in Proc. of the XXIIème Congrès INFORSID, Biarritz, France, 25-28 May 2004

Intégration de sources de données autonomes par articulation a priori d'ontologies

Ladjel Bellatreche, Guy Pierra, Dung Nguyen Xuan, Dehainsala Hondjack

LISI/ENSMA -Téléport2 - 1, Avenue Clément Ader
86960 Futuroscope – France
{bellatreche, pierra, nguyenx, hondjack}@ensma.fr

*RÉSUMÉ. L'intégration de sources données réparties est une méthode qui vise à offrir aux utilisateurs l'accès à des multiples sources de données à travers des requêtes sur un schéma global. Pour traiter l'hétérogénéité sémantique, considérée comme la plus importante difficulté, l'utilisation d'ontologies est apparue comme l'approche la plus prometteuse pour une possible automatisation. Deux types d'approches d'intégration à base d'ontologies ont jusque là été proposées. Soit des ontologies linguistiques sont utilisées. Cette approche nécessite toujours une intervention humaine. Soit le schéma global est supposé constituer lui-même une ontologie complète du domaine. Les sources n'ont alors plus aucune autonomie. L'approche que nous proposons s'inscrit dans cette dernière approche avec deux différences. (1) Chaque source de donnés contient a priori à la fois sa propre ontologie et les relations sémantiques qui l'articulent avec une ou des ontologie(s) de domaine. (2) Les sources gardent leur autonomie : chaque ontologie locale peut étendre l'ontologie de domaine. Dans notre approche, appelée **a priori**, l'intégration se déroule alors de façon complètement automatique et peut donc se réaliser à très grande échelle. Elle est actuellement prototypée dans plusieurs environnements SGBDOO, SGBDR.*

*ABSTRACT. Data integration is the process that gives users access to multiple data sources though queries against a global schema. To deal with semantic heterogeneity considered as the most important and toughest problem, utilization of ontologies is considered as the most suitable approach for a possible automation. Two categories of data integration approaches based on ontologies were proposed. The first one uses linguistic ontologies to facilitate concepts conciliation. This category requires an human intervention. In the second approach, the global schema is supposed to constitute a complete domain ontology. Consequently, sources do not have any autonomy. Our approach follows the second approach with two major differences. (1) It supposes that each data source contains **a priori** a conceptual ontology and the semantic relations that articulate each local ontology with domain ontology(ies). (2) Sources keep their autonomy : each source ontology may extend the domain ontology. In our a priori approach, data integration is done automatically and scales. It is currently prototyped in various environments.*

MOTS-CLÉS : Intégration, ontologie, bases de données à base ontologique, articulation d'ontologies, PLIB

KEYWORDS: Integration, ontology, ontology based database, articulation of ontologies, PLIB

1 Introduction

L'explosion du nombre de sources d'information accessibles via le Web multiplie les besoins de techniques d'intégration des sources de données autonomes et hétérogènes. L'intégration des données est le processus par lequel plusieurs sources de données autonomes, réparties et sous forme hétérogène (où chaque source est associée à un schéma local) sont intégrées sous forme de source unique représentée par un schéma global. Des exemples de besoins d'intégration de données se trouvent par exemple dans le "Peer-to-Peer" (Abiteboul et al., 2002) les entrepôts de données (Bellatreche et al., 2001), et le commerce électronique (Omelayenko et al., 2001).

Formellement, un système d'intégration de données est un triplet I : <G, S, M>, où G représente le schéma global (défini sur un alphabet A_G) qui modélise le schéma intégré, S est l'ensemble des schémas des sources (définis sur un alphabet A_S) décrivant la structure des sources participantes au processus d'intégration, et M est une correspondance entre G et S qui établit la connexion entre les éléments du schéma global et ceux des sources. Pour interroger le système intégré, les requêtes sont exprimées en termes de constructions du schéma global G.

De nombreux systèmes d'intégration ont été proposés dans la littérature (Castano et al., 1997, Lawrence et al. 2001, Chawathe et al. 1994, Reynaud et al., 2003, Levy et al. , 2001). La principale difficulté pour ces systèmes est d'interpréter automatiquement la signification (la sémantique) des données *hétérogènes* et *autonomes* (différents conflits) (Goh et al. 1999, Pitoura et al., 1995). Dans la première génération de systèmes d'intégration (e.g., TSIMMIS (Chawathe et al. 1994)), la signification des données n'est pas représentée explicitement. Les correspondances entre le schéma global et les schémas locaux sont réalisés manuellement et encodés dans des définitions de vues. Avec l'avènement des ontologies (Wache et al., 2001), un progrès important a pu être réalisé dans l'automatisation du processus d'intégration de sources hétérogènes grâce à la représentation explicite de la signification des données. Plusieurs types d'ontologies ont été utilisés dans les systèmes d'intégration: linguistiques (Castano et al., 1997) ou formelles (Hakimpour et al., 2002). Les ontologies linguistiques permettent une automatisation partielle du processus d'intégration avec la supervision d'un expert humain. Les ontologies formelles permettent une automatisation effective pour autant que chaque source référence exactement la même ontologie, sans possibilité d'extension ou d'adaptation. Quelques systèmes ont été développés autour de cette hypothèse comme le projet Picsel2 (Reynaud et al., 2003) pour intégrer les services Web, le projet COIN pour échanger les données financières (Goh et al., 1999). Les limitations de ces systèmes est qu'une fois l'ontologie partagée définie, chaque source doit utiliser le vocabulaire commun. L'ontologie partagée est en fait un schéma global, et, en conséquence, chaque source locale a moins d'autonomie. Dans de nombreux domaines comme le Web service,

l'e-procurement, la synchronisation des bases de données réparties, le *nouveau défi* est de permettre une intégration entièrement automatique des sources de données gardant une autonomie significative. La transformation du schéma à travers lequel une information est représentée sous forme de données nécessitant d'interpréter la signification des différentes données, nous pensons que ceci n'est possible qu'à deux conditions : (1) chaque source doit représenter explicitement la signification de ses propres données ; c'est la notion d'ontologie locale qui doit exister dans chaque source ; et (2) il existe une ontologie partagée (ontologie de domaine) et chaque ontologie locale référence explicitement l'ontologie partagée pour définir les relations sémantiques existantes entre les concepts des ontologies locales et globale (articulation).

Notre approche n'élimine pas la nécessité d'une réflexion humaine pour identifier deux conceptualisations différentes d'une même réalité. Mais, elle demande que cette réflexion soit faite a priori, lors de la mise à disposition de la source de données, et non a posteriori, pendant la phase d'intégration. Notre approche d'intégration est ainsi basée sur trois principes : (1) chaque source participante dans le processus d'intégration *doit* contenir sa propre ontologie ; une telle source est appelée base de données à base ontologique (BDBO), (2) chaque ontologie locale s'articule a priori avec une (ou des) ontologie(s) partagée(s), et (3) chaque ontologie locale étend l'ontologie partagée pour satisfaire ses besoins.

Dans ce contexte, le processus d'intégration est automatique et se décompose en deux étapes : une intégration des ontologies puis une intégration des données. A notre connaissance, notre travail est le premier à traiter le problème d'intégration en proposant qu'une ontologie formelle soit explicitement représentée dans chaque source de données, et que les articulations avec les ontologies partagées soient définies a priori. Dans cet article, comme dans (Mitra et al., 2000), l'articulation entre l'ontologie globale et les ontologies locales sont exprimées par un ensemble de relations de subsomption.

Notre travail s'inscrit dans le contexte du projet *OntoDB*, lancé depuis plusieurs années dans notre laboratoire pour permettre la gestion, l'échange, l'intégration et l'interrogation de données structurées associées à leurs ontologies. Le domaine d'application privilégié est le commerce électronique professionnel.

Le reste de cet article est organisé en quatre sections. La section 2 présente le domaine cible pour lequel notre approche a été développée. La section 3 présente la structure que nous proposons de donner a priori aux différentes sources pour rendre leur intégration automatique faisable. On présente d'abord le modèle d'ontologie utilisée, puis le modèle de base de données à base ontologique. La section 4 décrit les algorithmes d'intégration dans deux scénarii correspondant à des cas pratiques dans le commerce électronique professionnel. Enfin la section 5 conclut notre présentation.

2. Notre domaine d'application cible

Notre domaine d'application cible est le commerce électronique professionnel et l'échange de données techniques dans le domaine des composants industriels. Dans ces domaines techniques, où les notions d'interchangeabilité et de normalisation sont très développées, un vocabulaire technique consensuel existe pour les termes essentiels de chaque domaine. L'intégration de données dans ce domaine présente néanmoins trois difficultés : **(1)** ces vocabulaires techniques n'existent pas, a priori, sous une forme exploitable automatiquement, **(2)** ces vocabulaires ne couvrent pas les innovations qui apparaissent de façon continue, et **(3)** chaque base de données, fournisseur ou utilisateur, et chaque catalogue électronique même s'il référence le vocabulaire commun, utilise une structure et une terminologie qui lui est propre.

Pour résoudre la première difficulté, nous avons proposé une approche, un modèle et des outils permettant de formaliser ces vocabulaires consensuels sous forme d'ontologies. Ce sont les dictionnaires de référence PLIB qui couvrent un nombre de plus en plus important de domaine. Ces ontologies sont logiques au sens du 2.2.3. Si deux classes ne sont connectées, directement ou indirectement, par aucune relation de subsomption, leur intersection est considérée comme vide. Deux propriétés sont ou identiques ou différentes, elles ne peuvent être "voisines" (seules des fonctions de conversions peuvent exister, non encore implémentées). A partir de ces ontologies, l'approche d'intégration que nous proposons est basée sur les deux hypothèses suivantes : (a) il existe une ontologie de domaine recouvrant la totalité des termes consensuels, et (b) chaque source de données peut se définir en terme d'une ontologie qui lui à propre.

L'approche d'intégration a priori est alors basée sur deux principes :

1. chaque source de données contient explicitement à la fois son ontologie et les correspondances existant entre une ontologie et ses sources, c'est la notion de base de données à base ontologique.

2. chaque source visant à être intégrée contient également l'articulation entre son ontologie et l'ontologie de domaine à travers des relations de subsomptions respectant certaines propriétés.

Cette approche orientée entité dans sa représentation de la correspondance existante entre le niveau global et le niveau local, permet alors l'adjonction complètement automatique d'une nouvelle source dans un système intégré, que ce soit dans une perspective d'entrepôt, ou dans une perspective médiateur. Nous présentons dans les deux sections suivantes, d'abord les éléments de base de l'approche d'intégration dans le projet OntoDB, puis les algorithmes d'intégration au sein d'un entrepôt pour deux scénarios particuliers.

3 Composant de la démarche d'intégration dans le projet OntoDB

Nous présentons dans cette section les deux éléments de base de notre approche d'intégration : le modèle d'ontologie utilisé et méthode d'articulation entre l'ontologie globale et les ontologies locales, le modèle de base de données à base ontologique.

3.1 Le modèle d'ontologie PLIB

Définir un modèle d'ontologie destiné à être utilisé pour produire ensuite des conceptualisations devant faire l'objet de consensus, suppose d'identifier d'abord les structures de modélisation qui, dans un domaine donné, peuvent ou non déboucher sur des consensus. Dans le domaine technique en tout cas, dans le domaine du commerce électronique professionnel aussi, et probablement dans beaucoup d'autres domaines, une différence très nette apparaît entre les entités du domaine et les propriétés qui les caractérisent, d'une part, les relations entre entités et les structures de classification permettant de les organiser, d'autre part. Tous les produits et services mettent en œuvre des composants ou des entités constituantes. Ces composants font l'objet de transactions entre acteurs humains et sont donc déjà identifiées et caractérisés (informellement) de façon consensuelle. Par contre, le fait que telle entité soit rangée à tel endroit, le fait qu'elle soit convertissable de telle façon en telle autre entité, ou la structure de classification permettant d'organiser l'ensemble des entités dans une entreprise particulière dépendant beaucoup plus du contexte, c'est-à-dire à la fois de l'application et de l'organisation cible.

Le modèle d'ontologie PLIB est un modèle orienté entité-propriété (pas d'associations) (Pierra, 2003), (Pierra et al., 2003). Il vise à décrire l'ensemble des entités, supposées consensuelles, existant dans un domaine donné par l'intermédiaire de propriétés permettant de caractériser toutes les entités du domaine. Chaque propriété est définie dans le domaine d'une classe d'entités. Elle n'a de sens que pour cette classe et ses éventuelles sous-classes.

Dans le but de s'affranchir du caractère contextuel des classifications, dans les ontologies PLIB ayant vocation à être partagées appelée ontologies de référence, une classe ne doit être créée que si elle est indispensable pour servir de domaine à une propriété qui ne serait pas compréhensible dans le contexte de sa superclasse. Inversement, une propriété peut être définie dans le contexte d'une classe même si elle ne s'applique pas à toutes les instances ou sous-classes de cette classe. La seule condition est qu'elle soit définissable de façon non ambigu. Les hiérarchies des ontologies de référence partagées PLIB sont donc extrêmement "plates". Elles ne définissent pas tous les termes possibles existant dans un domaine, mais au contraire elles visent à définir un

vocabulaire canonique minimal. Ce vocabulaire doit seulement permettre de décrire d'une façon unique, par une classe d'appartenance et par un ensemble de valeurs de propriétés, toutes les entités qui font l'objet d'une compréhension commune par les experts d'un domaine. Toute entité existant dans un domaine peut donc ensuite se décrire, soit en termes de l'ontologie de référence du domaine, soit comme une spécialisation d'une des classes de l'ontologie en lui rajoutant éventuellement des propriétés supplémentaires (la classe spécialisée peut, dans le pire des cas, être une autre classe racine du domaine si l'entité décrite est complètement nouvelle par rapport au domaine). Le Modèle d'ontologie PLIB offre donc une relation d'extension particulière, appelée "case_of" (est_un_cas_de), permettant à un utilisateur de définir sa propre ontologie à partir d'une ontologie de référence. Une classe case_of d'une autre classe est subsumée par celle-ci. Elle peut importer de celle-ci un sous-ensemble quelconque de propriétés qui y sont définies. Ceci permet de définir à partir d'une même ontologie de référence, des ontologies utilisateurs de structures très différentes.

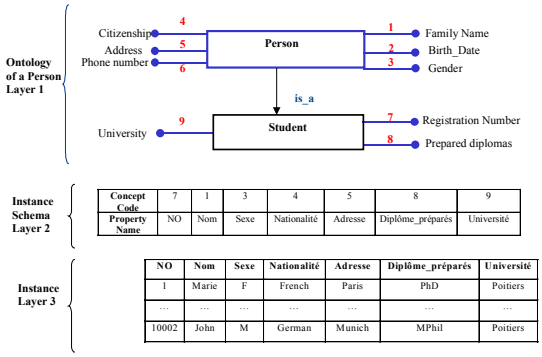

Figure 1: Un exemple de spécialisation d'une ontologie globale

3.1.2 Principe d'engagement sur une ontologie de référence (SSCR)

Quand une ontologie PLIB est partagée par plusieurs sources (par ce que les sources s'engagent sur des définitions ontologiques qui ont été acceptées et éventuellement normalisées) chaque source garde une grande autonomie. Elle peut définir sa propre hiérarchie de classes, et, si besoin est, rajouter les propriétés qui n'existent pas dans l'ontologie partagée.

Néanmoins, s'engager sur une ontologie de référence signifie respecter la double contrainte suivante (appelée SSCR smallest subsuming class reference).

- Toute classe locale doit référencer, par la relation case_of, la plus petite class subsumante existante dans la hiérarchie de référence si ce n'est pas la même que celle de

sa propre super classe; elle ne référence aucune classe que s'il s'agit d'un concept complètement nouveau pour le domaine,

- Toute propriété nécessaire à l'ontologie locale et existant dans l'ontologie de référence doit être importée à travers la relation case_of.

3.1.3 Une définition formelle d'une ontologie PLIB

Formellement, une ontologie PLIB peut être définie comme un quadruplet : O <C, P, Sub, Applic>, avec : C: l'ensemble des classes utilisées pour décrire les concepts d'un domaine donné (comme les service de voyages (Reynaud et al. 2003), les pannes des équipements, les composants électroniques, etc.) ; P: l'ensemble des propriétés utilisées pour décrire les instances de l'ensemble des classes C. Nous supposons que définit toutes les propriétés consensuelles dans le domaine susceptibles d'être représentées dans une base de données, Sub est la relation de subsomption (is-a et case-of) (Figure 1, 2) de signature Sub : $C \rightarrow 2^{C}$ [1], qui, à chaque classe de l'ontologie, associe ses classes subsumées directes [2]. Sub définit un ordre partiel sur C, et Applic est une fonction Applic : $C \rightarrow 2^{P}$, qui associe à chaque classe de l'ontologie les propriétés qui sont applicables pour chaque instance de cette classe. Les propriétés qui sont applicables sont héritées à travers la relation is-a et (partiellement) importées à l'aide de la relation de case-of.

Notons comme déjà indiqué que les définitions ontologiques sont intentionnelles. Le fait qu'une propriété soit applicable pour une classe signifie qu'elle est rigide, c'est-à dire essentielle pour chaque instance de la classe (Guarino et al., 2000). Cela ne signifie pas qu'une valeur sera explicitement représentée pour chaque instance dans la base de données. Dans notre approche, le choix des propriétés effectivement représentées est fait parmi les propriétés applicables au niveau du schéma.

Exemple 2 *Figure 2 donne un exemple d'une ontologie de deux classes = { Person et PhD Student }. Soit l'ensemble de propriétés qui caractérise ces classes. Les propriétés dans P seront affectées aux classes de l'ontologie. Cette affectation garantie que chaque classe aura ses propres propriétés rigides. La fonction de subsomption Sub définit la relation case-of entre les classes (par exemple, la classe Person subsume la classe Phd Student).*

[1] on utilise le symbole 2^{C} pour dénoter la power set de C .

[2] C_1 subsume C_2 si et seulement si $\forall x \in C_2, x \in C_1$

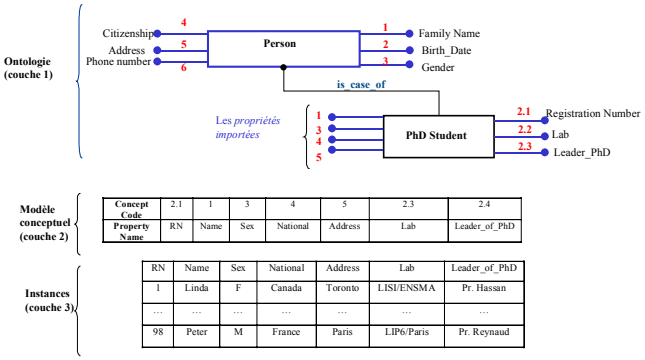

Figure 2: Un exemple d'une extension d'une ontologie globale

3.2 Les bases de données à base ontologique

Avec la croissance exponentielle du nombre de sources de données apparaissant sur le Web, les méthodes d'intégration traditionnelles imposant une activité manuelle de l'administrateur central apparaissent de moins en moins faisable ou acceptable. En ce qui concerne les données semi-structurées, le besoin d'intégration automatique est adressé à travers la notion de meta-données représentée par exemple en RDF ou RDFS. L'idée qui préside à cette approche est que si le travail sémantique d'intégration ne peut être réalisé a posteriori, alors elle doit être effectué a priori par les auteurs de sources documentaires. Ceci est effectué en ajoutant à des documents (sémantiquement et terminologiquement hétérogènes) des meta-données qui référencent une ontologie commune et fournissant donc une interface intégrée et homogène pour la recherche de documents pertinents pour une requête appartenant au domaine de l'ontologie.

L'approche d'intégration que nous proposons correspond à la mise en œuvre de la même idée dans l'univers des bases de données.

Dès lors qu'un responsable de base de données connecte celle-ci au Web, c'est bien pour en rendre le contenu facilement accessible. A partir du moment où des ontologies de domaine existent et sont acceptées (e.g., normalisées) c'est bien dans les termes de ces ontologies que les utilisateurs vont rechercher l'information. L'approche que nous proposons consiste alors à représenter dans les bases de données non seulement leurs ontologies et leurs schémas propres, mais également l'articulation avec la ou les ontologie(s) partagée(s) sur lesquelles elles s'engagent, et la capacité de répondre à une requête en terme de cette ou ces ontologies. C'est la notion de base de données à base

ontologique BDBO ou ontology-based database "OBDB" que nous avons développé et sommes en train de valider dans différents environnements.

Afin de représenter explicitement leurs ontologies, les BDBO doivent avoir une structure différente de celle des bases de données usuelles. Ce modèle d'architecture, appelé OntoDB permet alors de traiter, de façon générique, aussi bien les données que les ontologies, mais aussi les liens entre ontologies et données. Comme les bases de données traditionnelles les BDBO possèdent deux parties (partie droite de la Figure 3): une partie contenu et une partie méta-données qui décrit les tables, les colonnes, les clés étrangères, etc. Mais une BDBO possède, en plus, deux autres parties (partie gauche de la Figure 3). Celles-ci représentent l'ontologie et la structure de l'ontologie qui permet tous les traitements génériques sur les ontologies.

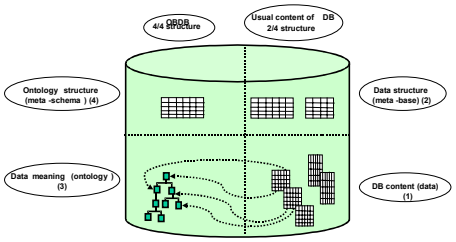

Figure 3: Architecture d'une base de données à base ontologique

3.2.1 Une définition formelle d'une base de données à base ontologique

Dans une BDBO, tout élément représenté dans le schéma, classe ou propriété doit appartenir à l'ontologie. De sorte que le schéma est un sous-ensemble de l'ontologie, chaque entité représentée correspondant à une classe et ses attributs correspondant aux propriétés applicables choisies (on fait abstraction ici du découpage éventuel résultant des opérations de normalisation, une vue étant, dans tous les cas, créée pour représenter la population de chaque classe). Pour simplifier le propos, nous supposons désormais que seules les classes feuilles sont directement instanciables. Les classes non feuilles sont supposées "abstraites", c'est-à-dire que leur population est l'union des populations de leur sous-classes.

Comment représenter la population de telles classes, sachant qu'un des services essentiels d'une BDBO est de pouvoir répondre à des requêtes portant sur une classe quelconque (feuille ou non-feuille) de la hiérarchie de subsomption? Cette question met en évidence une différence essentielle existant entre une base de données objet et une BDBO.

Dans une base de données objet, les propriétés de et qui sont également des propriétés de sont évidentes: ce sont les propriétés définies au niveau de et héritées par et puisque c'est la seule manière de partager des propriétés (dans la figure 4). Dans une BDBO, s'il est clair que les propriétés qui n'ont pas de sens pour ne doivent pas être retenues (i.e., et), deux réponses restent néanmoins possibles pour définir la population de :

1. soit c'est la projection sur les propriétés applicable de de l'ensemble des instances des sous-classes de ; selon la sous-classe dont elles proviennent: différentes instances pourront alors être décrites par différentes propriétés. Par exemple : et ; (certaines valeurs étant à nulle.
2. Soit c'est la projection de la population ci-dessus sur les seuls attributs valués dans toutes les sous-classes (c'est-à-dire).

Le choix entre ces deux représentations doit être laissé à l'utilisateur.

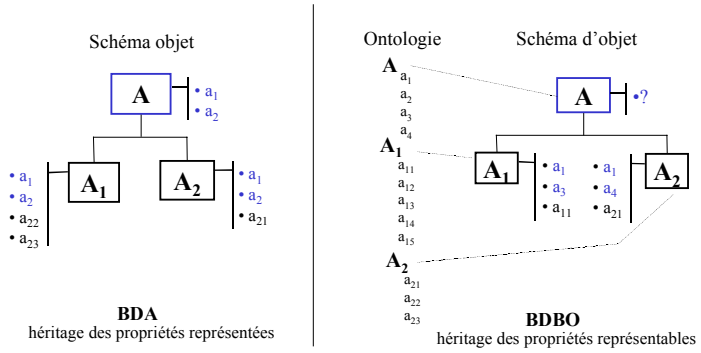

Figure 4: Différence entre une BDOO et une BDBO

Le schéma d'une classe abstraite non feuille étant aussi défini, on peut alors définir formellement la notion de OBDB. Une OBDB est un quadruplet <O I, Sch, Pop>, avec: O représente son ontologie (O : <C, P, Sub, Applic>); I représente l'ensemble des instances de la base de données; Sch : $C \rightarrow 2^P$ associe à chaque classe de l'ontologie c_i de C les propriétés qui sont effectivement utilisées pour décrire les instances de la classe c_i. Sch a deux définitions se basant sur la nature de chaque classe (si elle est feuille ou pas).

Le schéma de chaque classe feuille c_i est explicitement défini. Il doit assurer l'équation suivante: $\forall c_i \in C$, $Sch(c_i) \subset Applic(c_i)$ [1]
(seules les propriétés applicables peuvent être utilisées pour décrire les instances de la classe).

Le schéma d'une classe non feuille est calculé. Il est défini comme l'intersection entre les propriétés applicables de c_j et l'intersection entre les propriétés associées aux valeurs dans toutes les sous classes $c_{i,j}$ de c_j (cela correspond au cas 2 ci dessus).

$$Sch(c_j) = Applic(c_j) \cap (\cap_i Sch(c_{i,j})) \qquad [2]$$

Une définition alternative (cas 1, ci dessus) peut également être utilisée pour créer le schéma d'une classe non feuille où les instances sont complétées par des valeurs nulles :

$$Sch'(c_j) = Applic(c_j) \cap (\cup_i Sch(c_{i,j})) \qquad [3]$$

$Pop : C \rightarrow 2^I$ associe à chaque classe (feuille ou non) ses propres instances.

4 Algorithmes d'intégration les bases de données à base ontologiques

Dans cette section, nous présentons deux exemples d'algorithmes permettant d'intégrer automatiquement des sources de données ayant la structure d'une BDBO au sein d'un entrepôt ayant lui même une structure d'une BDBO. Un cas d'application typique est celui des offres de voyages proposées par des "tour opérators" consolidés au sein d'une grosse base de données telle que celle du Havas-Americain Express. Chaque fournisseur référence la même ontologie de domaine (correspondant par exemple à la DTD définie par l'organisme de normalisation de la profession (Reynaud et al., 2003) et il peut avoir ses propres extensions.

Soit $S = \{SB_1, SB_2, \ldots, SB_n\}$ l'ensemble des sources de données participant au processus d'intégration. Chaque source SB_i est définie comme suit: $SB_i : <O_i, I_i, Sch_i, Pop_i>$. Nous supposons que chaque source de données référence une ontologie partagée O, en respectant la condition (SSCR). Chaque source a ainsi été conçue en trois étapes:

(1) le DBA de chaque source a défini sa propre ontologie $O_i : <C_i, P_i, Sub_i, Applic_i>$, **(2)** le DBA de chaque source a choisi pour chaque classe feuille les propriétés qui seront valuées en définissant $Sch_i : C_i \rightarrow 2^{Pi}$; et **(3)** Le DBA choisit une implémentation de chaque classe feuille c_i (e.g., afin d'assurer la troisième forme normale), et il défini ensuite $Sch(c_i)$ comme une vue sur l'implémentation de c_i.

Deux scénarii d'intégration sont ici présentés :

- **FragmentOnto**: l'ontologie partagée ayant été définie avant que les opérateurs ne construisent leurs bases de données, celles-ci référencent directement l'ontologie partagée: chaque ontologie locale est un sous ensemble de l'ontologie partagée.

- **ExtendOnto**: source définit sa propre ontologie (elle n'instancie aucune classe de l'ontologie partagée). Par contre la condition SSCR est respectée et l'on souhaite enrichi automatiquement l'ontologie partagée.

4.1 Algorithme d'intégration pour FragmentOnto

Cette approche d'intégration suppose que l'ontologie partagée est entièrement suffisante pour couvrir toutes les sources locales. Une hypothèse de ce type est utilisée par exemple dans le projet Picsel2 (Reynaud et al., 2003) pour intégrer des services Web (agences de voyage) ou dans le projet COIN (Goh et al., 1999). Dans ce cas l'autonomie des sources se limite à (1) sélectionner un sous ensemble pertinent de l'ontologie partagée (classes et propriétés) et (2) concevoir le schéma local de la base de données.

L'ontologie O_i de chaque source SB_i étant un fragment de l'ontologie partagée O; elle se définit comme le quadruplet O_i : $<C_i, P_i, Sub_i, Applic_i>$, avec :

$C_i \subseteq C$;

$P_i \subseteq P$;

$\forall c \in C_i$, $Sub_i(c) \subseteq Sub(c)$;

$\forall c \in C_i$, $Applic_i(c) \subseteq Applic(c)$.

Pour intégrer ces sources au sein d'une BDBO il suffit de trouver l'ontologie, le schéma et la population du système intégré. Le système intégré est donc défini comme le quadruplet Int : $<O_{Int}, Sch_{Int}, Pop_{Int}>$. Maintenant la question à laquelle il convient de répondre porte sur la structure de chaque élément de Int?

L'ontologie du système intégré est O ($O_{Int} = O$), la population du système intégré est l'union des populations des différentes sources : ($I_{In} = \cup_i I_i$), le schéma du système intégré est défini pour chaque classe comme suit:

$$Sch_{Int}(c) = (\cap_{i \in \{1..n \mid Schi(c) \neq \phi\}} Sch_i(c)) \qquad [4]$$

Cette définition assure que les instances du système intégré ne seront pas complétées par des valeurs nulles (cf. (2) de 3.2.1). Pour chaque classe, seules les propriétés valuées dans toutes les sources de données seront préservées. Si dans certaines sources on trouve des classes vides, elles ne seront pas prises en compte pour calculer les propriétés fournies par toutes les sources [3] .

La population de chaque classe du système intégré est définie comme suit:

$$Pop\ Int(c) = proj_{sch(c)} \cup_i Pop_i(c) \qquad [5]$$

où proj représente l'opération de projection définie dans les BDDRs.

[3]Cette approche correspond à la formule (2). Une approche basée sur la formule (3) peut être également exprimée.

4.2 Algorithme d'intégration pour ExtendOnto

De nombreuses applications conçues autour de l'approche OntoDB exigent plus d'autonomie, même dans le domaine du commerce électronique professionnel qui est le notre:

- la classification de chaque source doit pouvoir être complètement différente de celle de l'ontologie partagée, et
- certaines spécialisations de classe et certaines propriétés n'existant pas dans l'ontologie partagée doivent pouvoir être ajoutées dans les ontologies locales.

Ce cas est très différent du précédent du fait que chaque source a sa propre ontologie et ses classes spécifiques. Néanmoins, chaque ontologie référence autant que possible (i.e., en respectant la condition SSCR) l'ontologie partagée .

Les articulations entre O_i et O peuvent être définies comme: $M : C \rightarrow 2^{Ci}$, avec $M_i(c) = \{$les plus grandes classes de C_i subsumées par $c_i\}$. Contrairement au cas précédent, chaque source SB_i est désormais définie comme quintuple: : $<O_i, I_i, Sch_i, Pop_i, M_i>$. Dans ce cas également, une automatisation du processus d'intégration est possible. Pour ce faire, nous devons trouver la structure finale de l'BDBO constituant le système intégré F : $<O_F, Sch_F, Pop_F>$.

Redéfinissons d'abord la structure de l'ontologie intégrée O_F : $<C^F, P^F, Sub^F, Applic^F>$, où chaque élément de O_F est défini comme suit:

$C^F = C \cup_{(i \mid 1 \le i \le n)} C_i$ -- on enrichit l'ontologie partagée
$P^F = P \cup_{(i \mid 1 \le i \le n)} P_i$
$\forall c \in C, Sub^F(c) = Sub(c) \cup_{(i \mid 1 \le i \le n)} M_i(c)$,

$$Applic^F(c) = \begin{cases} Applic(c), if : c \in C \\ Applic_i(c), if : c \in C_i \wedge c \notin C \end{cases}$$

Définissons ensuite la population et le schéma du système intégré:

- La population totale est $I^F = \cup_i I_i$
- Ensuite, la population Pop_F de chaque classe (c) est calculée d'une manière récursive en utilisant un parcours post fixé de l'arbre C_F. Si appartient à une C_i et n'appartient pas à C, sa population est donnée par : $Pop_F(c) = Pop_i(c)$

sinon (i.e., appartient à l'ontologie partagée), $Pop_F(c)$ est définie par l'équation suivante:

$$Pop_F(c) = \cup_{(cj \in SubF(c))} Pop_F(c_j) \qquad [6]$$

- Finalement, le schéma de chaque classe du système intégré est calculé en utilisant le même principe que la population de en considérant les classes feuilles et non

feuilles. Les schémas des classes feuilles sont explicitement définies (cf. (1) de 3.2.1). Concernant les classes non feuilles, si c n'appartient pas à C, mais à l'un des ensembles de classes C_i , le schéma peut être calculé en utilisant la formule (2) (resp. 3). Sinon (i.e., si c appartient à l'ontologie partagée), son schéma est calculé d'une manière récursive en utilisant un parcours post fixé de l'arbre C_F

$$Sch_F(c) = Applic(c) \cap (\cap_{(cj \mid cj \in SubF(c) \wedge PopF(ci) \neq \phi)} Sch^F(c_j)) \qquad [7]$$

Cela montre qu'il est possible d'offrir aux sources locales une large autonomie et permet également une construction automatique du système intégré d'une manière déterministique et exacte. A notre connaissance notre approche d'intégration est la première qui réconcilie les deux exigences.

Il est important de noter que lorsque toutes les sources de données utilisent une ontologie indépendante sans référencer une ontologie partagée, d'une part, l'intégration automatique peut néanmoins se produire (i.e., lecture de toutes les données dans le même entrepôt), et d'autre part, la tâche d'articulation de ces ontologies sur l'ontologie du système receveur peut être faite manuellement, par le DBA. Ensuite une nouvelle intégration peut être réalisée automatiquement comme dans le cas ExtendOnto.

5 Conclusion

Dans ce travail nous avons proposé une méthode complètement automatique d'intégration de sources de données structurées hétérogènes et autonomes. Cette approche, appelée intégration par articulation a priori d'ontologies, suppose l'existence d'une (ou plusieurs) ontologie(s) de domaine mais, elle laisse chaque source autonome quand à la structure de sa propre ontologie.

Au lieu de réaliser l'intégration des ontologies a posteriori, comme c'est le cas dans toutes les approches classiques, notre approche exige de l'administrateur de chaque source à intégrer (1) que sa base de données contienne une ontologie et (2) qu'il ajoute a priori à cette ontologie les relations (articulations) existantes entre celle-ci et l'ontologie de domaine. Cette hypothèse est réaliste dans tous les secteurs où des ontologies de domaines existent ou apparaissent, et où chaque administrateur qui publie sa base de données souhaite à la fois lui conserver sa structure propre, et la rendre accessible à des usagers de façon homogène à travers une ontologie de domaine. C'est en particulier le cas dans le cadre du commerce électronique professionnel. Elle exige également que chaque source publie non seulement ses données, mais également son ontologie, ce qui correspond à une généralisation de l'approche de type meta-données utilisée pour les sources semi-structurées. Nous avons proposé une mise en œuvre de cette approche pour les données structurées à travers notre modèle de base de données à base ontologique.

Nous avons présenté ici la mise en œuvre de cette approche dans une perspective d'intégration physique des données au sein d'un entrepôt. Nous utilisons des ontologies PLIB qui sont bien adaptés pour représenter les entités d'un domaine fortement structuré et les propriétés intrinsèques qui les caractérisent. Ce type d'ontologie semble bien adapté au domaine du commerce électronique professionnel. De plus, ce modèle d'ontologie, formellement défini dans le langage EXPRESS, est associé à une structure d'échange permettant de représenter de façon neutre tant des ontologies que des instances référençant ces ontologies. Les algorithmes que nous avons présentés permettent alors, à travers un tel échange, l'intégration automatique du contenu de toute nouvelle source autonome au sein d'un entrepôt déjà constitué à partir de l'ontologie de domaine. Cette intégration peut même étendre automatiquement, si on le souhaite, l'ontologie de domaine en respectant la compatibilité ascendante. La même approche peut cependant être mise en œuvre avec d'autre modèle d'ontologie, telle que OWL. C'est la seule approche, à notre connaissance, permettant d'intégrer de façon automatique des sources autonomes.

Beaucoup de questions, apparaissent actuellement ouvertes et nécessitent parallèlement aux travaux en cours, d'être explorées : (1) outre les domaines cibles, quels sont les domaines d'application de l'approche proposée avec des ontologies PLIB (2) faisabilité et intérêt d'utiliser la même approche avec des modèles d'ontologie différents, par exemple OWL, (3) enrichir notre modèle d'ontologie et d'articulation entre ontologie afin d'y représenter explicitement les dépendances fonctionnelles pouvant exister entre propriétés, et les expressions de classes, voire le lien existant entre ontologie locale et les schémas relationnels finaux existant après normalisation.

Bibliographie

Abiteboul S., Benjelloun O., Manolescu I., Milo T., Weber R., "Active xml: Peer-to-peer data and web services integration", *Proceedings of the International Conference on Very Large Databases*, 2002, p. 1087-1090

Bellatreche L., Karlapalem K., Mohania M., "Some issues in design of data warehousing systems", *Developing Quality Complex Data Bases Systems: Practices, Techniques, and Technologies, Edited by Dr. Shirley A. Becker,* Idea Group Publishing, 2001, , p. 125-172.

Castano S, Antonellis V., "Semantic dictionary design for database interoperability", *Proceedings of the International Conference on Data Engineering (ICDE)*, , April 1997, p. 43-54

Castano S., Antonellis V., Vimercati S.D.C., "Global viewing of heterogeneous data sources", *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, n° 2, 2001, p. 277-297

Chawathe S.S., Garcia-Molina H., Hammer J., Ireland K., Papakonstantinou Y., Ullman J.D., Widom J., "The tsimmis project: Integration of heterogeneous information sources",

Proceedings of the 10th Meeting of the Information Processing Society of Japan, Marsh 1994, p. 7-18.

Goh C.H., Bressan S., Madnick E., Siegel M.D., "Context interchange: New features and formalisms for the intelligent integration of information", *ACM Transactions on Information Systems*, vol. 17, n°3, 1999, p. 270-293.

Guarino N., Welty C.A., "Ontological analysis of taxonomic relationships", *Proceedings of 19th International Conference on Conceptual Modelling (ER'00)*, October 2000, p. 210-224,.

Hakimpour F., Geppert A., "Global schema generation using formal ontologies", *Proceedings of 21st International Conference on Conceptual Modelling (ER'02)*, October 2002, p. 307-321.

Lawrence R., Barker K., "Integrating relational database schemas using a standardized dictionary", *Proceedings of the ACM Symposium on Applied Computing (SAC)*, Marsh 2001, p. 225-230.

Levy A.Y., Rajaraman A., Ordille, J.J., "The world wide web as a collection of views: Query processing in the information manifold", *Proceedings of the International Workshop on Materialized Views: Techniques and Applications (VIEW'1996)*, June 1996, p. 43-55,.

Omelayenko B., Fensel, D., "A two-layered integration approach for product information in b2b e-commerce", *Proceedings of the Second International Conference on Electronic Commerce and Web Technologies*, September 2001, p. 226-239.

Pierra G., "Context-explication in conceptual ontologies: The plib approach", Special track "Data Integration in Engineering, Concurrent Engineering (CE'2003) - the vision for the Future Generation in Research and Applications, July 2003, p. 243-254 (to appear in special issue of JAMS - Journal of Advanced Manufacturing Systems).

Pierra G., Potier, J.C., Sardet, E., "From digital libraries to electronic catalogues for engineering and manufacturing", *International Journal of Computer Applications in Technology (IJCAT)*, vol. 18, 2003, p. 27-42.

Pitoura E., Bukhres O.A., Elmagarmid A.K., "Object orientation in multidatabase systems", *ACM Computing Surveys*, vol. 27, n° 2, June 1995, p. 141-195

Reynaud C., Giraldo G., "An application of the mediator approach to services over the web", *Special track "Data Integration in Engineering, Concurrent Engineering (CE'2003) - the vision for the Future Generation in Research and Applications*, July 2003, p. 209-216

Wache H., Vögele T., Visser U., Stuckenschmidt H., Schuster G., Neumann H., Hübner S., "Ontology-based integration of information - a survey of existing approaches", *Proceedings of the International Workshop on Ontologies and Information Sharing, August 2001, p. 108-117.*

Mitra P., Wiederhold G., Kersten M. L., "A Graph-Oriented Model for Articulation of Ontology Interdependencies'', Proceedings of the 7th International Conference on Extending Database *Technology (EDBT'00), 2000, p. 86-100*