Mappings pour l'intégration de documents XML

Chantal Reynaud, Brigitte Safar

Université Paris-Sud XI, CNRS (L.R.I.) & INRIA (Futurs)
91405 Orsay cedex
{Chantal.Reynaud, safar}@lri.fr
http://www.lri.fr/~cr

Résumé. Ce travail se situe dans le domaine de l'intégration de sources d'informations hétérogènes. Il vise l'intégration d'une nouvelle source XML à un serveur d'information en exploitant une ontologie, c'est-à-dire une description explicite et déclarative de la sémantique d'un domaine d'application. Nous proposons d'automatiser la génération des mises en correspondance, ou mappings, entre l'ontologie et la source à intégrer. Nous montrons que ce processus repose non seulement sur l'exploitation des schémas des modèles à connecter mais aussi sur les données associées. Un prototype, DétectHybrideMap, développé en Java, est à l'origine d'expérimentations sur des données réelles fournies par France Télécom R&D dans le domaine du tourisme.

1 Introduction

L'intégration d'information est une problématique importante du fait du nombre croissant de sources d'information disponibles via le Web. Appliquée à un serveur d'information, elle donne l'impression à un utilisateur qu'il interroge un système unique et centralisé grâce à une interface d'interrogation uniforme basée sur un modèle du domaine d'application, l'ontologie. Notre travail se situe dans le cadre d'un tel serveur, intégrant les schémas de sources externes déjà intégrées au sein d'un entrepôt local[1]. Des mises en correspondance (ou mappings) ont déjà été établies entre l'ontologie du domaine et les schémas intégrés au serveur. Notre objectif porte sur la génération automatisée de mappings entre l'ontologie et une nouvelle source à intégrer.

La source à intégrer est une source XML structurée à l'aide du vocabulaire de l'ontologie du serveur d'information. En revanche, la structure de ce document est particulière dans le sens où elle ne correspond pas forcément à la structure des documents déjà intégrés.

L'approche de découverte de mappings que nous adoptons est guidée par l'ontologie du domaine : recherche des mises en correspondances entre concepts, puis entre propriétés avant de s'intéresser aux mises en correspondance entre relations. Elle s'appuie sur les mappings déjà établis entre l'ontologie et les sources intégrées au sein du serveur. Le processus de génération des mappings se veut être le plus automatisé possible. Le concepteur ne doit intervenir qu'en fin de processus pour valider les résultats trouvés. Enfin, l'approche doit être générique c'est-à-dire utilisable quel que soit le domaine d'application. Notre objectif, au travers de ce papier, est de montrer que la génération automatisée de mappings dans ce

[1] Travail de recherche développé dans le cadre du projet Picsel3 (2005-2008) en partenariat avec France-Telecom R&D.

contexte nécessite l'exploitation à la fois du schéma des sources XML (les DTDs) et des données des sources à connecter.

Ce papier est organisé de la façon suivante. Dans la section suivante, nous présentons quelques travaux portant sur la détection de mappings et situons notre approche. Nous décrivons ensuite l'environnement de travail, puis les techniques utilisées pour la détection de mappings. La section 5 est consacrée aux expérimentations réalisées dans le domaine du tourisme. Enfin, nous concluons et présentons quelques perspectives.

2 Travaux proches

Il existe de nombreux travaux qui visent à automatiser la génération de mappings. Une synthèse des techniques utilisées est présentée dans Rahm et Bernstein (2001)] et Shvaiko et Euzenat (2004). Nous distinguerons deux types de techniques, celles qui exploitent le schéma des sources et celles qui exploitent les données associées aux schémas.

Les techniques basées sur le schéma des sources sont variées. Il peut s'agir de comparer les éléments des schémas. Les méthodes utilisées sont : analyse de chaînes de caractères (comparaison des préfixes, des suffixes, distance de Levenshtein, technique des n-grammes (Euzenat et Valtchev (2004), Melnik et al. (2002), Do et Rahm (2001), Noy et Musen (2001), Giunchiglia et Shvaiko (2004)), techniques de traitement de langage naturel (décomposition, lemmatisation, etc. (Do et Rahm (2001), Giunchiglia et Shvaiko (2004), Euzenat et Valtchev (2004)) ou utilisation de ressources linguistiques (Giunchiglia et Shvaiko (2004), Euzenat et Valtchev (2004)). Il peut aussi s'agir d'exploiter la structure des schémas comparés, souvent représentés sous forme de graphes. Les algorithmes mettant en œuvre ces techniques implémentent diverses heuristiques. Une heuristique consiste, par exemple, à considérer que des éléments de deux schémas distincts sont similaires si leurs sous-concepts directs, et/ou leurs sur-concepts directs et/ou leurs concepts frères sont similaires (Do et Rahm (2001), Noy et Musen (2001) (Thanh Le et al. 2004)). Ces techniques structurelles peuvent être basées sur la notion de point fixe (Melnik et al. (2002)). Dans S-Match (Giunchiglia et Shvaiko (2004)), le problème de matching est vu comme un problème de satisfiabilité d'un ensemble de formules du calcul propositionnel. Les graphes et les correspondances à tester sont traduits en formules de la logique propositionnelle en considérant la position des concepts dans le graphe et non seulement leur nom. Basé sur l'utilisation d'un modèle (SAT), ce système est également une illustration de l'application de techniques sémantiques.

D'autres travaux se basent sur les données. Ainsi, FCA-Merge (Stumme et Maedche (2001)) exploite des instances dans le but de fusionner des ontologies locales. Les auteurs proposent d'extraire ces instances de documents textes puis d'appliquer l'Analyse en Concepts Formels. Chaque nœud du treillis de concepts obtenu est associé à un ensemble de concepts des ontologies locales lorsque les instances associées sont contenues dans les mêmes documents. L'étape finale d'analyse du treillis pour construire l'ontologie globale est à la charge du concepteur. OLA (Euzenat et Valtchev (2004)) est une classe d'algorithmes d'alignement d'ontologies qui exploite toutes les caractéristiques possibles des ontologies représentées en OWL-Lite et qui inclut, de ce fait, la comparaison des extensions connues des concepts. Enfin, certains systèmes appliquent des techniques d'apprentissage automatique. Ils exploitent différents types d'informations : les mots, leur position, leur format, leur fréquence, les caractéristiques de la distribution de valeurs prises. C'est le cas de GLUE

(Doan et al. 2003) qui porte sur l'identification de mises en correspondance entre un schéma global et le schéma (DTD) de sources d'information XML.

Notre travail se distingue des travaux précédemment décrits. Nous avons été guidés par l'intégration de données XML réelles dont l'étude a montré que la définition de certains mappings faisait explicitement référence aux données des sources à mettre en correspondance. Une exploitation limitée aux schémas de ces sources n'était donc pas suffisante, c'est pourquoi nous proposons une approche hybride combinant exploitation de schémas et de données qui a été mise en œuvre dans DétectHybrideMap (Bisgin (2005)).

3 Environnement de travail

Dans cette partie, nous décrivons l'ontologie, l'entrepôt de sources XML, la source à intégrer et les mappings.

3.1 L'ontologie

L'ontologie est composée d'un ensemble de classes ayant des propriétés et d'un ensemble de relations de subsumption ou du domaine entre classes. Une ontologie O est définie comme un tuple (C, R) où $C = \{c_1, c_2, ..., c_n\}$ est l'ensemble des classes et où $R = \{r_1, r_2,...,r_m\}$ regroupe l'ensemble des relations entre classes (R_1) et les propriétés des classes (R_2). Une relation $r_1 \in R_1$ met en relation deux noms de classes, ($\forall r_1 (t_1 , t_2) \Rightarrow t_1 \in C, t_2 \in C$), et une propriété $r_2 \in R_2$ relie un nom de classe à un littéral, ($\forall r_2 (t_1 , t_2) \Rightarrow t_1 \in C, t_2 \in \lambda$ où λ est un littéral). Dans le cadre du projet Picsel 3, le formalisme de représentation des connaissances adopté pour représenter l'ontologie est RDF-S (Antoniou et Van Harmelen (2004)) dont les primitives permettent la représentation de classes, de propriétés, de hiérarchies de classes, et de hiérarchies de propriétés. Il s'écrit à l'aide de triplets RDF de la forme <sujet, prédicat, objet>. Ainsi les relations $r(t_1 , t_2)$ sont traduites en RDF-S sous la forme de triplet (t_1 , r , t_2).

3.2 L'entrepôt local

L'entrepôt local comprend un ensemble de sources XML pour lesquelles des mises en correspondance avec l'ontologie du domaine ont déjà été définies. Ces mises en correspondance sont établies entre les schémas des sources, des DTDs représentées sous la forme d'arbres, et l'ontologie. La représentation des DTDs est simplifiée. Elle ne prend pas en compte le format des éléments ni des attributs, elle ignore leurs caractéristiques (EMPTY, REQUIRED, etc.) et les cardinalités.

Un arbre T représentant une DTD est un ensemble tel que $T = \{N, A, AT, ATN\}$ où :

- N est l'ensemble fini des nœuds : tous les éléments de cet ensemble sont des éléments de la DTD associée.
- A est l'ensemble fini des couples NxN représentant des arêtes entre deux nœuds non vides. Ils correspondent aux liens de composition entre éléments de la DTD.
- AT est l'ensemble fini des attributs de la DTD. Les attributs de cet ensemble ne sont pas des nœuds de N mais chaque attribut est lié à un élément de l'ensemble N.
- ATN est l'ensemble fini des couples $(e_i, <at_1, at_2,..., at_n>)$ tel que $i \in \{1,n\}$. Si un élément e_x n'a pas d'attribut, il n'y aura pas de couple de cet élément dans ATN. S'il a deux attributs at_y, et at_z, le couple correspondant dans ATN sera $(e_x, <at_y, at_z>)$.

Un exemple de DTD et sa représentation sous forme d'arbre sont donnés FIG 1. Dans la suite, nous appellerons *domaine* d'un nœud, l'ensemble de ses nœuds fils dans l'arbre. Ainsi, le domaine du nœud *a* de la DTD de FIG.1 est {b,c}.

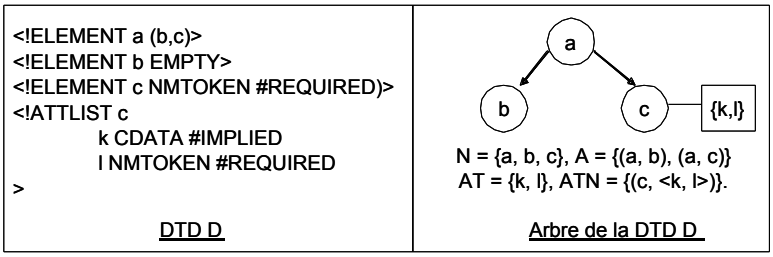


FIG. 1 - *Une DTD et sa représentation sous forme d'arbre.*

3.3 La source à intégrer

La source à intégrer dans l'entrepôt est une source XML à laquelle est associée une DTD représentée également sous forme d'arbre comme les DTDs des documents de l'entrepôt local (cf. section 3.2). Elle est structurée à l'aide du vocabulaire de l'ontologie à condition, bien entendu, que les termes pertinents soient présents. En revanche, la structure de la DTD associée au document ne correspond pas forcément à la structure des DTDs des documents déjà dans l'entrepôt. A titre d'illustration, FIG.2 présente deux sous arbres représentant des DTDs. Le premier correspond à une partie de l'arbre de la DTD d'une nouvelle source à intégrer et le deuxième à une partie de l'arbre d'une DTD de l'entrepôt (les attributs ne sont pas représentés pour plus de simplicité). Ces DTDs contiennent des nœuds aux noms identiques (en gris dans la figure) : « *poi, name, contact, nb, format, datum, unit, postal-address*» mais leur position, dans chaque DTD, est différente.

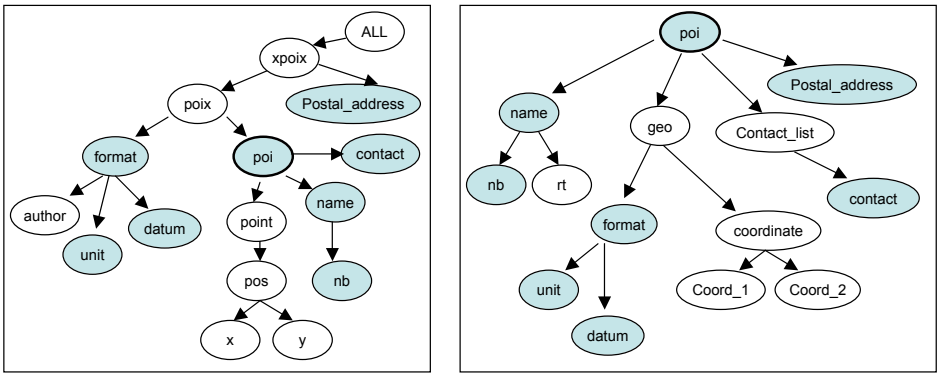


FIG. 2 - Sous arbre T_1 de la DTD à intégrer et sous arbre T_2 d'une DTD de l'entrepôt.

3.4 Les mappings

Un mapping est une mise en correspondance entre un élément de l'ontologie *O* et un élément (pris au sens général du terme) de l'arbre *T* d'une DTD. Pour les mappings déjà existants, il s'agit des DTDs des sources XML de l'entrepôt déjà constitué. Pour les map-

pings à générer, il s'agit de la DTD de la nouvelle source XML à intégrer. Nous définissons un mapping comme une fonction qui fait correspondre à tout concept ou propriété de *O*, l'élément ou l'attribut avec lequel ce concept ou cette propriété est mis en correspondance, l'élément ou l'attribut étant défini par son chemin dans *T*.

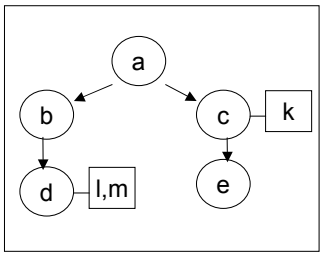
Cette mise en correspondance peut être conditionnelle. Les conditions qui doivent être éventuellement satisfaites portent sur les valeurs des éléments ou des attributs dans les documents XML satisfaisant la DTD d'arbre *T*. Supposons, par exemple, que l'on recherche un mapping pour le concept *email* spécialisation du concept *contact* de l'ontologie. Un mapping conditionnel le reliera à l'attribut *type* de l'élément *contact* de la DTD, à condition que la valeur de *type* dans le document XML associé à la DTD soit *email*.

Nous utiliserons le symbole "↔" pour représenter la liaison entre les éléments mis en correspondance. Nous explicitons ci-dessous les notations utilisées pour représenter chacune des parties d'un mapping, puis les formats des différents mappings en Fig. 4.

Notation de la composante ontologie. Soient c_1, c_2 deux concepts de l'ontologie O, $r_1 \in R_1$ une relation entre c_1 et c_2 et $r_2 \in R_2$ une propriété de c_1.

- Les concepts de O seront désignés par leur nom, celui-ci les identifiant de manière unique. Les mappings concernant les concepts seront de la forme " c_1 ↔…" et " c_2 ↔…".
- Les relations de R_1 seront désignées par leur nom suivi des concepts liés. Les mappings les concernant seront de la forme "$r_1(c_1, c_2)$ ↔…".
- Les relations de l'ensemble R_2 (les propriétés) seront désignées par leur nom associé au concept caractérisé. Les mappings les concernant seront de la forme : "r_2 de c_1 ↔…" et "r_2 de c_2 ↔…".

Notation de la composante source La deuxième entité liée au sein d'un mapping correspond à un élément ou à un attribut d'un arbre de DTD. Chaque élément ou attribut d'un arbre est identifié, de façon unique, par le chemin qui le relie à la racine de l'arbre. Nous utilisons Xpath, le langage de requête de XML standardisé par le W3C, comme langage de représentation de ces chemins (cf. FIG.3).



/a : désigne la racine de l'arbre
//c : désigne un élément c à un endroit quelconque du document.
/a/b : désigne un élément b dont le père est a, a étant la racine.
//b/d : désigne un élément d dont le père est b, b n'étant pas la racine.
//c/@k : désigne l'attribut k de l'élément c

FIG. 3 - *Représentations en Xpath*.

Ainsi, les mappings établissant un lien avec un élément dont le nom est e seront de la forme : …↔ //e. Les mappings établissant un lien avec un attribut (de nom att) d'un élément (de nom e) de la DTD seront de la forme : ... ↔ //e/@att. Les mappings conditionnels établissant un lien avec un attribut d'un élément de la DTD suivant la valeur de cet attribut dans le document XML associé à la DTD, seront de la forme :... ↔//e/[@ att='valeur']/@att

concept $c_1 \in O$	$c_1 \leftrightarrow$ //e $c_1 \leftrightarrow$ //e/@att $c_1 \leftrightarrow$ //e/[@att = 'val']/@att
relation r_1 entre c_1 et $c_2 \in O$ tels que $\exists$ $c_1 \leftrightarrow$ //a et $c_2 \leftrightarrow$ //b	$r_1(c_1, c_2) \leftrightarrow$ //a/ …/b
propriété r_2 de $c_1 \in O$ tel que $\exists$ $c_1 \leftrightarrow$ //a et b est le correspondant de r_2 dans T	r_2 de $c_1 \leftrightarrow$ //a/ …/b

Fig. 4 – *Format des différents mappings*

Pour les mappings établissant un lien avec une relation de l'ontologie, la partie du mapping concernant la source désignera le chemin, s'il existe, reliant les correspondants dans la DTD des éléments de l'ontologie reliés par la relation.

4 Techniques de recherche des mappings

Le but de notre travail est de détecter semi-automatiquement des mappings entre une ontologie représentée en RDF-S et une DTD d'une nouvelle source à intégrer, dont les balises reprennent, dans la mesure du possible, le vocabulaire de l'ontologie. Les entrées du système sont donc (1) l'ontologie, (2) la DTD de la nouvelle source et (3) la liste des mappings pré-existants entre l'ontologie et les DTDs des sources déjà intégrées à l'entrepôt.

L'approche proposée est guidée par l'ontologie. Dans un premier temps, on cherche à identifier les correspondants des concepts de l'ontologie. On s'appuie ensuite sur les correspondances de concepts trouvées pour chercher les mappings de leurs propriétés. On s'intéresse, dans un troisième temps, aux relations entre concepts. Afin d'illustrer le processus de découverte, composé de techniques basées à la fois sur les schémas des documents rapprochés mais aussi sur l'exploitation des données, nous nous focalisons sur les techniques permettant d'identifier les correspondants des concepts de l'ontologie. Ce sont ces mêmes techniques qui sont utilisées pour trouver les mappings sur les propriétés.

4.1 Exploitation des schémas des documents à rapprocher

L'objectif est ici de trouver les correspondants des différents concepts *c* de l'ontologie *O* parmi les éléments d'une DTD d'arbre T_1. Deux techniques sont utilisées, cf. l'algorithme Fig. 5, l'une exploitant l'ontologie, la seconde exploitant les mappings pré-existants, la première précédant la seconde dans son application.

La première technique appliquée consiste à poser que si, pour un concept *c* de l'ontologie *O*, il existe dans T_1 un élément de même nom que *c*, un mapping peut être établi entre le concept et l'élément considéré. Cette technique est justifiée par le fait que la DTD est construite à l'aide du vocabulaire fixé dans l'ontologie *O*, que l'ontologie, telle qu'elle est structurée, correspond à une description des connaissances d'un certain point de vue, que le concepteur de la DTD connaît ce point de vue et choisit, en conséquence, les termes de l'ontologie pour structurer son document. Si un même terme est utilisé dans l'ontologie et le schéma d'une source, c'est que le concepteur de la nouvelle source est d'accord avec l'interprétation proposée par le concepteur de l'ontologie.

La deuxième technique s'applique quand la première a échoué, c'est-à-dire quand il n'existe pas de termes dans T_1 de même nom qu'un concept *c* de l'ontologie. Elle consiste à utiliser la liste des mappings pré-existants entre *O* et les DTDs des sources déjà intégrées dans l'entrepôt local. On recherche à quel terme le concept *c* a été connecté dans les autres DTDs. Si pour une DTD d'arbre T_2, un mapping existe pour le concept *c* avec un élément *e* présent également dans T_1, l'approche consiste à étudier si ce mapping ne pourrait pas aussi s'appliquer dans le contexte de T_1. Le mapping n'est pas immédiatement réutilisé car il n'est pas certain que les deux termes de même nom mais utilisés dans des contextes différents correspondent au même concept. Une vérification s'impose. Elle est basée sur la comparaison du domaine (D_2) du terme considéré dans l'arbre T_2 avec le domaine de ce même terme ou de ses ancêtres dans l'arbre T_1. Une telle comparaison permet de trouver l'élément *e*'de T_1 le plus proche du *e* de T_2, c'est-à-dire dont le domaine a le plus d'éléments en commun avec le domaine D_2 de *e* dans T_2. L'élément *e*' retourné est soit l'élément *e* initial, soit un ancêtre de *e* dans T_1.

Recherche du correspondant d'un concept c de l'ontologie parmi les éléments d'une DTD $T_1 = \{N_1, A_1, AT_1, ATN_1\}$

Si $c \in N_1$ Alors retourner $c \leftrightarrow //c$
Sinon
  Si $\exists\ T_2 = \{N_2, A_2, AT_2, ATN_2\}$ déjà intégré et
    $\exists$ e, tel que $e \in N_2$
      et $e \in N_1$
      et $\{c \leftrightarrow //e\} \in$ ListeMappingEtablis
  Alors $D_2 \leftarrow$ Domaine(e,T_2) ;
    e' $\leftarrow$ Cdom(e, T_1, D_2, 0, e) ;
    retourner $c \leftrightarrow //e'$
  FinSi
FinSi

Cdom(e, T_1, D_2, Seuil, Origine)
  $D_1 \leftarrow$ Domaine(e,T_1) ; $N_{inter} \leftarrow D_1 \cap D_2$;
  Selon le cas N_{inter}
    $N_{inter} = |D_2|$: retourner e ;
    $N_{inter} >$ Seuil : Cdom(père(e), T_1, D_2, N_{inter}, e) ;
    $N_{inter} =$ Seuil : retourner Origine
  FinSelon

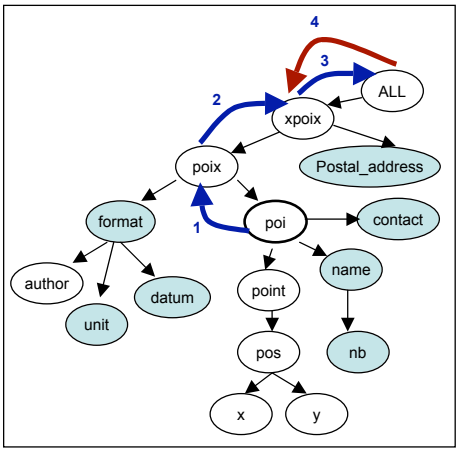


FIG. 5 - Algorithme d'exploitation des schémas et illustration de son déroulement

La comparaison s'effectue de la façon suivante. Au départ, le nombre de termes communs aux deux domaines comparés, N_{inter}, est initialisé à 0. En cours d'exécution, si N_{inter} est égal au nombre d'éléments de D_2, cela signifie que les deux domaines comparés sont égaux, et que l'élément le plus proche du *e* de T_2 est l'élément avec lequel il est comparé dans T_1. Dans le cas contraire, si le nombre d'éléments communs lors d'une étape est supérieur à celui de l'étape précédente, le processus est réitéré avec le père de l'élément courant dans T_1. Lorsque le nombre d'éléments communs ne progresse plus, l'élément recherché retourné est l'élément courant de l'étape précédente.

A titre d'illustration, considérons les arbres de DTD T_1 et T_2 présentés FIG. 2 et suivons l'exécution de l'algorithme pour *poi* (abréviation de ***po***int d'***i***ntérêt), nœud commun aux deux DTDs. Le domaine D_2 de *poi* dans T_2, {name, nb, rt, geo, format, unit, datum, coordinates, coord_1, coord_2, contact_list, contact, postal_address}, de cardinalité 13, est comparé au domaine de l'élément courant *poi* dans T_1, {point, pos, x, y, name, nb, contact}. Trois éléments, {name, nb, contact}, sont communs aux deux domaines. N_{inter} étant différent du cardinal de D_2 mais supérieur à N_{inter} initialisé à 0, le processus est réitéré en comparant D_2 au domaine du nœud père de *poi* dans T_1, l'élément *poix* (flèche numéro 1 sur FIG.5). Six éléments sont

communs : {*name, nb, contact, format, unit, datum*}. N_{inter} ayant augmenté, le processus est réitéré à partir du père de *poix, xpoix* (flèche 2) puis à partir du père de *xpoix*, *ALL* (flèche 3). A ce stade, le nombre d'éléments communs n'augmente plus, l'élément de T_1 dont le domaine est le plus proche de celui de *poi* dans T_2 est donc *xpoix,* l'élément courant de l'étape précédente (flèche 4). De cette façon, le concept *cultural-object* de *O* qui n'a pas de correspondant dans T_1 mais pour lequel un mapping a déjà établi avec le terme *poi* dans T_2 sera mis en correspondance avec *xpoix* dans T_1.

4.2 Exploitation des données des documents à rapprocher

Les techniques considérées dans cette section exploitent les données contenues dans les documents XML. Cette nécessaire étude des données s'explique par le fait que les concepteurs des sources XML n'utilisent pas toujours des noms de balise individualisés aussi précis que ceux proposés dans l'ontologie. Une pratique relativement courante consiste à caractériser les éléments décrits au travers de valeurs d'attributs d'éléments, de ce fait, assez généraux. En revanche, dans l'ontologie, si des concepts ont des spécificités, ils sont généralement modélisés sous forme de concepts liés par une relation de spécialisation à des concepts plus généraux. Par exemple, dans l'ontologie OntoTourism avec laquelle nous avons travaillé, plusieurs concepts (*email, fax, tel, web)* ont été définis comme des spécialisations du concept *Contact*. Ces noms de concepts sont utilisés par certains concepteurs de sources comme des noms de balises explicites, mais d'autres concepteurs peuvent préférer représenter les spécificités des contacts via la valuation d'un attribut (en général *Type* ou *Subtype*). Ainsi, un email pourra être représenté comme un contact dont la valeur de l'attribut *Type* est *email*.

Ce choix de représentation a un impact sur les mappings à générer car les noms de concept ou de propriétés qui n'interviennent pas explicitement en tant que balise n'apparaissent pas dans les arbres de DTD. Cet état de fait est pris en compte dans DétectHybrideMap par les mappings conditionnels, qui mettent en correspondance un concept de l'ontologie et un élément d'une DTD lorsque qu'un de ses attributs prend une valeur donnée. Ainsi par exemple, le mapping concernant *email*, est représenté de la façon suivante : Email ↔ // contact[@type = 'email']/@type, ce qui correspond à mettre en correspondance *email* de l'ontologie avec *contact* de la source à intégrer à condition que la valeur de l'attribut *type* de *contact* soit *email*.

La recherche d'éventuels mappings conditionnels avec l'arbre T_1 d'une nouvelle source à intégrer s'appuie sur les mappings conditionnels déjà établis. L'identification de nouveaux mappings de ce type impose de vérifier qu'il existe dans T_1 un élément dont les valeurs dans les documents XML sont celles exprimées dans les conditions des mappings conditionnels déjà établis. Cette vérification sera faite parmi les termes de T_1 pour lesquels aucun mapping n'a encore été établi.

Pour des raisons d'efficacité, l'approche proposée pour effectuer la vérification consiste à partitionner l'ensemble des mappings conditionnels déjà établis en différentes classes selon l'élément avec lequel la mise en correspondance est établie. On obtiendra ainsi, par exemple, la classe des mappings conditionnels établis avec *contact*. A chaque classe de mappings sera associé un ensemble de valeurs, celles intervenant dans les conditions des mappings. {*tel, fax, email, web*} sera l'ensemble de valeurs associées à la classe *contact*. La construction de ces ensembles de valeurs nécessite d'exploiter les données des documents XML proprement dites. Cherchant à minimiser le volume des données à exploiter, seul un sous-ensemble des

documents de l'entrepôt local sera considéré. La taille du sous-ensemble optimal à considérer a été déterminée de manière empirique par expérimentation (cf. section 5).

Cette troisième technique consiste ainsi à travailler sur chaque concept *c* de l'ontologie qui n'a pas encore de correspondant dans T_1 mais pour lequel existe, dans la liste des mappings déjà établis, un mapping conditionnel avec un élément d'une DTD d'arbre T_2. Soit la classe de mapping Cl_i à laquelle appartient le mapping considéré et son ensemble de valeurs V_i, l'élément de T_1 retenu pour le mapping sera celui dont le score, rapport entre le nombre de valeurs appartenant à V_i prises par un attribut de cet élément et le cardinal de V_i, sera le plus élevé. Une proposition de mapping est faite à l'expert pour tous les éléments de T_1 dont le score est non nul. Par exemple, soit le concept *museum* de *O* qui n'a pas encore de correspondant dans T_1 mais pour lequel a déjà été établi le mapping conditionnel suivant dans T_2 : Museum ↔//category[@subtype='museum']/@subtype. Ce mapping appartient à une classe dont l'ensemble des valeurs est {*museum, touristic-site*}. L'attribut *typePOI* de l'élément *xpoix* de T_1 ayant aussi pour liste de valeurs l'ensemble {*museum, touristic-site*}, le score de *xpoix* est 2/2, et les 2 mappings conditionnels ci-dessous sont proposés à l'expert pour validation :

Museum ↔ //*xpoix*[@*typePOI* ='museum']/@*typePOI*
Touristic-site ↔ //*xpoix*[@*typePOI* ='touristic-site']/@*typePOI*

D'autres techniques d'exploitation des données des documents sont nécessaires. Un premier travail a été initié. Il consiste à rapprocher des éléments de noms syntaxiquement différents, qui n'apparaissent pas dans des mappings conditionnels, mais dont les données associées ont le même format (date, monétaire, entier, réel, alpha-numérique, booléen). Une telle technique peut être intéressante pour des concepts ou des attributs ayant un format particulier, par exemple le format date, monétaire, etc. Ainsi, la propriété *coord_1* du concept *lieu_géographique* de l'ontologie n'a pas de correspondant dans l'arbre T_1 de la nouvelle source à intégrer. Il existe pour cette propriété un mapping avec l'élément *coor_x* de T_2. L'élément *coor_x* n'existe pas dans T_1 mais il a le même format de valeurs que l'attribut *X* de l'élément *position* de T_1 sans correspondant dans l'ontologie. Selon cette technique, un mapping entre *coord_1* et *X* pourrait être proposé à l'expert pour validation. Dans l'état actuel de l'implémentation, cette proposition n'est pas faite car deux candidats au mapping sont possibles : *X et Y,* et le choix a été fait de ne faire aucune proposition en cas de mappings possibles multiples.

5 Expérimentations

Les expérimentations de DétectHybrideMap ont été réalisées à partir de données réelles, relatives à des sites culturels, fournies par France Télécom Recherche & Développement (FT) et par le Comité Régional du Tourisme de l'Île-de-France (CRT). L'ontologie OntoTourism relative aux données étudiées comprend 34 concepts, 27 relations (8 sont des relations entre concepts, 19 sont des propriétés de concepts).

5.1 Génération des DTDs et des arbres associés

Les données nous ont été transmises sous forme de documents XML. Les données du CRT, à l'origine représentées dans un format propre, ont été traduites par FT qui y a introduit des balises identiques pour la plupart à celles utilisées dans ses propres documents. Elles se

présentent sous la forme d'un grand fichier XML qui décrit 730 sites culturels de Paris, tous structurés de la même façon et encadrés par une balise, nommée « *ALL* ». Une DTD a été générée à partir de ce fichier. Les données propres à FT se présentent sous la forme de 536 fichiers XML, tous structurés de la même façon et décrivant chacun un site culturel de Paris. Une DTD compatible avec l'ensemble de ces fichiers a été générée.

Les DTDs générées ont été simplifiées afin d'obtenir des arbres ne comprenant que les noms des éléments et des attributs sous forme de balise. Il s'agit d'une copie exacte des DTDs des sources sans cardinalité et sans indication des types.

5.2 Établissement des mappings

Nous avons établi manuellement des mappings entre l'ontologie OntoTourism et, d'une part, l'arbre de DTD des données propres à FT, et d'autre part, l'arbre de DTD des données du CRT. La première liste de mappings (pour les données FT) représente l'ensemble de mappings supposés déjà établis dans notre approche, et la deuxième liste, (pour les données du CRT), ceux que notre système doit être capable de générer automatiquement, à partir de l'arbre de DTD de la nouvelle source. 58 mappings ont été établis entre l'ontologie et l'entrepôt local (couvrant 95 % des termes de l'ontologie) et 51 mappings avec la nouvelle source à intégrer (couvrant 83 % des termes).

5.3 Tests réalisés

L'Algorithme DétectHybrideMap a réussi à détecter 90,1 % des mappings entre les termes de OntoTourism et ceux de CRT. La distribution des mappings détectés suivant la technique utilisée par DétectHybrideMap est donnée dans TAB.1. Les catégories 1 et 2 correspondent aux mappings portant sur des concepts ou des propriétés de l'ontologie identifiés par exploitation du schéma, les catégories 3 et 4 à ceux identifiés par exploitation des données des sources et la dernière technique (non présentée dans le papier) établit un mapping pour une relation de l'ontologie quand celle-ci relie deux concepts de l'ontologie pour lesquels des mappings ont été identifiés. Le mapping consiste à associer la relation au chemin reliant les correspondants des deux concepts dans l'arbre s'il existe (cf. 3.4).

Pour les mappings conditionnels, nous avons fait plusieurs essais afin de trouver la taille minimale de l'échantillon à exploiter pour qu'il soit représentatif. Nous avons d'abord calculé les scores de chaque classe de mappings en exploitant tous les documents XML. Nous avons ensuite réduit le nombre des documents utilisés par pas de cinq pour cent. Nous avons constaté qu'au-dessous de 25%, les scores de chaque classe de mappings diminuaient beaucoup. La taille de l'échantillon a donc été fixée à 25% de l'ensemble des documents.

Les catégories	1	2	3	4	5	Total des mappings
Mappings de France Telecom	19	2	25	7	5	58
Mappings de CRT	16	2	25	5	3	51

Les catégories	1	2	3	4	5	Total des mappings
Mappings de CRT établis manuellement	16	2	25	3	5	51
Mappings détectés	15	2	25*	0	4	46

TAB. 1 – *Répartition des mappings manuels et détectés par catégorie.*

Les mappings conditionnels détectés sont corrects. Néanmoins, le système considère qu'il ne s'agit que de propositions. Il les présente, accompagnés de leur score, au concepteur du système qui doit les confirmer ou les refuser.

Trois mappings dépendant de la technique de comparaison du format des valeurs (catégorie 4) n'ont pas été trouvés par le système. Il s'agit du mapping rapprochant les concepts *appreciation* de l'ontologie et *quality* de la DTD de CRT, et ceux rapprochant les propriétés *coord_1* et *coord_2* du concept *lieu_géographique*, des attributs *X* et *Y* de l'élément *pos* de la DTD de CRT. Pour détecter les mappings de ce type, il sera indispensable de faire une analyse plus profonde, comme une analyse sémantique afin d'améliorer le taux de détection des mappings.

Les deux mappings de relation qui dépendaient de l'existence d'un mapping pour le concept *appreciation* n'ont de ce fait pas été détectés non plus : un mapping de catégorie 1 entre la propriété *provider* du concept *appreciation* de l'ontologie et l'attribut *provider* de la DTD de CRT et un mapping de catégorie 5, pour la relation *is_appreciated_as* reliant les concepts *cultural_object* et *appreciation*.

L'expérimentation montre que DétectHybrideMap a réussi à détecter 21 mappings avec les techniques automatiques de catégorie 1, 2 et 5, et 25 mappings conditionnels avec la technique 3 en interaction avec l'utilisateur, ce qui fait au total 46 mappings sur 51 (90 %). Ce résultat est encourageant.

Enfin, nous avons mesuré le temps d'exécution total du programme en utilisant deux machines différentes. Le temps d'exécution était d'environ 30 secondes sur le serveur Compaq et de 21 secondes sur le terminal Aopen, temps que nous considérons comme satisfaisant.

6 Conclusion

A travers l'intégration au sein d'un entrepôt d'une nouvelle source de données, ce papier a montré comment il était possible de rapprocher différents modèles de connaissances, l'un représentant en RDF-S l'ontologie du domaine, et le second étant un document XML, représentant la nouvelle source à intégrer. Ce rapprochement s'effectue en définissant explicitement des mappings entre les éléments des deux modèles, mappings que nous avons spécifiés et choisis de représenter en Xpath. Le papier montre, d'une part, que les techniques utilisables pour générer ces mappings correspondent à des comparaisons des noms des termes, associées à l'exploitation de la structure des modèles, techniques qu'il serait vraisemblablement intéressant de compléter par l'utilisation de ressources linguistiques. D'autre part, ce papier met l'accent sur la nécessaire exploitation des schémas des modèles comparés mais aussi des données associées. Pour cela, nous proposons des techniques qui exploitent uniquement un sous-ensemble des données. Les expérimentations réalisées sur des données XML réelles fournies par France Telecom R&D donnent des résultats tout à fait satisfaisants.

Ce travail, intégré au projet Piscel3, va se poursuivre, l'objectif étant de parvenir à automatiser la génération de wrappers. A court terme, il nécessite d'approfondir la spécification des mappings établissant un lien avec une relation entre deux concepts de l'ontologie, et leur représentation. Actuellement, pour une relation, la représentation de la partie du mapping concernant la source désigne le chemin reliant les correspondants dans la DTD des éléments de l'ontologie reliés par la relation (cf. section 3.4). Ce choix semble être bien adapté aux relations entre un concept et une propriété. Il n'est pas certain que les relations entre deux concepts doivent être représentées de la même façon.

-

Références

Antoniou, G., F. Van Harmelen (2004). *A Semantic Web Primer*. MIT Press, p. 63-100.

Bisgin O. E. (2005). *Détection semi-automatique de mappings*. Rapport de stage de master recherche en informatique, Université Paris-Sud XI.

Doan, A., J. Madhavan, P. Domingos, A. Halevy (2003). *Ontology matching: A machine learning approach*. In Steffen Staab and Rudi Studer, editors, Handbook on Ontologies in Information Systems, pp. 397–416. Springer-Verlag, Heildelberg (DE).

Do, H. H., E. Rahm (2001). *COMA – a system for flexible combination of schema matching approaches*. VLDB, pp. 610-621.

Euzenat, J., P. Valtchev (2004). *Similarity-based ontology alignment in OWL-Lite*. ECAI, Valencia (ES).

Giunchiglia, F., P. Shvaiko (2004). *Semantic Matching*. The Knowledge Engineering review, 18(3), pp. 265-280.

Melnik, S., H. Garcia-Molina, E. Rahm (2002). *Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching*. ICDE, San Jose CA.

Noy, N. F., M. A. Musen (2001). *Anchor-PROMPT: Using Non-Local Context for Semantic Matching*. Workshop on Ontologies and Information Sharing, IJCAI, Seattle, WA.

Rahm, E., P. Bernstein (2001). *A survey of approaches to automatic schema matching*. VLDB Journal: Very Large Data Bases, 10(4), pp. 334-350.

Shvaiko, P., J. Euzenat (2004). *A survey of Schema-based Matching Approaches*. Technical Report DIT-04-087, Informatica e Telecomunicazioni, University of Trento.

Stumme, G., A. Maedche (2001). *FCA-MERGE: Bottom-Up Merging of Ontologies*. IJCAI.

Thanh Le, B., R. Dieng-Kuntz, F. Gandon (2004). *On Ontology Matching Problems for building a Corporate Semantic Web in a Multi-Communities Organization*. ICEIS (4), pp. 236-243.

Summary

This work deals with integration of heterogenous information sources.The objective is to add a new XML source to an information server thanks to an ontology, which is an explicit and declarative description of the semantic of an application domain. We propose to generate automatically mappings between the ontology and the source to be integrated. We show that this process is based not only on the schemas of the sources but also on data contained in the sources. A prototype, DetectHybrideMap, developed in java, has been used to experiment the approach on real data in the tourism domain provided by France Telecom R&D.