Prise en compte des ontologies non canoniques dans les BDBO : le modèle ONTODB2.

Chimène Fankam
LISI/ENSMA
1 avenue clément - BP 40109 - 86960 Futuroscope Chasseneuil Cedex France
chimene.fankam@ensma.fr

Les ontologies conceptuelles permettent de décrire de manière consensuelle l'ensemble des informations pertinentes d'un domaine d'application à l'aide d'un ensemble de concepts. La modélisation à base d'ontologies permet de décrire à la fois les données et leur sémantique exprimée par des concepts ou des expressions de concepts de l'ontologie. Ainsi, est apparue la nécessité de rendre persistantes aussi bien les ontologies que leurs instances. Plusieurs modèles de base de données ont été proposés à cet effet. Il s'agit des Bases de Données à Base Ontologique (BDBO). Des exemples de telles bases sont : SESAME [Sesame], ONTOMS[Myung et al., 2005], DLDB[Pan et al., 2003], etc. Au LISI/ENSMA, un modèle de BDBO fondé sur le modèle d'ontologie canonique PLIB a été défini : OntoDB[Dehainsala, 2007]. Dans sa version actuelle, OntoDB ne prend en compte que les concepts canoniques et ne permet pas de représenter des concepts non canoniques (dérivation, inférence, caractéristiques des relations telles que la symétrie, la transitivité, l'équivalence, etc.) présents dans des modèles d'ontologies tels que OWL ou Flogic. Nos travaux visent à (1) étendre le modèle OntoDB de BDBO pour pouvoir supporter la représentation des concepts non canoniques ainsi que les traitements associés. En conséquence, cette extension autorisera la représentation d'équivalences conceptuelles et de mécanismes d'inférence ; (2) proposer des techniques permettant d'optimiser OntoDB.

1. Extension du modèle d'ontologie dans OntoDB

La solution que nous proposons est de représenter, dans la partie ontologie de OntoDB, un modèle d'ontologie pivot. Ce modèle pivot regroupe l'ensemble des constructions communes à tous les modèles d'ontologies que notre architecture supportera (notion de classe décrite par des propriétés, relation de subsumption, …). Puis, nous avons défini des règles de transformation entre chaque modèle et le modèle pivot. Les ontologies de différents domaines peuvent ainsi être représentées dans OntoDB2, grâce à des techniques de transformation de modèles, comme instances du modèle pivot. Il est évident que le modèle pivot ainsi défini, ne permet pas de couvrir totalement la sémantique de chaque modèle. C'est pourquoi et en vue d'enrichir ce modèle pivot, nous avons également défini des transformations des constructions orthogonales spécifiques à chaque modèle [Fankam et al., 2007] ; parmi elles, a) les caractéristiques algébriques des relations(transitivité, inverse, etc.) ; b) les expressions algébriques (exemple : diamètre = rayon *2) ; et c) l' extension de classes non canoniques. Au niveau ontologique, la sémantique des ces constructions est représentée par une correspondance syntaxique en prédéfinissant la valeur de certaines propriétés. Au niveau des données, un raisonneur, qui sera intégré à OntoDB, interprétera ces propriétés et

sera associé à des mécanismes permettant d'automatiser la génération des comportements (triggers pour les expressions, génération de vues pour les classes non canoniques, etc.) associés à ces propriétés. Dans la pratique, les applications n'utilisent qu'un sous ensemble de la sémantique du modèle du domaine correspondant. Grâce au noyau commun et aux constructions orthogonales des modèles PLib, OWL et Flogic représentées, OntoDB sera une architecture flexible et plus expressive supportant des applications issues de divers domaines.

2. Optimisation de la représentation du modèle au sein d'OntoDB

Les modèles d'ontologies que nous souhaitons traiter suivent une méthodologie de conception orientée objet, or, OntoDB est implémenté sous l'environnement PostgresSQL sur une base de données relationnelle. La représentation actuelle du modèle d'ontologie PLIB dans OntoDB est complexe avec des relations d'héritage et de nombreuses jointures, ce qui ralentit les traitements au niveau ontologique. Le modèle OntoDB doit donc être optimisé.

– Le modèle pivot (proche de PLib) implémenté, doit être un modèle simple. Le modèle : FlatLib[Pierra et al., 2005] a été obtenu par la mise à plat des hiérarchies dans PLIB.

– Afin de maintenir la consistance, tant au niveau ontologique qu'au niveau des données, nous comptons permettre la vérification des contraintes définies par les différents modèles (subsumption acyclique), les contraintes de typage et de cardinalité des propriétés au niveau des données. Egalement, nous fournirons des interfaces de spécification couplées à des compilateurs pour traiter les expressions de dérivation et certaines contraintes d'intégrité de niveau ontologique (exemple : âge_étudiant >12) qui seront par la suite évaluées ou vérifiées par des triggers.

– Concernant les requêtes, de bonnes performances sont attendues grâce à une technique de marquage permettant, en une seule requête de sélection, de calculer les propriétés, les sous-classes ou super-classes d'une classe donnée.

En conclusion, nous proposons une BDBO supportant différents modèles d'ontologie en se basant sur un modèle pivot enrichi par les constructions orthogonales de chaque modèle. L'implémentation de ce modèle pivot sera également simplifiée et optimisée dans cette évolution de OntoDB.

Dehainsala H., Une proposition d'un modèle d'architecture pour les bases de données à base ontologique : le modèle OntoDB , Thèse 2007.

Fankam C.., Ait-Ameur Y., Pierra G., Exploitation of ontology languages for both reasoning and persistency purposes, *Webist 2007*, pp 254-262.

Myung P., Ji-Hyun L., Chun-Hee L.,Jiexi L., Serres O., Chin-Wan C., ONTOMS : An Efficient and Scalable Ontology Management System , 2005.

Pan, Z., Heflin J., DLDB: Extending Relational Databases to Support Semantic Web Queries., ISWC 2003, pp. 109-113.

Pierra G., Dehainsala H., Ngabiapsi N., Bachir M., Transformation relationnelle d'un modèle objet par prise en compte des contraintes d'intégrité de niveau instance, *Inforsid 2005*, pp 455-470.

Sesame, http://www.openrdf.org/.