Fouille de données dans les bases relationnelles pour l'acquisition d'ontologies riches en hiérarchies de classes

Farid Cerbah*

*Dassault Aviation
Département des études scientifiques
78, quai Marcel Dassault 92552 Saint-Cloud Cedex
farid.cerbah@dassault-aviation.fr

Résumé. De par leur caractère structuré, les bases de données relationnelles sont des sources précieuses pour la construction automatisée d'ontologies. Cependant, une limite persistante des approches existantes est la production d'ontologies de structure calquée sur celles des schémas relationnels sources. Dans cet article, nous décrivons la méthode RTAXON dont la particularité est d'identifier des motifs de catégorisation dans les données afin de produire des ontologies plus structurées, riches en hiérarchies. La méthode formalisée combine analyse classique du schéma relationnel et fouille des données pour l'identification de structures hiérarchiques.

1 Introduction

Dans les entreprises qui ont à produire et à gérer des données techniques très spécialisées pour la définition de produits complexes, comme dans les secteurs de l'aéronautique et de l'automobile, les entrepôts de données reposent pour une large part sur des bases de données relationnelles. Du fait de leur caractère structuré, ces entrepôts sont des sources à privilégier dans les processus de construction d'ontologies. Cependant, entreprendre un travail d'acquisition d'ontologies à partir de telles sources de données sans disposer d'une aide logicielle adaptée peut s'avérer très vite rédhibitoire.

La thématique d'acquisition d'ontologies à partir de bases de données relationnelles n'est pas nouvelle. Plusieurs méthodes et outils ont été développés pour tirer parti de ces données structurées, avec souvent pour objectif d'assurer l'intégration de bases de données hétérogènes. Cependant, on constate qu'une limite persistante des méthodes proposées est la dérivation d'ontologies de structure calquée sur les schémas des bases de données sources. Ces résultats peuvent difficilement convaincre des utilisateurs attirés par le pouvoir d'expression des formalismes du web sémantique et qui ne peuvent se satisfaire « d'entrepôts sémantiques » ressemblant fortement à leurs bases de données relationnelles. Une attente légitime est d'obtenir en retour des modèles qui rendent mieux compte de la structure conceptuelle sous-jacente aux données stockées.

La dérivation d'ontologies faiblement structurées est le propre des méthodes qui se contentent d'exploiter les méta-données définies dans les schémas sans examiner les données. Une analyse même sommaire de bases de données existantes montre que des motifs de catégorisation

complémentaires peuvent être identifiés avec robustesse dans les données pour affiner de manière significative la structure de l'ontologie-cible. Plus précisément, des hiérarchies de classes peuvent être induites des données pour spécialiser des classes dérivées du schéma relationnel. C'est l'idée-clé qui est exploitée dans la définition de la méthode RTAXON décrite dans cette article.

Dans la suite de cet article, nous commencerons par introduire la problématique en déroulant le processus de transformation sur un exemple représentatif. Nous ferons ensuite un survol de l'état de l'art en nous focalisant sur les méthodes d'acquisition qui traitent spécifiquement de bases de données relationnelles. En section 4, nous décrivons en détail la méthode RTAXON qui est implémentée dans le logiciel RDBToOnto[1]. La section 5 est consacrée à l'évaluation et la conclusion donne quelques pistes de prolongement de ces travaux.

2 Un exemple représentatif

La problématique peut être introduite à travers un exemple. La figure 1 illustre les données d'entrée et le résultat du processus de transformation. Les dérivations appliquées pour obtenir l'ontologie cible peuvent être réparties en deux groupes. Les dérivations du groupe **(a)** résultent de l'exploitation de meta-données issues du schéma de la base. A chaque définition de relation correspond une classe de l'ontologie. Ces mises en correspondance simples de relations à classes sont souvent pertinentes, même si plusieurs exceptions sont à prendre en compte. Pour compléter la définition des classes, certains attributs de relation sont directement traduits en attributs de classes (datatype properties). Les clés externes (liens entre relations dans le schéma) constituent la source la plus fiable pour introduire des associations entre classes (object properties) et, dans cet exemple, chacune des quatre clés est à l'origine d'une association dans l'ontologie. Les dérivations appliquées pour obtenir cette partie haute de l'ontologie sont bien couvertes par les méthodes existantes et la plupart d'entre elles, appliquées sur ce cas, produiraient cette structure calquée sur le schéma comme résultat final. Cependant, l'exploration des données fait apparaître la possibilité d'affiner la structure de l'ontologie. En particulier, les dérivations du groupe **(b)** montrent comment la classe Produit peut être affiner en sous-classes en exploitant les données issues de la colonne Catégorie de la relation Produits. De façon similaire, la classe Fournisseurs peut être étendue par une hiérarchie à deux niveaux en combinant les valeurs des colonnes Pays et Ville de la relation source.

Ces exemples illustrent des motifs de catégorisation très largement employés par les concepteurs de bases de données et pour lesquels des procédés d'identification robustes peuvent être élaborés.

3 Travaux connexes

L'acquisition automatique d'ontologies à partir de bases de données est une problématique de recherche relativement récente. Cependant, elle s'inscrit en continuité de la problématique de rétro-ingénierie de modèles relationnels. L'objectif des travaux menés dans ce cadre était d'extraire des modèles conceptuels à partir de modèles relationnels. Une part importante des

[1] www.tao-project.eu/researchanddevelopment/demosanddownloads/RDBToOnto.html

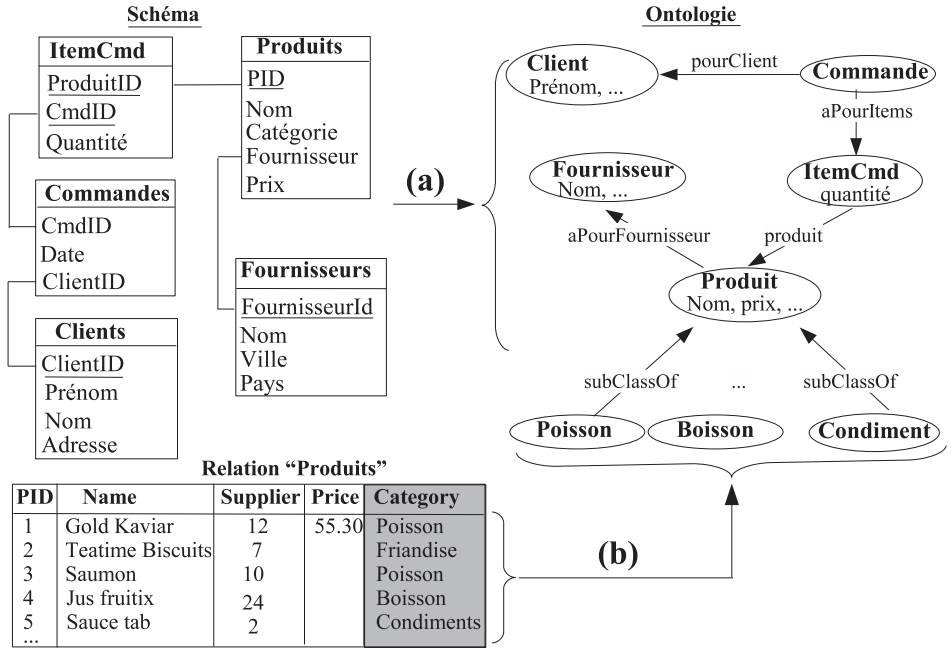

FIG. 1 – *Un exemple de dérivation d'un modèle d'ontologie par exploitation du schéma et du contenu d'une base relationnelle*

règles de transformation définies dans ce domaine précurseur restent pertinentes dans une perspective de construction d'ontologies et on retrouve les règles les plus fiables dans plusieurs approches ayant des ontologies pour cibles (Stojanovic et al. (2002); Astrova (2004); Li et al. (2005)).

La plupart des approches, aussi bien en rétro-ingénierie de modèles que de construction d'ontologies, se contentent d'exploiter les méta-données issues du schéma relationnel. On note toutefois quelques tentatives visant à exploiter aussi les données. L'identification de corrélations entre tuples est envisagée dans (Tari et al. (1998); Astrova (2004)) mais en ne considérant que les valeurs de clés (i.e. identifiant de tuples). Les rapports d'inclusion entre clés peuvent révéler des relations d'héritage. Il faut souligner qu'en pratique les règles basées sur les corrélations entre clés sont peu productives car les motifs sous-jacents ne sont rencontrés que dans les bases de données dont la conception a été optimisée.

L'approche proposée dans Lammari et al. (2007) et celle que nous définissons dans cet article se rejoignent dans la volonté d'aller plus loin dans l'exploration des données. Dans les deux approches, le processus de fouille n'est pas restreint aux seules valeurs de clés. Lammari et al. (2007) s'appuie sur une interprétation précise de la sémantique des valeurs nulles. D'une certaine manière manière, il s'agit ici de profiter des défauts de modélisation qui résultent de l'absence de constructions dédiées à l'héritage de concepts dans le modèle relationnel. Typiquement, lorsque toutes les instances (tuples) d'un concept complexe sont rassemblées dans

une même relation, certains attributs peuvent s'avérer n'être pertinents que pour certains sous-concepts et dès lors n'être renseignés que pour les instances de ces sous-concepts et laisser vides pour les autres. Par exemple, dans une relation Employés sensée contenir tout l'effectif d'une compagnie aérienne, les attributs Heures de vols et Numéro de licence ne seraient renseignés que pour les entrées correspondant à des membres du personnel naviguant. Un partitionnement de la relation sur la base des valeurs nulles présentes dans ces attributs permettrait de rendre explicite la structure hiérarchique sous-jacente.

La définition de techniques de mise en correspondance entre bases relationnelles et ontologies (Bizer (2003); Barrasa et al. (2004); Auer et al. (2009)) est une thématique apparentée. Le but est de proposer des moyens déclaratifs pour associer des modèles relationnels à des ontologies pré-définies et d'offrir des procédés d'instanciation à la volée des ontologies à partir de données extraites de bases susceptibles de subir des mises a jour.

4 Combiner analyse du schéma relationnel et fouille de données

Notre motivation première dans la conception de la méthode RTAXON est de combiner les règles les plus robustes exploitant le schéma avec un procédé de fouille, également soumis à de fortes exigences de robustesse, et ciblé sur l'identification de hiérarchies de concepts.

Dans cette partie centrale de l'article, nous commencons par donner quelques définitions préliminaires. Nous présentons ensuite les étapes du processus d'acquisition (section 4.2) avant de nous focaliser sur la phase de fouille des données (section 4.3).

4.1 Définitions préliminaires

Un schéma de base de données relationnelle D est un ensemble fini de schémas de relation $D = \{R_1, \ldots, R_n\}$ où chaque R_i est caractérisé par un ensemble d'attributs $\{A_{i1}, \ldots, A_{im}\}$. La fonction $pkey$ associe à chaque relation sa clé primaire qui est un ensemble d'attributs $K \subseteq R$. Une relation r sur un schéma R (i.e. une *instance* de R) est un ensemble de tuples (séquences de $|R|$ valeurs). De manière similaire, une base de données d sur D est définie par un ensemble de relations $d = \{r_1, \ldots, r_n\}$. Par convention, nous considérons que lorsqu'une lettre capitale désigne un schéma de relation, la lettre minuscule correspondante désigne une instance de ce schéma. Nous avons recours à la notion de *dépendance d'inclusion* (par ex., de Marchi (2003)) pour rendre compte des corrélations entre relations. Une dépendance d'inclusion est une expression de la forme $R[X] \subseteq S[Y]$ où X et Y sont respectivement des séquences d'attributs des schémas de relation R et S, et satisfaisant la contrainte $|X| = |Y|$. La dépendance est établie entre deux instances de r et s si à chaque tuple u de r, on peut associer un tuple v de s tel que $u[X] = v[Y]$.

Les clés externes (ou étrangères) peuvent être vues comme une forme particulière de dépendances d'inclusion où $Y = pkey(S)$. Nous utiliserons la notation $R[X] \subseteq S[pkey(S)]$ pour les dépendances de ce type.

Les éléments d'ontologies seront exprimés dans la syntaxe dite abstraite du formalisme OWL (Patel-Shneider et al. (2004)).

4.2 Le processus global

Le processus de transformation défini ici est constitué d'étapes automatisées.

Les principales étapes sont : la normalisation de la base, l'identification des classes et des relations inter-classes et l'instanciation de l'ontologie.

Il convient de souligner que certaines phases de traitements impliquées dans la mise en œuvre de cette méthode sollicitent des moyens réutilisables de la plate-forme RDBToOnto (cf. Cerbah (2008) et site dédié sur la Toile) et qui peuvent être exploités pour implémenter d'autres méthodes. RDBToOnto automatise l'intégralité du processus de transformation, de l'extraction des informations dans les bases de données à la génération des ontologies. Plus précisément, la réutilisation concerne l'étape de normalisation décrite ci-après, un certain nombre de traitements récurrents dans la manipulation des sources de données et des ontologies ainsi que les moyens interactifs de configuration et de contrôle du processus.

4.2.1 Normalisation de la base

Dans les méthodes et outils existants, cette étape n'est pas intégrée dans le processus d'acquisition. Il est habituel de considérer que la base d'entrée est sous une forme normalisée (2NF ou 3NF). Ce choix part de l'hypothèse que le processus de transformation peut être aisément étendu en y incorporant une étape de normalisation. Si cette hypothèse est en théorie acceptable, elle est plus contestable sur le plan pratique, à fortiori dans une démarche expérimentale. On observe en effet que nombre de bases de données potentiellement exploitables pour la construction d'ontologies requièrent un effort important de normalisation pour constituer des sources acceptables. En particulier, les problèmes de redondance sont fréquents. Les cas de duplication de données entre relations peuvent être formalisées par des dépendances d'inclusion. La redondance liée à ces duplications est éliminée en transformant toutes les dépendances d'inclusion en dépendances fondées sur les clés externes. Plus formellement, toute dépendance attestée $R[X] \subseteq S[Y]$ telle que $Y \neq pkey(S)$ est remplacée par la dépendance $R[A] \subseteq S[pkey(S)]$ où A est un nouvel attribut désignant une clé externe. Les attributs de X ainsi que les données associées (redondantes) dans r sont supprimés.

A noter que dans le processus global mis en oeuvre dans RDBToOnto, cette étape préliminaire n'est que partiellement automatisée. Les dépendances d'inclusion sont définies manuellement alors leur interprétation pour transformer la base est automatisée.

4.2.2 Identification des classes et des propriétés

Cette étape est centrale dans le processus d'acquisition. Les relations de la base sont explorées pour dériver des éléments de définition de l'ontologie. Le schéma est la première source exploitée à travers l'application de règles traduisant des correspondances typiques entre motifs de schéma et éléments d'ontologies, impliquant en particulier la définition de classes, de propriétés élémentaires (datatype properties) et d'associations inter-classes (object properties). Nous donnons dans le tableau 1 la définition de trois des règles les plus fiables que l'on retrouve dans plusieurs approches de rétro-ingénierie et d'apprentissage d'ontologies. La première règle, triviale, indique que toute relation peut potentiellement donner lieu à l'introduction d'une classe (même si la relation peut être consommée par des règles plus spécifiques comme la troisième règle). La seconde règle exprime également une correspondance simple

Relation ⟶ Classe

Source	Préconditions	Cible
$R \in D$	$\neg\, \exists\, C \mid R = source(C)$	class($\boldsymbol{C}_R$)

Dépendance sur clé ⟶ Association inter-classes fonctionnelle

Source	Préconditions	Cible
		ObjectProperty($\boldsymbol{P}_A$
$R_0[A] \subseteq R_1[pkey(R_1)]$	$R_0 = source(C_0)$	domain(C_0)
	$R_1 = source(C_1)$	range(C_1)
		Functional)

Relation à clé primaire composite ⟶ Association inter-classes

Source	Préconditions	Cible
$R_0 \in D$		
$\lvert R_0 \rvert = 2$	$R_1 = source(C_1)$	ObjectProperty($\boldsymbol{P}_R$
$pkey(R_0) = \{K_1, K_2\}$	$R_2 = source(C_2)$	domain(C_1)
$R_0[K_1] \subseteq R_1[pkey(R_1)]$		range(C_2))
$R_0[K_2] \subseteq R_2[pkey(R_2)]$		

TAB. 1 – *Trois règles fiables exploitant des motifs identifiés dans le schéma. Dans la partie cible de la règle, la variable en caractère gras désigne l'identifiant de l'élément définitoire ajouté dans l'ontologie. Le prédicat «* **source** *» est employé pour fournir des informations de traçabilité utiles pour contrôler le processus.*

où une dépendance sur clé externe est traduite par une propriété inter-classes fonctionnelle. La troisième règle s'appuie sur un motif plus complexe définissant une relation dont la clé primaire est composée de deux clés externes. Les relations conformes à ce motif se traduisent naturellement en propriétés à cardinalité multiple.

La seconde source exploitée pour construire la structure de classes, essentielle dans notre approche, est le contenu de la base. Les données stockées sont explorées pour identifier des sous-classes permettant de spécialiser les classes qui résultent de l'application préalable des règles exploitant les motifs de schéma. Cette partie, qui constitue le coeur de notre contribution, est décrite en section 4.3.

4.2.3 Instanciation de l'ontologie

La dernière étape consiste à créer les instances de classes et de propriétés à partir des données. Pour une classe donnée, une instance est dérivée de chaque tuple de la relation source. De plus, si la classe a été spécialisée, les instances dérivées des tuples doivent aussi être réparties dans les sous-classes (cf. section 4.3.2).

4.3 Identification de hiérarchies dans les données

Notre exemple introductif (cf. section 2) nous a permis de présenter de manière informelle des motifs de catégorisation utilisés dans les bases de données pour pallier tant bien que mal les limites du modèle relationnel en matière de structuration de données. Les motifs les plus fréquents consistent à doter les relations d'un ou plusieurs attributs dédiés spécifiquement à la structuration des tuples. Ces *attributs catégorisants* sont particulièrement intéressants dans une optique de construction d'ontologies dans la mesure où des hiérarchies peuvent être induites du contenu de ces attributs.

Notre méthode d'identification de hiérarchies est focalisée sur l'exploitation de motifs basés sur ces attributs catégorisants. Même si cette restriction nous permet de viser un bon niveau de robustesse, la tâche de repérage de ces motifs reste difficile.

Nous décrivons ci-après la phase de repérage des motifs puis la génération des hiérarchies à partir des motifs identifiés.

4.3.1 Repérage des attributs catégorisants

Deux types d'informations sont impliqués dans cette tâche : les noms des attributs et la diversité dans l'extension des attributs (i.e. dans les colonnes). Ces deux sources fournissent des indicateurs permettant d'identifier avec une certaine fiabilité les attributs candidats et de sélectionner le plus plausible.

- Identification d'indicateurs lexicaux dans les noms d'attributs

Lorsqu'ils sont introduits à des fins de catégorisation, les attributs portent souvent des noms révélateurs. Dans l'exemple de la figure 1, l'attribut catégorisant dans la relation Produits est clairement identifié par son nom (Catégorie). La marque lexicale révélant le rôle de l'attribut peut être une partie d'un nom composé ou d'une forme abrégée (ex : Type Avion, CatId).
La méthode de filtrage lexicale que nous avons mise en oeuvre est relativement simple. Elle repose sur une liste prédéfinie d'indicateurs lexicaux. L'identification des indicateurs dans les noms implique une segmentation fondée sur une analyse des formes les plus fréquentes de formation de noms d'attributs.

- Filtrage par estimation de la diversité des données

Avec une liste étendue d'indicateurs lexicaux, la première phase de filtrage s'avère efficace. Cependant, nos expérimentations montrent que sur des bases de données complexes, il résulte souvent de cette étape plusieurs candidats. Pour les départager, nous introduisons une étape de filtrage complémentaire fondée sur une estimation de la diversité dans les extensions des attributs candidats. Notre hypothèse est qu'un bon candidat doit exhiber un niveau caractéristique de redondance qui peut être approché formellement par la notion d'entropie issue de la théorie de l'information.

L'entropie est une mesure de l'incertitude d'une source de données. Dans notre contexte, les attributs dont le contenu est très répétitif se caractérisent par une entropie faible. Inversement, parmi les attributs d'une même relation, la clé primaire possède l'entropie la plus élevée puisque toutes les valeurs stockées sont distinctes.
De manière informelle, le principe qui sous-tend cette seconde phase de sélection est de fa-

voriser le candidat qui imposerait la répartition la mieux équilibrée des instances dans les sous-classes.

Nous donnons ci-après une description formelle de cette étape. Pour un attribut donné A d'une relation de schéma R instanciée par une relation r, la diversité de A est estimée par :

$$H(A) = - \sum_{v \in \pi_A(r)} P_A(v) \; . \; \log P_A(v) \tag{1}$$

$$P_A(v) = \frac{|\sigma_{A=v}(r)|}{|r|} \tag{2}$$

- $\pi_A(r)$ est la *projection* de r sur A définie par $\pi_A(r) = \{t[A] \mid t \in r\}$. Cet ensemble représente le *domaine actif* de A. Autrement dit, $\pi_A(r)$ est l'ensemble des valeurs attestées dans l'extension de A. Chaque valeur v de l'ensemble $\pi_A(r)$ est une catégorie potentielle (et donc une sous-classe potentielle de l'ontologie-cible).

- $\sigma_{A=v}(r)$ est une sélection sur r définie par $\sigma_{A=v}(r) = \{t \in r \mid t[A] = v\}$. Cette sélection extrait de la relation r le sous-ensemble de tuples dont l'attribut A a pour valeur v. Dans ce contexte précis, la sélection extrait de la relation toutes les entrées auxquelles la catégorie potentielle v a été assignée.

- $P_A(v)$ est la probabilité qu'un tuple de r ait la valeur v assignée à l'attribut A. Ce paramètre rend compte du poids de v dans A. Il peut être estimé par la fréquence de v dans l'extension de A.

Soit $C \in R$ le sous-ensemble des attributs pré-sélectionnés à l'aide des indicateurs lexicaux. Un premier élagage est appliqué pour exclure les candidats dont l'entropie est située aux marges :

$$C' = \{ \; A \in C \mid H(A) \in [\; \alpha, \; H_{max}(R) \; . \; (1 - \beta) \;] \; \} \tag{3}$$

- $H_{max}(R)$ désigne l'entropie la plus élevée dans l'ensemble des attributs de la relation ($H_{max}(R) = \max_{A \in R} H(A)$)

- α et β sont des paramètres d'entrée ($\alpha, \beta \in [0, 1]$).

Comme précisé précédemment, $H_{max}(R)$ est souvent l'entropie de l'attribut jouant le rôle de clé primaire.

Si, après cet élagage, plusieurs candidats sont encore en lice[2], nous sélectionnons en définitive l'attribut catégorisant qui produira l'organisation la mieux équilibrée des instances. Cela revient à sélectionner l'attribut dont l'entropie est la plus proche de l'entropie maximale pour le nombre de catégories potentielles en jeu :

$$\tilde{H}_{max}(A) = - \log \frac{1}{|\pi_A(r)|} \tag{4}$$

[2] A noter que l'élagage peut exclure tous les candidats. Dans ce cas, le choix se porte arbitrairement sur l'un d'entre eux.

Cette valeur de référence, dérivée de l'expression (1), est représentative d'une structure *parfaitement* équilibrée de $|\pi_A(r)|$ catégories avec le même nombre d'éléments dans chaque catégorie. Précisons que cette valeur est indépendante du nombre total d'éléments ($|r|$).

La décision finale consiste à sélectionner dans C' l'attribut A^* dont l'entropie est la plus proche de cette entropie maximale :

$$A^* = arg \min_{A \in C'} \delta(A) \tag{5}$$

où

$$\delta(A) = \frac{|H(A) - \tilde{H}_{max}(A)|}{\tilde{H}_{max}(A)} \tag{6}$$

4.3.2 Génération et instanciation de l'ontologie

Comme le montre la première règle de la table 2, le passage des valeurs de l'attribut catégorisant aux sous-classes peut être assez direct. Une sous-classe est ici déduite de chaque élément du domaine actif de l'attribut. Toutefois, des mises en correspondance plus complexes peuvent s'imposer. La seconde règle s'appuie notamment sur un motif de catégorisation où les catégories à exploiter pour produire les classes sont définies dans une autre relation. On retrouve ce motif plus complexe dans de nombreuses bases de données. Nous en donnons un exemple d'application en figure 2. Dans cet exemple, l'attribut catégorisant CatId dans la relation Albums est associé via une clé externe à la relation Catégories où toutes les catégories autorisées sont recensées. Des appellations de classes plus appropriées peuvent être assignées en exploitant les valeurs du second attribut Description de la relation Catégories plutôt que de reprendre les valeurs de clés de type numérique. Qui plus est, une hiérarchie plus complète peut être obtenue en considérant aussi les catégories auxquelles aucun tuple n'a été associé dans la relation Albums. C'est le cas dans cet exemple de la catégorie Tango.

Les classes de la hiérarchie sont instanciées en exploitant les tuples issus de la même relation source. Une instance est déduite de chaque tuple. Par ailleurs, la tâche complémentaire de répartition des instances dans les sous-classes est basée sur un partitionnement de la relation relativement aux valeurs de l'attribut catégorisant.
Formellement, pour chaque valeur v de $\pi_{A^*}(r)$, les instances de la classe correspondante sont dérivées des tuples du sous-ensemble $\sigma_{A^*=v}(r) = \{t \in r \mid t[A] = v\}$.

5 Evaluation

RTAXON a été évaluée sur un ensemble de 38 bases de données de différents domaines. Une soixantaine d'attributs catégorisants ont été recensés dans ces bases. Sur ce corpus d'expérimentation, la procédure d'identification des attributs catégorisants produit des résultats d'une précision de 65% et d'un rappel de 60%. Dans 30% des cas, l'étape de sélection exploitant les indicateurs lexicaux n'est pas suffisamment discriminante. La résolution des conflits entre les candidats restants est assurée en invoquant l'étape complémentaire de filtrage fondée sur l'estimation de la diversité dans les données. Une précision de 59% a été atteinte par cette étape de résolution de conflits.

Valeurs d'attribut catégorisant ⟶ Sous-classes

Source	Préconditions	Cible
$r \in d$	$R = source(C)$	$\forall v \in \pi_A(r)$
$A = catAtt(r)$		class($\mathbf{C}_v$ partial C)

Valeurs (indirectes) d'attribut catégorisant ⟶ Sous-classes

Source	Préconditions	Cible
$r \in d$		
$A = catAtt(r)$		
$R[A] \subseteq S[pkey(S)]$	$R = source(C)$	$\forall v \in \pi_{B_1}(r)$
$pkey(S) = \{B_0\}$		class($\mathbf{C}_v$ partial C)
$S = \{B_0, B_1\}$		
$\lvert\pi_{B_0}(r)\rvert = \lvert\pi_{B_1}(r)\rvert$		

TAB. 2 – *Règles complexes pour la génération de hiérarchies de classes fondée sur l'identification d'attributs catégorisants ($A = catAtt(r)$). Dans la partie cible de la règle, la variable en caractère gras désigne l'identifiant de l'élément définitoire ajouté dans l'ontologie*

Pour mieux évaluer la pertinence de la sélection fondée sur l'entropie, nous avons expérimenté trois autres modes de sélection. Les deux premiers modes, très simples, consistent à sélectionner l'attribut qui a le domaine actif de plus petite ou de plus grande cardinalité. Le troisième mode consiste à rechercher l'attribut dont le domaine actif a une cardinalité proche de la moyenne des cardinalités sur l'ensemble des attributs candidats (i.e. recherche d'un attribut qui a un nombre « intermédiaire » de valeurs distinctes). Dans les trois cas, nous commençons par éliminer les candidats de cardinalité à valeur marginale, ce qui conduit à exclure les colonnes dont toutes les valeurs sont identiques ou, inversement, toutes distinctes (comme les attributs de clés primaires).

Nous obtenons une meilleure performance globale avec notre méthode de sélection qui implique une caractérisation plus fine de la diversité informationnelle des attributs candidats. Toutefois, si les deux premiers modes élémentaires sont nettement moins performants, l'écart avec le troisième est moins significatif et, dans un nombre non négligeable de cas, la moyenne des cardinalités s'avère être une meilleure valeur de référence pour situer l'attribut catégorisant. Dans la dernière étape de sélection, la maximisation de l'entropie a pour effet de privilégier l'équilibre de la structure tout en favorisant les attributs avec un plus grand nombre de classes potentielles[3], mais ce critère, aussi pertinent soit-il, reste partiel. La présence d'un nombre « intermédiaire » de catégories est aussi un critère plausible, bien que difficile à caractériser. Nous envisageons de définir et d'expérimenter une solution combinant ces deux critères.

La méthode RTAXON et le logiciel RDBToOnto dans lequel elle est implémentée ont été utilisés dans le cadre d'un projet relativement conséquent de migration de données techniques vers un référentiel sémantique à base d'ontologies. Dans ce projet, la source principale est un entrepôt de données du domaine de la maintenance aéronautique comportant plusieurs dizaines

[3] Rappelons toutefois que l'effet de maximisation du nombre de classes est atténué par la première phase d'élagage.

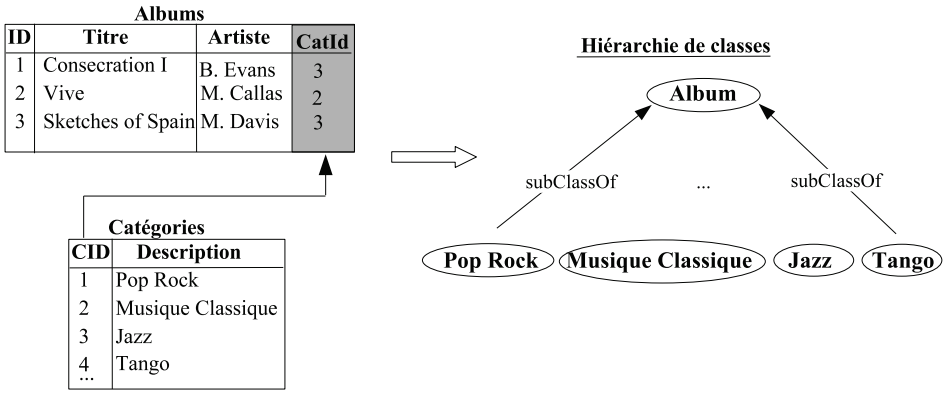

FIG. 2 – *Un exemple de motif de catégorisation où les catégories à extraire pour produire les sous-classes sont définies dans une relation externe.*

de milliers d'items relatifs aux activités de maintenance (outillage, rechanges, composants, ...). Dans cette base complexe, les motifs de catégorisation étudiés ici sont très largement représentés et RTAXON produit à partir de cette base une ontologie de 110 classes (60K instances) où une dizaine de classes sont affinées en hiérarchies relativement étendues (en largeur). Une évaluation par des experts du domaine a permis de valider la pertinence de la structure induite automatiquement et de confirmer la capacité à bien rendre compte de la variété des concepts sous-jacents à ce domaine. Des informations complémentaires sur cette étude de cas sont disponibles sur le site du projet européen TAO[4].

6 Conclusion

Nous avons présenté une nouvelle méthode d'acquisition automatique d'ontologies à partir de bases de données relationnelles. Cette méthode montre comment construire des ontologies plus riches en hiérarchies en combinant une analyse classique des méta-données et une tâche de fouille dédiée à l'identification de motifs de catégorisation dans les données.
Plusieurs extensions peuvent être envisagées. Un premier prolongement de la méthode serait de viser l'identification de motifs de même forme mais plus complexes. En particulier, les motifs combinant deux attributs catégorisants, aussi très fréquents, permettraient de produire des hiérarchies à deux niveaux (voir notre exemple des fournisseurs en section 2). RDBToOnto supporte en partie ces motifs à deux attributs, mais seule la phase de génération est automatisée. Les deux attributs du motif sont fournis en entrée via l'interface utilisateur. Dans l'optique d'automatisation de la phase d'identification, ces motifs complexes pourraient être révéler par une analyse des cooccurrences dans les attributs. Une autre extension qui nous parait prometteuse serait de diversifier les sources en recherchant d'autres formes de motifs identifiables dans le contenu, comme les motifs fondés sur les valeurs nulles (Lammari et al. (2007)).

[4]http ://www.tao-project.eu

Références

Astrova, I. (2004). Reverse engineering of relational databases to ontologies. In *The Semantic Web : Research and Applications, First European Semantic Web Symposium (ESWS 2004)*, Greece. Stringer-Verlag.

Auer, S., S. Dietzold, J. Lehmann, S. Hellmann, et D. Aumueller (2009). Triplify – Light-weight linked data publication from relational databases. In *soumis à WWW 2009*.

Barrasa, J., O. Corcho, et A. Gomez-Pérez (2004). R2O, an extensible and semantically based database-to-ontology mapping language. In *Second Workshop on Semantic Web and Databases (SWDB2004)*, Toronto, Canada.

Bizer, C. (2003). D2R MAP - a database to RDF mapping language. In *Proceedings of WWW03*, Budapest.

Cerbah, F. (2008). Learning highly structured semantic repositories from relational databases - rdbtoonto tool. In *The Semantic Web : Research and Applications – Proceedings of the 5th European Semantic Web Conference (ESWC 2008)*. Springer.

de Marchi, F. (2003). *Découverte et visualisation par l'exemple des dépendances fonctionnelles et d'inclusion dans les bases de données relationnelles*. Ph. D. thesis, Université Blaise Pascal – Clermont II.

Lammari, N., I. Comyn-Wattiau, et J. Akoka (2007). Extracting generalization hierarchies from relational databases. A reverse engineering approach. *Data and Knowledge Engineering 63*, 568–589.

Li, M., X. Du, et S. Wang (2005). Learning ontology from relational database. In *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, Volume 6, pp. 3410 – 3415. IEEE.

Patel-Shneider, P. F., P. Hayes, et I. Horrocks (2004). OWL web ontology language semantics and abstract syntax. Technical report, W3C, http ://www.w3.org/TR/owl-absyn.

Stojanovic, L., N. Stojanovic, et R. Volz (2002). Migrating data-intensive web sites into the semantic web. In *Proc. of the ACM Symposium on Applied Computing (SAC 02)*, Madrid.

Tari, Z., O. A. Bukhres, J. Stokes, et S. Hammoudi (1998). The reengineering of relational databases based on key and data correlations. In *Searching for Semantics : Data Mining, Reverse Engineering, etc*, pp. 7–10. Chapman and Hall.

Summary

Relational databases are valuable sources for ontology learning. Previous work showed how formal ontologies can be learned from such structured input. However, a major persisting limitation of the existing approaches is the derivation of ontologies with flat structure that simply mirror the schema of the source databases. In this paper, we present the RTAXON learning method that shows how the content of the databases can be exploited to identify categorization patterns from which class hierarchies can be generated. This fully formalized method combines a classical schema analysis with hierarchy mining in the data. RTAXON is one of the methods implemented in the RDBToOnto tool.