

Un modèle et une algèbre pour les systèmes de gestion d'ontologies

Gilles Falquet* Claire-Lise Mottaz-Jiang* Jacques Guyot *

*Centre universitaire d'informatique, Université de Genève
falquet, mottaz, guyot@cui.unige.ch

Résumé. Nous présentons ici une approche pour la gestion de bases d'ontologies basée sur un modèle comprenant, outre la définition formelle des concepts (sous forme d'axiomes de logique de description), d'autres éléments descriptifs (termes, commentaires et arguments), ainsi que leurs liens d'alignement avec des concepts d'autres ontologies. L'adaptation ou la combinaison d'ontologies se font grâce à une algèbre comprenant des opérations telles que la sélection, la projection, l'union ou la jointure d'ontologies. Ces opérations agissent au niveau des axiomes, des éléments descriptifs et des liens d'alignement.

1 Introduction

L'interconnexion croissante des systèmes d'information, de même que des initiatives telles que le Web sémantique requièrent la création de nombreuses ontologies pour assurer la cohérence sémantique des opérations. Il devient donc nécessaire de développer des systèmes de gestion qui permettent non seulement de les stocker mais également de les aligner et de les combiner pour créer de nouvelles ontologies adaptées à des besoins particuliers, favorisant ainsi la réutilisation.

Contrairement à une démarche d'intégration où l'on ne cherche à obtenir qu'une seule ontologie homogénéisée, notre approche s'attache à conserver au sein d'une même base les différents points de vue (c'est-à-dire les différentes ontologies), mettant ainsi en évidence les apports de chaque contributeur. Cependant, les outils utilisant des ontologies ont besoin d'ontologies "normales" (mono-point de vue) pour fonctionner. Nous proposons donc un ensemble d'opérations et laissons le soin aux utilisateurs de les utiliser pour extraire de la base une ontologie "sur mesure", dans un contexte et un but spécifiques.

2 Un modèle de bases d'ontologies

Une ontologie est composée d'un vocabulaire $V = V_C \cup V_P \cup V_I \cup V_L$ (contenant les identifiants de concepts, de propriétés, d'individus (instances de concepts) et les littéraux), d'un ensemble T d'axiomes terminologiques (TBox) et d'un ensemble A d'annotations (terminologiques et argumentatives). Dans le vocabulaire V_P nous distinguerons les propriétés ontologiques (servant aux définitions) des propriétés d'annotation. Nous ne considérerons pas, dans cet article, les axiomes assertionnels (ABox).

Axiomes. Un axiome établit une relation de subsomption entre deux expressions de concepts E_1 et E_2 . Dans le cas où E_1 est un concept atomique on parle de définition.

Une expression de concept (ici nous prendrons le langage *SHOIN* qui est celui de OWL, Patel-Schneider et al. (2003), mais la démarche s'applique à toute logique descriptive) obéit à la syntaxe

$$\begin{aligned}
 E &\rightarrow \text{atome} \mid \text{and}(E, \dots) \mid \text{or}(E, \dots) \mid \text{not}(E) \mid \text{some}(R, E) \mid \text{all}(R, E) \mid \\
 &\quad \text{atLeast}(n, R, E) \mid \text{atMost}(n, R, E) \mid \text{oneOf}(i_1, \dots, i_n), \text{hasValue}(R, i_n) \mid \dots \\
 R &\rightarrow \text{nom de propriété} \\
 i_k &\rightarrow \text{nom d'un individu}
 \end{aligned}$$

Le graphe d'une telle expression est son arbre syntaxique étiqueté de manière à distinguer les descendants de chaque noeud. Pour un noeud *and*, *or* ou *not*, les arcs vers les descendants sont étiquetés par *Arg*. Pour les noeuds *some* et *all* on utilisera les étiquettes *Prop* et *Val* pour distinguer la propriété sur laquelle porte la restriction et le concept dans lequel la propriété prend ses valeurs. Pour *atLeast* et *atMost* on ajoute des arcs *Num* pour le nombre et pour *oneOf* et *hasValue* des arcs *Ind* pour indiquer les individus.¹

Un axiome de définition est représenté par un arc étiqueté \sqsubseteq ou \equiv entre un concept atomique et une expression. La figure 1 montre le graphe de l'axiome $A \sqsubseteq \text{and}(D, \text{atLeast}(2, P, C))$ (en blanc au centre).

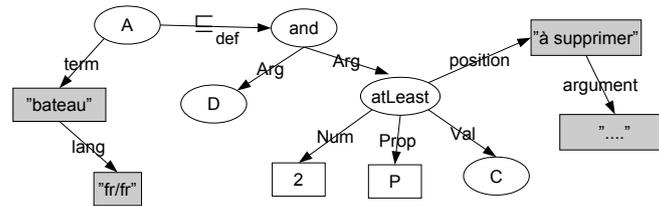


FIG. 1 – Graphe d'un axiome avec ses annotations (en gris)

Annotations. Les annotations constituent la partie non formalisée de l'ontologie. On distingue des annotations d'ordre terminologique, qui lient un concept atomique à un ou plusieurs termes, éventuellement dans plusieurs langues, ainsi qu'à des définitions ou commentaires en langue naturelle. Les annotations peuvent également être d'ordre argumentatif, par exemple pour soutenir ou au contraire critiquer une définition ou une partie de définition, Falquet et Mottaz Jiang (2000). Dans notre modèle, les annotations sont des objets possédant une valeur (en général un texte) qui sont liés aux concepts atomiques où aux expressions de concepts par des arcs étiquetés.

Graphe d'une ontologie. Le graphe d'une ontologie est un graphe orienté étiqueté dont les noeuds sont les éléments du vocabulaire de l'ontologie et les noeuds des arbres d'expressions de concepts. Les arcs proviennent des arbres d'expressions de concepts, des annotations

¹Nous n'utiliserons pas la représentation en arbres XML de OWL car les choix effectués pour cette représentation et le fait que les arbres XML n'ont pas d'arcs étiquetés conduisent à des expressions complexes qui compliquent à la fois l'exposé et les expressions de sélections. Il est cependant facile de traduire notre représentation en OWL/XML et vice versa

et des axiomes. On distinguera les axiomes définitoires (entre un concept atomique et une expression), qui seront étiquetés \sqsubseteq_{def} et \equiv_{def} et les axiomes "structurels" entre deux concepts atomiques, étiquetés \sqsubseteq ou \equiv .

Vocabulaire et structure d'alignement. Chaque ontologie d'une base d'ontologie est munie d'une structure d'alignement qui est un ensemble d'arcs (A, t) où A est un concept atomique et t est *objet d'alignement* appartenant au *vocabulaire d'alignement* de la base.

On dira que deux concepts A de O_1 et A' de O_2 sont alignés s'il existe un objet d'alignement t et deux arcs (A, t) et (A', t) dans les structures d'alignement des deux ontologies. La figure 2 montre une base composée de trois ontologies avec des objets d'alignement (en noir).

Contrairement aux approches basées sur des ontologies d'articulation, nous n'imposons aucune structure sur les objets d'alignements. Ceux-ci ne forment pas une ontologie, mais un vocabulaire universel. Nous nous situons plutôt dans la ligne des travaux sur l'alignement d'ontologies décrits dans Euzenat et Shvaiko (2007), bien que notre but ne soit pas de fournir des algorithmes d'alignement automatique d'ontologies. De plus, nous ne considérons que des alignements qui représentent des équivalences sémantiques et non des implications. Notons

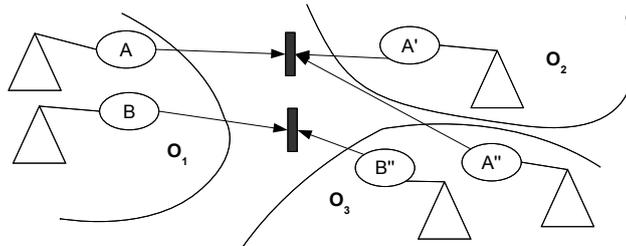


FIG. 2 – Structure d'alignement de plusieurs ontologies

que cette structure de base d'ontologie peut servir soit pour stocker des ontologies indépendantes que l'on souhaite combiner et réutiliser, soit pour représenter différents points de vues sur un domaine. Dans ce dernier cas, chaque point de vue correspond à une ontologie. C'est l'approche suivie dans la logique de description distribuée, Borgida et Serafini (2003) ou dans C-OWL, Bouquet et al. (2003).

3 Opérations sur les ontologies

Les opérations que nous présentons ci-dessous ont été conçues pour réaliser de manière simple des tâches typiques d'adaptation ou de combinaison d'ontologies que nous avons observées lors de la réalisation de différents projets basés sur des ontologies.

Projection. On définit la projection d'une ontologie sur un ensemble X de propriétés en distinguant ce qui se passe pour les arbres d'expression de concepts et pour le reste du graphe de l'ontologie.

La projection sur X d'un arbre d'expression de concept de racine e est l'arbre obtenu en supprimant tous sous-arbres de racine f tels que le chemin de f à e contient un noeud *all*,

some, *atLeast* ou *atMost* avec un arc *Val* qui fait référence à une propriété n'appartenant pas à X .

Par exemple, la projection sur $\{R, S\}$ de l'expression

$$\text{and}(\text{all}(R, C), \text{some}(T, \text{all}(R, D)), \text{all}(S, \text{or}(A, \text{all}(T, A)))) \quad (1)$$

donnera

$$\text{and}(\text{all}(R, C), \text{all}(S, \text{or}(A)))$$

Il est parfois utile de limiter la projection à un seul niveau. Dans ce cas, tous les sous-arbres d'un noeud faisant référence à une propriété appartenant à X seront conservés. La projection sur $\{R, S\}$ limitée au premier niveau de l'expression 1 sera

$$\text{and}(\text{all}(R, C), \text{all}(S, \text{or}(A, \text{all}(T, A))))$$

La projection d'une ontologie est obtenue en remplaçant chaque arbre d'expression par sa projection. Pour le reste de l'ontologie (les annotations et arcs d'axiomes) on élimine les arcs dont l'étiquette n'appartient pas à X et on supprime les éventuelles composantes connexes ainsi créées qui ne contiennent aucun axiome (et ne participent donc plus à l'ontologie).

Il est clair qu'en général les expressions obtenues par projection ne seront pas équivalentes, par contre les alignements sont conservés car on considère qu'il s'agit de nouvelles définitions (moins précises) des mêmes concepts.

Sélection. La sélection a pour but d'extraire un sous-ensemble de l'ensemble des concepts définis dans une ontologie. Pour spécifier une sélection on définit une expression logique portant sur (le graphe de) l'ontologie, c'est-à-dire sur les définitions de concepts, leurs annotations et éventuellement leurs instances.

Un expression de sélection est une expression de logique du premier ordre possédant une variable libre et définie sur le vocabulaire constitué de prédicats unaires correspondant aux types de noeuds du graphe d'une ontologie (*And*, *Or*, *Some*, ...) de prédicats binaires correspondant aux types d'arcs (*Arg*, *Pro*, *Val*, *Num*, plus tous les noms de propriétés d'annotation) et de constantes les noms de propriétés et les littéraux.

Par exemple, pour sélectionner dans O_1 tous les concepts qui ont une restriction *some* sur la propriété S au premier niveau on écrira : $O_2 = \sigma_E O_1$ avec

$$E(x) = \exists y(\text{Arg}(x, y) \wedge \text{Some}(y) \wedge \text{Prop}(y, S))$$

Etant donné un prédicat de sélection $S(x)$, la sélection $\sigma_S O'$ de l'ontologie O par S a pour graphe le sous-graphe de O qui contient tous les concepts atomiques (noeuds) satisfaisant S , les arcs d'alignement et d'annotation de ces concepts, leurs expressions de définition, les concepts atomiques référencés (à travers des arcs *Arg* et *Val* depuis ces définitions et les noeuds).

Les concepts non-sélectionnés mais référencés doivent être conservés, mais sans leur définition, pour maintenir la consistance des définitions des concepts sélectionnés. Ils passent donc du statut de concept défini à concept de base.

Etant donné que les ontologies sont interconnectées, l'expression de sélection peut utiliser d'autres ontologies. Par exemple, si l'on veut sélectionner les concepts de O_1 qui sont alignés avec un concept de O_2 , on écrira $\sigma_{\exists(y:\$1)\text{Aligned}(x,y)} O_1(O_2)$ où $y : \$1$ indique que la variable y doit être un objet de la première ontologie passée en paramètre (ici O_2)

Union. Le graphe de l'union $O_1 \cup O_2$ est obtenu en faisant l'union disjointe des noeuds et arcs de O_1 et O_2 . Les objets d'alignement étant communs, si deux concepts A_1 de O_1 et A_2 de O_2 sont alignés, on obtient dans l'union deux concepts distincts qui sont liés au même objet d'alignement.

L'union-importation $O_1 \cup_{import} O_2$ est une opération asymétrique. On procède comme pour l'union standard, mais si un noeud (concept atomique) A_1 de O_1 est aligné avec un noeud A_2 de O_2 , A_1 est supprimé, ainsi que les axiomes et annotations qui lui sont liés. Comme son nom l'indique, cette opération a pour but d'importer des définitions de concepts d'une ontologie dans une autre.

Jointure. Contrairement à l'union, la jointure combine les définitions de concepts par une opération ensembliste (union, inter). L'idée est que pour toute paire d'axiomes $A_1 \equiv_{\text{def}} E_1$ de O_1 et $A_2 \equiv_{\text{def}} E_2$ de O_2 tels que A_1 et A_2 sont alignés, $O_1 \bowtie O_2$ contiendra un unique axiome $A \equiv_{\text{def}} (E_1 \sqcap E_2)$. Il en va de même pour les axiomes de la forme $A_1 \sqsubseteq_{\text{def}} E_1$. Si l'un des axiomes est de la forme $A_i \sqsubseteq_{\text{def}} E_i$, l'axiome obtenu sera $A \sqsubseteq_{\text{def}} (E_1 \sqcap E_2)$. On procède de la même manière s'il y a plus de deux concepts alignés sur le même objet d'alignement. On remplace également tous les arcs de ou vers l'un des A_i par un arc de ou vers A .

Si l'on élimine les concepts qui ne sont pas alignés avec au moins un autre concept, on obtient la jointure stricte. Si on les garde on peut obtenir la jointure externe à gauche ou à droite.

4 Exemples d'utilisation

Dans le projet UNL (Universal Networking Language www.unl.org), des équipes dispersées dans le monde développent indépendamment des parties de l'UNLKB (ontologie linguistique du langage UNL). Ces parties, qui portent sur des domaines disjoints ou non, sont ensuite importées dans la base d'ontologies UNLPLAZA. Chaque concept est associé à un "mot universel" qui joue le rôle d'objet d'alignement entre ontologies. L'une des fonctions de l'UNLPLAZA est de produire (exporter) des ontologies construites par union, dans le cas où il existe plusieurs définitions de concepts pour le même mot universel, un ordre de priorité détermine laquelle choisir. Si l'ordre de priorité des ontologies est $O_1 > O_2 > \dots > O_k$, l'exportation correspond à l'expression $O_k \cup_{import} (O_{k-1} \cup_{import} (\dots \cup_{import} O_1) \dots)$

Dans le cas où la base contient des ontologies portant sur un même domaine, mais développées par des personnes qui s'intéressent à des caractéristiques différentes, on peut utiliser la jointure pour regrouper les caractéristiques en des définitions plus complètes. Par exemple : une ontologie des aliments selon le point de vue des cuisiniers (propriétés : saveur, temps de cuisson, prix, saison) pourra être jointe avec une ontologie selon le point de vue des diététiciens (valeur énergétique, vitamines, graisse, hydrate de carbone, protéines)

Pour importer dans O_1 uniquement les parties de définition concernant une propriété P on projette O_2 sur P puis on joint (à gauche) le résultat à O_1 : $O_1 \times \pi_P O_2$

Le remplacement de définitions s'effectue de la manière suivante. Soit $E(x)$ l'expression permettant de sélectionner les concepts de O_1 dont on veut remplacer la définition par celles existant dans O_2 . l'expression de remplacement est $O_1 \times \sigma_{x.\exists(y:\$1) \text{Aligned}(x,y)} O_2 (\sigma_{E(x)} O_1)$

où l'on commence par sélectionner les concepts de O_2 alignés avec les concepts sélectionnés de O_1 , puis on fait une jointure externe gauche des deux ontologies.

Enfin, une opération intéressante consiste à remplacer la hiérarchie de subsomption (les axiomes d'inclusion entre concepts atomiques) d'une ontologie O par une autre (par exemple WordNet). Ce qui s'obtient par $\pi_{\text{toutes les propriétés sauf } \sqsubseteq} O \times \pi_{\sqsubseteq} \text{WordNet}$

5 Conclusion et situation par rapport à d'autres travaux

L'approche que nous proposons se situe dans la ligne des travaux de Mitra et al. (2000). Ceux-ci proposent une algèbre pour un modèle d'ontologie de type réseau sémantique (dont la sémantique n'est pas formellement définie). Dans ce modèle, les ontologies sont interconnectées à travers une ontologie d'articulation munie de liens "d'implication sémantique" avec les autres ontologies. L'ontologie d'articulation est générée à partir de règles de ponts sémantiques ("semantic bridge rules") de la forme $O_1.A \Rightarrow O_2.B$. Les opérations proposées sont la sélection, l'union et l'intersection d'ontologies. Contrairement à ce que nous proposons, il n'y a pas d'opérations, telle la jointure, qui produisent de nouvelles définitions de concepts.

Nous travaillons actuellement à l'extension des opérations algébriques de l'UNLKB à tout le modèle OWL pour arriver à un système de gestion d'ontologies OWL.

Références

- Borgida, A. et L. Serafini (2003). Distributed description logics : Assimilating information from peer sources. *Journal of Data Semantics 1*, 153–184.
- Bouquet, P., F. Giunchiglia, F. van Harmelen, L. Serafini, et H. Stuckenschmidt (2003). Cowl : Contextualizing ontologies. In *Second International Semantic Web Conference (ISWC-2003)*, Volume 2870 of LNCS, pp. 164–179. Springer.
- Euzenat, J. et P. Shvaiko (2007). *Ontology Matching*. Springer-Verlag.
- Falquet, G. et C.-L. Mottaz Jiang (2000). Conflict resolution in the collaborative design of terminological knowledge bases. In *International Conference on Knowledge Engineering and Knowledge Management (EKAW 2000)*, LNCS. Springer.
- Mitra, P., G. Wiederhold, et M. Kersten (2000). A graph-oriented model for articulation of ontology interdependencies. In *Conference on Extending Database Technology 2000 (EDBT'2000)*, Konstanz, Germany, Volume 1777 of LNCS. Springer.
- Patel-Schneider, P. F., P. Hayes, I. Horrocks, et F. van Harmelen (2003). Web ontology language (owl) abstract syntax and semantics. Technical report, W3C (www.w3.org/TR/2003/WD-owl-ref-20030331).

Summary

We present an ontology algebra that is intended for managing ontology bases. The ontology model we consider is comprised of formal definitions (axioms in a description logics language); other descriptive elements (terms, comments, arguments) and alignment links between ontologies. We introduce projection, selection, union and join operations and define them in terms of the graph representation of the ontologies.