

Modéliser des connaissances ontologiques dans le cadre du modèle des Graphes Conceptuels

Frédéric Fürst

LINA - FRE 2729

22 rue de la Houssinière, BP 92208, 44322 Nantes

furst@lina.univ-nantes.fr

Résumé. Cet article présente OCGL (Ontology Conceptual Graph Language), un langage de représentation d'ontologie basé sur le modèle des Graphes Conceptuels. Il décrit en détail la façon dont une ontologie est modélisée en OCGL, et présente l'implémentation de ce langage dans l'atelier d'ingénierie ontologique TooCoM.

1 Introduction

L'ingénierie des ontologies est née de la volonté de diversifier les applications des Systèmes à Base de Connaissances (SBC), et de permettre des représentations de connaissances indépendantes de ces diverses applications (Gomez-Perez et al., 2003). L'intégration d'un tel composant dans un Système à Base de Connaissances suppose alors d'adapter les représentations qu'il intègre à l'objectif opérationnel du système, adaptation qui est l'objet du processus d'opérationnalisation des ontologies (Fürst et al., 2004). D'autre part, les ontologies ont vocation à intégrer toute la sémantique des différents domaines de connaissances, c'est-à-dire des propriétés de base comme la subsumption entre concepts, mais également toute propriété permettant d'exprimer la sémantique du domaine considéré. Les ontologies évoluent ainsi des ontologies légères (lightweight ontologies), n'intégrant qu'un nombre restreint de propriétés, vers des ontologies lourdes (heavyweight ontologies), visant la modélisation de toutes les propriétés nécessaires à la représentation de toute la sémantique d'un domaine (Gomez-Perez et al., 2003).

Dans cet article, nous présentons OCGL (Ontology Conceptual Graph Language), un langage de représentation d'ontologies lourdes, basé sur le modèle des Graphes Conceptuels (GCs) (Sowa, 1984). OCGL est implémenté dans l'outil TooCoM (a Tool to Operationalize an Ontology in the Conceptual Graph Model), dédié à la modélisation et l'opérationnalisation d'ontologies lourdes dans le cadre du modèle des Graphes Conceptuels¹. Nous détaillons ici le modèle de représentation utilisé dans TooCoM, mais ne présentons pas le processus d'opérationnalisation qu'il implémente, renvoyant pour cela le lecteur à (Fürst et al., 2004).

1. Cet outil est disponible sous licence GPL sur le site <http://sourceforge.net/projects/toocom/>

2 OCGL : Ontology Conceptual Graphs Language

Notre objectif consiste à offrir la possibilité de modéliser toutes les connaissances d'un domaine dans le cadre du paradigme Entité/Relation, pour construire des ontologies lourdes dédiées au raisonnement. Le choix du modèle des GCs repose, d'une part, sur l'aspect graphique de ce formalisme, plus intuitif et facilement manipulable par des experts du domaine (qui ne sont a priori pas experts du modèle), et d'autre part sur l'expressivité des GCs, qui permettent de représenter à la fois des connaissances terminologiques et des propriétés de base des ontologies (subsumption, signature des relations, etc), et des propriétés ne correspondant pas à des propriétés classiques, sous forme d'implications. De plus, il existe dans le modèle des GCs des mécanismes de raisonnement qui permettent de mettre en œuvre les représentations pour raisonner.

Ainsi, OCGL reprend en bonne partie le modèle des Graphes Conceptuels, sa syntaxe graphique et sa sémantique formelle, ainsi que ses extensions, notamment la SG-family (Baget and Mugnier, 2002). Il en diffère cependant en ce qu'il est seulement un langage de représentation d'ontologie, et non pas un langage de représentation de connaissances. Ainsi, seules les connaissances ontologiques ont leur place en OCGL, et non les connaissances factuelles portant sur des cas particuliers. De même, seules les instances ontologiques des concepts sont représentées, c'est-à-dire celles qui participent à la définition de la sémantique du domaine.

De plus, les représentations de connaissances dans OCGL ne sont pas opérationnelles, de manière à assurer une réutilisabilité maximum aux ontologies en préservant leur indépendance vis-à-vis des applications. Les représentations du modèle des GCs dotées d'une sémantique opérationnelle (règles et contraintes de graphes conceptuels), c'est-à-dire dont le mode opératoire est fixé, ne sont donc pas reprises dans OCGL. De plus, il est nécessaire d'intégrer aux ontologies exprimées en OCGL des propriétés des primitives conceptuelles (concepts et relations) classiquement utilisées dans les ontologies, et qui ne sont pas explicitement intégrées au modèle des GCs, telles que les propriétés algébriques des relations (Staab and Maedche, 2000). OCGL étend donc une partie du modèle des Graphes Conceptuels, en lui ajoutant un ensemble de propriétés portant sur les types de concepts et les types de relations.

Le langage OCGL comporte un **niveau terminologique**, composé d'un ensemble de primitives conceptuelles (concepts et relations) et d'un ensemble d'instances ontologiques, et un **niveau axiomatique**, où est exprimée la sémantique des primitives conceptuelles. Ce niveau axiomatique se décompose en un **ensemble de schémas d'axiome** correspondant aux propriétés classiques des ontologies légères et un **ensemble d'axiomes** (appelés également axiomes de domaine) complétant si besoin est l'expression de la sémantique des primitives (cf. figure 1).

Un schéma d'axiome possède une forme prédéfinie fixe, instanciée avec des primitives conceptuelles, et porte sur une ou plusieurs primitives conceptuelles (généralement une seule primitive, parfois deux). Par opposition, les axiomes n'ont pas de forme prédéfinie, même s'ils s'écrivent tous sous la forme d'un couple de graphes liées par des liens entre sommets concepts. Ils présentent ainsi la même forme que les règles GC, mais ne font que spécifier une connaissance, et ne sont pas forcément destinés à produire des connaissances. De plus, ils peuvent ne pas exprimer une propriété portant sur une ou deux primitives en particulier, mais ils représentent plutôt des propriétés mettant

en jeu plusieurs primitives. Ainsi, l'axiome « *Les amis de mes amis sont mes amis* » peut être vu comme une propriété de transitivité de la relation *ami*(*Humain*,*Humain*). Mais l'axiome « *Les ennemis de mes ennemis sont mes amis* » peut difficilement être exprimé comme une propriété algébrique ou une cardinalité d'une primitive conceptuelle particulière². Ce dernier axiome sera donc représenté dans une ontologie par un axiome³.

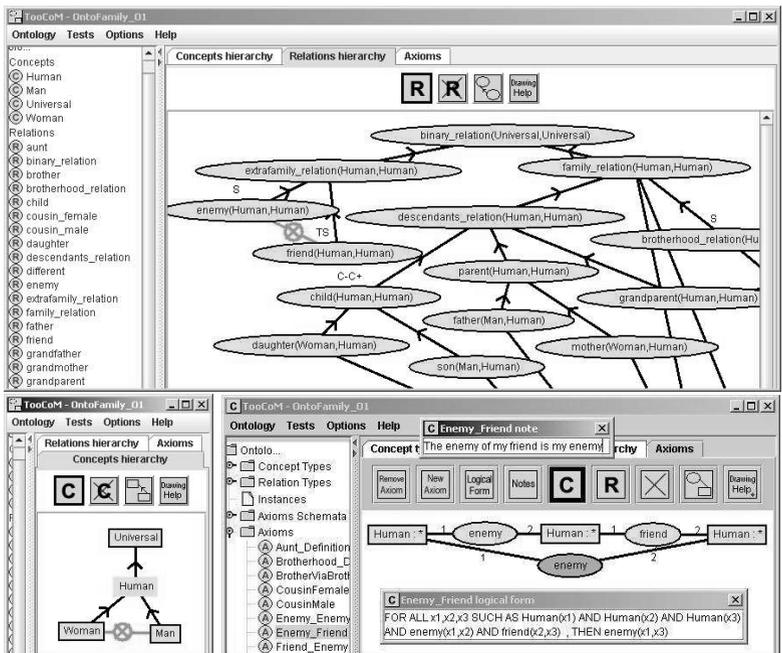


FIG. 1 – Représentation d’une ontologie des relations familiales en OCGL dans l’atelier TooCoM. La partie supérieure présente la hiérarchie de relations, la partie inférieure gauche, la hiérarchie des concepts et la partie inférieure droite, un axiome. Dans les hiérarchies, les flèches représentent des liens sorte-de, et les propriétés sont indiquées par des symboles (un cercle barré pour la disjonction entre concepts ou l’exclusivité entre relations, un S pour la symétrie d’une relation, un T pour la transitivité, etc.). Dans la partie axiome, les nœuds clairs représentent la partie antécédent, les nœuds sombres la partie conséquent. Chaque partie contient des nœuds concepts (rectangles) et des nœuds relations (ellipses).

Les schémas d’axiome spécifiés sont représentés par des symboles décorant les

2. A la rigueur, cet axiome peut être vu comme une sorte de combinaison entre l’inverse d’une relation et une transitivité.

3. Les axiomes et schémas d’axiome ont pour but commun l’expression de la sémantique du domaine. C’est pourquoi, dans la suite, et quand cela ne prête pas à confusion, le terme axiome sera utilisé pour désigner à la fois les axiomes (de domaine) et les schémas d’axiome.

hiérarchies. Ceci permet à l'utilisateur d'avoir une vue aussi globale et synthétique que possible de l'ontologie. Les schémas d'axiome portant sur les concepts sont : le lien **sorte-de** entre concepts, l'**abstraction** d'un concept (un concept c est abstrait si toute instance de c est aussi instance d'un de ses concepts fils), la **disjonction** entre deux concepts c_1 et c_2 (deux concepts c_1 et c_2 sont disjoints si aucune instance commune aux deux concepts n'existe. Les propriétés combinées d'abstraction d'un concept et de disjonction deux à deux entre ses concepts fils indiquent que les concepts fils constituent une partition du concept père.).

Les schémas d'axiome portant sur les relations sont : le lien **sorte-de** entre relations, la **signature** d'une relation, qui précise les concepts les plus spécifiques qu'elle peut lier, l'**incompatibilité** de deux relations ayant même signature, qui indique qu'il est impossible que ces relations lient le même ensemble d'instances, l'**exclusivité** de deux relations ayant même signature, qui indique que si une des relations est niée entre individus, alors l'autre est établie, les **propriétés algébriques** d'une relation binaire liant les mêmes concepts (symétrie, transitivité, réflexivité, anti-réflexivité, anti-symétrie), les **cardinalités maximum et minimum** portant sur un concept donné de la signature d'une relation.

Le seul schéma d'axiome portant sur les instances est le **type** d'une instance : le type d'une instance est le concept le plus spécifique auquel elle appartient.

Les schémas d'axiome ne permettant pas toujours d'exprimer toute la sémantique d'un domaine, il est nécessaire d'ajouter au langage des **axiomes** (ou axiomes de domaine) offrant la possibilité d'exprimer graphiquement des propriétés. Un axiome est constitué d'un couple de graphes conceptuels, liés par des nœuds concepts, et construits sur le support (au sens du modèle des GCs) constitué par les ensembles de concepts, de relations et d'instances ontologiques, les relations sorte-de, les signatures des relations, et les types des instances. Les sommets concepts sont, soit des concepts génériques, au sens des Graphes Conceptuels⁴, soit des instances ontologiques.

3 Conclusion

Dans cet article, nous présentons OCGL, un langage de représentation d'ontologie utilisant le modèle des Graphes Conceptuels. L'un des principaux avantages d'OCGL par rapport aux langages de représentation d'ontologie existants est son aspect graphique qui offre un formalisme intuitif et facilement manipulable. D'autre part, OCGL s'appuie sur la SG-Family pour représenter tout type de connaissances axiomatiques, sans se limiter aux propriétés classiquement considérées dans les ontologies légères, comme c'est à l'heure actuelle le cas dans les langages du Web sémantique, RDF et OWL. Enfin, OCGL permet de spécifier ces connaissances axiomatiques sans leur imposer de sémantique opérationnelle, ce qui augmente la réutilisabilité des ontologies. Cette approche diffère donc de celles qui visent à ajouter aux ontologies légères des règles (par exemple dans le langage SWRL) ou des contraintes (par exemple dans le langage PAL de l'outil Protégé) dont la sémantique opérationnelle conditionne l'usage.

4. Ils sont alors étiquetés par un type et par le marqueur *.

Références

- Baget, J. and Mugnier, M. (2002). Extensions of Simple Conceptual Graphs: the Complexity of Rules and Constraints. *Journal of Artificial Intelligence Research*, 16:425–465.
- Fürst, F., Leclère, M., and Trichet, F. (2004). Operationalizing domain ontologies: a method and a tool. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI'2004)*, volume 110, pages 318–322. IOS Press.
- Gomez-Perez, A., Fernandez-Lopez, M., and Corcho, O. (2003). *Ontological Engineering*. Springer-Verlag, Advanced Information and Knowledge Processing.
- Sowa, J. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley.
- Staab, S. and Maedche, A. (2000). Axioms are objects too: Ontology engineering beyond the modeling of concepts and relations. Research report 399, Institute AIFB, Karlsruhe.

Summary

This article presents OCGL (Ontology Conceptual Graph Language), a language dedicated to the representation of ontologies and based on the Conceptual Graphs model. The article describes in detail how an ontology is designed in OCGL, and shows the implementation of this language in the ontological engineering tool TooCoM.

