OWL-SG : un sous-langage OWL pour la famille SG

Frédéric Comte, Michel Leclère, Marie-Laure Mugnier

LIRMM
Université de Montpellier II
161, rue Ada, F-34392 Montpellier cedex - France,
{frcomte,leclere,mugnier}@lirmm.fr

1 Introduction

Le web est constitué de ressources identifiées par des URI et indexées automatiquement par des moteurs de recherche permettant de retrouver une ressource à partir de mots-clés contenus dans ces ressources. Le web sémantique vise à étendre le web par la prise en compte de méta-données sémantiques sur ces ressources. L'idée est de représenter formellement le contenu d'une ressource. Une telle représentation est appelée une annotation sémantique. Dès lors que le web a pour vocation d'être utilisé par de nombreux utilisateurs, il est nécessaire que le formalisme de représentation utilisé ainsi que le vocabulaire conceptuel soit partagé par les différents acteurs du web sémantique.

RDF offre une première réponse en proposant un formalisme de représentation de méta-données à base de triplets *(sujet, prédicat, objet)* [RDF04c]. RDF Schéma étend ce premier formalisme en proposant des primitives spécifiques d'expression de connaissances ontologiques séparées des connaissances factuelles [RDF04a]. OWL [BHH+04] étend lui-même RDFS en proposant un formalisme plus complet d'expression de connaissances factuelles et ontologiques. Enfin, SWRL [HPSB+04] vise à étendre OWL par la prise en compte d'un langage de règles.

Ces différents langages disposent d'une syntaxe et d'une sémantique formelle permettant de définir des notions de satisfiabilité et de déduction mais ne fournissent pas de mécanismes d'inférences permettant la mise en œuvre de raisonnements. L'exploitation des connaissances représentées dans ces langages passe par une transformation dans des formalismes disposant d'inférences opérationnelles associées telles que les logiques de description (testant l'insatisfiabilité par la méthode des tableaux), les modèles relationnels des bases de données (cf. RACER [HM01], Triple [SD02], Sesame [BKH02]...

Notre objectif est d'étudier le potentiel de la famille SG [BM01] (un paradigme issu des graphes conceptuels) comme formalisme d'opérationalisation des langages et problèmes du web sémantique.

2 RDF et la famille SG

Des travaux précédents [CDH00,Bag04], se sont intéressés au langage RDFS en poursuivant deux objectifs différents :

- O. Corby et al. ont exploité les similitudes graphiques entre RDF et les graphes conceptuels et ont ainsi définis une transformation des triplets RDFS en graphes simples. Cette transformation conserve les aspects intuitifs des formalismes respectifs (une classe est transformée en type de concept, une propriété en type de relation, une annotation simple en un graphe conceptuel simple) mais ne permet de représenter de manière correcte qu'un sous-ensemble du langage RDF/RDFS ;

– J.F. Baget, s'appuyant sur la sémantique formelle [RDF04b]et sur les liens établis entre déduction et test d'existence de sous-graphe (cf. interpolation lemma sur RDF sans interprétation du vocabulaire), définit une transformation qui conserve la sémantique formelle de RDFS (la déduction (testée par projection) sur les graphes conceptuels associés aux documents RDFS est adéquate et complète par rapport à celle définie sur RDFS). Cependant cette transformation ne conserve pas les aspects visuels intuitifs (les classes et les propriétés sont transformées en sommets concepts, un unique type de relation ternaire particulier représentant les triplets).

Il est intéressant d'analyser pourquoi la recherche de la complétude avec la déduction RDFS entraîne une perte des qualités graphiques intuitives du modèle :

– la première raison est due à la non partition du vocabulaire conceptuel : en effet les sujet, prédicat et objet RDF peuvent tous être identifiés par des URI ; ainsi il est possible qu'une URI utilisée comme prédicat, soit utilisée dans un autre triplet comme sujet ou objet ;
– la deuxième raison est l'aspect réflexif du formalisme qui code le niveau méta (les classes, et les propriétés) dans le formalisme lui-même. Ainsi si une URI est utilisée comme prédicat d'un triplet RDF, elle doit dès lors être considérée comme sujet d'un autre triplet RDF dont le prédicat est *rdf :type* et l'objet *rdf :property*. De même, si une URI est objet d'un triplet dont le prédicat est *rdf :type*, alors cette URI doit être sujet d'un autre triplet dont le prédicat est *rdf :type* et l'objet est *rdfs :class*.

Les cas où la transformation de O. Corby et al. ne permettrait pas de détecter une déduction entre documents RDFS correspondent à des cas où on atteste dans un document RDF des connaissances à plusieurs (ou à différents) niveaux (modèle, méta-modèle, méta-méta-modèle...) que ce soit du à la non-partition du vocabulaire conceptuel en classe, propriété et individu ou à la création de chemin d'arête étiquetée *rdf :type* dans le graphe RDF.

Si il est pertinent que des langages d'échange de connaissances rassemblent dans un même formalisme les représentations de différents niveaux, on peut vouloir limiter les raisonnements effectués sur ces représentations à un nombre limité de niveaux. Dès lors, on peut se demander combien de niveau de connaissances manipule-t-on dans le modèle des graphes conceptuels ? Nous en voyons trois : celui des individus (ou n-uplets d'individus) représentés par les sommets concepts des graphes de la base de connaissance, celui des ensembles d'individus (ou n-uplets d'individus) représentés par les types de concepts et de relations et les paramètres formels des règles de graphes, et celui du formalisme lui-même qui impose certaines contraintes sur les individus, et ensembles d'individus, comme par exemple la partition du vocabulaire conceptuel.

Nous ne pouvons transformer dans la famille SG que deux niveaux RDF (un niveau assertionnel et un niveau ontologique) en imposant que le niveau supérieur soit en accord avec la sémantique du formalisme des graphes conceptuels (en particulier la partition du vocabulaire conceptuel). Selon les documents RDFS considérés et les raisonnements que l'on souhaite mener, ces niveaux peuvent être choisis de différentes façons. Nous nous focaliserons sur le cas (classique) suivant où :

– le vocabulaire ontologique est constitué des URI sujets des triplets dont le prédicat est *rdf :type* et l'objet est *rdfs :class* que nous nommons classes (ou *rdf :property* que nous nommons propriétés), sans considérer toutes les URI issues du niveau méta (préfixées par rdf, ou rdfs) : *rdfs :class*, *rdfs :property*, *rdfs :range....*
– le vocabulaire assertionel est constitué des URI, que nous nommons individus, sujets des triplets dont le prédicat est *rdf :type* et l'objet est une URI sujet d'un triplet ayant *rdf :type* comme prédicat et *rdfs :class* comme objet. Des individus indéfinis étant représentés par des "blank nodes" sujets de triplets similaires.

Les connaissances assertionnelles sont alors constituées de l'ensemble des triplets liant des individus par des prédicats propriétés (appartenant donc au vocabulaire ontologique) et par les triplets liant les individus à leurs classes au moyen du prédicat *rdf :type*. Seules ces connaissances seront transformées en graphes simples.

Les connaissances ontologiques font l'objet de notre étude et sont constituées de l'ensemble des triplets liant des classes et propriétés dont le prédicat est une URI appartenant au vocabulaire *rdfs* ou *owl* disposant d'une sémantique particulière exprimable dans le modèle des graphes conceptuels (*rdfs :subclassOf, rdfs :range, owl :someValuesFrom...*).

3 Un sous-langage OWL pour la famille SG

Notre objectif est d'étendre les travaux précédents aux primitives d'OWL en :

- privilégiant les raisonnements réalisés sur les connaissances assertionnelles sur ceux que l'on peut faire sur les connaissances ontologiques (en particulier permettre l'expression et le traitement de requêtes conjonctives) ;
- conservant les aspects visuels intuitifs du modèle (que ce soit dans les représentations mais aussi lors des raisonnements afin de faciliter les explications) ;
- limitant notre transformation à un sous-ensemble du langage OWL ;
- assurant, pour ce sous-ensemble, consistance et complétude des raisonnements proposés dans la SG-family par rapport à la sémantique définie en OWL [PSHH04].

Contrairement à ce qu'indique son nom, OWL n'a pas pour seule vocation de représenter des ontologies. Il permet tout autant de représenter des connaissances assertionnelles que des connaissances ontologiques mais fournit un cadre unifié pour ces deux types de connaissances.

OWL a été défini comme une extension à RDFS en ce sens que tout document OWL bien formé est un document RDF bien formé et que la sémantique d'OWL est compatible (respecte et étend) avec la sémantique de RDFS. En fait OWL dispose de deux sémantiques formelles : l'une héritée des logiques de description ayant servies de base à la définition d'OWL, et l'autre issue de la sémantique de RDFS duquel OWL réutilise la syntaxe. Une des difficultés de définition d'OWL a justement été de rendre ces deux sémantiques compatibles [HPSvH03].

3.1 La famille SG

La famille SG est issue des travaux du LIRMM autour du formalisme des graphes conceptuels [CM92,BM01]. L'objectif est de disposer d'un formalisme de représentation et de raisonnement à base de graphes. Les principaux composants étudiés (tous ne sont pas complètement définis et l'utilisation conjointe de ces composants n'a pas été étudiée) actuellement dans la famille SG sont :

- un support composé d'une hiérachie de types de concepts, d'une hiérarchie de types de relations (chaque type de relation disposant d'une arité fixée), et d'un ensemble de marqueurs individuels (avec un type de concept associé) ;
- des graphes simples construits sur un support donné permettant de représenter des faits ;
- des couples de lambda abstraction (graphes simples dont certains sommets concepts génériques sont identifiés comme paramètres formels) de même arité utilisés comme règles, contraintes positives, contraintes négatives ;
- des définitions de types (association d'un type à une lambda de même arité) ;
- la possibilité de mettre des liens de coréférences entre sommets concepts d'un graphe simple ;

– la possibilité de construire des types de concept conjonctifs et de définir des types conjonctifs interdits [CM04] ;
– la possibilité d'utiliser des graphes emboîtés ;
– la possibilité d'utiliser une négation atomique dans les graphes simples.

Après avoir tenté de traduire globalement les sous-langages DL et Lite, nous pensons que l'approche graphes conceptuels nécessite de définir un sous-langage OWL-SG qui n'est pas directement comparable à Lite ou DL : certaines primitives d'OWL sont difficilement exprimables dans le modèle graphes conceptuels (cardinalités, *owl :allValuesFrom...*) ; par ailleurs OWL Lite et DL rejettent a priori la possibilité d'exprimer des cycles de triplets ne comportant que des "blank" nodes alors que cette caractéristique est naturelle dans le modèle des graphes conceptuels. Ce dernier point est particulièrement important pour l'expression de requêtes conjonctives où les individus seront souvent réduits à des "blank nodes".

L'essentiel du travail en cours consiste donc à sélectionner certaines directives OWL, à fixer une éventuelle restriction syntaxique d'utilisation à cette directive, puis à définir une transformation dans la famille SG telle que les sémantiques formelles des deux modèles coïncident. Nous menons ce travail de définition du sous-langage OWL-SG :

– de manière syntaxique, en traduisant intuitivement les directives OWL dans le formalisme des graphes conceptuels dotant ainsi OWL d'une sémantique SG. Cette transformation est réalisée en XSLT d'OWL vers CogXML. L'idée étant de permettre d'opérationaliser rapidement dans la SG-family des problématiques du web sémantique : aide à la construction d'annotations ou de requêtes relativement à une base de connaissances ontologiques donnée, vérification de la cohérence d'ontologie, vérification de la cohérence d'une base d'annotations, interrogation d'une base d'annotations par expression de requête construites relativement à une base de connaissances ontologiques donnée ;
– de manière plus théorique, en définissant un sous-langage OWL-SG pour lequel la transformation dans la famille SG des directives OWL respectant ce sous-langage permet de disposer d'une sémantique équivalente à celle définie dans OWL et permettant donc d'assurer adéquation et complétude des notions de satisfiabilité et déduction respectives.

3.2 Les caractéristiques d'OWL-SG

OWL propose trois sous-langages imbriqués : Lite, DL et Full. Lite et DL ont été définis pour correspondre aux logiques de description nécessaires à leur opérationnalisation : SHIF(D) et SHION(D). Ils imposent une partition du vocabulaire conceptuel en classe, propriété et individu. Full n'impose pas cette contrainte. OWL-SG doit lui aussi imposer une partition du vocabulaire conceptuel.

OWL Lite et DL ne disposent pas d'une expressivité suffisante permettant l'expression de requêtes conjonctives quelconques. RDFS semblent disposer de tels langages de requêtes. La famille SG permet l'expression et le traitement des requêtes conjonctives. C'est une des différences importantes des graphes conceptuels par rapport aux logiques de descriptions. OWL-SG doit permettre l'expression de requêtes conjonctives.

A côté d'un ensemble de "directives logiques" de représentation de connaissances, OWL dispose de directives n'ayant pas vocation à servir de base à un raisonnement : telles que des directives d'annotations des représentations faites, des directives de liens entre ontologies, des directives de gestion de version... Leur seule sémantique est celle obtenue par leur codage en RDF. Elles peuvent donc être interprétées comme de simples triplets RDF mais il ne semble pas opportun de s'intéresser à ces directives dans OWL-SG.

Par ailleurs, OWL permet d'associer des types de donnés XML à des propriétés particulières. La prise en compte de la sémantique de ces types nécessite de compléter les raisonneurs logiques par des raisonneurs adaptés au type utilisé. Il ne semble pas important de se focaliser sur les types de données utilisés dans OWL. Dans OWL-SG, les propriétés utilisant ces types comme objets seront traitées commes les autres propriétés, les types de données comme des classes, et les littéraux comme des individus.

OWL étant prévu pour le web sémantique, il a été défini pour une utilisation distribuée et extensible (d'où la définition de primitives d'import d'ontologies). Cela amène : (i) l'hypothèse du monde ouvert (i.e. les ontologies OWL ne fournissent qu'une connaissance partielle du monde modélisé à la différence de l'hypothèse du monde clos des BD), (ii) le rejet de l'hypothèse du nom unique pour ses individus (deux individus identifiés par deux URI différents ne doivent pas être considérés nécessairement comme distincts) et (iii) le principe de monotonie (aucune connaissance assertée ne peut-être "retirée" par une nouvelle connaissance). Ainsi un ensemble de directives OWL doit être interprété de manière conjonctive.

Ces caractéristiques fondamentales d'OWL sont également présentes dans la famille SG sauf pour l'hypothèse du nom unique. Une des difficultés de représentation des connaissances OWL en GC vient de l'absence de notion de différence dans la famille SG. L'hypothèse du nom unique permet cependant d'asserter la différence entre individus nommés ; dès lors qu'il faut relâcher cette contrainte, aucune différence n'est exprimable. Une solution consiste à utiliser une relation particulière de différence (c'est la solution choisie dans TooCom [Für04]) mais cela pose de nombreux problèmes comparables à ceux de l'introduction d'une négation atomique sur les relations [Mug00]. L'introduction de la différence dans OWL-SG impose l'ajout d'une forme de négation à la famille SG qui entraîne des changements importants dans le modèle dès lors que l'on veut garantir la complétude des raisonnements effectués. Cette absence de différence n'est pas anodine, car la relation de différence est la clé de représentation des axiomes de cardinalités [LTF02].

4 Conclusion

Un premier noyau pour un langage OWL-SG a été balisé [CL05]. Il permet d'exprimer les hiérarchies de classes et propriétés et d'exprimer que des classes sont disjointes (par la notion de types interdits). Il permet l'expression des caractéristiques des propriétés (symétrie, transitivité...) par des règles de graphes. Il permet également l'expression de connaissances assertionnelles (par des graphes simples utilisants des types conjonctifs [BMT99]). La prise en compte d'autres axiomes nécessitent des modifications du modèle (relacher l'hyothèse du nom unique pour affirmer l'équivalence entre individus, autoriser une relation de préordre dans le support pour l'expression d'équivalence entre classes ou propriétés). La définition de classes ou propriétés par le biais des restrictions doit pour certaines catégories de restriction être exprimable à l'aide de règles de graphes. Enfin la prise en compte de règles comme envisagé dans SWRL ne pose pas de problème puisqu'elles correspondent directement aux règles de graphes.

Références

[Bag04] J.F Baget. Homomorphismes d'hypergraphes pour la subsomption en rdf/rdfs. In *Conference LMO'04*, 2004.

[BHH+04] S. Bechhofer, F. Van Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, and L.A. Stein. Owl web ontology language reference. Recommendation, W3C, 2004.

[BKH02] J. Broekstra, A. Kampman, and F. Van Harmelen. Sesame : a generic architecture for storing and querying rdf and rdf schema. In *Proceedings of ISWC, LNCS*, volume 2342, pages 54–68, 2002.

[BM01] J.-F. Baget and M.-L. Mugnier. The $\mathcal{SG}$ Family : Extensions of Simple Conceptual Graphs. In *Proc. IJCAI'01*, volume 1, pages 205–210, 2001.

[BMT99] F. Baader, R. Molitor, and S. Tobies. Tractable and Decidable Fragments of Conceptual Graphs. In *Proc. of ICCS'99*, volume 1640 of *LNAI*, pages 480–493. Springer, 1999.

[CDH00] O. Corby, R. Dieng, and C. Herbert. A conceptual graph model for w3c resource description framework. In *springer verlag LNAI*, volume 1867, pages 468–482, 2000.

[CL05] F. Comte and M. Leclère. Opérationalisation des ontologies owl dans la famille sg. In *poster à paraître dans actes des 7èmes rencontres des jeunes chercheurs en intelligence artificielle (RJCIA)*, 2005. poster.

[CM92] M. Chein and M.-L. Mugnier. Conceptual Graphs : Fundamental Notions. *Revue d'Intelligence Artificielle*, 6(4) :365–406, 1992. Available at http ://www.lirmm.fr/˜mugnier/.

[CM04] M. Chein and M.-L. Mugnier. Types and coreference in simple conceptual graphs. In *ICCS'04*, volume 3127/2004 of *LNAI*, pages 303 – 318. Springer, 2004.

[Für04] F. Fürst. *Contribution à l'ingénierie des ontologies : une méthode et un outil d'opérationalisation.* PhD thesis, Université de Nantes, 2004.

[HM01] Volker Haarslev and Ralf Möller. RACER system description. *Lecture Notes in Computer Science*, 2083, 2001.

[HPSB+04] I. Horrocks, P.F. Patel-Schneider, H. Boley, S Tabet, B. Grosof, and M. Dean. Swrl : A semantic web rule language combining owl and ruleml. Member submission, W3C, 2004.

[HPSvH03] I. Horrocks, P.F. Patel-Schneider, and F. van Harmelen. From shiq and rdf to owl : the making of a web ontology language. *journal of web semantics*, Web semantics : Science, Services, and Agents on the World Wide Web 1 :7–26, 2003.

[LTF02] M. Leclère, F. Trichet, and F. Furst. Construction d'une ontologie de domaine de la géométrie projective. In *13e congrès francophone de reconnaissance des formes et d'intelligence artificielle (RFIA)*, volume 3, pages 1013–1022, Angers, 2002. AFRIF-AFIA.

[Mug00] M.-L. Mugnier. Knowledge Representation and Reasoning based on Graph Homomorphism. In *Proc. ICCS'00*, volume 1867 of *LNAI*, pages 172–192. Springer, 2000. Revised version available at http ://www.lirmm.fr/˜mugnier/.

[PSHH04] P.F. Patel-Schneider, P. Hayes, and I. Horrocks. Owl web ontology language semantics and abstract syntax. Recommendation, W3C, 2004.

[RDF04a] Rdf vocabulary description language 1.0 : Rdf schema. Recommandation, W3C, 2004.

[RDF04b] Rdf semantics. Recommendation, W3C, 2004.

[RDF04c] Resource description framework (rdf) : Concepts and abstract syntax. Recommendation, W3C, 2004.

[SD02] M. Sintek and S. Decker. Triple : A query, inference, and transformation language for the semantic web. In *Proceedings of ISWC, LNCS*, volume 2342, pages 364–378, 2002.