MuMIe: Une Approche Automatique pour l'Interopérabilité des Métadonnées

Samir Amir, Ioan Marius Bilasco, Thierry Urruty et Chabane Djeraba

LIFL UMR CNRS 8022, Université de Lille1, Telecom-Lille1,
50 avenue Halley Parc scientifique de la Haute-Borne, Villeneuve d'Ascq, France
{samir.amir, marius.bilasco, thierry.urruty, chabane.djeraba}@lifl.fr

Résumé. Avec l'explosion du multimedia, l'utilisation des métadonnées est devenue cruciale pour assurer une bonne gestion des contenus. Cependant, il est nécessaire d assurer un accès uniforme aux métadonnées. Plusieurs techniques ont ainsi été développées afin de réaliser cette interopérabilité. La plupart d'entre elles sont spécifiques à un seul langage de description. Les systèmes de matching existants présentent certaines limites, en particulier dans le traitement des informations structurelles. Nous présentons dans cet article un nouveau système d'intégration qui supporte des schémas provenant de langages descriptifs différents. De plus, la méthode de matching proposée a recours à plusieurs types d'information de façon à augmenter la précision de matching.

1 Introduction

L'omniprésence des ressources numériques pose le problème de l'efficacité de leur gestion, ce qui a généré un intérêt croissant pour l'utilisation des métadonnées destinées à améliorer la gestion de ces types de contenu. Les métadonnées peuvent véhiculer divers types d'information décrivant les types de contenu multimédia, les informations sémantiques de ces contenus et les caractéristiques des terminaux utilisant ces contenus.

Si on anticipe la croissance des métadonnées, il sera certainement de plus en plus difficile d'obtenir un accès uniforme aux objets multimédia en raison du nombre de communautés indépendantes de métadonnées. En effet, chaque communauté combine les termes à partir de plusieurs vocabulaires spécifiques, et utilise différentes structures pour organiser les métadonnées. L'interopérabilité des métadonnées devient donc un enjeu crucial.

L'intégration manuelle des métadonnées entraîne un coût temps assez important. En plus, elle doit être mise à jour à chaque apparition d'un nouveau format de métadonnées. Ainsi, plusieurs techniques automatiques ont été développées pour faciliter le processus d'intégration. Les techniques de *matching* jouent un rôle central dans le cadre de ces approches. Cependant, peu de systèmes de matching supportent des schémas issus de différents langages. On trouve notamment les systèmes suivants : Similarity Flooding (Melnik et al. (2002)), Cupid (Do et Rahm (2002)). Ces approches n'utilisent pas la majorité des informations structurelles et sémantiques (ex : propriétés d'équivalence, caractéristiques de généralisation, etc.), ce qui rend la détection des matchings complexes (n:m matching) difficile. De plus, un des inconvénients majeurs de ces méthodes est leur façon d'utiliser les informations structurelles.

Motivés par les défis présentés ci-dessus, nous abordons par la suite une approche simplifiée que nous baptisons MuMIe (Multi-level Metadata Integration). Cette approche permet d'obtenir une interopérabilité à deux niveaux par l'integration de plusieurs schémas hétérogènes définis dans différents langages.

Le reste de ce papier est organisé comme suit : dans la Section 2 nous présentons en détail notre solution en insistant sur l'homogénéisation des langages de description, sur les mesures des similarités linguistiques et structurelles que nous mettons en oeuvre. Dans la Section 3 nous analysons les résultats obtenus dans les expérimentations réalisées sur des standards issues du monde multimédia. Nous concluons le travail en rappelant les avancées réalisées dans cet article par rapport à l'état de l'art, ainsi qu'en précisant les travaux futurs.

2 L'approche MuMIe (Multi-Level Metadata Integration)

Cette section montre les deux principales étapes de notre approche. La première est celle de la projection qui consiste à obtenir une interopérabilité au niveau des langages de description (Section 2.1). La seconde étape est celle du matching qui sert à trouver toutes les correspondances entre les métadonnées hétérogènes.

2.1 Espace commun de représentation

En raison de l'hétérogénéité des langages de définition des schémas, il est impossible de trouver un modèle de représentation unique supportant toutes les caractéristiques des schémas. Dans notre approche, nous modélisons ces schémas en tant que graphes orientés et classifiés représentant uniquement les concepts de schémas et les propriétés qui les relient entre eux. Ces deux entités sont des informations de base pour tous les langages descriptifs. Pour ce faire, les concepts descriptifs relatifs aux langages de définition des schémas doivent être connus. Par exemple, toutes les classes (*rdfs :class*) et les propriétés (*rdf :property*) des schémas RDF doivent être représentées par des nœuds. Ces nœuds sont classifiés en *nœud simple* et *nœud complexe*. Les nœuds simples correspondent aux concepts (ex : rdfs :class, xsd :complexType, xsd :element, etc.), et les nœuds complexes correspondent aux propriétés reliant les concepts de schémas (ex : *rdf :property*, *owl :ObjectProperty*, etc.). Quant aux informations sémantiques et structurelles (ex : *rdf* :*equivalentProperty*, *owl* :*DifferentFrom*, etc) elles ne sont pas représentées dans le graphe mais elles sont considérées comme des propriétés des nœuds.

2.2 Calcul de la similarité entre les nœuds

Dans cette étape, nous calculons la similarité entre tous les nœuds formant les graphes. Nous commençons par le calcul de la similarité entre les nœuds en utilisant leur noms et commentaires, les informations sémantiques et structurelles décrites dans la section 2.1 sont également utilisées afin de détecter les matchings complexes. Le détail du calcul linguistique est donné dans (Amir et al. (2010)). Une fois que le calcul de similarité linguistique est effectué, nous utilisons l'information structurelle pour filtrer les faux candidats. Pour ce faire, on considère que les nœuds possèdent trois niveaux contextuels (Lee et al. (2002)). Le premier concerne la similarité des ancêtres. Le second sert à mesurer la similarité des fils immédiats. L'analyse de feuilles constitue le troisième niveau d'analyse. La similarité des nœuds est obtenue en combinant ces trois niveaux.

2.2.1 Similarité des ancêtres

Les ancêtres d'un nœud n_i sont définis par le chemin p_i s'étendant de n_i jusqu'à la racine du graphe. Par conséquence, afin de calculer la similarité entre deux contextes correspondants aux nœuds (n_i, n_j), la similarité entre les chemins (p_i, p_j) doit être calculée. Pour ce faire, nous utilisons dans cet article, l'approche proposée dans (D.Carmel et al. (2003)). Les auteurs proposent une méthode flexible permettant de calculer les similarités entre les chemins de nœuds en tenant compte de quatres paramètres : $lcs_n(p_i, p_j)$ est la plus longue sous-séquence commune entre p_i et p_j normalisée par la longueur de p_i. $pos(p_i, p_j)$ considère que le matching idéal entre deux chemins (p_i, p_j) est celui qui commence du premier nœud de p_i sans discontinuité. $gap(p_i, p_j)$ est utilisé pour s'assurer que les occurrences de p_i et p_j sont proches les unes des autres. $ld(p_i, p_j)$ donne des valeurs importantes aux chemins (p_i, p_i) dont la valeur de la longueur est proche. La combinaison des paramètres mentionnés précédemment donne la similarité *ps* entre deux chemins (p_i, p_i) :

$$ps(p_i, p_j) = \delta lcn_n(p_i, p_j) + \varphi pos(p_i, p_j) - \theta gap(p_i, p_j) - \lambda ld(p_i, p_j) \quad (1)$$

où δ, φ, θ et λ sont déterminées sur la base les expérimentations effectuées dans (D.Carmel et al. (2003)). δ = 0.75 ; φ = 0.25 ; θ=0.25 ; λ=0.2. Dans notre approche nous introduisons une nouvelle relaxation des contraintes définies dans (D.Carmel et al. (2003)) comme suit :

- Pour les quatre paramètres, la plus longue sous-séquence commune (lcs) est calculée en fonction de la matrice de similarité linguistique *lSim* calculée dans la section précédente. Cela signifie que deux nœuds (n_i, n_j) sont considérés identiques si la valeur de *lSim*(n_i, n_j) est supérieure à un seuil donné e.g. 0.80.
- Les paramètres dans (D.Carmel et al. (2003)) ont été définis pour les schémas XML qui sont basés sur une classification taxonomique. Pour cette raison, l'utilisation de ces paramètres dans un graphe étiqueté nécessite une autre relaxation pour le calcul de lcs. Par exemple, si on considère les deux triplets RDF *(Object, ContentOn, URL)* et *(Object, HasReference, URL)*, on peut remarquer que le nom de la propriété dans le premier triplet (*Content*) correspond à *Object*. Par contre, le nom de la propriété dans le deuxième triplet (*Reference*) correspond à *URL*. A ce stade, nous introduisons une nouvelle relaxation en considérant les nœuds de type propriété comme des nœuds qui peuvent être permutés avec leur fils ou parent immédiat.

Finalement, la similarité entre deux contextes des ancêtres est la similarité entre les chemins pondérée par la linguistique similarité des nœuds correspondant (n_i, n_j) :

$$ancSim(n_i, n_j) = ps(n_i, n_j) * lSim(n_i, n_j) \quad (2)$$

2.2.2 Similarité des fils immédiats

La similarité des fils immédiats *immSim* entre deux nœuds (n_i, n_j) est faite par la comparaison des deux sous ensembles des fils immédiats $S = \{s_1, s_2, \cdots, s_n\}$ et $S^{'} = \{s_1^{'}, s_2^{'}, \cdots, s_m^{'}\}$ (les fils immédiats désignent uniquement les nœuds simples). Cela est fait par l'utilisation de la similarité *SimI* entre chaque paire de nœuds appartenant aux deux ensembles, où :

$$SimI(s_i^{'}, s_j) = ps(s_i^{'}, s_j) * lSim(s_i^{'}, s_j) \quad (3)$$

$ps(s_i^{'}, s_j)$ est la similarité entre les deux chemins allant de $(s_i^{'}, s_j)$ à (n_i, n_j). Puis, les paires ayant une valeur maximale de *SimI* sont sélectionnées. la moyenne des meilleures similarités est prise afin da calculer la valeur de similarité *immSim*

2.2.3 Similarité des feuilles

Le contexte des feuilles d'un nœud simple n_i est l'ensemble des feuilles reliées à ce nœud. Si on considère que $l_i \in$ *leaves(n_i)* est un nœud de type feuille, alors la contexte de l_i est le chemin p_i allant de n_i à l_i. A ce stade, le contexte des feuilles est donné par :

$$\textit{leafSim}(l_i, l_j) = ps((p_i, p_j)) * \textit{lSim}(l_i, l_j) \tag{4}$$

Afin de mesurer la similarité entre deux feuilles $l_i \in$ *leaves(n_i)* et $l_j \in$ *leaves(n_j)*, on calcule la similarité des feuilles *leafSim* entre chaque paire des feuilles dans les deux ensembles des feuilles. Puis on sélectionne la paire ayant la valeur maximale de similarité. La moyenne des meilleures similarités est prise.

2.2.4 Similarité des nœuds

La similarité des nœuds est obtenue par la combinaison des trois similarités décrites précédemment : similarité des ancêtres, similarité des fils immédiats et similarité des feuilles.

$$\textit{nodeSim}(n_i, n_j) = \alpha * \textit{ancSim}(n_i, n_j) + \beta * \textit{immSim}(n_i, n_j) + \gamma * \textit{leafSim}(n_i, n_j) \tag{5}$$

où $\alpha + \beta + \gamma = 1$ and $(\alpha, \beta, \gamma) \geq 0$

Une fois que la similarité structurelle est effectuée, le système retourne pour chaque nœud n_i les K nœuds correspondant aux K-plus grandes valeurs de *nodeSim*. Ces nœuds doivent avoir une valeur de *nodeSim* supérieure à un seuil donné.

3 Expérimentations

Dans cette section nous étudions l'influence des paramètres α, β et γ. Nous montrons également les tests effectués permettant d'évaluer la qualité de matching du système proposé.

3.1 Réglage des paramètres

Afin de connaître l'influence de chaque contexte sur le matching, nous avons effectué plusieurs expérimentations en utilisant des combinaisons différentes des paramètres α, β et γ. La Figure 1 montre les résultats du matching en terme de *F-mesure* (Madhavan et al. (2001)) entre les standards MPEG7-DIG35 (à droite) et DIG35-EXIF (à gauche) respectivement. Nos expérimentations ont montré qu'une partie importante de l'information structurelle est contenue dans le contexte des nœuds parents où les valeurs implorantes de *F-mesure* sont localisées pour $\alpha \in$ [0.45 0.6] ; cela explique l'intérêt de quelques stratégies de matching qui considèrent que le contexte d'un nœud dépend uniquement de ces parents. En outre, les expérimentations ont montré que les contextes des fils immédiats et feuilles ont un rôle important dans le processus de matching ($\beta \in$ [0.1 0.15] et $\gamma \in$ [0.25 0.4]). Cependant, la fixation des paramètres α, β et γ pour chaque paire de nœuds, en fonction de leur position dans le graphe pourra sans doute augmenter la qualité de matching. Cela fera parti de nos futures travaux.

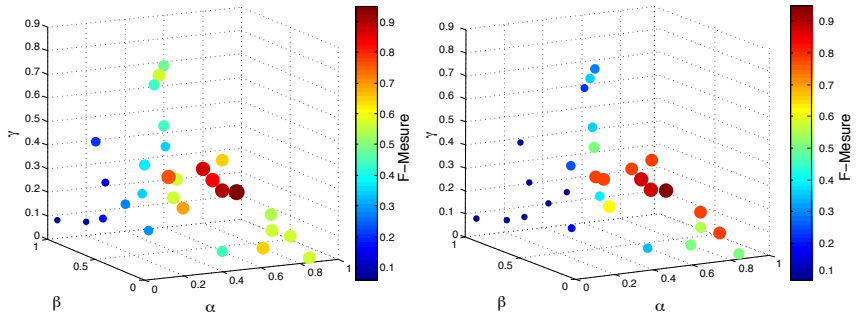

FIG. 1 – *Influence de α, β et γ sur la qualité du matching*

3.2 Qualité de matching

Le système a été testé sur plusieurs standards de métadonnées (Figure 2). Ces standards présentent une hétérogénéité significative aux deux niveaux. Les paramètres α, β et γ ont été choisis suite aux expériences effectuées dans la section précédente ($\alpha = 0.55, \beta = 0.15, \gamma = 0.30$). Le matching a été calculé entre CAM4Home (Bilasco et al. (2010)) et le reste des standards, le choix de CAM4Home comme un schema médiateur est fait car ce dernier contient plusieurs informations communes avec le reste des standards. Notre solution a été évaluée en fonction des matchings correctes détectés entre CAM4Home et les autres standards. La Figure 2 montre les résultats obtenus en terme de *F-mesure* décrite dans (Melnik et al. (2002)).

Les expérimentations ont montré que notre solution est meilleure que SF pour tous les standards utilisés. Cela est dû à l'utilisation d'un seul contexte de similarité pour SF. Alors que, notre approche prend compte des trois contextes de similarité. De plus, notre système n'est pas limité à un ou deux langage de description. Il a également détecté avec succès 68% des matchings complexes.

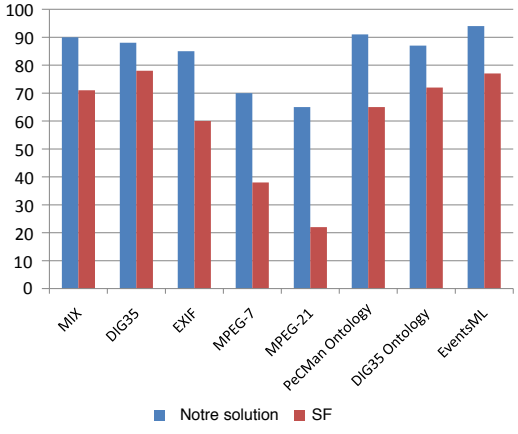

FIG. 2 – *Etude comparative (F-mesure).*

4 Conclusion

En raison de la forte utilisation de métadonnées, il est important de développer un système d'intégration pour assurer leur interopérabilité. L'existence d'un tel système est devenue cruciale afin d'uniformiser l'accès aux métadonnées. Pour cela, nous avons proposé une nouvelle technique réalisant une interopérabilité des métadonnées quelque soit le langage de description utilisé. Nous avons essentiellement proposé une stratégie de matching qui agit sur les deux niveaux d'hétérogénéité. Cela est fait par l'utilisation de plusieurs types d'information : syntaxique, sémantique et structurelle. Notre expérimentation a montré que la combinaison de ces informations augmente de manière significative la détection des mappings corrects entre les métadonnées. Dans nos travaux en cours, nous prévoyons d'améliorer le système d'intégration proposé par une meilleure exploitation de l'information structurelle. Nous explorerons principalement l'utilisation des relations d'adjacence entre les nœuds afin de détecter des alignements qui ne peuvent pas être détectés par la stratégie actuelle.

Références

Amir, S., I. M. Bilasco, T. Danisman, I. E. sayad, et C. Djeraba (2010). Multimedia metadata mapping : Towards helping developers in their integration task. In *ACM MoMM*, pp. 213–220.

Bilasco, I. M., S. Amir, P. Blandin, C. Djeraba, J. Laitakari, J. Martinet, E. Martínez-Gracía, D. Pakkala, M. Rautiainen, M. Ylianttila, et J. Zhou (2010). Semantics for intelligent delivery of multimedia content. In *SAC*, pp. 1366–1372.

D.Carmel, Y. S. Maarek, M. Mandelbrod, Y. Mass, et A. Soffer (2003). Searching xml documents via xml fragments. In *SIGIR*, pp. 151–158.

Do, H. H. et E. Rahm (2002). Coma - a system for flexible combination of schema matching approaches. In *VLDB*, pp. 610–621.

Lee, M.-L., L. H. Yang, W. Hsu, et X. Yang (2002). Xclust : clustering xml schemas for effective integration. In *CIKM*, pp. 292–299.

Madhavan, J., P. A. Bernstein, et E. Rahm (2001). Generic schema matching with cupid. In *VLDB*, pp. 49–58.

Melnik, S., H. Garcia-Molina, et E. Rahm (2002). Similarity flooding : A versatile graph matching algorithm and its application to schema matching. In *ICDE*, pp. 117–128.

Summary

The recent growth of digital resources requires an extensive use of metadata. However, a uniform access is necessary in order to take advantage from metadata. In this context, several techniques for achieving interoperability have been developed. Most of these techniques focus on matching schemas defined using one schema description language. The few existent matching systems that support schemas from different languages present some limitations. In this paper, we presents a new integration system supporting schemas from different description languages. Moreover, the proposed matching process makes use of several types of information in a manner that increases the matching accuracy.