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

Résumé

Corese est un moteur de recherche sémantique pour le langage RDF (Resource Description Framework), langage standard du web sémantique proposé par le W3C. Ce moteur est dédié à des applications de web sémantique communautaire ou d'entreprise. Le moteur Corese permet la recherche d'information basée sur des modélisations de connaissances, des modèles conceptuels ou des référentiels métier et organisationnel. Corese est une implémentation de RDF/S sous forme de graphes conceptuels.

I. Introduction

Le moteur Corese[1] charge des ontologies au format RDF Schema, des meta donnés décrivant des ressources sous forme d'énoncés RDF. Il comprend également un langage de règles adapté à RDF. Corese construit une représentation interne de ces informations sous forme de graphes conceptuels. Il réalise des inférences grâce aux règles et répond aux requêtes de recherche d'information dans la base de meta données RDF. Les résultats de requêtes sont retournés en RDF/XML et peuvent être traités par des feuilles de style XSLT pour engendrer des formats de présentation lisibles par les utilisateurs (HTML, SVG, etc.). Ce papier montre l'intérêt des graphes conceptuels pour RDF/S et le web sémantique.

II. Mapping RDF CG

RDF/S a de nombreux points communs avec les graphes conceptuels. Nos travaux antérieurs 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 ensuite proposé une correspondance entre les GC et RDF/S [2]. En 2001, Tim Berners-Lee publiait une note soulignant la similarité entre RDF et les graphes conceptuels [1].

Les deux modèles distinguent connaissance ontologique et assertionnelle. Cette dernière est positive, conjonctive et existentielle et est représentée par des graphes orientés, étiquetés bipartis. Dans Corese, un graphe RDF représentant une annotation ou une requête est traduit en un GC. En ce qui concerne la connaissance ontologique, la hiérarchie de classes (resp. relations) 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. Ce traitement similaire des propriétés/relations rend le mapping pertinent, contrairement à celui des langages à objets où les propriétés appartiennent aux classes.

Il y a toutefois des différences entre RDF et les GC. RDF permet d'associer plusieurs classes à une ressource. Ceci est traduit en GC en engendrant le type correspondant au plus général sous-type commun des types correspondant aux classes. De la même manière, la signature d'une propriété peut comporter plusieurs classes en domain et en range. On engendre également le plus grand sous-type commun des types correspondant aux classes de la signature, pour le domain et pour le range.

On peut ainsi construire un moteur de recherche RDF en GC en compilant la hiérarchie des types et en associant un type compilé aux ressources RDF et en utilisant l'opération de projection des GC pour la recherche. Il faut noter que tout schema RDFS, en tant que document RDF, est également chargé sous forme de graphe conceptuel et peut être interrogé par le moteur de recherche. L'ontologie est ainsi représentée sous forme d'un graphe.

III. Moteur de recherche

Dans cette section nous présentons les principales fonctions du moteur de recherche Corese.

A. Langage de requête

Interroger une base de graphes peut se faire 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. Le OU booléen est traduit sous forme normale conjonctive. 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 triplet de la forme `ressource propriété valeur` ou une expression évaluable. Le noyau du langage est compatible avec la proprosition SPARQL du W3C.

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

Le moteur Corese exploite la projection pour calculer les réponses aux requêtes. Il propose quelques extensions de la projection.

B. Variable de propriété

Il est possible d'utiliser des variables pour désigner des propriétés dans une requête. La ligne (4) 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

Le langage de requête et le moteur permettent de rechercher des chemins, orientés ou non orientés, de longueur supérieure à un, 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 retourne 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: $\forall (t_1, t_2) \in H^2$,
$D_H(t_1, t_2) = min_t(l_H(\langle t_1, t\rangle) + l_H(\langle t_2, t\rangle)) =$

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

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

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 au moteur 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 (grouper par auteur et compter le nombre de documents). Enfin le moteur peut trier les résultats selon certains critères (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 règle

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>
```

V. Architecture

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

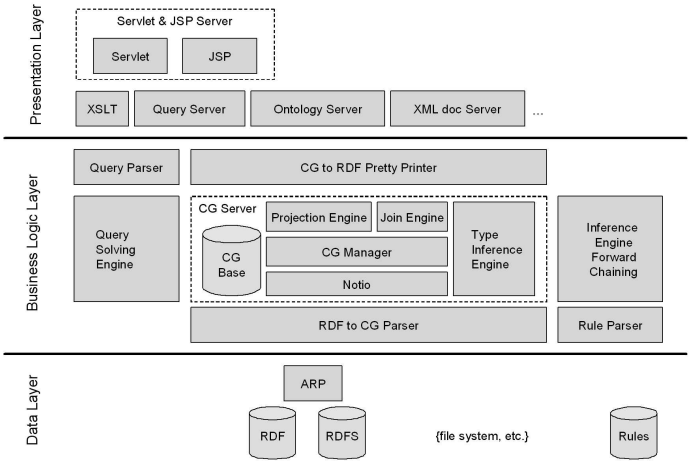

Fig. 1. Corese : architecture 3-tier

- un serveur de graphes conceptuels utilisant les structures de données de Notio [5],
- un traducteurs RDF/CG et CG/RDF
- un parser des langage de requêtes et de règles
- 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'extractions 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 dont voici quelques unes :

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 soutient 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 Artifi cial 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 Artifi cial 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`