}}, V_{LAB_{5,4}}, V_{LAB_{5,5}}\}$,
  - $V_{LAB_{i,LastCol}} = \{V_{LAB_{6,1}}, V_{LAB_{7,1}}, V_{LAB_{8,1}}, V_{LAB_{9,1}}, ..., V_{LAB_{16,1}}\}$,
  - $V_{LAB_{i,j}} = \{V_{LAB_{1,1}}, V_{LAB_{2,1}}, V_{LAB_{4,2}}, V_{LAB_{4,4}}, V_{LAB_{18,1}}, V_{LAB_{20,1}}\}$,

- $V_{NBR_{i,j}} = \{V_{NBR_{6,2}}..., V_{NBR_{16,5}}\}$,
- E_{Fact} contient par exemple $(V_{NBR_{6,2}}, V_{LAB_{6,1}})$ et $(V_{NBR_{6,2}}, V_{LAB_{5,2}})$,
- E_{Dim} contient par exemple $(V_{LAB_{5,2}}, V_{LAB_{4,2}})$ et $(V_{LAB_{5,3}}, V_{LAB_{4,2}})$.

Les figures des graphes dans cet article sont issues d'un prototype en cours de développement au sein de notre équipe. Les technologies utilisées sont : Java v.1.6, JGraph v.0.8.3, Oracle v.11g, eclipse v3.7.0. La Figure 3 illustre le graphe de l'exemple présenté ci-dessus. Dans la zone étiquetée par $V_{LAB_{LastLine,j}}$ et E_{Dim} , nous avons les sommets et les arcs de ces deux ensembles. L'exemple donné pour E_{Dim} correspond à l'arborescence ("accidents de trajet", "total", "dont accidents avec arrêt"). La zone étiquetée par $V_{NBR_{i,j}}$ et E_{Fact} contient les sommets des données numériques de $V_{NBR_{i,j}}$ ainsi que les arcs de E_{Fact}. Enfin, la zone étiquetée par $V_{LAB_{i,LastCol}}$ contient les nœuds des années.

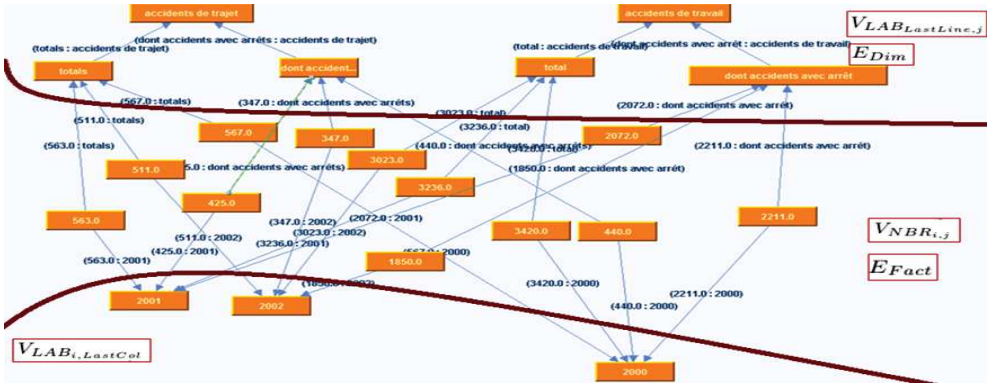

FIG. 3: Exemple de construction automatique de graphe.

5 Enrichissement automatique du graphe d'Open Data

Les graphes de la section 4 manquent de généralisation sur le domaine d'application des concepts identifiés. A titre d'exemple, il peut être intéressant de classifier les concepts "Accidents de service" et "Accidents de trajet" comme sous-concepts du concept "Accidents". Ceci permet en effet de construire des dimensions munies de hiérarchies plus profondes. Nous proposons alors l'enrichissement par généralisation des graphes d'open data en utilisant les Treillis de Galois (Birkho, 1967) qui est une technique de classification conceptuelle.

Définition 3. Un treillis de Galois représente un contexte formel $C = (O, A, I)$, tel que : O un ensemble fini d'objets, A un ensemble fini d'attributs et I une relation binaire entre O et A avec $I \subseteq O \times A$. Un contexte formel sera présenté par un tableau binaire $Objets \times Attributs$ selon que le couple (o, a) de $O \times A$ appartient à I ou non.
Le tableau 1 illustre un exemple de contexte formel pour des libellés du deuxième dataset de la Figure 1 où $O = \{accident\,de\,service, accident\,de\,trajet, maladie\,professionnelle\}$ et $A = \{accident, service, trajet, maladie, professionnelle\}$. Les ensembles sont obtenus en appliquant les phases 1(a) et 1(b). A partir du tableau 1, nous construi-

sons le diagramme de Hasse correspondant au treillis comme le montre la Figure 4.

	accident	service	trajet	maladie	professionnelle
accident de service	1	1	0	0	0
accident de trajet	1	0	1	0	0
maladie professionnelle	0	0	0	1	1

TAB. 1: Exemple de contexte formel.

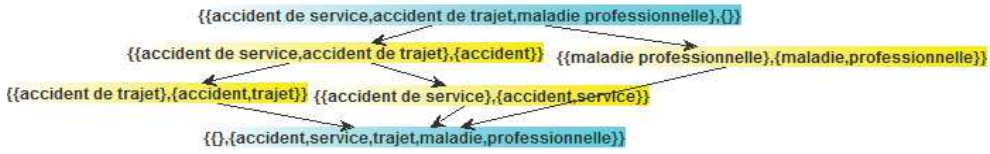

FIG. 4: Diagramme de Hasse du Treillis.

Description du processus. Le processus d'enrichissement de graphe est comme suit :

1. Nous construisons le contexte formel tel que :
   (a) $O = V_{LAB}$: les objets sont constitués à partir de l'ensemble des sommets du graphe de type V_{LAB},
   (b) $A = \{v \in V_{LAB}\}$: les attributs sont les mots v extraits à partir des sommets V_{LAB}. L'extraction des mots se fait par élimination des mots vides. Elle pourra être améliorée dans nos futurs travaux avec par exemple une lématisation des mots.
2. Nous calculons les concepts du treillis formant le diagramme de Hasse ;
3. Nous choisissons les concepts du treillis ayant un seul attribut et au moins deux objets tel que le concept $\{accident\,de\,service, accident\,de\,trajet \times accident\}$ de la Figure4 ;
4. Pour chaque concept choisi, nous enrichissons l'ensemble V_{LAB} du graphe avec un nouveau sommet correspondant à l'attribut et nous rajoutons dans l'ensemble des arcs V_{Dim} les arcs liants l'attribut avec les objets du concept.

Exemple. Enrichissement du graphe de la Figure3. Pour la phase 1, le contexte formel comporte : $O = \{accidents\,de\,travail, accidents\,de\,trajet, total, dont\,accidents\,avec\,arret, 2000, .., 2010\}$ et $A = \{accidents, travail, trajet, arret, 2000, .., 2010\}$.
Après avoir construit le treillis, nous enrichissons la zone étiquetée par $V_{LAB_{LastLine,j}}$ et E_{Dim} avec le concept{{accidents de travail, accidents de trajet, dont accidents avec arrêt},accidents} tout en gardant un seul parent par objet. Nous obtenons alors une généralisation des deux concepts "accidents de travail" et "accidents de trajet".

6 Intégration automatique des graphes

La phase d'intégration consiste à rassembler les différents graphes des sources dans un unique graphe. Toutefois nous envisageons dans nos prochains travaux de résoudre les problèmes d'hétérogénéité (sémantique, structurel,..) au niveau de cette phase. La Figure 5 montre

le graphe intégré des deux datasets de la Figure 1. Les nœuds encadrés et les arcs en gras proviennent de la deuxième source de données construits à partir des processus que nous avons présenté. Les autres éléments correspondent à la première source traitée dans les sections 4 et 5. Dans ce graphe le nœud "accidents de trajet" rassemble les données issues des deux sources avec différents niveaux de granularité. L'exploitation du graphe intégré par l'utilisateur sera plus simple que l'exploitation des sources de données.

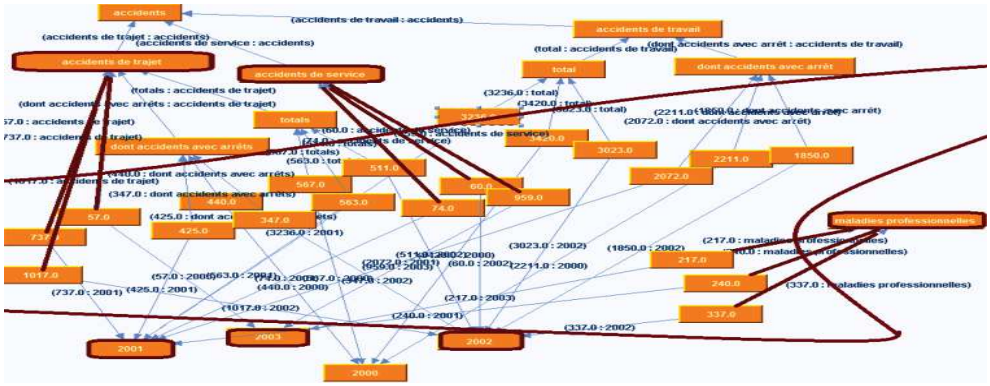

FIG. 5: Exemple d'intégration de graphes.

7 Définition incrémentale du schéma multidimensionnel

Cette phase permet de produire un schéma multidimensionnel (Golfarelli et al., 1998) en se basant sur des interactions de l'utilisateur avec la structure des sources de données. En effet, la décomposition des sommets du graphe en V_{LAB} et V_{NBR} nous permet de proposer à l'utilisateur uniquement la structure (les sommets V_{LAB}) pour la construction du schéma multidimensionnel. Nous proposons une interface en deux fenêtres comme le montre la Figure 6 : la fenêtre de gauche contient les sommets V_{LAB} et celle de droite contient le schéma multidimensionnel. L'utilisateur peut :(a) construire des éléments du schéma (dimensions, mesures, fait, ...) par des opérations (ajouter/supprimer/éditer/mettre à jour..),(b) déplacer des sommets du graphe dans les éléments du schéma multidimensionnel.

Exemple. L'utilisateur crée la dimension $DTEMPS$ et le niveau de dimension $ANNEE$ puis déplace dans ce dernier les sommets 2000,.., 2003. Par la suite, il crée une deuxième dimension $DCAUSES$ avec deux niveaux de dimension $TYPE\_ACC$ et $CLASS\_ACC$ puis déplace le sommet "avec arrêt de maladie" dans $TYPE\_ACC$ et les sommets "accidents de service", "accident de trajet" et "accident de travail" dans "$CLASS\_ACC$". Enfin, il crée le fait $FACCIDENTS$ et la mesure associé NBR.

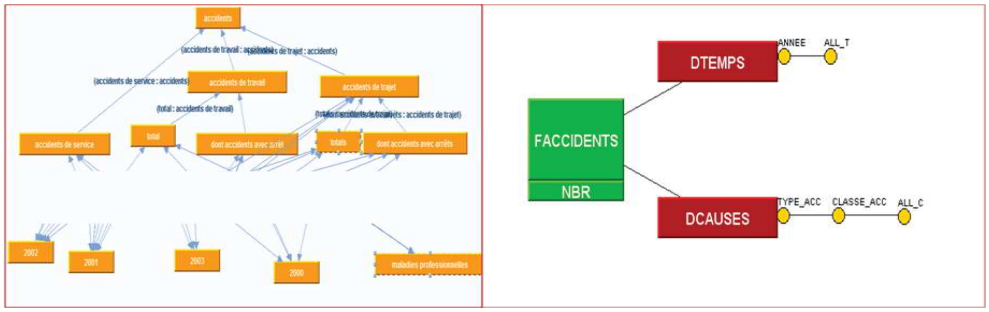

FIG. 6: Exemple d'un schéma multidimensionnel

8 Conclusion

Nous avons présenté une nouvelle approche incrémentale d'intégration des open data basée sur la construction et l'enrichissement automatique des graphes représentatifs des sources open data. Ces graphes permettent : (a) une intégration simplifiée et plus flexible de la structure et les données des sources, (b) une facilité de l'intervention de l'utilisateur dans la phase de construction du schéma multidimensionnel. Nous envisageons une généralisation de l'heuristique de construction du graphe pour les données matricielles à N dimensions ainsi qu'une amélioration de la phase d'enrichissement des graphes par des entités spatio-temporelles. Nous avons aussi comme perspectives de résoudre les problèmes d'hétérogénéité et de définir un langage OLAP d'interrogation des graphes.

Références

Birkho, G. (1967). *Lattice Theory* (3rd edition, 1967. 1st ed., 1940 ; 2nd ed., 1948 ed.), Volume 25. ASM Colloquium Publications.

Bizer, C., T. Heath, et T. Berners-Lee (2009). Linked data - the story so far. In *Int. J. Semantic Web Inf. Syst.*, pp. 1–22.

Bohm, C., M. Freitag, A. Heise, C. Lehmann, A. Mascher, F. Naumann, V. Ercegovac, M. Hernández, P. Haase, et M. Schmidt (2012). GovWILD : integrating open government data for transparency. In *WWW (Companion Volume)*, pp. 321–324.

Coletta, R., E. Castanier, P. Valduriez, C. Frisch, D. Ngo, et Z. Bellahsene (2012). Public data integration with WebSmatch. In *CoRR*.

Eberius, J., M. Thiele, K. Braunschweig, et W. Lehner (2012). DrillBeyond : enabling business analysts to explore the web of open data. In *PVLDB*, pp. 1978–1981.

Fellbaum, C. (1998). *WordNet : An Electronic Lexical Database*. Bradford Books.

Golfarelli, M., D. Maio, et S. Rizzi (1998). Conceptual design of data warehouses from e/r schemes. In *International Conference on Systems Science, IEEE*, Hawaii, pp. 166–181.

Husemann, B., J. Lechtenborger, et J. Vossen (2000). Conceptual data warehouse modelling. In *2 nd international workshop on Design and Management of Data Warehouse*, pp. 6.1–6.11.

Ma, Y., B. Xu, Y. Bai, et Z. Li (2011). Building linked open university data : Tsinghua university open data as a showcase. In *JIST*, pp. 385–393.

Mazón, J., J. Zubcoff, I. Garrigós, R. Espinosa, et R. Rodríguez (2012). Open business intelligence : on the importance of data quality awareness in user-friendly data mining. In *EDBT/ICDT Workshops*, pp. 144–147.

NER, S. (2013). Stanford.

Pivk, A., P. Cimiano, et y. Sure (2004). From tables to frames. In *International Semantic Web Conference*, pp. 166–181.

Plu, J. et F. Scharffe (2012). Publishing and linking transport data on the web. In *CoRR*.

Prat, N., J. Akoka, et I. Comyn-Wattiau (2006). A uml-based data warehouse design method. In *Decision Support Systems*, Volume 42(3), pp. 1449–1473.

Ravat, F., O. Teste, R. Tournier, et G. Zurfluh (2010). Finding an application-appropriate model for xml data warehouses. In *Information Systems Journal, Elsevier Science Publisher*, Volume 36(6), pp. 662–687.

RDF, W. (2013). W3c working draft.

Rizzi, S., A. Abelló, J. Lechtenbörger, et J. Trujillo (2006). Research in data warehouse modeling and design : dead or alive ? In *DOLAP*, pp. 3–10.

Romero, O. et A. Abello (2010). Automatic validation of requirements to support multidimensional design. In *Data & Knowledge Engineering*, Volume 69(9), pp. 917–9427.

Sabou, M., A. Brasoveanu, et I. Arsal (2012). Supporting tourism decision making with linked data. In *I-SEMANTICS*, pp. 201–204.

Sais, F. (2007). *Intégration sémantique des données guidée par une ontologie*. Thèse de doctorat, Université Paris-Sud.

vocabulary, W. T. R. D. C. (2012). W3c working draft.

Zapilko, B. et A. Harth (2011). Enriching and analysing statistics with linked open data. In *Eurostat (Hrsg.) NTTS*. S8 Papaer 1, Brüssel.

Zepda, L., L. Celma, et R. Zatarain (2008). Mixed approach for data warahouse conceptual design with mda. In *International Conference on Computational Science and Its Applications*, pp. 1204–1217.

Summary

In spite of their emergence, Open Data are characterized by an important heterogeneousness and a scattering on Web what makes difficult their integration within a datawarehouse. The current works propose not automated processes of integration based on Linked Open Data. In this paper, we propose a process to automate the multidimensional datawarehousing of Open Data. Our approach is based on the transformation of Open Data in a generic graph favoring their integration and serves as support for the semi-automatic and incremental definition of the multidimensional schema.