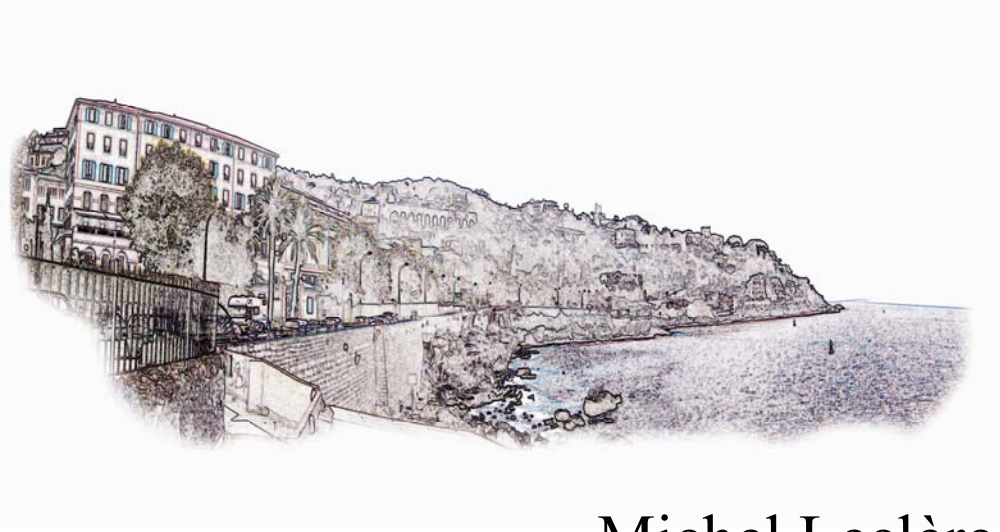
Plate-forme AFIA / Nice, du 30 mai au 3 juin 2005

JOURNEE THEMATIQUE:

Raisonner le web sémantique avec des graphes



Michel Leclère
Francky Trichet
Olivier Corby
Catherine Faron-Zucker

Introduction

Depuis quelques années, une certaine effervescence règne autour du concept de « web sémantique », effervescence qui a conduit à l'émergence de plusieurs standards « web sémantique » de représentation de connaissances : RDF, le langage de représentation d'annotations ; OWL, le langage de représentation d'ontologies ; RuleML le langage de représentation de règles. Ces langages disposent pour certains d'une sémantique formelle mais ne disposent pas de mécanismes de raisonnement. Ce sont des langages d'échange de connaissances sur le web, qui n'ont pas vocation à offrir de tels services. Cependant la mise en place d'un service inférentiel pour le web sémantique nécessite de définir des mécanismes d'opérationnalisation de ces langages, , i.e. des mécanismes visant à ajouter une sémantique opérationnelle à la sémantique formelle des primitives et de leurs constructeurs. Si les logiques de descriptions sont la cible privilégiée pour opérationnaliser OWL, elles ne sont à l'heure actuelle pas capables de supporter RDF ou RuleML.

Le web sémantique véhicule également l'idée d'un web compréhensible à la fois par les hommes et les machines. Les travaux sur les Topic Maps répondent à cette idée en permettant une représentation aisée des connaissances par des graphes bâtis autour de notions simples. Cependant, ils ne disposent pas d'une sémantique formelle permettant de définir les raisonnements que l'on pourrait mettre en place sur eux

Il existe en France une communauté de chercheurs travaillant autour de la représentation de connaissances et des raisonnements à partir de graphes. Cette communauté partage un même formalisme de représentation des connaissances : les graphes conceptuels.

La demi-journée thématique « raisonner le web sémantique avec des graphes » doit permettre de réunir les chercheurs et industriels intéressés par ce thème afin de croiser les différents points de vues, d'échanger sur les projets en cours et d'envisager la naissance d'une collaboration scientifique sur ce thème. On cherchera à cibler quelles inférences sont utiles pour le web sémantique, et en quoi les modèles de graphes permettent de répondre à ces inférences. Les présentations pourront aborder des problèmes théoriques aussi bien qu'applicatifs et pourront concerner des travaux passés, en cours ou en projet

Les présentations porteront sur les sujets suivants (liste non-exhaustive):

- Annotation : construction, vérification ;
- Ontologies : construction, validation, alignement, fusion ;
- Requêtes : formulation, reformulation, extension de requêtes ;
- Moteurs d'inférences : recherche d'information, navigation sémantique, services sémantiques, implémentation (passage à l'échelle, aspects distribués) ;
- Liens entre langages du web sémantique (RDF/S, OWL, RuleML, Topic Maps…) et langages de graphes ;
- Applications ;
- Interface et visualisation des connaissances.

Sommaire:

Le moteur de recherche sémantique Corese **1-3**
O. Corby, R. Dieng-Kuntz, C. Faron-Zucker, F. Gandon, A. Giboin
INRIA Sophia Antipolis, I3S Université de Nice

OWL-SG : un sous-langage pour la famille OWL **5-10**
F. Comte, M. Leclère, M.L. Mugnier
LIRMM

Raisonner sur les ontologies au niveau meta **11-13**
F. Trichet, F. Fürst
LINA

Proximité conceptuelle et distances de graphes **15-17**
F. Gandon, O. Corby, R. Dieng-Kuntz, A. Giboin
ACACIA - INRIA

Raisonner le web sémantique avec des graphes : application à un cas industriel **19-24**
Olivier Carloni
LIRMM, Mondeca

Flexibilité de la recherche d'information dans les projets de conception **25-28**
C. Djaiz, N. Matta, O. Corby, W. Léone, W. Lesguillier
Tech-Cico (UTT), ACACIA-INRIA

MIEL++ : un entrepôt intégrant des données floues exprimées en graphes conceptuels, bases de données relationnelles et XML **29-31**
P. Buche, J. Dibie-Barthélemy, D. Doussot, O. Haemmerlé, G. Hignette, R. Thomopoulos
INRA-Mét@risk, INA P-G/INRA MIA, LRI, INRA

Le moteur de recherche sémantique Corese

Olivier Corby[1], Rose Dieng[1], Catherine Faron-Zucker[2], Fabien Gandon[1] et Alain Giboin[1]
(1) INRIA Sophia Antipolis, (2) I3S Université de Nice - Sophia Antipolis

I. Introduction

Corese [1] est un moteur de recherche sémantique pour RDF, le langage standard du web sémantique recommandé par le W3C. Ce moteur est dédié à des applications de web sémantique communautaire ou d'entreprise. Il permet la recherche d'information basée sur des modélisations de connaissances, des modèles conceptuels ou des référentiels métier et organisationnel. Il implémente RDF/S dans le modèle des graphes conceptuels. Cet article montre l'intérêt des graphes conceptuels pour le web sémantique, pour raisonner avec RDF/S.

Corese charge des meta donnés RDF, des ontologies au format RDF Schema, et comprend un langage de règles adapté à RDF. Il construit une représentation interne de ces informations sous forme de graphes conceptuels sur lesquels il raisonne par projection de graphes pour répondre à des requêtes de recherche d'information dans la base de meta données RDF. Les résultats de requêtes sont retournés au format RDF/XML et sont traités par des feuilles de style XSLT pour engendrer des formats de présentation lisibles par les utilisateurs (HTML, SVG, etc.).

II. Correspondance entre RDF et CG

Les modèles de RDF/S et des Graphes Conceptuels (GC) présentent de nombreuses similarités. Nos travaux passés sur les GC nous ont permis de comprendre la nature et l'utilité de RDF/S associé à XML en 1999. Dès cette année nous disposions d'une première maquette de RDF en GC. Nous avons proposé une correspondance entre les GC et RDF/S en 2000[2]. En 2001, Tim Berners-Lee publiait une note soulignant la similarité entre RDF et les graphes conceptuels [1].

Essentiellement, les deux modèles distinguent connaissances assertionnelles et connaissances ontologiques. Les connaissances assertionnelles sont positives, conjonctives et existentielles; elles sont représentées par des graphes orientés, étiquetés bipartites. Dans Corese, un graphe RDF représentant une annotation ou une requête est traduit en un GC. Quant aux connaissances ontologiques, la hiérarchie de classes (resp. propriétés) du schema RDFS est traduite sous forme de hiérarchie de types de concept (resp. relations) du support de GC. Les propriétés RDF sont des entités de premier ordre comme les classes RDFS, de la même manière que les types de relation sont déclarés séparément des types de concept dans le support des GC. Ce traitement similaire des propriétés/relations rend la mise en correspondance des formalismes particulièrement pertinent, au regard en particulier des langages à objets où les propriétés appartiennent aux classes.

Les différences entre les deux modèles résident essentiellement en ce que RDF permet d'associer plusieurs classes à une ressource et de déclarer plusieurs classes en domain et en range dans la signature d'une propriété. Ceci est traduit en GC en engendrant le plus général sous-type commun des types correspondant aux classes.

Ainsi dans Corese la hiérarchie des types est compilée et un type compilé est associé à chaque ressource RDF. Pour répondre à une requête sur les métadonnées RDF, les algorithmes de raisonnement de Corese sont alors basés sur l'opération de projection de graphe. En outre, une ontologie représentée par un schema RDFS, en tant que document RDF, est également chargé sous forme de graphe conceptuel. Cela permet à Corese de répondre à des requêtes portant sur l'ontologie elle-même.

III. Moteur de recherche

A. Langage de requête

L'interrogation d'une base de graphes se fait dans Corese en écrivant des graphes requêtes et en les projetant sur la base de graphes. Toutefois, en utilisant la forme de graphe pour exprimer des requêtes on atteint rapidement des limites. Les problèmes rencontrés sont : le OU booléen, les expressions évaluables et l'introduction d'opérateurs comme optional, not, select, group, etc. Pour ces raisons nous avons conçu un langage de requêtes basé sur le modèle de triplets de RDF et traduit en graphe conceptuel. Par exemple la requête suivante recherche les personnes (1) ayant un nom (2) qui sont les auteurs (3) d'une thèse (4) qui a un titre (5) qui contient le mot 'web' (6).

```
(1) ?p rdf:type c:Person
(2) ?p c:name ?n
(3) ?p c:author ?doc
(4) ?doc rdf:type c:Thesis
(5) ?doc c:Title ?title
(6) ?title ~ 'web'
```

Chaque élément de la requête est un triplet de la forme `ressource propriété valeur` ou une expression évaluable. Le noyau du langage de requête de Corese est compatible avec la proprosition SPARQL du W3C.

Corese exploite la projection de graphes pour calculer les réponses aux requêtes et implémente en outre quelques extensions de de cette opération.

[1] http://www.inria.fr/acacia/corese

B. Variable de propriété

Le langage de requête de Corese permet d'utiliser des variables pour désigner des propriétés dans une requête. La ligne (4) de la requête suivante recherche toute valeur de propriété, désignée par `?p`, du document. Plusieurs occurrences de la même variable sont liées à la même propriété par projection.

```
(1) ?doc rdf:type c:Thesis
(2) ?doc c:title ?title
(3) ?title ~ 'web'
(4) ?doc ?p ?value
```

C. Chemins

Corese permet de rechercher des chemins de relations, orientés ou non orientés reliant deux ressources. Par exemple, la requête suivante recherche une personne reliée à une ressource de type science sociale par un chemin non orienté de longueur au plus 5, par des relations de type (ou sous-types de) c:Relation :

```
(1) ?person rdf:type c:Person
(2) ?person c:Relation{5} ?topic
(3) ?topic rdf:type c:SocialScience
```

D. Projection approchée

Dans certains cas, une requête peut ne pas avoir de réponse exacte dans la base de graphes. Corese implémente un algorithme de recherche approchée qui calcule les meilleures approximations des types demandés. L'évaluation des réponses est faite en calculant et minimisant une distance sémantique dans l'ontologie [3].

La distance sémantique entre un concept requête et un concept cible est la plus petite somme des longueurs des chemins entre chacun des types des concepts et leur plus précis super type commun. La longueur d'un chemin est la somme des longueurs élémentaires entre les types des concepts. La longueur d'un arc entre un fils et un père de profondeur d est $1/2^d$. Ainsi la distance décroit avec la profondeur.

Définition 1: $\forall (t_1, t_2) \in H^2$,

$$D_H(t_1, t_2) = min_t(l_H(\langle t_1, t\rangle) + l_H(\langle t_2, t\rangle)) =$$

$$min_t(\sum_{\{x \in <t_1,t>, x \neq t_1\}} 1/2^{d_H(x)} + \sum_{x \in <t_2,t>, x \neq t_2} 1/2^{d_H(x)})$$

où t est un super type commun de t_1 and t_2 de profondeur maximale.

Ainsi, le moteur Corese est capable de trouver *un support de cours sur la programmation par objets* comme réponse approchée à une requête demandant *un rapport sur Java*. En effet, rapport et support de cours sont proches sémantiquement, de même que Java et programmation par objets. Le moteur retourne les meilleurs résultats par ordre de distance croissante.

L'énoncé suivant permet de demander une recherche approchée :

```
select more where
?doc rdf:type c:Report
?doc c:concern ?topic
?topic rdf:type c:Java
```

E. Présentation des résultats

Des directives au moteur permettent de paramétrer la présentation des résultats. On peut demander à Corese de ne retourner qu'une partie des résultats calculés (clause `select` similaire à SQL). Il est possible de grouper les réponses partageant les mêmes concepts (e.g. grouper les documents par auteur et par date). Il est possible de compter le nombre de réponses différentes (e.g. grouper par auteur et compter le nombre de documents). Enfin le moteur peut trier les résultats selon certains critères (e.g. trier par année décroissante et par ordre alphabétique de noms d'auteur).

Le moteur construit des graphes réponses correspondant aux projections, en suivant ces directives. Puis il engendre un format RDF/XML suivant la correspondance GC vers RDF. Ces documents RDF/XML peuvent être servis tels quels dans le standard du Web sémantique. Ils peuvent également être traités par un moteur de feuilles de style XSLT pour produire du code HTML présentable dans un navigateur web standard.

IV. Langage de règles

Le modèle des graphes conceptuels introduit la notion de règle de graphes [4]. Nous avons conçu et intégré un moteur de règles de graphes fonctionnant en chaînage avant sur la base de graphes. Le test des conditions se fait par projection. L'application de la règle consiste à joindre la conclusion sur les concepts trouvés par projection. Nous avons également proposé un langage de règles avec une syntaxe compatible avec RDF et reposant sur un sous-ensemble du langage de requête de Corese. Voici un exemple de règle stipulant qu'une personne à la tête d'une équipe dirige les membres de l'équipe et est un manager :

```
<cos:rule>
 <cos:if>
    ?m rdf:type s:Person
    ?m s:head ?t
    ?t rdf:type s:Team
    ?t s:hasMember ?p
    ?p rdf:type s:Person
 </cos:if>
<cos:then>
    ?m s:manage ?p
    ?m rdf:type c:Manager
</cos:then>
</cos:rule>
```

Ce langage de règles permet d'exprimer plusieurs propriétés d'ontologies similaires à celles du langage OWL: propriétés algébriques (transitivité d'une propriété, symétrie, inverse), restrictions de valeurs, intersection de classes...

V. Architecture

Le moteur est écrit en Java, il est constitué des composants logiciels principaux suivants :

- un serveur de graphes conceptuels utilisant les structures de données de Notio [5],
- un traducteurs RDF/CG et CG/RDF
- un parser des langages de requêtes et de règles

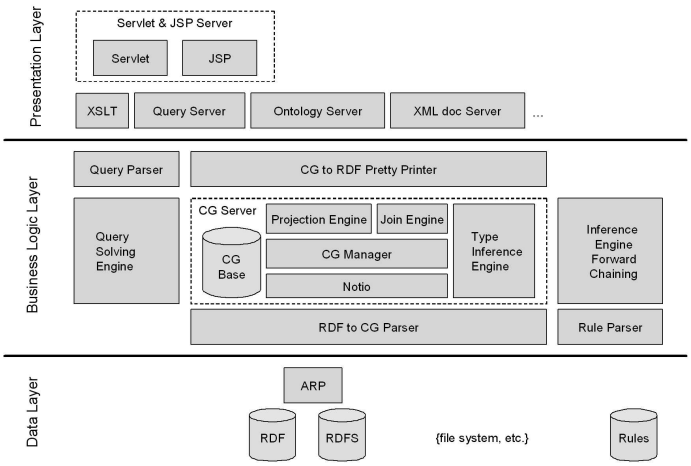


Fig. 1. Corese : architecture 3-tier

- un moteur de recherche basé sur la projection
- un moteur de règles d'inférence en chaînage avant
- un serveur web sémantique

L'architecture du serveur web sémantique (SWS) intégrant Corese et basé sur Tomcat est celle d'une couche supplémentaire autour du moteur de recherche gérant les échanges avec le Web et avec les ressources locales. L'application web implantant le SWS permet à un *Web Master* de concevoir un site reposant, entièrement ou en partie, sur des requêtes posées au moteur de recherche Corese et sur la transformation du résultat de ces requêtes en pages HTML, JSP ou SVG en utilisant des feuilles de style XSLT.

Afin d'assurer une indépendance maximale de l'application au domaine, l'architecture du serveur implante une forme de MVC (Model-View-Controler) pour séparer les aspects données, les aspects calculs et les aspects rendu. Pour ce faire, l'application web se décompose en plusieurs composants réutilisables, notamment:

- Des servlets implantant des tâches récurrentes dans nos scénarios d'utilisation d'un serveur web sémantique: résolution d'une requête personnalisée, modification d'une annotation, traitement d'un formulaire, gestion des gabarits des pages, etc.
- Des feuilles de style implantant des transformations XSLT dans un langage déclaratif et donc modifiable par l'utilisateur. Elles sont utilisées pour des tâches de rendu (format d'affichage d'une réponse, génération du JSP d'un formulaire, génération d'une vue de l'ontologie, etc.) et des tâches de modification de fichiers XML (édition d'une annotation, d'une classe, etc.). Corese est utilisé pour étendre les feuilles de style XSLT en fournissant des fonctions sémantiques supplémentaires. De plus les feuilles XSLT sont aussi utilisées pour décrire des patrons d'extraction de données pour des bases de données légataires par exemple.
- Un Tag JSP permet de faire appel à Corese au sein d'un page JSP et permet ainsi de générer des éléments au vol comme, par exemple, un menu, une liste de requêtes contextuelles ou la présentation d'une sous-partie de l'ontologie.

VI. Applications

Corese a été testé sur plus d'une dizaine d'applications parmi lesquelles :

1) Weblearn : action spécifique du CNRS sur le eLearning 2004-2005.
2) Meat : Description d'expériences de puces à ADN avec l'IPMC, 2003-2005.
3) KMP : Knowledge Management Platform, projet RNRT 2003-2004 avec INRIA, Rodige, Latapses, Telecom Valley, et le GET.
4) EADS CCR : prototype de mémoire de laboratoire de recherche industriel. 2004 et 2005.
5) Samovar : mémoire de projet de conception automobile avec Renault, 2001.
6) Comma : Corporate Memory Management through agents, projet IST, 2000-2001.

Conclusion

Le moteur Corese est une implémentation de RDF/S basée sur les graphes conceptuels. C'est une application d'envergure des graphes conceptuels tirant partie des formats standards du web sémantique. Cette approche s'est avérée très féconde car permettant de découpler les aspects syntaxique, sémantique et pragmatique. Elle permet d'exploiter la puissance du formalisme des GC en interne tout en parlant un langage standard, RDF/XML, pour les échanges avec le monde extérieur. Elle a également permis l'intégration forte des traitements sémantiques de RDF dans la chaîne de traitement XML pour le Web.

Remerciement

Corese a bénéficié d'un soutien de l'Inria sous forme d'une opération de développement logiciel de deux ans, merci à Olivier Savoie. Le développement de Corese a également été financé par le projet européen Comma.

References

[1] Reflections on Web Architecture. Conceptual Graphs and the Semantic Web, 2001 `http://www.w3.org/DesignIssues/CG.html`

[2] O. Corby, R. Dieng, C. Hebert. A conceptual graph model for W3C Resource Description Framework. In Proc. of the 8th International Conference on Conceptual Structures, ICCS'00, LNCS 1867, Springer-Verlag, pp. 468-482, Darmstadt, Germany, 2000

[3] O. Corby, R. Dieng-Kuntz, C.Faron-Zucker. Querying the Semantic Web with the Corese Search Engine, To appear in Proc. of the 3rd Prestigious Applications Intelligent Systems Conference, PAIS2004, in conjunction with the 16th European Conference on Artificial Intelligence, ECAI2004, Valencia, Spain, 23-27 August 2004.

[4] E. Salvat. Theorem Proving Using Graph Operations in the Conceptual Graph Formalism, In Proc. of the 13th European Conference on Artificial Intelligence, ECAI98, pp. 356-360, Brighton, UK, 1998

[5] F. Southey and J. G. Linders, Notio - A Java API for Developing CG Tools, In Proc. of the 7th International Conference on Conceptual Structures, ICCS'99, LNAI 1640, Springer-Verlag, pp. 262-271, 1999

[6] Eric Prud'hommeaux, Andy Seaborne. SPARQL Query Language for RDF W3C Working Draft 2005 `http://www.w3.org/2001/sw/DataAccess/rq23`

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.

Raisonner sur les ontologies au niveau meta

Francky Trichet, Frédéric Fürst

LINA - FRE CNRS 2729 - Université de Nantes
2 rue de la Houssinière - BP 92208
44322 Nantes Cedex 3, France

{trichet,furst}@univ-nantes.fr

1 Introduction

Les ontologies sont actuellement au cœur de nombreuses applications de l'Ingénierie des Connaissances, en particulier le Web Sémantique, car elles permettent de raisonner sur des domaines de connaissances. Cependant, il apparait de plus en plus nécessaire de pouvoir raisonner sur les ontologies elles-mêmes, pour les évaluer, les aligner ou les fusionner. Dans ce cadre, nous proposons ici d'opérationnaliser des ontologies de représentation pour raisonner sur des ontologies de domaine. Nos travaux ont pour cadre le langage OCGL de représentation d'ontologie, inspiré du modèle des Graphes Conceptuels.

2 Operationnalisation : les principes

Nous nous plaçons ici dans le cadre de l'utilisation d'ontologies *lourdes* (heavyweight ontologies), c'est-à-dire d'ontologies intégrant toute la sémantique d'un domaine, à travers des axiomes, et non uniquement des ontologies légères (lightweight ontologies) qui ne prennent en compte que quelques propriétés telles que la subsomption [3, 4]. De telles ontologies, et les langages de représentation adéquats, sont nécessaires pour utiliser les connaissances pour raisonner, et non uniquement pour partager des terminologies. Dans les ontologies lourdes, les connaissances axiomatiques (non terminologiques) sont représentées via des *axiomes*, qui peuvent correspondre à des propriétés classiques (subsomptions, propriétés algébriques), que nous appellons ici schéma d'axiome, ou a des propriétés particulières au domaine, que nous appellons ici axiomes de domaine.

Pour garantir l'indépendance des ontologies vis-à-vis des applications, et donc leur réutilisabilité, la représentation des axiomes doit seulement préciser leur *sémantique formelle*, qui contraint l'interprétation des primitives conceptuelles (concepts et relations), sans fixer leur *sémantique opérationnelle*, qui spécifie la façon dont les axiomes sont utilisés pour raisonner [1]. Cependant, l'utilisation des connaissances axiomatiques dans un Système à Base de Connaissances (SBC) nécessite que soit précisée cette sémantique opérationnelle. C'est tout l'objet de l'*opérationnalisation* d'une ontologie, qui consiste à transcrire celle-ci dans un formalisme opérationnel de représentation des connaissances, doté de représentations à la sémantique opérationnelle fixée (essentiellement règles et contraintes), et ceci en fonction de l'objectif du SBC. La spécification de cet objectif se fait au travers d'un *scénario d'usage*, qui décrit l'usage des connaissances exprimées dans les axiomes dans le SBC. Ce scénario peut être raffiné en *contextes d'usage*, chaque contexte d'usage décrivant l'usage d'un axiome de l'ontologie à opérationnaliser (cf. figure 1).

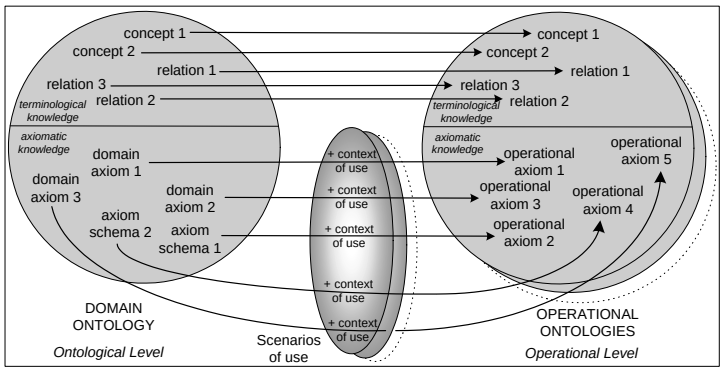


FIG. 1 – *Le processus d'operationnalisation d'une ontologie lourde. Les connaissances terminologiques sont représentées de la même façon au niveaux ontologique et opérationnels. La représentation des connaissances axiomatiques au niveau opérationnel dépend du scénario d'usage.*

3 Operationnalisation des ontologies de représentation

Opérationnaliser une ontologie de domaine correspond donc à la construction d'un SBC qui peut être utilisé pour raisonner sur des instances du domaine. Si on considère l'ontologie de domaine elle-même comme une instance du langage de représentation utilisé, raisonner sur cette ontologie peut être réalisé dans un SBC intégrant une version opérationnelle d'une ontologie du langage de représentation. Ainsi, opérationnaliser une ontologie d'un langage, c'est participer à la construction d'un SBC dédié au raisonnement sur les ontologies exprimées dans ce langage (cf. figure 2).

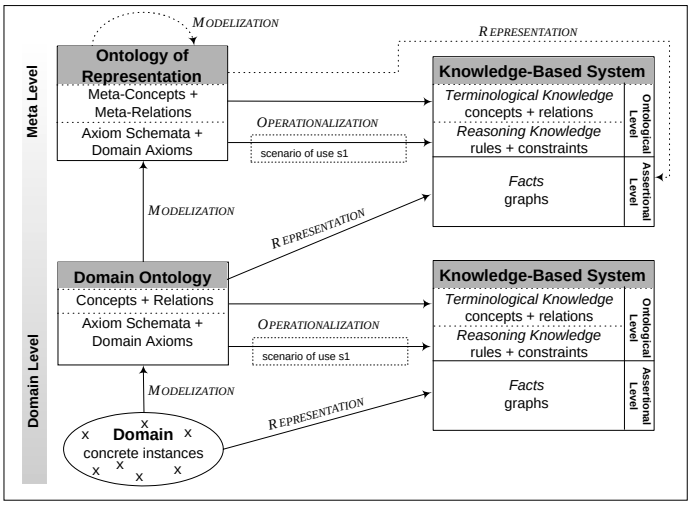


FIG. 2 – *Opérationnalisation des ontologies de domaine et des meta-ontologies.*

Nous avons utilisé ce principe pour mener des tests sur une ontologie exprimée en OCGL, un langage de représentation d'ontologie issu du modèle des Graphes Conceptuels, et implémenté dans l'outil TooCoM d'édition et d'opérationnalisation d'ontologie [1]. Après avoir construit l'ontologie d'OCGL en OCGL, nommée MetaOCGL, nous l'avons opérationnalisée dans un scénario d'usage dédié à la validation, ce qui nous a permis d'utiliser l'ontologie opérationnelle ainsi générée pour évaluer des ontologies exprimées en OCGL. De même, opérationnaliser MetaOCGL dans un scénario dédié à la déduction a permis de compléter automatiquement les ontologies de domaine [2].

4 Conclusion

Utiliser les ontologies de représentation pour raisonner sur les ontologies de domaine permet d'éviter l'implémentation de mécanismes de raisonnement particuliers à chaque formalisme de représentation. Cette approche permet d'étendre l'exigence de portabilité et de généricité des ontologies vis-à-vis des SBC dédiés au raisonnement au niveau domaine, aux SBC dédiés au raisonnement sur les ontologies elles-mêmes. Un parallèle peut être établi avec l'ingénierie des modèles et l'architecture MDA (Model Driven Architecture) (cf. figure 3). Dans les deux cas, les ontologies (resp. modèles) évoluent du statut de descriptions d'un domaine (resp. d'un système), vers des ontologies supports du raisonnement dans les SBC (heavyweight ontologies) (resp. supports à la construction automatique de systèmes). Et dans les deux cas, l'utilisation de représentations de plus en plus abstraites permet d'augmenter la portabilité des méthodes et d'améliorer leur efficacité.

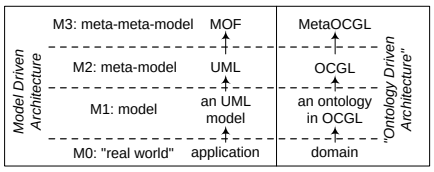


FIG. 3 – *Architecture MDA et son pendant en ingénierie des ontologies.*

Références

[1] F. Fürst, M. Leclère, and F. Trichet. Operationalizing domain ontologies: a method and a tool. In *European Conference on Artificial Intelligence (ECAI'2004)*, pages 318–322, 2004.

[2] F. Fürst and F. Trichet. Axiom-based ontology matching: a method and an experiment. Research report 05-02, LINA, Nantes, 2005.

[3] A. Gomez-Perez, M. Fernandez-Lopez, and O. Corcho. *Ontological Engineering.* Springer, Advanced Information and Knowledge Processing, 2003.

[4] S. Staab and A. Maedche. Axioms are objects too: Ontology engineering beyong the modeling of concepts and relations. Research report 399, Institute AIFB, Karlsruhe, 2000.

Proximité Conceptuelle et Distances de Graphes

Fabien Gandon, Olivier Corby, Rose Dieng-Kuntz, Alain Giboin
ACACIA, INRIA Sophia Antipolis
Fabien. Gandon@sophia.inria.fr

1 Simuler la Proximité de Concepts

Intuitivement nous sommes tous portés à dire que le concept de *voiture* est plus proche du concept de *camion* que de celui d'*avion* ; cependant nous pensons aussi que le concept de *voiture* est plus proche du concept d'*avion* que du concept de *livre*. Ces distances intuitives peuvent être simulées par exemple pour améliorer les moteurs de recherche du Web dans leurs algorithmes de filtrage et de tri des réponses.

L'idée d'évaluer la proximité conceptuelle sur des réseaux sémantiques remonte au travaux de [Quillian, 1968] et [Collins & Loftus, 1975] sur la mémoire sémantique humaine. La proximité de deux concepts peut venir d'une complémentarité fonctionnelle (ex: un clou et un marteau), d'une similarité fonctionnelle (ex: un marteau et un tournevis), etc. Ce dernier exemple appartient à la famille des similarités sémantiques dans laquelle la proximité est basée sur une caractéristique définitionnelle partagée (ex: être un outil).

Une structure supportant naturellement le résonnement sur les similarités sémantiques est la hiérarchie des types telle que l'on peut la trouver dans un support en graphes conceptuels. En effet, dans cette structure, les liens de subsomption groupent les types suivant les caractéristiques définitionnelles qu'ils partagent. Lorsqu'elle est appliquée au graphe d'une hiérarchie, une proximité calculée par propagation donne une distance sémantique, la première et la plus simple étant celle qui compte les arcs [Rada et al., 1989].

Deux grandes familles d'approches peuvent être identifiées pour le calcul de telles distances: (A) celles qui incluent des informations externes à la hiérarchie, ex: des statistiques sur l'utilisation des types de concepts [Resnik, 1995] [Jiang & Conrath, 1997], et (B) les approches reposant uniquement sur la hiérarchie [Rada et al., 1989][Wu & Palmer, 1994]. Dans le domaine des graphes conceptuels, seule la deuxième approche est utilisée en particulier pour proposer une projection ne prenant plus uniquement des valeurs booléennes *i.e.* une similarité $S:C^2 \rightarrow [0,1]$ où 1 correspond à la valeur *vraie* de la projection classique et toute autre valeur donne une idée de la similarité entre le graphe projeté et le graphe source. L'utilisation initiale faite par Sowa visait à permet des déplacements de côté dans le treillis des types. [Ralescu & Fadlalla, 1990] l'ont utilisé pour relaxer les contraintes de l'opérateur de jointure. Plus récemment, [Zhong et al., 2002] ont utilisé une distance atténuée par la profondeur des types dans l'ontologie pour construire une mesure de similarité entre graphes conceptuels.

Dans la suite, nous donnons un aperçu des applications et des travaux que nous menons autour de cette notion de proximité ou distance sémantique.

2 Proximité Conceptuelle et Distribution

Dans un Web sémantique d'entreprise les scénarios amènent souvent la contrainte de bases d'annotations (assertions à propos de ressources documentaires) distribuées. Pour gérer cette distribution nous avons proposé une architecture et des protocoles permettant en particulier de maintenir la spécialisation des bases d'annotations quant aux sujets abordés dans leurs assertions.

Chaque archive de notre architecture maintient une structure appelée ABIS (Annotation Base Instances Statistics) décrivant des statistiques sur les types de triplets (relations binaires imposées par le modèle [RDF]) présents dans leur base d'annotations. Par exemple si, dans l'ontologie, il existe une propriété Auteur avec la signature:

[Document] → (Auteur) → [Personne]

L'ABIS pourra contenir des statistiques sur l'existence des instances de triplets suivantes:

[Article] → (Auteur) → [Etudiant]
[Livre] → (Auteur) → [Philosophe] …

L'ABIS est construit lors de la transformation des annotations RDF en graphes conceptuels et capture la contribution d'une archive à la mémoire globale en terme de types de connaissances. Ainsi il fournit un moyen de comparer deux bases et nous l'utilisons pour maintenir la spécialisation des bases d'annotations grâce à une distance sémantique.

Pour comparer deux types primitifs, nous utilisons la distance de [Rada et al., 1989] comptant le nombre d'arc sur le chemin le plus court qui relie ces deux types à travers la hiérarchie; voir formule (1). En utilisant cette distance on peut définir une distance entre deux triplets RDF (ou deux instances d'une relation binaire), comme étant la somme des distances entre: les types des deux relations, les types des deux concepts en premier argument (domain) et les types des deux concepts en deuxième argument (range); voir formule (2). La distance entre un triplet et un ABIS est alors définie comme la distance minimale entre ce triplet et les triplets recensés par l'ABIS; voir formule (3). Et finalement,

la distance entre une annotation et un ABIS est la somme des distances entre chaque triplet de l'annotation en l'ABIS; voir formule (4).

$$dist(t_1,t_2) = length(t_1, lcst(t_1,t_2)) + length(t_2, lcst(t_1,t_2)) \quad (1)$$

où , $lcst(t_1,t_2))$ est le plus proche supertype commun de t_1 et t_2.

$$dist(triple_1, triple_2) = dist(domain(triple_1), domain(triple_2)) + dist(predicate(triple_1), predicate(triple_2)) + dist(range(triple_1), range(triple_2)) \quad (2)$$

$$dist(triple, ABIS) = \min_{triple_i \in ABIS} (dist(triple, triple_i)) \quad (3)$$

$$dist(An, ABIS) = \sum_{triple_j \in An} dist(triple_j, ABIS) \quad (4)$$

Cette distance donne une fonction d'évaluation / fonction de coût utilisée comme critère dans un protocole de mise aux enchères des nouvelles annotations à archiver: chaque nouvelle annotation est mise aux enchères entre les archives existantes; chaque archive fait une offre qui correspond à la distance entre son ABIS et l'annotation; l'archive avec l'offre la plus petite gagne l'annotation. Ce protocole permet de maintenir la spécialisation des bases et ainsi de faciliter l'optimisation de la résolution de requêtes distribuées en utilisant les ABIS pour la décomposition et le routage des projections. Il reste à faire l'étude de l'influence des constructeurs de combinaisons utilisés dans ces distances (minimum, maximum, moyenne, somme, etc.). Ici la définition d'une distance conceptuelle sur la hiérarchie des types permet de construire un consensus calculatoire (distance) au dessus du consensus ontologique (support), et de l'utiliser dans un consensus protocolaire (enchères).

3 Proximité Conceptuelle et Approximation

La plateforme CORESE [Corby et al, 2004] intègre une fonctionnalité de recherche approchée qui démontre une autre application des inférences simulant la proximité conceptuelle. CORESE utilise une extension de la distance atténuée par la profondeur [Zhong et al., 2002] des types dans le treillis de l'ontologie; voir formules (5) et (6).

$$\forall (t_1,t_2) \in H^2; t_1 \le t_2 \text{ on a } l_H(t_1,t_2) = \sum_{\{t \in \langle t_1,t_2 \rangle, t \ne t_1\}} \left[\frac{1}{2^{depth(t)}} \right] \quad (5)$$

avec H la hiérarchie des types de concepts, $\langle t_1, t_2 \rangle$ le chemin le plus court entre t_1 et t_2, et *depth(t)* la profondeur de t dans l'ontologie *i.e.* le nombre d'arcs sur le chemin le plus court entre t et la racine $\top$

$$\forall (t_1,t_2) \in H^2 \text{ on a } dist(t_1,t_2) = \min_{\{t \ge t_1, t \ge t_2\}} \left(l_H(t_1,t) + l_H(t_2,t) \right) \quad (6)$$

En utilisant cette distance on peut relaxer la contrainte d'égalité ou de spécialisation des types lors de la projection en la remplaçant par une contrainte de proximité utilisant une distance conceptuelle comme celle définie en (6). On obtient alors une projection approchée.

4 Proximité Conceptuelle et Regroupement

Dans le cadre du projet KmP [KMP] nous nous sommes intéressés à la construction d'un algorithme de regroupement (clustering) des compétences présentes sur la Télécom Valley de Sophia Antipolis et annotées en RDF. Le regroupement normalement effectué manuellement par les experts en management et économistes s'est révélé être un algorithme de regroupement monothétique (monothetic clustering). La représentation recherchée demandait de pouvoir fournir des moyens de contrôler simplement le niveau de détail et de granularité choisi pour générer le regroupement. En analyse de données [Jain et al., 1999], une structure classique supportant le choix des niveaux de détail est le dendrogramme, un arbre qui, à chaque niveau de coupure, donne une solution de regroupement plus ou moins fin.

Un dendrogramme repose sur une ultra-métrique c'est-à-dire une distance avec une inégalité triangulaire sur contrainte: $dist(t_1,t_2) \le max(dist(t_1,t'), dist(t_2,t'))$ pour tout t'

Nous avons donc cherché à construire cette ultramétrique à partir de la distance sémantique de CORESE. Nous n'avons considéré pour cela que les structures d'arbres ce qui nous donne une distance exacte ayant pour formule (7).

$$dist(t_1,t_2) = \frac{1}{2^{depth(lcst(t_1,t_2))-2}} - \frac{1}{2^{depth(t_1)-1}} - \frac{1}{2^{depth(t_2)-1}} \quad (7)$$

Nous avons en suite proposé une transformation produisant une ultramétrique et améliorant le nombre de niveaux de détail disponibles dans de dendrogramme obtenu en favorisant le regroupement des classes ayant une descendance peu profonde, formule (8) et figure 1.

$$dist_{CH}(t_1,t_2) = \max_{\forall st \le lcst(t_1,t_2)} \left(dist(st, lcst(t_1,t_2)) \right) \text{ quand } t_1 \ne t_2 \quad (8)$$

$$dist_{CH}(t_1,t_2) = 0 \text{ quand } t_1 = t_2$$

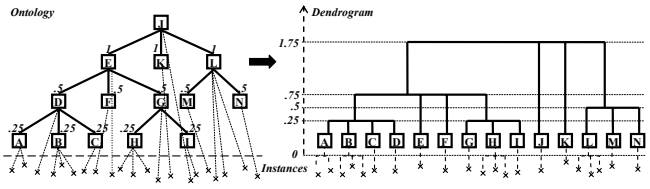


Figure 1. Transformation de la distance en ultramétrique.

Comme le regroupement suit la hiérarchie des types, on peut nommer chaque regroupement. Un exemple de regroupement des compétences sur la Télécom Valley est donné en Figure 2.

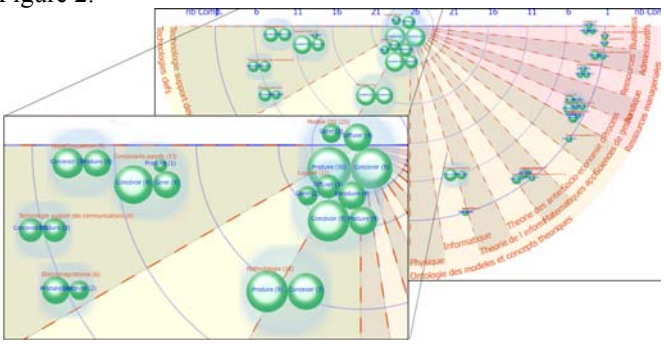


Figure 2. Radar view of the clusters on 180°

5 Proximité Simulée vs. Proximité Naturelle

En parallèle avec notre exploration des caractéristiques, interprétations et applications des distances conceptuelles. Nous avons commencé à questionner la valeur de ces distances et leur fidélité par rapport aux proximités naturellement ressenties par les humains. Pour cela nous avons commencé une étude empirique et statistique. Une première hypothèse testée est "Est-il juste de considérer que les frères sont à égale distance du père et à égale distance les uns des autres ou est-ce un effet secondaire du fait que l'on repose sur la structure des chemins de subsomption?".

Afin d'étudier ces distances dans leur milieu naturel et de les comparer avec leurs simulations informatiques, nous avons conçu une plateforme permettant de réaliser, gérer, et d'analyser des expériences où les participants organisent et regroupent spatialement des concepts selon leur proximité intuitive [Boutet et al., 2005]; voir figure 3.

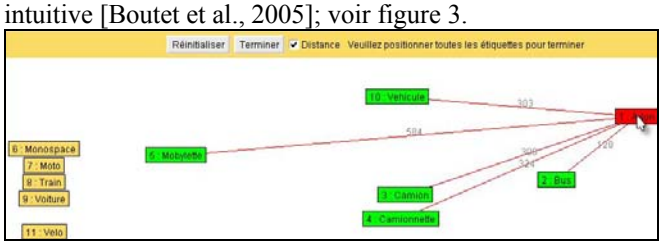
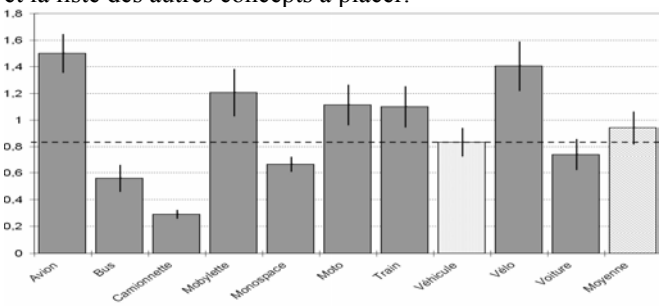


Figure 3. Applet de l'exercice de placement

A partir de ces exercices, des analyses statistiques sont faites pour étudier les distances conceptuelles naturelles. Prenons l'exercice de la figure 3 effectué par 30 participants de 13 à 50 ans. Les distances capturées ont été normalisées avant d'en calculer la moyenne, l'écart type et la variance. Le graphique 1 montre les distances entre le concept *camion* et la liste des autres concepts à placer.



Graphique 1. Distances entre *Camion* et d'autres véhicules

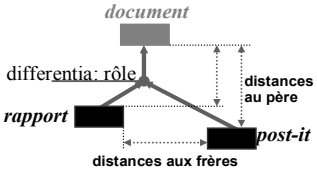
On peut lire sur ce graphique que le concept *camionnette* est en moyenne très proche de *camion* et, étant donné que la variance est très faible, qu'il s'agit d'un consensus. Le concept *véhicule* est particulier dans cette liste puisque dans une ontologie il serait naturellement placé comme père (ou ancêtre) des autres concepts et que ces autres concepts seraient entre eux des frères (ou des cousins). On voit de façon intéressante que la distance entre *camion* et ses frères est parfois plus petite (4 cas) et parfois plus grande (5 cas) que la distance à son père *véhicule*. Il reste beaucoup à faire dans cette étude mais si ces résultats se confirmaient ils montreraient qu'une structure de subsomption seule ne permet pas de simuler de tels comportements.

6 Perspectives des Distances Ontologiques

Les distances conceptuelles laissent de nombreuses questions de recherche nourries par un ensemble grandissant d'applications, en particulier:

Proximité naturelle vs. distance mathématique: la distance utilisée par CORESE dans sa recherche approchée est une semi-distance i.e. elle ne vérifie pas l'inégalité triangulaire. Quelle est la valeur des conditions nécessaires de la définition mathématique des distances? Devons nous à tout prix essayer de les respecter ou est-ce simplement une limite de la métaphore des distances? Quels seraient sinon les caractéristiques définitionnelles d'une distance conceptuelle?

Distance conceptuelle vs. chemin de subsomption: les structures de la hiérarchie de types sont le terrain favori pour la définition des distances conceptuelles. Faut-il considérer des représentations plus riches incluant, par exemple, des liens frère-frère, qui permettraient de mieux simuler de telles distances? Comment mieux définir ces distances? Comment étudier les différentes familles de distances qui semblent cohabiter dans nos inférences au quotidien? Comment les capturer, les apprendre, pour les utiliser dans des inférences de recherche d'information?



Références

[Boutet et al., 2005] Boutet, M., Canto, A., Roux, E., Plateforme d'étude et de comparaison de distances conceptuelles, Rapport de Master, Ecole Supérieure En Sciences Informatiques, 2005

[Collins & Loftus, 1975] Collins, A., Loftus, E., A Spreading Activation Theory of Semantic Processing. *Psychological Review*, vol. 82, pp. 407-428, 1975

[Corby et al, 2004] Corby, O., Dieng-Kuntz, R., Faron-Zucker, C., Querying the Semantic Web with the Corese Search Engine, In Proc. of *ECAI*, IOS Press, pp.705-709, 2004

[Jain et al., 1999] Jain, A.K., Murty, M.N., and Flynn, P.J. (1999): Data Clustering: A Review, *ACM Computing Surveys*, 31(3) 264-323.

[Jiang & Conrath, 1997] Jiang, J., Conrath, D., Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. In Proc. of *International Conference on Research in Computational Linguistics*, Taiwan, 1997

[KMP] http://www-sop.inria.fr/acacia/soft/kmp.html

[Quillian, 1968] Quillian, M.R., Semantic Memory, in: M. Minsky (Ed.), *Semantic Information Processing*, M.I.T. Press, Cambridge, 1968.

[Rada et al., 1989] Rada, R., Mili, H., Bicknell, E., Blettner, M., Development and Application of a Metric on Semantic Nets, *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 19(1), pp. 17-30, 1989.

[Ralescu & Fadlalla, 1990] A.L.Ralescu,A. Fadlalla. The Issue of Semantic Distance in Knowledge Representation with Conceptua Graphs, In Proc. Of AWOCS90, pp. 141-142, 1990

[RDF] http://www.w3.org/RDF/

[Resnik, 1995] Resnik, P., Semantic Similarity in a Taxonomy: An Information-Based Measure and its Applications to Problems of Ambiguity in Natural Language. In *Journal of Artificial Intelligence Research*, vol 11, pp. 95-130, 1995

[Sowa, 1984] Sowa., J.F., *Conceptual structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, Massachusetts, 1984

[Zhong et al., 2002] J. Zhong, H. Zhu, J. Li, Y. Yu. Conceptual Graph Matching for Semantic Search, In Proc. of 10th International Conference on Conceptual Structures, ICCS2002, LNCS 2393, Springer Verlag, pp. 92-106, Borovets, Bulgaria, 2002

Raisonner le Web Sémantique avec des graphes : Application à un cas industriel

Olivier Carloni

LIRMM, Université de Montpellier II,
161, rue Ada, F-34392 Montpellier cedex - France
carloni@lirmm.fr

Mondeca,
3, cité Nollez, 75018 Paris
http://www.mondeca.com

Abstract. Ce travail décrit un projet de recherche qui sera réalisé dans le cadre d'une thèse CIFRE en collaboration entre la société Mondeca et l'équipe RCR (Représentation des Connaissances et Raisonnements) du LIRMM.

1 Le contexte

1.1 La société

Mondeca est un éditeur de logiciel impliqué dans le Web Sémantique. Elle a développé ITM (Intelligent Topic Manager), un logiciel de gestion documentaire et d'ingénierie des connaissances. Un processus d'annotation semi-automatique permet d'extraire les informations pertinentes contenues dans les documents. Ses dernières sont alors représentées dans une base de connaissances que les utilisateurs peuvent consulter et enrichir. L'objectif de Mondeca pour les années à venir est d'améliorer la gestion des connaissances qu'offre déjà ITM, notamment en automatisant certaines vérifications et en munissant son outil de facultés d'inférence.

1.2 Le principal formalisme de description de la connaissance dans ITM

Les informations extraites des textes sont décrites dans la base de connaissances en utilisant le formalisme des Topic Maps. Une topic map [FIG01] se construit à partir de trois primitives typées : les topics, les roles et les associations. Les topics peuvent être reliés à des associations d'arité quelconque par l'intermédiaire de roles (tous les roles sont binaires). Chaque topic joue un rôle dans l'association qui le connecte à un autre topic. Les roles permettent, en quelque sorte, de nommer les "paramètres" des associations. Le type d'une association n'impose pas à son instance d'arité fixe. Chaque primitive du réseau topic map possède un identifiant unique et la structure d'un topic et d'une association permet de comporter des attributs.

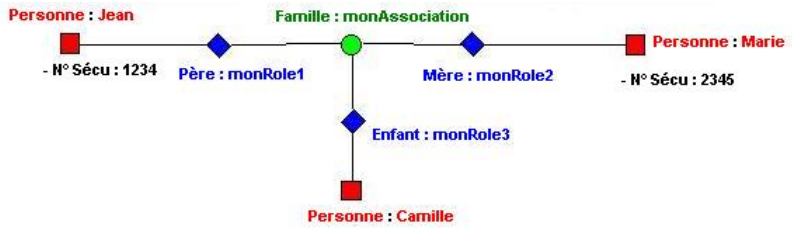


Fig. 1. Une topic map représentant une famille

1.3 Le modèle M_{ITM} de représentation de la connaissance dans ITM

Le modèle M_{ITM} de représentation des connaissances défendu par Mondeca comporte trois niveaux de topic maps :

1. Le niveau "méta", qui se définit lui même (définition réflexive) et comporte tous les constructeurs permettant de créer des ontologies de domaine pour chaque client. Ce niveau évolue très peu et est commun à tous les clients.
2. Le niveau "ontologique". C'est à ce niveau que sont créées les ontologies de domaine pour chaque client en utilisant les constructeurs du niveau méta. On définit ici des types et des propriétés générales qui seront utilisées au niveau "factuel". Il existe plusieurs sortes d'"ontologies" différentes : les ontologies de domaine spécifiques au client, mais aussi des ontologies construites pour gérer les services offerts par le logiciel. On retrouve, par exemple, les ontologies de "gestion documentaire", de "gestion terminologique", de "gestion des connaissances".
3. Le niveau "factuel". Les types et propriétés de l'ontologie permettent de définir les éléments de ce niveau. Les descriptions faites peuvent être du genre organisationnel (se réfère à l'ontologie de gestion documentaire), terminologique ou "faits du monde réel".

La hierarchie entre les niveaux est établie par une relation d'instanciation qui permet de typer les éléments d'un niveau en utilisant les types définis dans le niveau supérieur.

L'exemple FIG02 illustre la contrainte qu'exerce le réseau des topics décrit à un niveau, sur la topologie du réseau et la structure des éléments du niveau inférieur. "Jean" ne peut être connecté à l'association3 que si dans l'ontologie la classe de "Jean" est bien connectée par "type_d'association_permis" au topic représentant le type de l'association3 (i.e. "appartenir_famille").

Les ontologies d'ITM sont d'abord définies en OWL en utilisant l'outil d'édition Protégé [PR01].

Une fois l'ontologie établie en OWL pour un client (après étude des besoins client), cette dernière est traduite en Topic Maps pour être transférée dans ITM.

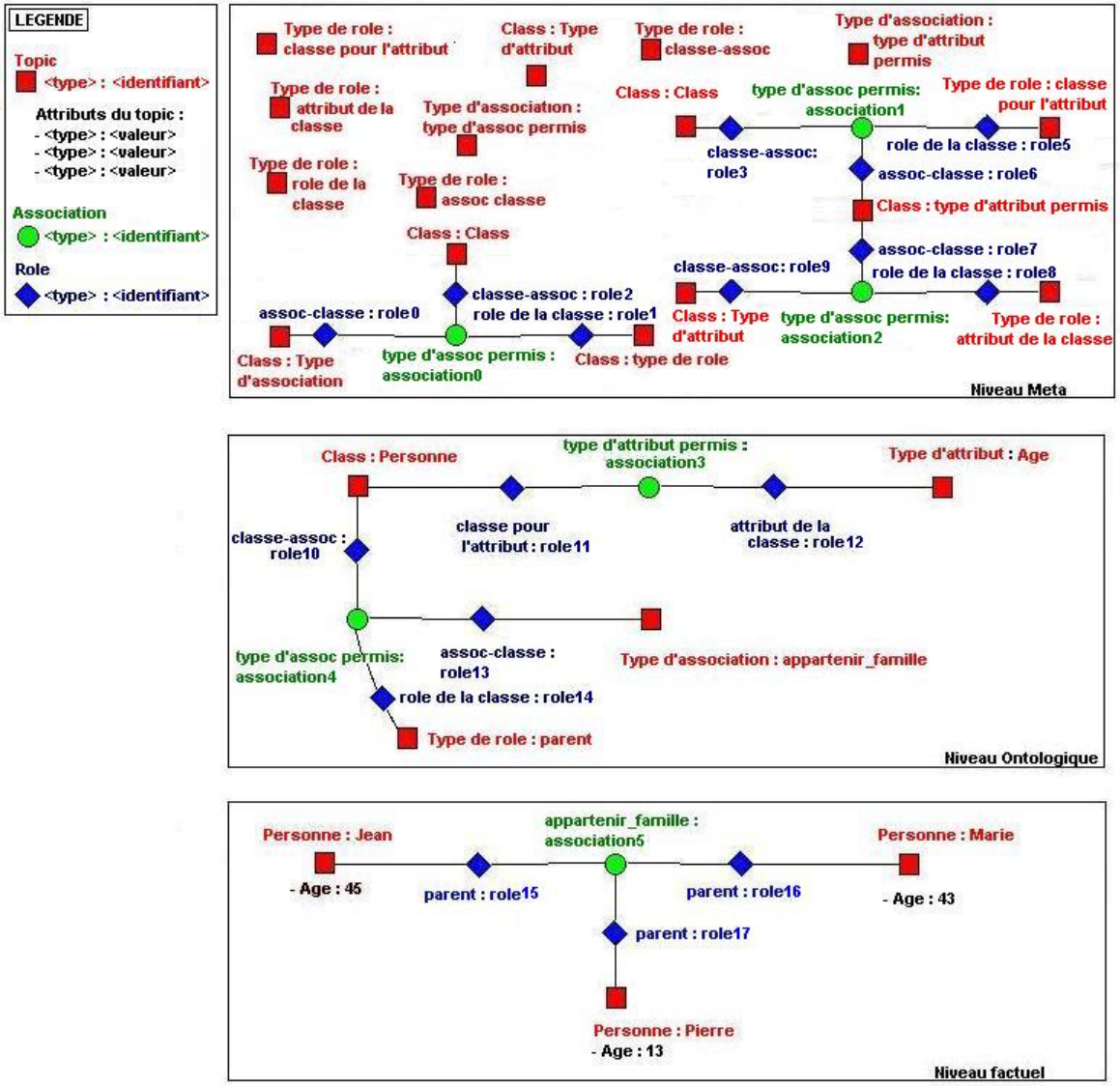


Fig. 2. Le modèle M_{ITM} avec son niveau méta, ontologique et factuel.

Le formalisme des Topic Maps a été délibérément conçu pour s'affranchir d'une sémantique trop contraignante et n'a donc pas de sémantique formelle. OWL devrait permettre de lui en fournir une.

Pourtant, Mondeca souhaite pourvoir son outil de capacités de raisonnement permettant notamment :

1. de contrôler la cohérence des descriptions faites dans la base par rapport au modèle (exemple : vérifier les faits par rapport à l'ontologie),
2. de déduire à partir de connaissances implicites exprimées sous forme de règles, de nouvelles représentations ou des réponses à des requêtes.

2 Les problématiques de recherche

2.1 L'hypothèse de recherche

Nous faisons l'hypothèse que les graphes conceptuels sont un bon candidat pour doter ITM de facultés d'inférence.

Le but de ce travail est donc d'évaluer dans quelle mesure la famille SG permet d'opérationnaliser les raisonnements dans ITM. La famille SG [BA01] des graphes conceptuels s'obtient à partir des graphes conceptuels simples essentiellement définis par :

- Le support : Il est composé d'une hiérarchie de types de concepts et de types de relations primitifs, d'ensembles de types incompatibles et d'un ensemble de marqueurs individuels typés.
- Les graphes conceptuels simples : Ce sont des graphes bipartis comportant des sommets relations et des sommets concepts qui sont étiquetés à partir du support. Les sommets concepts sont étiquetés par une conjonction correcte de types primitifs (pas de types incompatibles) et par un marqueur qui est soit le marqueur individuel, soit le marqueur générique.
- La projection : C'est un homomorphisme de graphes. La projection permet d'effectuer des raisonnements logiquement fondés car elle est adéquate et complète avec la déduction en logique du premier ordre sur les formules existencielles conjonctives.

A partir de ce modèle de base, on peut obtenir d'autres modèles plus expressifs en introduisant des règles ou des contraintes.

2.2 Sémantique formelle

Un des travaux théoriques importants est d'établir la sémantique formelle des représentations faites dans ITM à partir du modèle intuitif de représentation des connaissances (M_{ITM}) et du modèle décrit en OWL. Les principaux constructeurs de OWL utilisés sont ceux de OWL-Lite, mais le découpage n'est pas si simple et une petite partie de OWL-DL et OWL-Full est nécessaire pour une

représentation des connaissances en OWL cohérente avec le modèle M_{ITM}. Cette formalisation est en cours.

Lorsque la sémantique formelle est bien définie, il faut établir quelle est la portion des descriptions de la base qui peut être transcrite en graphes conceptuels tout en préservant cette sémantique. Il semble, à ce stade de l'étude, qu'une grande partie des descriptions peuvent être importées sans problème dans le modèle des graphes conceptuels simples.

2.3 Transformations en graphes conceptuels

Pour pouvoir raisonner en graphes conceptuels sur la connaissance décrite dans ITM, il faut définir des transformations syntaxiques du langage des topic maps vers le langage des graphes conceptuels tout en préservant le modèle de représentation des connaissances de Mondeca.

La transformation n'est pas unique car le modèle M_{ITM} comporte 3 niveaux conceptuels (les niveaux méta, ontologique et factuel) alors que celui des graphes conceptuels en comporte 2 (le support et le graphe), il existe alors trois ou quatre transformations potentiellement correctes et utiles.

2.4 Extensions pour ITM

Les graphes conceptuels permettront aussi d'enrichir le langage de représentation des connaissances de ITM. Par exemple, les graphes conceptuels simples permettent de représenter un individu non identifié par le marqueur générique, ce qui n'est pas possible pour l'instant dans ITM. Les règles et les contraintes de la famille SG augmenteront l'expressivité du modèle. Les extensions réalisées devront se positionner par rapport aux standards existants ou émergents du Web Sémantique (RDF, OWL, le tout jeune RuleML, ...)

2.5 Passage à l'échelle

ITM gère des bases de connaissances dont la taille peut atteindre le million de topics. Il est important de garantir un temps de réponse acceptable. Il faudra déterminer si le passage à l'échelle du moteur d'inférence se fait correctement.

CoGITaNT [CO01] est un des outils qui a été développé pour permettre des inférences en graphes conceptuels et TooCoM [TO01] permet de visualiser, d'éditer et d'opérationnaliser des ontologies. L'étude doit démontrer si dans l'état actuel de CoGITaNT (architecture, organisation des données, algorithmes), on peut gérer des graphes massifs en un temps raisonnable.

TooCoM permettra de visualiser et d'éditer les axiomes des ontologies avant d'évaluer les performances, dans CoGITaNT, d'une inférence à partir de ces axiomes.

References

[PR01] Editeur d'ontologies Protégé. Reference web : http://protege.stanford.edu

[CO01] CoGITaNT (COnceptual Graphs Integrated Tools Allowing Nested Typed Graphs) Reference web : http://cogitant.sourceforge.net

[BA01] Jean-François Baget, Marie-Laure Mugnier. The SG Family: Extensions of Simple Conceptual Graphs, Proceedings of the seventeenth International Joint Conference on Artificial Intelligence (IJCAI'2001), Seattle, Washington, USA, August 2001, vol. I, pages 205-210, Morgan Kaufmann, 2001.

[TO01] Frederic Fürst, Michel Leclère, Francky Trichet: Operationalizing Domain Ontologies: A Method and a Tool. ECAI 2004: 318-322.

Flexibilité de la recherche d'information dans les projets de conception

Chaker Djaiz[1], Nada Matta[1], Olivier Corby[2], William Léone[1], William Lesguillier[1]

1Tech-CICO, Université de Technologie de Troyes

12 rue Marie Curie, BP. 2060, 10010 Troyes Cedex, France

E-mail: {Chaker.Djaiz, Nada.Matta, william.leone, willaim.lesguiller}@utt.fr

2 INRIA-ACACIA

2004 route de Lucioles, BP. 2060, Sophia-Antipolis Cedex, France

Olivier.corby@inria.fr

Résumé. Le partage de connaissances demeurant un problème à traiter et particulièrement l'accès contextuel à l'information. Nous offrons un environnement de recherche d'information qui d'une part, peut inférer la connaissance en utilisant un réseau sémantique et d'autre part, il est basé sur le contexte et les objectifs d'une activité spécifique (la conception). L'environnement de recherche d'information utilisé est l'outil Corese qui permet notamment d'établir un lien entre le formalisme RDF et graphes conceptuels.

Mots-clés : Recherche d'informations, contexte de conception, Corese, graphe conceptuel, XML, RDF

1 Introduction

Le partage de la connaissance reste toujours un problème important à traiter dans les organisations. Des études en Web sémantique et les interfaces homme machine fournissent quelques techniques pour guider l'accès à l'information, ces techniques ne sont pas suffisantes pour donner un accès contextuel aux besoins de l'utilisateur. En fait, avec les techniques du Web sémantiques [Berners-Lee, et al, 01] l'accès à l'information par l'utilisateur est guidé par l'ontologie du domaine. Cependant ce type d'ontologie est établi comme une définition consensuelle de concepts du domaine qui peut être considéré comme une référence des concepts utilisés dans un domaine donné. Généralement l'utilisateur a sa propre représentation du domaine qui est plus au moins proche de l'ontologie définie. Notons également, que la structure de cette ontologie est généralement décrite par un ingénieur de connaissance qui introduit des biais d'interprétation. L'activité de l'utilisateur change à travers le temps ainsi que ses besoins d'accès à l'information. Plusieurs travaux en interface homme machine [Jing et autres, 02] étudient des techniques pour capturer l'activité de l'utilisateur afin d'offrir une recherche d'information contextuelle. Des études en ergonomie montrent que nous avons besoin de connaître les objectifs de l'activité et son environnement avant d'observer cette activité, ce qui n'est pas facile si la traçabilité des informations est réalisée d'une façon générique indépendamment de la tâche de l'utilisateur. Dans notre approche, nous considérons ces types de problèmes. Nous nous focalisons sur une activité de conception dans laquelle le processus de tâche peut fournir quelques informations sur les objectifs de l'activité de conception. Une mémoire de projets telle qu'elle est définie dans [Bekhti et al, 03] (considérant aussi bien le contexte du projet que sa logique de conception) peut être considérée comme un référent dans une organisation (Figure 1 .). Ce référent est structuré comme une ontologie donnant des liens aux documents et liant également le contexte aux concepts de résolution de problème. La recherche d'information peut être basée sur les relations entre ces concepts, qui peuvent offrir un accès contextuel à l'information. Nous exploitons les graphes conceptuels pour représenter le réseau sémantique et nous utilisons le moteur de recherche d'information Corese [Corby et al, 02] qui se base sur des fonctions de spécialisation, de généralisation et de projection pour offrir une recherche d'information en profondeur. Cet article décrit dans la seconde session, le besoin en terme de recherche d'information contextuelle dans le domaine de la conception. La quatrième session présente comment une mémoire de projet est représentée en graphes conceptuels et avec le formalisme RDF. La recherche d'information utilisé dans Corese (cinquième session) est basée sur la relation entre les graphes conceptuels et la représentation avec le formalisme RDF.

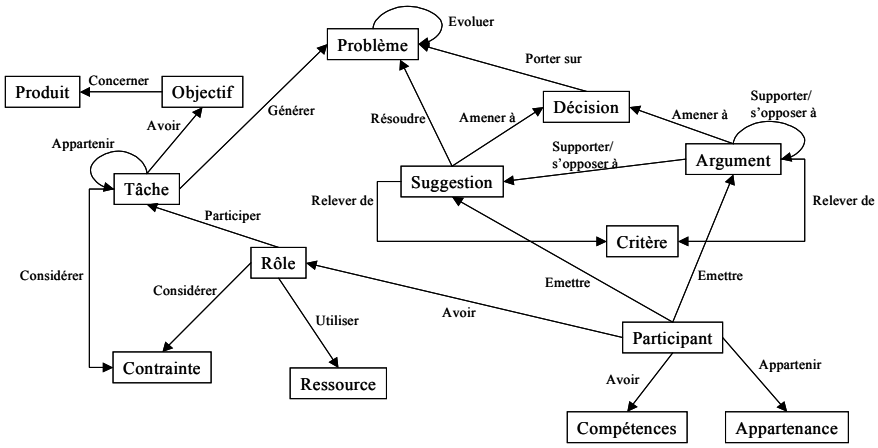


Figure 1 . Structure d'une mémoire de projet [Bekhti et al, 03]

2 Besoin d'information dans une activité de conception

Les concepteurs ont besoin d'apprendre des projets passés afin de traiter de nouveaux problèmes dans leurs activités. Leurs besoins sont principalement focalisés sur différents éléments des projets passés (comme les problèmes traités dans ces projets, les décisions prises, les contraintes, les arguments, les critères, etc.). La structure définie dans la figure 1 peut aider à la recherche de ce type d'information. Le processus de recherche d'information peut être guidée non seulement par des classifications des problèmes, des projets, etc. mais aussi par les critères de similitude entre les descriptions de projets. Ainsi, une recherche suivant une classification de concepts n'est pas suffisante. Les relations entre les éléments de projet doivent jouer un rôle important dans cette recherche d'information. Les concepteurs doivent savoir pourquoi une telle suggestion n'a pas été considérée, qui a décidé que, sous quelles contraintes, quels types de ressources ils avaient, avec quelles compétences, etc. Nous privilégions l'utilisation de la fonction de projection sur les graphes représentants la mémoire de projet plus que la spécialisation et la généralisation qui vont être utilisés plutôt en complément à la projection. Les résultats ne sont pas une liste de documents ou une partie de ces documents mais un réseau montrant d'une part, les relations entre certains éléments de projet et d'autre part, pointant vers les documents correspondants. Les concepteurs ont donc l'information contextuelle et peuvent l'utiliser efficacement dans leur activité de conception.

3 Recherche d'information basée sur les relations sémantiques

L'indexation des documents peut être simple, par méta-données et/ou par mots-clés (actuellement recommandés dans les documents HTML) ou plus complexe en utilisant des ontologies (assez soutenu par le consortium du Web sémantique). Nous avons choisi pour cela les graphes conceptuels pour représenter notre réseau sémantique. Les graphes conceptuels sont un formalisme qui offre une structure mettant en œuvre les relations sémantiques entre les concepts. Dans ce formalisme, il existe un certain nombre de fonctions d'inférence [Sowa, 84] ce sont pour la plupart des mécanismes d'aides à la recherche d'information. Nous notons en particulier des fonctions de projection, de jointure, de généralisation et de spécialisation. Ce formalisme a été employé et validé dans un certain nombre d'applications. Ainsi son efficacité est généralement approuvée. Autrement, la structure des graphes conceptuels (particulièrement structures et graphes) est proche de RDF (Resource Framework Description) et de RDFs (Resource Framework Description Schema). Elle peut être facilement traduite en RDF et en XML respectant les recommandations du Web sémantique [RDFS, 04]. Ce passage mutuel entre les graphes conceptuels, le RDF et le XML peut fournir un certain nombre de fonctions qui donnent une certaine puissance à la recherche d'information. Le Moteur de recherche Corese développé à l'INRIA [Corby et al, 04] est basé sur ce principe.

3.1 Corese: un moteur de recherche sémantique:

Le moteur de recherche Corese [Corby et al, 04] est dédié aux requêtes en Web sémantique dans lesquels des documents sont décrits avec des annotations RDF. Corese interprète les méta-données de RDF dans le modèle des graphes conceptuels afin d'exploiter les possibilités d'inférence de ce formalisme. Les RDFS [RDFS, 04] et les modèles des graphes conceptuels partagent beaucoup de dispositifs communs et des passages peuvent être facilement établi entre RDFS et les graphes conceptuels. La recherche d'information dans Corese est basée sur ce passage. En fait, le langage de requête de Corese est en RDF auxquels certaines conventions ont été ajoutées pour présenter des variables et des opérateurs. Une requête RDF est interprétée en une requête type graphe conceptuel. Elle est traitée par une projection conceptuelle. Corese permet ainsi

de traiter des requêtes de recherche d'information sur les données aussi bien que sur le schéma (la structure de la mémoire de projet).

3.2 Représentation d'une mémoire de projet:

La connaissance est directement extraite à partir des projets de conception est structurée dans des fichiers XML. Un secrétaire de projet peut conserver les connaissances liées au contexte et les décisions prises lors de réunions de projet en utilisant un outil (DYPKM) [Bekhti et al, 03] qui produit directement un fichier XML. Un fichier XSL est défini afin de permettre la sérialisation en fichier RDF correspondant. Ces deux fichiers sont alors intégrés dans la base de connaissance de Corese afin d'offrir un moteur de recherche efficace.

```
2 réponses
Critère 2

1 Critère http://www.utt.fr/Critere/cr123.html
  Argumenthttp://www.utt.fr/Discussion/Argum/ar342.html
      Relatedto http://www.utt.fr/Critere/cr123.html
  Argument http://www.utt.fr/Discussion/Argum/ar10.html
      Relatedto http://www.utt.fr/Critere/cr123.html
  Suggestion http://www.utt.fr/Discussion/Sugg/sug11.html
      Relatedto http://www.utt.fr/Critere/cr123.html
  Problem http://www.utt.fr/Discussion/Prob/pb2.html
      Appelation string Definition of the ERP focus
2 Critère http://www.utt.fr/Critere/cr13.html
  Argument http://www.utt.fr/Discussion/Argum/ar32.html
      Relatedto http://www.utt.fr/Critere/cr13.html
  Suggestion http://www.utt.fr/Discussion/Sugg/sug11.html
      Relatedto http://www.utt.fr/Critere/cr13.html
  Problem http://www.utt.fr/Discussion/Prob/pb2.html
      Appelation string Definition of the ERP focus
```

Figure 2 . Résultat de la recherche exploratoire

4 Recherche d'infromation avec Corese

La structure de la mémoire de projet (Figure 1 .) est mise en application dans un RDF schéma. Le moteur d'inférence de Corese utilise le modèle RDFs de la mémoire de projet et l'explore afin d'effectuer une recherche d'information à partir des fichiers RDF qui contiennent les informations relatives aux projets. Nous montrons dans les deux exemples suivants ce type de recherche d'information. Le premier exemple montre une recherche d'exploration effectuée par Corese. Le deuxième exemple, présente comment le moteur d'inférence peut être utilisé pour effectuer une recherche en profondeur et construire des liens entre des concepts qui ne sont pas directement liés.

4.1 Exemple d'une recherche d'exploration

Quand un concepteur traite un problème, il doit par exemple, savoir si ce problème a été résolu dans des projets passés et comment ? Sa demande peut être : "quels sont les arguments et les suggestions avancés pour résoudre un problème donné ainsi que les critères qui caractérisent cette argumentation ? " Ce besoin est traduit en une requête RDF. S'accordant avec notre représentation de RDFs dés lors une projection en graphe conceptuel est effectué grâce au moteur de projection de Corese: RDF - > GC - > RDF. Cette requête RDF est ainsi interprétée par le moteur Corese comme une requête graphe conceptuel. Le résultat est obtenu en utilisant la fonction de généralisation et en exploitant les fichiers RDF. L'affichage des résultats en format RDF est rendu plus facile grâce à l'utilisation des données de formatage du fichier XSLT (Figure 2 .). En cliquant sur les URLs, nous pouvons avoir comme résultat (exemple extrait d'une mémoire de projet de définition de principes d'évaluation de risques professionnels à l'INRS [Bekhti, 03]):

PB2: ERP focus
  Critère 123: *Terminologie*
    Argument 342: *La non clarté du périmètre d'évaluation*
    Argument 10: *Le périmètre d'évaluation dans le paragraphe 3 n'est pas en relation dans le texte*
    Suggestion 11: *utilisation du "Champ d'intervention" au lieu du "paramètre d'évaluation"*
    Critère 13: Exhaustivité
    Argument 32: *Le chef de compagnie doit informer les employés au sujet de l'état d'évaluation*
    Suggestion 11: *Accentuer le rôle du chef de compagnie dans processus d'évaluation*

4.2 Exemple d'une recherche en profondeur

Un concepteur veut par exemple savoir " quelles sont les compétences qui ont contribués à résoudre un problème passé et sous quels critères ? " La particularité de cette requête est que ces concepts (problèmes et compétences), ne sont pas directement reliés dans la structure de la mémoire de projet. En fait, il y a une relation entre un problème et les suggestions et les arguments. Les arguments et les suggestions sont liés aux critères qui les caractérisent. Il y a également un lien entre les arguments, les suggestions et les participants qui les ont énoncés. Pour chaque participant, ses compétences sont définies (Figure 1 .). Le moteur de recherche Corese effectue une projection approximative qui est basée sur une relation de jointure afin d'établir une méta-relation entre les concepts donnés. Ainsi, le moteur Corese infère le document détaché ou bien lié par transitivité. Les résultats obtenus sont toujours des fichiers RDF (Figure 3 .).

1 réponse
1 Criteria, Membership, Participant http://www.utt.fr/Critere/cr123.html
Appelation string
Competences http://www.utt.fr/Competences/Comp/cm13.html
Problem http://www.utt.fr/Discussion/Prob/pb2.html
Appelation string Definition of the ERP focus

Les résultats obtenus:
PB2: ERP focus
Critère 123: *Terminologie*
Compétence 13: *Consultant*

Figure 3 . Résultat de la recherche inférentielle

5 Conclusion

Les concepteurs ont besoin d'apprendre des projets passés afin de traiter de nouveaux problèmes de conception. Une mémoire de projet représentant des relations entre problème, suggestions, arguments, critères, participants de projet, tâches, etc. fournit une traçabilité utile de la connaissance produit dans la réalisation d'un projet. Cette mémoire montre les relations qui existent entre le contexte du projet (organisation, environnement) et la résolution des problèmes. La recherche d'information peut exploiter ces types de relations existantes afin de permettre une recherche contextuelle d'information. Nous montrons dans cet article comment un réseau sémantique (représenté sous forme de graphes conceptuels) peut fournir une recherche d'information contextuelle. Pour cela, le moteur de recherche Corese est utilisé, il permettra d'établir le passage entre le formalisme RDF et les graphes conceptuels pour offrir une recherche d'informations en profondeur. Le processus de recherche est non seulement basé sur une spécialisation mais également sur des inférences en utilisant des relations d'influence entre les concepts. Les résultats aident le concepteur à comprendre des situations de résolution de problèmes passées et donc d'apprendre de ces situations. Les exemples présentés dans cet article sont extraits d'un projet de définition de principes d'évaluation professionnelle à l'Institut National de Recherche en Sécurité. Notre approche doit être validé sur d'autres projets de conception. Nous visons également à définir une interface graphique pour présenter les résultats qui illustrent les relations entre les éléments de projet de conception. En conclusion, l'environnement que nous avons défini peut également être utilisé pour établir des regroupements des éléments de projet afin d'établir des classifications des tâches, des problèmes, des arguments, etc. produites dans une entreprise. Ces classifications peuvent montrer l'évolution des stratégies de conception dans l'entreprise.

6 Références

[Bekhti et al, 03] Bekhti S., Matta N., Project memory: An approach of modelling and reusing the context and the de design rationale, Proceedings of IJCAI'03 (International joint of conferences of Artificial Intelligence) Workshop on knowledge management and organisational memory, Accapulco, 2003.

[Berners-Lee et al, 01] Berners-Lee T., Hendler J., Lassila O., the Semantic Web, In: Scientific American, May 2001.

[Corby et al., 04] O. Corby, R. Dieng, C. Faron-Zucker Querying the Semantic Web with Corese Search Engine, Prestigious Applications of Intelligent Systems PAIS, ECAI, Valencia, 2000.

[Jing et al, 02] Jing Y., Taylor N., Brown K., An intelligent Inference Approach to user Interaction Modelling in a generic Agent Based Interface System, In proceedings of ECAI2002 (European Conference on Artificial Intelligence), July, Lyon, 2002.

[RDFS, 04] RDF Vocabulary Description Language 1.0: RDF Schema W3C Recommendation, February 10, 2004 Dan Brickley, R.V. Guha, eds.

[Sowa, 84] Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading,MA.

MIEL++ : un entrepôt intégrant des données floues exprimées en graphes conceptuels, bases de données relationnelles et XML

Patrice Buche[1], Juliette Dibie-Barthélemy[2], David Doussot[2], Ollivier Haemmerlé[2,3], Gaëlle Hignette[2], Rallou Thomopoulos[4]

[1]INRA - Mét@risk, 16 rue Claude Bernard, F-75231 Paris Cedex 05, France
[2]UMR INA P-G/INRA MIA, 16 rue Claude Bernard, F-75231 Paris Cedex 05, France
[3]LRI (UMR CNRS 8623 - Université Paris-Sud) / INRIA (Futurs), Bâtiment 490, F-91405 Orsay Cedex, France
[4]INRA - UMR IATE - bat. 31, 2 Place Viala 34060 Montpellier Cedex 1

Mots-clés Intégration de données, graphes conceptuels, sous-ensembles flous, validation.

1 Introduction, domaine d'application

Nous présentons dans ce résumé les travaux menés par l'équipe de recherche en informatique de l'Institut National Agronomique Paris-Grignon au cours des 5 dernières années. Ces travaux ont trouvé leur champ d'application dans le domaine de la prévention du risque microbiologique dans les aliments. La sécurité des aliments est une préoccupation grandissante, les crises passées ayant la plupart du temps eu des répercussions catastrophiques d'un point de vue sanitaire mais également économique. L'évaluation du risque microbiologique ne peut se faire qu'à partir d'informations sur le comportement des germes pathogènes lors d'épisodes de contamination. Les équipes travaillant en microbiologie prévisionnelle ont été amenées à construire des bases de données permettant une capitalisation de ces connaissances.

Notre équipe travaille sur la construction d'un entrepôt de données, qui a vocation à contenir des données extraites de la littérature scientifique en microbiologie ou des données fournies par des partenaires industriels. Les données à stocker présentent un certain nombre de caractéristiques : (i) les données sont hétérogènes ; (ii) les données peuvent être imprécises ; (iii) l'entrepôt de données est incomplet par nature ; (iv) les sources de données ne sont pas de qualité égale.

Pour prendre en compte ces caractéristiques, nous travaillons sur les thèmes suivants :

- intégration de données contenues dans des bases fondées sur des formalismes différents ;
- représentation de données imprécises par le biais de sous-ensembles flous ;
- mécanismes d'interrogation élargie des bases palliant l'incomplétude ;
- enrichissement automatique de l'entrepôt de données par des données automatiquement extraites du Web ;
- validation de données.

2 Intégration de données, vue d'ensemble du système MIEL++

Nous avons choisi de traiter l'hétérogénéité de nos données en construisant un entrepôt de données intégrant trois sources de données distinctes par le biais d'une même interface d'interrogation (Cf. Fig. 1) : une base de données relationnelle, une base de graphes conceptuels, et une base de données XML.

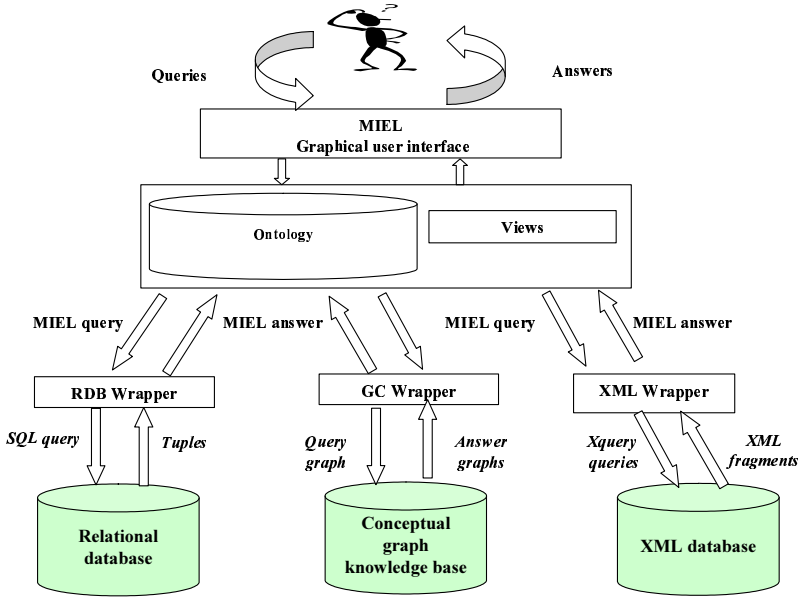


FIG. 1 – Le schéma global de l'entrepôt de données MIEL++

3 Données imprécises dans le modèle des graphes conceptuels

Les données expérimentales en microbiologie prévisionnelle présentent certaines formes d'imprécision : (i) variabilité liée à la complexité intrinsèque des processus biologiques ; (ii) imprécision des capteurs ; (iii) imprécision de l'expression des résultats dans les sources de données (publications scientifiques).

Nous avons choisi de représenter les données imprécises à l'aide de distributions de possibilités. Nous utilisons pour cela le modèle des sous-ensembles flous proposé par Zadeh. Nous avons étendu le modèle des graphes conceptuels autorisant une représentation de sous-ensembles flous dans les sommets concepts. Un sommet concept peut être porteur d'un type de concept flou ou d'un marqueur flou. L'opération de projection des graphes conceptuels reste définie comme un homomorphisme de graphes autorisant la spécialisation des étiquettes des sommets, en utilisant la nouvelle définition de la spécialisation des étiquettes des sommets concepts. L'extension du modèle à la représentation de sous-ensembles flous est utilisée pour représenter des données imprécises, mais également pour permettre à l'utilisateur d'exprimer des préférences dans les critères de sélection de ses requêtes à l'entrepôt de données.

4 Ouverture de l'entrepôt de données sur le Web

Le projet RNTL e.dot[1] a pour objectif la construction d'un entrepôt de données stockées au format XML alimenté automatiquement à partir du Web. Le domaine d'application choisi est également la prévention du risque microbiologique dans les aliments. Dans le cadre de ce projet, nous travaillons sur : (i) l'extraction d'informations à partir de publications scientifiques contenant des tableaux de données. Ces documents *pdf* sont transformés en documents XML étiquetés sémantiquement à partir de l'ontologie du système afin de permettre leur interrogation ; (ii) l'interrogation de ces documents étiquetés sémantiquement qui ne répondent pas nécessairement à un schéma prédéfini. Nous proposons une interrogation flexible de ces documents ; (iii) l'intégration des données aux bases de l'entrepôt.

Ces différentes parties de notre travail ont pour point commun d'être fondées sur l'utilisation d'une ontologie qui est proche de celle que l'on utilise dans le modèle des graphes conceptuels (un ensemble de types de concepts correspondant à une taxonomie de termes, un ensemble de relations, et des contraintes d'utilisation de ces concepts et relations).

5 Validation de graphes conceptuels

Ce travail a porté sur la validation syntaxique d'une base de connaissances exprimée en termes de graphes conceptuels. Nous avons proposé un ensemble de propriétés et avons montré que leur satisfaction globale par une base de connaissances garantissait la validité syntaxique de cette base. Nous avons également présenté des règles de réparation associées à chacune des propriétés.

Nous avons ensuite étudié la validation sémantique d'une base de connaissances relativement à des contraintes, que nous considérons comme des connaissances expertes extérieures à la base, uniquement fournies à des fins de validation et que nous supposons fiables. Ces contraintes sont classées en deux catégories : (i) des contraintes négatives qui permettent, dans une situation donnée, de s'assurer qu'une certaine conclusion ne peut pas être déduite. Cette forme de validation peut être comparée à l'étude de la cohérence de la base ; (ii) des contraintes positives qui permettent, dans une situation donnée, de s'assurer qu'une certaine conclusion peut être déduite. Cette forme de validation peut être comparée à l'étude de la complétude de la base.

[1] Entrepôt de Données Ouvert sur la Toile, mené avec l'équipe de Marie-Christine Rousset (LRI), l'équipe de Serge Abiteboul (INRIA) et la société Xyleme