



# Ontology-based Approximate Query Processing for Searching the Semantic Web with Corese

Olivier Corby, Rose Dieng-Kuntz, Catherine Faron Zucker, Fabien Gandon

## ► To cite this version:

Olivier Corby, Rose Dieng-Kuntz, Catherine Faron Zucker, Fabien Gandon. Ontology-based Approximate Query Processing for Searching the Semantic Web with Corese. [Research Report] RR-5621, INRIA. 2006, pp.36. inria-00070387

**HAL Id: inria-00070387**

**<https://inria.hal.science/inria-00070387>**

Submitted on 19 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# *Ontology-based Approximate Query Processing for Searching the Semantic Web with Corese*

Olivier Corby — Rose Dieng-Kuntz — Catherine Faron-Zucker — Fabien Gandon

**N° 5621**

July 2005

Thème SYM



*rapport  
de recherche*



## Ontology-based Approximate Query Processing for Searching the Semantic Web with Corese

Olivier Corby<sup>\*</sup>, Rose Dieng-Kuntz<sup>\*</sup>, Catherine Faron-Zucker<sup>†</sup>, Fabien Gandon<sup>\*</sup>

Thème SYM — Systèmes symboliques  
Projet ACACIA

Rapport de recherche n° 5621 — July 2005 — 36 pages

**Abstract:** The semantic web relies on ontologies representing domains through their main concepts and the relations between them. Such a domain knowledge is the keystone to represent the semantic contents of web resources and services in metadata associated to them. These metadata then enable us to search for information based on the semantics of web resources rather than their syntactic forms. However, in the context of the semantic web there are many possibilities of executing queries that would not retrieve any resource. The viewpoints of the designers of ontologies, of the designers of annotations and of the users performing a Web search may not completely match. The user may not completely share or understand the viewpoints of the designers and this mismatch may lead to missed answers. Approximate query processing is then of prime importance for efficiently searching the Semantic Web. In this paper we present the Corese ontology-based search engine we have developed to handle RDF(S) and OWL Lite metadata. We present its theoretical foundation, its query language, and we stress its ability to process approximate queries.

**Key-words:** Semantic Web, Web Search, RDF(S), OWL Lite, Ontology, Approximate Queries

<sup>\*</sup> INRIA National Institute of Research in Computer Science of Sophia Antipolis, France

<sup>†</sup> I3S Institute of Research in Computer Science of the University of Nice Sophia Antipolis, France

## Approximation de requête basée sur une ontologie, pour la recherche sur le Web sémantique avec Corese

**Résumé :** Le web sémantique repose sur des ontologies pour représenter un domaine à travers ses principaux concepts et leurs inter-relations. Une telle connaissance du domaine est la clé pour représenter le contenu sémantique des ressources et services web dans des métadonnées associées. Ces métadonnées permettent une recherche d'information basée sur la sémantique des ressources web plutôt que sur leur forme syntaxique. Cependant, dans le contexte du web sémantique, il y a de multiples occasions d'exprimer des requêtes pour lesquelles aucune ressource n'est trouvée. Les points de vue de l'auteur des ontologies, de l'auteur des annotations et des utilisateurs effectuant une recherche sur le web peuvent ne pas coïncider complètement. L'utilisateur peut ne pas complètement partager ou comprendre le point de vue des auteurs, ce qui peut conduire à des réponses perdues. Le traitement de requêtes approchées devient de ce fait un enjeu pour rechercher efficacement sur le web. Dans cet article, nous présentons le moteur de recherche sémantique Corese, que nous avons développé pour raisonner sur des métadonnées RDF(S) et OWL Lite. Nous présentons ses fondements théoriques, son langage de requête et nous mettons l'accent sur sa capacité à exécuter des recherches approchées.

**Mots-clés :** Web sémantique, Recherche sur le Web, RDF(S), OWL Lite, Ontologies, Requêtes approchées

## Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>A Concrete Example</b>	<b>7</b>
<b>3</b>	<b>Ontology based Web Search</b>	<b>10</b>
3.1	A Logic based Approach . . . . .	10
3.2	Theoretical Foundations of Corese . . . . .	11
3.3	Corese Ontology Representation Language . . . . .	13
3.4	Corese Query Language . . . . .	14
<b>4</b>	<b>Approximate Semantic Web search</b>	<b>16</b>
4.1	Ontological Approximation . . . . .	16
4.1.1	Ontological Distance . . . . .	16
4.1.2	Contextual Closeness . . . . .	19
4.1.3	Approximate Projection . . . . .	20
4.2	Structural Approximation . . . . .	22
<b>5</b>	<b>Software, Applications and Evaluation</b>	<b>24</b>
5.1	Architecture . . . . .	24
5.1.1	Presentation Layer . . . . .	24
5.1.2	Application Layer . . . . .	24
5.1.3	Persistent Layer . . . . .	25
5.2	Real World Applications . . . . .	25
5.2.1	SAMOVAR . . . . .	25
5.2.2	CoMMA IST project . . . . .	26
5.2.3	ESCRIRE . . . . .	26
5.2.4	KMP . . . . .	26
5.2.5	Ligne-de-Vie . . . . .	26
5.2.6	MEAT . . . . .	26
5.3	Evaluation . . . . .	26
5.3.1	Corese Performance . . . . .	27
5.3.2	Scenario-based evaluation . . . . .	27
5.4	Related Work . . . . .	28
5.5	Query Languages for RDF . . . . .	28
5.6	Ontology-Based Web Search Applications . . . . .	29
5.7	Ontology Alignment or Versioning . . . . .	30
5.7.1	Other Domain-Specific Web Search Approaches . . . . .	30

## List of Figures

1	Conceptual clustering of the competences of the Sophia Telecom Valley . . .	8
2	Corese answer presentation . . . . .	9
3	Corese general principle . . . . .	10
4	Corese 3-tier architecture . . . . .	25

## 1 Introduction

The present Web comprises a huge amount of heterogeneous data (structured data, semi-structured data, textual data, multimedia data), dedicated to human users. The Semantic Web [4] aims at representing the semantic contents and characteristics of Web resources in formalisms understandable by automated tools as well as by humans. It can be seen as a semantic network of web resources: web resources associated with semantic descriptions and linked by semantic relations. It relies on rich metadata, also called semantic annotations, offering explicit semantic descriptions of Web resources. These semantic annotations are built on ontologies, representing domains through their concepts and the semantic relations between these concepts. Ontologies are the foundations of the so called Semantic Web and the keystone of the automation of tasks on the web: searching, merging, sharing, maintaining, customizing, monitoring, etc.

In this paper, we focus on searching the Semantic Web. This specific kind of Web search is needed in web applications such as Digital Libraries, e-Learning, Web Intelligence, Expertise Networks, Corporate Webs and Intranets used in particular for Knowledge Management, etc. This last application calls for a domain specific Web search: it is characterized by the delimitation of 'corporate' webs for which ontologies can be designed to represent the application domains. Publishing languages like HTML enable us to retrieve web documents based on the structure of their presentation and their textual contents; structuring languages like XML or SGML enable us to access web resources based on their data structure. Semantic annotations, by providing richer descriptions of web resources and by representing their semantic contents based on ontologies, improve the Web search and enable us to access web resources based on their semantic descriptions.

Searching the Semantic Web can be addressed according to three different points of view corresponding to three different classes of actors of the Semantic Web: developers of ontologies, annotators of Web resources and end-users of the Semantic Web.

(1) Developers of ontologies are focusing on the representation of domain knowledge but when building ontologies they are guided by the projected uses of these ontologies, one of them being to support the Semantic Web search. WebODE [2], KAON [40] [41] and Protege-2000 [45] are the most recent and complete workbenches dedicated to the design, development and management of ontologies for the Semantic Web; the OWL API [3] is a high-level programmatic interface for accessing and manipulating OWL ontologies. OILed, OntoEdit, Ontolingua, WebOnto were precursor tools for ontology design. Upstream, to support the ontology design process, languages are specified according to the expressivity required for the representation of ontology features. A roadmap of the ontology languages for the semantic Web is given in [26]. OWL [47] and RDFS [58] are the two emerging web languages recommended by the W3C. Both are built upon RDF [57]; OWL is descending from DAML and OIL [39] and is built upon RDFS. In OWL, the desired expressivity is obtained by choosing between OWL Lite, OWL DL and OWL full.

(2) Once ontologies are designed and represented in a dedicated language, annotators of web resources create semantic annotations based on these ontologies. This annotation process is also guided by the same goal of supporting the Semantic Web search. MnM



[61] and Ont-O-Mat [23] are the most succeeding tools for ontology-driven automatic and semi-automatic annotation of Web pages; they both are based on the Amilcare information extraction tool. KIM [33] is another tool for automatic semantic annotation of Web pages based on light-weight upper level ontologies.

(3) Finally, search engines and other applications process ontology-based queries and solve them against bases of annotations and associated ontologies. They rely on inferences, and operators guided by the goal of supporting search (improving recall, decreasing noise, etc.). In this paper, we focus on this query processing point of view and address the problem of a dedicated ontology based query language; section E will survey related works.

In this paper, we focus on the end-user point of view and address the problem of a dedicated ontology-based query language. We show how ontologies ensure an efficient retrieval of Web resources by enabling inferences based on domain knowledge and we emphasize the prime interest of semantic approximations for efficiently searching the Semantic Web. The vision of the Semantic Web implicitly relies on the (strong) hypothesis that an ontology designed to describe a domain is usable both to annotate web resources and to retrieve them. Reality is more contrasted.

Usually, an ontology is built by specialists of the domain, not by specialists of the Web search task in this domain, *i.e.* the users. The user may not completely share or understand the viewpoints of the ontology designers. There may be some mismatch between the need of a clean reusable formal ontology and an effective guideline for Semantic Web search. Some experiments of the Corese semantic search engine we have developed give us good examples of misunderstanding, misuse by the user of concepts stated by the ontologist: in the CoMMA project the *Commerce* concept could be used instead of the *Business* one, *TechnicalReport* instead of *ResearchReport*. Users may not use the *right* concepts - from the viewpoint of the ontologist - when writing a query, and this mismatch may lead to missed answers. Moreover, a user asking for a *person* working on a *subject* may appreciate, instead of a failure, the retrieval of a *research group* working on that subject, even if a research group is not exactly a person. Lastly, a user may search for some related resources without knowing how their possibly complex relation is stated in the annotations. For instance, a user may search for organizations related to human sciences while ignoring the diversity of relations used to express this relationship in the annotations. All these examples illustrate the prime interest of semantic approximations for efficiently searching the Semantic Web.

In the next section, we present a concrete scenario of semantic search with Corese to highlight its strong point: ontology-based search, approximate search. In section 3 we present the Corese Semantic Web search engine we have developed, its theoretical foundations, its ontology representation language and its query language dedicated to the retrieval of web resources annotated in RDF(S). In section 4 we focus on the approximate query processing provided by Corese; we show how the Corese query language enables both ontological and structural approximations. Finally, we present the software architecture and some concrete experiments which demonstrated its interest in several real world projects.

## 2 A Concrete Example

KMP<sup>1</sup> is an on-going project for which we have built upon Corese a knowledge management platform for cartography of skills in telecommunications for Sophia Antipolis firms members of the Telecom Valley association.

The goal of KMP is to build an innovative solution of knowledge management shared within a community, in order to foster synergies and partnerships by providing a dynamic map of the competences of the different stakeholders. The solution relies on the specification, design, building and evaluation of an online customizable service. This service is becoming the main component of a portal for the community of the industries, the academic institutes and the institutional organizations involved in the Telecom Valley of Sophia Antipolis. The project is a real-world experiment and the steering committee is composed of eleven pilot companies including: Amadeus, Philips Semiconductors, France Telecom R&D, Hewlett Packard, IBM, Atos Origin, Transiciel, Elan IT, Qwam System, Cross Systems.

In KMP, an algorithm provides clustering views to analyze the competences present in the Telecom Valley. The screenshot in figure 1 shows one of these views called the "Clusters". It presents a distribution of grapes corresponding to resources involved in the competence and each grape contains bubbles representing actions (e.g. *"produce"*, *"design"*) involved in the competence. The SVG view is dynamically generated from the integrated RDF/S data collected. It provides very powerful representation to analyze the diversity of the Telecom Valley. The grouping of competences relies on our ontology-based distance to evaluate the conceptual similarities between the competences.

The following concrete motivational scenario will clearly show the strong points of Corese. Let us consider a query submitted to Corese and asking for persons both expert in Java programming and interested in XML. Without rules and with exact projection, there is no answer to the query. However, after applying the set of forward chaining rules of the ontology on the annotation base, Corese retrieves one exact answer, thanks to the following rule stating that a person author of a Thesis on a given subject is an expert of this subject.

```
<cos:rule>
  <cos:if>
    ?p rdf:type s:Person
    ?p s:hasCreated ?doc
    ?doc rdf:type s:Thesis
    ?doc s:concern ?s
  </cos:if>
  <cos:then>
    ?p s:isExpertIn ?s
  </cos:then>
</cos:rule>
```

---

<sup>1</sup>[http://www.telecom.gouv.fr/rnrt/projets/res\\_02\\_88.htm](http://www.telecom.gouv.fr/rnrt/projets/res_02_88.htm)

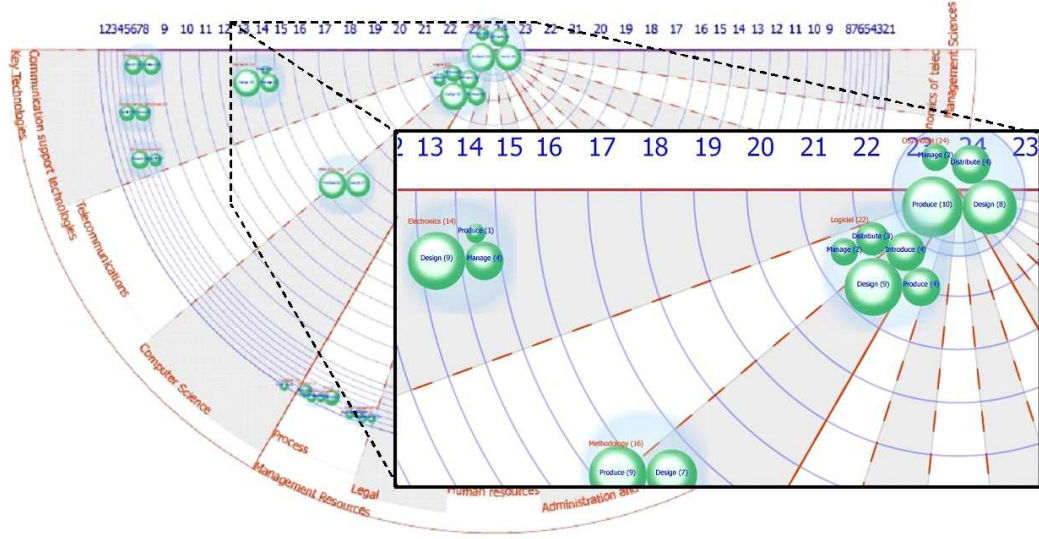


Figure 1: Conceptual clustering of the competences of the Sophia Telecom Valley

The exact answer which is retrieved is a researcher expert in EJB programming and skilled in XML. In the O'CoMMA ontology, `EJBProgramming` is a subclass of `JavaProgramming` and `IsSkilledIn` is a subproperty of `IsInterestedIn`. Therefore there exists a projection of the above query into the following annotation on Yvonne Duchard having a thesis on EJB programming.

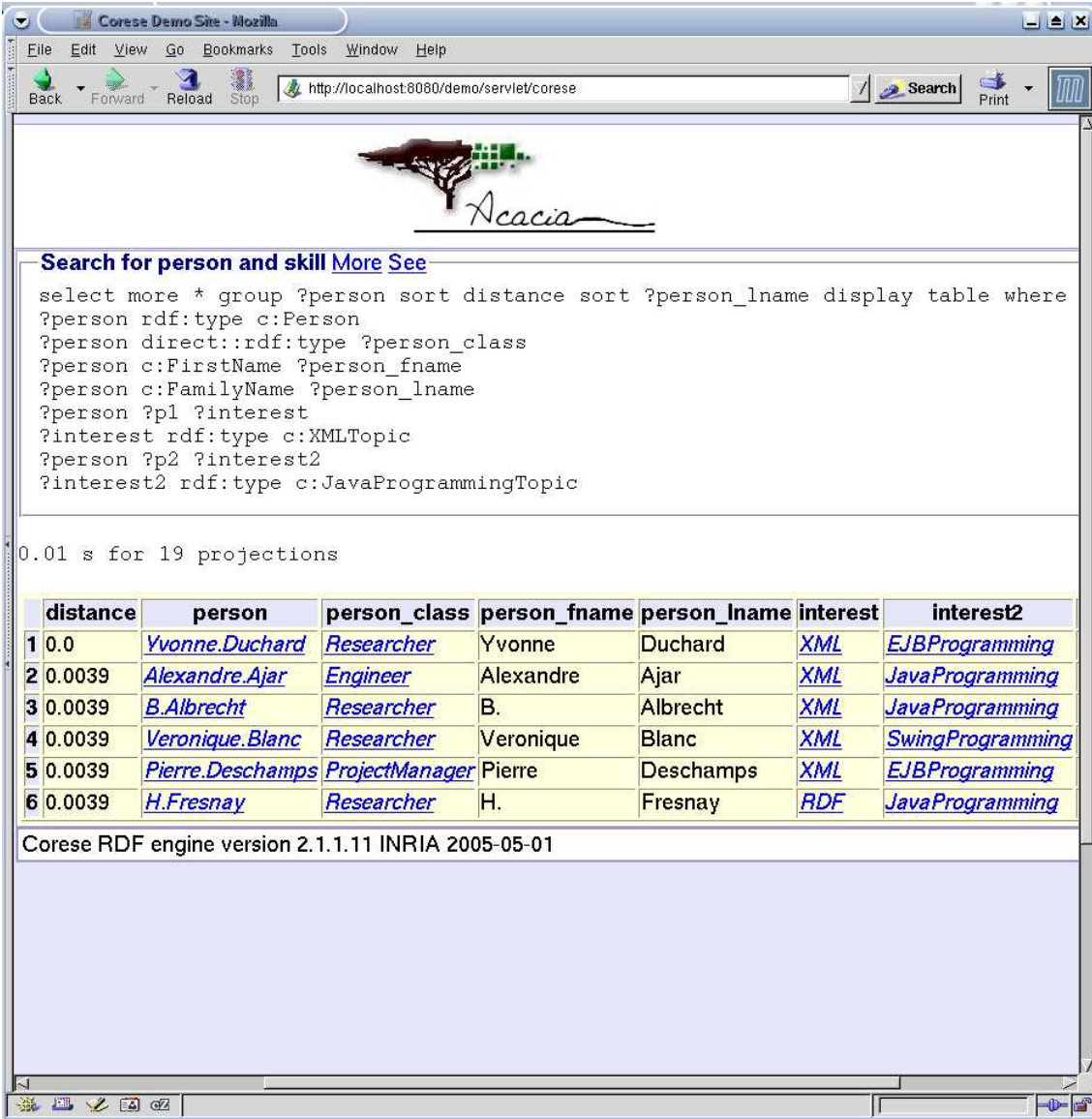
```

Researcher s:Yvonne.Duchard
  IsExpertIn
    EJBProgrammingTopic
      s:EJBProgrammingTopic
  IsSkilledIn
    XMLTopic s:XMLTopic

```

In approximate mode, Corese retrieves eight annotations and Figure 2 presents a screen shot of the Corese interface displaying the retrieved answers.


The first answer shows how Corese supports serendipity. It is the exact answer described above which is there extended with an interesting approximation: in addition to XML, Yvonne Duchard is also interested in (aware of) Wap which is close enough to XML: these topics both are subclasses of Telecommunication. The seven other answers approximately match the query. For instance, the second answer is an engineer skilled in both XML and Java programming and the third answer is a project manager skilled in both XML and EJB programming. These two annotations have the same similarity to the query: in both cases, the `isExpertIn` property is approximated by `isSkilledIn` which is its ancestor. As



Corese Demo Site - Mozilla

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop http://localhost:8080/demo/servlet/corese Search Print

 Acacia

**Search for person and skill [More See](#)**

```
select more * group ?person sort distance sort ?person_lname display table where
?person rdf:type c:Person
?person direct::rdf:type ?person_class
?person c:FirstName ?person_fname
?person c:FamilyName ?person_lname
?person ?p1 ?interest
?interest rdf:type c:XMLTopic
?person ?p2 ?interest2
?interest2 rdf:type c:JavaProgrammingTopic
```

0.01 s for 19 projections

	distance	person	person_class	person_fname	person_lname	interest	interest2
1	0.0	<a href="#">Yvonne.Duchard</a>	<a href="#">Researcher</a>	Yvonne	Duchard	<a href="#">XML</a>	<a href="#">EJBProgramming</a>
2	0.0039	<a href="#">Alexandre.Ajar</a>	<a href="#">Engineer</a>	Alexandre	Ajar	<a href="#">XML</a>	<a href="#">JavaProgramming</a>
3	0.0039	<a href="#">B.Albrecht</a>	<a href="#">Researcher</a>	B.	Albrecht	<a href="#">XML</a>	<a href="#">JavaProgramming</a>
4	0.0039	<a href="#">Veronique.Blanc</a>	<a href="#">Researcher</a>	Veronique	Blanc	<a href="#">XML</a>	<a href="#">SwingProgramming</a>
5	0.0039	<a href="#">Pierre.Deschamps</a>	<a href="#">ProjectManager</a>	Pierre	Deschamps	<a href="#">XML</a>	<a href="#">EJBProgramming</a>
6	0.0039	<a href="#">H.Fresnay</a>	<a href="#">Researcher</a>	H.	Fresnay	<a href="#">RDF</a>	<a href="#">JavaProgramming</a>

Corese RDF engine version 2.1.1.11 INRIA 2005-05-01

Figure 2: Corese answer presentation

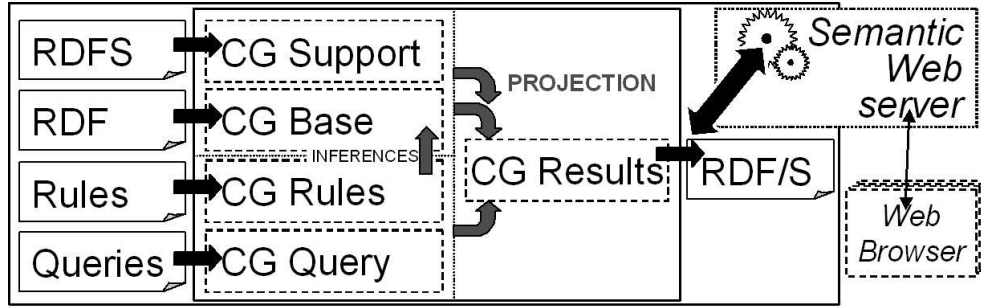


Figure 3: Corese general principle

an application, this rule is used in suggesting profiles to build project teams or manage mobility in a compagny.

### 3 Ontology based Web Search

#### 3.1 A Logic based Approach

Ontologies enable us to take into account during the query processing some background knowledge implicit in the annotations. This comprises subsumption links between concept types or relation types, signatures of relations, axioms or rules enabling deductions, etc. This knowledge supports inferences improving the efficiency of the matching process.

The use of ontological knowledge in web search approaches is expressed in the following logical model, descendin from [52]. Given (1) a model for ontologies, (2) a model for annotations of web resources based on ontologies, (3) a model for queries based on ontologies, and (4) a matching function defining how a query is matched with any annotation, a web resource  $R$  is relevant for a query  $Q$  according to the ontology  $O$  from which they are built *iff* the annotation of  $R$  and the ontology  $O$  together logically imply  $Q$  (noted  $R \wedge O \rightarrow Q$ ).

The query is viewed as a set of constraints on the description of the web ressources to be retrieved and then correspond to a search problem to be solved. The matching function implements the strategy chosen for solving this problem. It differs from one search system to another, depending on the formalism chosen for the descriptions, the types of query and the characteristics to be met by the result. Corese [12] is an ontology-based semantic search engine for the Semantic Web that implements such a matching function using the projection operator defined in the Conceptual Graphs(CG) formalism[56]. Its general principle is presented in figure 1.

### 3.2 Theoretical Foundations of Corese

The Corese engine internally works on conceptual graphs. When matching a query with an annotation, according to a shared ontology, these are translated in the conceptual graph model [56] [7]. Through this translation, Corese takes advantage of the existing work of this knowledge representation community, in particular the results on operators and reasoning capabilities of the Conceptual Graphs formalism.

Conceptual Graph (CG) and RDF(S) models share many common features and a mapping can easily be established between RDF(S) and a large subset of the CG model. An in-depth comparison of both models has been the starting point of the development of Corese [10] [16].

Both models distinguish between ontological knowledge and assertional knowledge. In both models, the assertional knowledge is positive, conjunctive and existential and it is represented by directed labeled bipartite graphs. In Corese, an RDF graph  $G$  representing an annotation or a query is thus translated into a conceptual graph CG. Regarding the ontological knowledge, the class (resp. property) hierarchy in a RDF Schema corresponds to the concept (resp. relation) type hierarchy in a CG support. RDF properties are declared as first class entities like RDFS classes, in just the same way that relation types are declared independently of concept types in a CG support. This is this common handling of properties that makes relevant the mapping of RDFS and CG models. In particular, it can be opposed to object-oriented language, where properties are defined inside classes.

There are some differences between the RDF(S) and CG models in their handling of classes and properties. However they can be quite easily handled when mapping both models. Mainly, the RDF data model supports multi-instantiation whereas the CG model does not and a RDF property declaration may specify several constraints for the domain (resp. range) whereas in the CG model, a relation type declaration specifies a single constraint for the domain (resp. range). However, the declaration of a resource as an instance of several classes in RDF can be translated in the CG model by generating the concept type corresponding to the most general specialization of the concept types translating these classes. Similarly, the multiple domain (resp. range) constraints of an RDF property can be translated into a single domain (resp. range) constraint of a CG relation type by generating the concept type corresponding to the most general specialization of the concept types constraining the domain (resp. range) of the property.

As a result, the management of RDF(S) through conceptual graphs consists in compiling the type hierarchies of the CG support, the association of a compiled type to each resource, and, finally, the use of the projection operation of the CG model as the keystone of an optimized query processing based on compiled type hierarchies.

This projection operation is the basis of reasoning in the conceptual graph model. A conceptual graph  $G_1$  logically implies a conceptual graph  $G_2$  iff it is a specialization of  $G_2$  (noted  $G_1 \leq G_2$ ). A graph  $G_1$  is a specialization of  $G_2$  iff there exists a projection of  $G_2$  into  $G_1$  such that each concept or relation node of  $G_2$  is projected on a node of  $G_1$  whose type is the same as the type of the corresponding node of  $G_2$  or a specialization of it, according to the concept type hierarchy and the relation type hierarchy.

Formally, let us define a CG as a labeled bipartite graph  $G = (C, R, E, l)$  where  $C$  and  $R$  are the sets of its concept nodes and of its relation nodes,  $E$  is the set of its edges and  $l$  is a mapping which labels each relation node  $r$  of  $R$  by a relation type  $type(r)$  of the relation type hierarchy  $\mathcal{T}_r$  and each concept node  $c$  of  $C$  by a couple  $(type(c), ref(c))$  where  $type(c)$  is a concept type of the concept type hierarchy  $\mathcal{T}_c$  and  $ref(c)$  is an individual marker or the generic referent  $*$ . The projection operation is then defined as follows [7]: A projection from a CG  $G = (C_G, R_G, E_G, l_G)$  to a CG  $H = (C_H, R_H, E_H, l_H)$  is a mapping  $\Pi$  from  $C_G$  to  $C_H$  and from  $R_G$  to  $R_H$  which:

- preserves adjacency and order on edges:  $\forall rc \in E_G, \Pi(r)\Pi(c) \in E_H$  and if  $c$  is the  $i^{th}$  neighbor of  $r$  in  $G$  then  $\Pi(c)$  is the  $i^{th}$  neighbor of  $\Pi(r)$  in  $H$ ;
- may decrease labels:  $\forall x \in C_G \cup R_G, l_H(\Pi(x)) \leq l_G(x)$ .

A query is thus processed in the Corese engine by projecting the corresponding conceptual graph into the conceptual graphs translated from RDF(S). The retrieved web resources are those for which there exists a projection of the query graph into their annotation graphs.

For example the following query graph enables us to search for documents about science and their authors.

```
[Document:*]-
  -(createdBy)-[Person:*]
  -(subject)-[Science:*]
```

When processing this query, Corese retrieves a technical report of a researcher about cognitive science and a book of a professor about social science: these documents are annotated with the following graphs upon which there exists a projection of the query graph.

```
[TechReport:#techr2871]-
  -(createdBy)-[Researcher:#john-smith]
  -(subject)-[CognitiveScience:*]
```

```
[Book:#book9638]-
  -(createdBy)-[Professor:#david-dupond]
  -(topic)-[SocialScience:*]
```

The node `[Document:*)` of the query graph is projected upon `[TechReport:#techr2871]` in the first graph and upon `[Book:#book9638]` in the second, the types `TechReport` and `Book` being subclasses of `Document` in the ontology shared by these annotation graphs and the query graph, and the uri `#doc1` and `#doc2` specializing the generic referent `*`; the node `[Person:*)` is projected upon `[Researcher:#john-smith]` and `[Professor:#david-dupond]`, their types being subclasses of `Person` and the uri `#john-smith` and `#david-dupond` specializing the generic referent `*`; the node `[Science:*)` is projected upon `[CognitiveScience:*)` and `[SocialScience:*)`, their types being subclasses of `Science`; the node `(createdBy)` is projected upon the node of the same type in both graphs; and the node `(subject)` is projected upon the node of the same type in the first graph and upon the node `(topic)` in the second, `topic` being a subtype of `subject` in the ontology.

### 3.3 Corese Ontology Representation Language

The first ontology representation language of Corese was RDFS. It has progressively been extended to handle some major features of OWL Lite. Our choice of RDFS is mainly historical: the first implementations of Corese with RDF(S) preceded the emergence of OWL. However the different projects in which Corese has been experimented have shown us that the expressivity of RDF(S) is sufficient in many applications - if extended with inference rules and approximation in the query language. We think that OWL Lite features are quite sufficient to handle most knowledge representation problems encountered in Semantic Web applications. Corese provides OWL value restrictions, intersection, subclass and algebraic properties such as transitivity, symmetry and inverse. It also provides the annotation, versioning and ontology OWL statements. Corese does not yet provide cardinality restrictions, property and class equivalences, `owl:sameAs` and loops in subsumption hierarchy.

These extensions to OWL features are based on domain axioms which are taken into account when matching a query with an annotation [11]. We have proposed an RDF Rule extension to RDF and Corese integrates an inference engine based on forward chaining production rules. The rules are applied once the annotations are loaded and before the query processing occurs: the annotation graphs are enriched before the query graph is projected. This is the key to the scalability of Corese to the web application in which we have used it.

The production rules of Corese implement conceptual graph rules [53]: a rule  $G_1 \Rightarrow G_2$  is a pair of lambda abstractions  $(\lambda x_1, \dots, \lambda x_n G_1, \lambda x_1, \dots, \lambda x_n G_2)$  where the  $x_i$  are co-reference links between generic concepts of  $G_1$  and corresponding generic concepts of  $G_2$  that play the role of rule variables.

For instance, the following CG rule states that if a person ?m is head of a team ?t which has a person ?p as a member, then ?m manages ?p :

```
[Person:?m] - (head) - [Team:?t] -
- (hasMember) - [Person:?p]
=> [Person:?m] - (manage) - [Person:?p]
```

A rule  $G_1 \Rightarrow G_2$  applies to a graph  $G$  if there exists a projection  $\pi$  from  $G_1$  to  $G$ , i.e.  $G$  contains a specialization of  $G_1$ . The resulting graph is built by joining  $G$  and  $G_2$  while merging each  $\pi(x_i)$  in  $G$  with the corresponding  $x_i$  in  $G_2$ . Joining the graphs may lead to specialize the types of some concepts, to create relations between concepts and to create new individual concepts (i.e. concepts without variable).

The Corese rule language is based on the triple model of RDF. The syntax of a rule is the following:

```
<cos:rule>
  <cos:if>
    a triple pattern
  </cos:if>
  <cos:then>
```



```

    a triple pattern
  </cos:then>
</cos:rule>

```

where `cos` is the prefix for the Corese namespace and where the triples correspond to RDF statements whose conjunction is translated into a conceptual graph.

For instance, the CG rule above is the translation of the following Corese rule:

```

<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
  </cos:then>
</cos:rule>

```

This triple syntax is shared with the Corese query language, which is further described in the next section.

### 3.4 Corese Query Language

The Corese query language is built upon the RDF triple model: a query is either a triple or a boolean combination of triples. For instance the following query retrieves all the persons (line 1) with their names (line 2) who are authors (line 3) of a thesis (line 4), and it returns their thesis title (line 5):

```

(1) ?p rdf:type kmp:Person
(2) ?p kmp:name ?n
(3) ?p kmp:author ?doc
(4) ?doc rdf:type kmp:Thesis
(5) ?doc kmp:Title ?t

```

The first element of a Corese triple is either a variable or a resource qualified name (an XML qname); the third element is either a variable, a value or a resource qname; the second element is either a property qname, a variable or a comparison operator. Class and property names are thus qnames whose namespaces are either standard and denoted by predefined prefixes (`rdf`, `rdfs`, `xsd`, `owl` and `cos` for the Corese namespace) or user-defined prefixes denoting namespaces, as shown in the following example.

```
dc as http://purl.org/dc/elements/1.1/
```

Variable names begin with a question mark. Values are typed with the XSD datatypes: numerical values, `xsd:string`, `xsd:boolean` and `xsd:date`. The language of the value of a literal can be specified by using the `@` operator and based on the specification of `xml:lang`. For instance, in the following example, we constrain the title to be in English.

```
?doc kmp:Title ?t@en
```

The comparison operators for equality and difference (`=`, `!=`), ordering (`<`, `<=`, `>`, `>=`) and string inclusion and exclusion (`~`, `!~`) enable us to compare a variable with a value or with another variable. For instance in the following example, we constrain the title so that it must include the word 'web'.

```
?t ~ "web"
```

Type comparators enable us to specify constraints on some types in a query: strict specialization (`<:`), specialization or same type (`<=:`), same type (`=:`), generalization or same type (`>=:`), strict generalization (`>:`). These operators can also be combined with a `!` negation operator (`!<:`, `!<=:`, etc.).

For instance, by using the `<:` operator in the following example, we constrain the document to be a strict specialization of a thesis (e.g. a PhD thesis, a MSc thesis, etc.).

```
?doc <: kmp:Thesis
```

By default, a list of triples is a conjunction. The `or` and `and` operators are also available and brackets enable us to combine conjunctions and disjunctions in a query. Corese handles such queries by putting them in disjunctive normal form, processing each conjunctive sub-query and juxtaposing all the results.

A limited form of negation is provided in the Corese query language; it is a negation as failure: a `not` operator is provided to prefix properties that should not be found in an annotation for it to be considered as an acceptable result. For instance, the following query retrieves all the documents which have not been graded yet.

```
?doc not::kmp:grade ?g
```

Let us note that the Corese query language supports ontological reasoning by querying ontologies just like annotations, since RDF Schemas are RDF data. For instance, the following query retrieves all the properties whose `domain` is a subclass of the `kmp:Document` concept.

```
?p rdf:type rdf:Property
?p rdfs:domain ?c
?c rdfs:subClassOf kmp:Document
```

Some SQL-like operators extend the core Corese query language to improve the presentation of the retrieved answers:

- By default, the matching of all the variables occurring in a query are returned from the retrieved annotations. A **select** operator allows to select the only variables whose matching are desired in the answers.

For instance, in the following example, we select only the title of the document and the name of its author.

```
select ?t ?n
```

- A **group** operator corresponding to the SQL **group-by** allows to group the retrieved answers according to one or more concepts instead of listing separately answers about the same concept(s) (in case an annotation is answering a query several times).

For instance, when querying for documents on a specific subject and written by an author, a **group** on the *document* variable will avoid that a document written by several authors appears several times, once for each of its authors. By default, a **group** is applied to the first variable of a query.

- A **count** operator, combined with a **group** allows the counting of the (different) documents retrieved. For instance, to mention the number of documents written on each subject, **count** is applied to the *document* variable and **group** to the *subject* one.

## 4 Approximate Semantic Web search

We have extended the core query language of Corese to address the problem of possible mismatch between end-user and ontologist concepts. Corese is able to cope with queries for which there is no exact answer by approximating the semantics of the query, its structure, or both.

### 4.1 Ontological Approximation

The first principle of the Corese semantic approximation is to evaluate semantic distances between classes in the ontology. Based on this ontological distance, Corese not only retrieves web resources whose annotations are *specializations* of the query, it also retrieves those whose annotations are *semantically close*.

#### 4.1.1 Ontological Distance

The idea of evaluating conceptual relatedness from semantic networks representation dates back to the early works on simulating the humans' semantic memory [48] [8]. Relatedness of two concepts can take many forms for instance, functional complementarity (e.g. nail and hammer) or functional similarity (e.g. hammer and screwdriver). The latter example belongs to the family of semantic similarities where the relatedness of concepts is based on the definitional features they share (e.g. both the hammer and the screwdriver are hand

tools). The natural structure supporting semantic similarities reasoning is the taxonomy of types where **is-a** links group types according to the characteristic they share (e.g. hammer, screwdriver, saw, plane, pliers, etc. are subtypes of hand tool). When applied to a semantic network using only **is-a** links, the relatedness calculated by a spreading algorithm gives a form of semantic distance. Rada *et al.* [49] defined the conceptual distance between two types A and B as the minimum number of edges separating A and B and they show it is a metric. They applied this distance to document retrieval based on Boolean queries. However, to compare Boolean queries and Boolean indexing of documents they used an averaging distance over the set of concepts in the query and the index, and this distance exhibited counter intuitive behaviors around zero.

Starting from here we can identify two main trends in defining a semantic distance over a type hierarchy: (1) the approaches that include additional external information in the distance, e.g. statistics on the use of a concept; (2) the approaches trying to rely solely on the structure of the hierarchy to tune the behavior of the distances. For the first approaches relying on additional external information we can quote Resnik [51] whose work was influenced by information theory and thus defines the notion of information content of a concept  $c$  as negative the log likelihood:  $IC(c) = -\log p(c)$ . Then the information shared by two concepts is indicated by the information content of the concepts that subsume them in the taxonomy. The author tried to apply this technique directly to words and encountered counter intuitive behaviors due to polysemia. This approach has been improved in [30] by integrating the value of the information content of the compared concepts to the calculation of the similarity. These techniques require statistical analysis of the corpus to evaluate the probabilities and thus require to find a relevant corpus to effectively approximate the probabilities by frequencies. A comparison of different distances on using WordNet, is proposed in [6]. Interestingly, there is a hybrid approach [36] bridging the first trend and the second one and Resnik found that this approach outperformed his: a negative log likelihood is still used to define the information content, but the normalized path length between the two concepts being compared is used rather than the probability of a subsuming concept.

The second trend essentially explores the different ways of combining the depths of the concepts and their deepest common super concept in the hierarchy. The simplest one is the one of Rada *et al.* presented before, where the algorithm just counts the number of arcs separating the two concepts compared. An alternative is proposed in [60] based on the ratio between the depth of the super type and the depth of the two compared types. In the domain of Conceptual Graphs, a use for such a distance is to propose a non binary projection, i.e. a similarity  $S : C^2 \rightarrow [0, 1]$  where 1 is the perfect match and 0 the absolute mismatch. The initial idea comes from Sowa [56] who applied it to similarity measures in CGs allowing sideways travel in the type lattice of the ontology. It has been applied in [50] to the join operation on graphs. More recently an equivalent distance has been proposed in [63] to build a similarity between two CGs and carry out semantic search. We will use here a simple yet efficient version of this similarity [20].

Starting from the fact that in an ontology, low level classes are semantically closer than top level classes (for instance *TechnicalReport* and *ResearchReport* which are brothers at

depth 10 are closer than *Event* and *Entity* which are brothers at depth 1), we want the ontological distance between types to decrease with depth: the deeper the closer. To express this closeness relativity, we define the length of a subsumption link  $(t, t')$  between a type  $t$  and a direct super type  $t'$  of it in an inheritance hierarchy  $H$  by  $1/2^{d_H(t')}$ , where  $d_H(t')$  is the depth of  $t'$  in  $H$ . Because of multiple inheritance,  $d_H$  will refer to the maximal depth of a type in  $H$  (with  $d_H(\top) = 0$ ,  $\forall x \in H$ ,  $d_H(x) \leq d_H(\perp)$ , and  $\forall (x, y) \in H^2$ ,  $y < x \Rightarrow d_H(y) < d_H(x)$ ). This definition supposes the type hierarchy  $H$  to be homogeneous in its choices of differentia at each and every level.

**Definition 1 (Subsumption Path Length  $l_H$ )** *The length of a subsumption path between a type  $t_1$  and one of its super types  $t_2$  in an inheritance hierarchy  $H$  is inductively defined by:*

- $\forall t \in H$ ,  $l_H(\langle t, t \rangle) = 0$ ,
- $\forall (t_1, t_2) \in H^2$ , with  $(t_1, t_2)$  a subsumption link ( $t_2$  direct super type of  $t_1$ ),  $l_H(\langle t_1, t_2 \rangle) = 1/2^{d_H(t_2)}$ ,
- $\forall (t_1, t_2) \in H^2$ , let  $t$  the direct super type of  $t_1$  in  $\langle t_1, t_2 \rangle$ , then  $l_H(\langle t_1, t_2 \rangle) = 1/2^{d_H(t)} + l_H(\langle t, t_2 \rangle) = \sum_{\{t \in \langle t_1, t_2 \rangle, t \neq t_1\}} 1/2^{d_H(t)}$

**Definition 2 (Ontological Distance  $D_H$ )** *The ontological distance between any two types of an inheritance hierarchy is the minimum of the sum of the lengths of the subsumption paths between each of them and a common super type:*

$$\forall (t_1, t_2) \in H^2, D_H(t_1, t_2) = \min_{\{t \geq t_1, t \geq t_2\}} (l_H(\langle t_1, t \rangle) + l_H(\langle t_2, t \rangle))$$

**Lemma 1**  $\forall (t_1, t_2) \in H^2$ ,  $\forall t$  common super type of  $t_1$  and  $t_2$  in  $H$ ,  $D_H(t_1, t_2) \leq 1/2^{d_H(t)-2}$

**Proof.**  $\forall (t_i, t) \in H^2$ , with  $t_i \leq t$ ,  $l_H(\langle t_i, t \rangle) \leq \sum_{n=d_H(t)}^{d_H(t_i)-1} 1/2^n$ .  
Therefore  $l_H(\langle t_i, t \rangle) \leq 1/2^{d_H(t)-1} - 1/2^{d_H(t_i)-1}$ .

The computation of the ontological distances between types is based on the following theorem:

**Theorem 1**  $\forall (t_1, t_2) \in H^2$ ,  
 $D_H(t_1, t_2) = \min_t (l_H(\langle t_1, t \rangle) + l_H(\langle t_2, t \rangle)) =$

$$\min_t \left( \sum_{\{x \in \langle t_1, t \rangle, x \neq t_1\}} 1/2^{d_H(x)} + \sum_{x \in \langle t_2, t \rangle, x \neq t_2} 1/2^{d_H(x)} \right)$$

with  $t$  a common super type of  $t_1$  and  $t_2$  of maximal depth  $d_H$ .

**Proof.**  $\forall t', t' \geq t_1$  and  $t' \geq t_2$  such that  $d_H(t') < d_H(t)$ , we have  $l_H(\langle t_1, t' \rangle) \geq 1/2^{d_H(t')}$  and for  $t$  as defined in theorem 1, we have  $l_H(\langle t_1, t \rangle) < 1/2^{d_H(t)+1} \leq 1/2^{d_H(t')}$ . Therefore  $l_H(\langle t_1, t \rangle) < 1/2^{d_H(t')}$ . Idem for  $t_2$ . So  $l_H(\langle t_1, t \rangle) + l_H(\langle t_2, t \rangle) < l_H(\langle t_1, t' \rangle) + l_H(\langle t_2, t' \rangle)$ .

**Theorem 2**  $D_H$  is a semi-distance.

**Proof.** The definition of  $D_H$  complies with the following:

- $\forall t \in H, D_H(t, t) = 2 * l_H(t, t) = 0$ ,
- $\forall (t_1, t_2) \in H^2, D_H(t_1, t_2) = 0 \Rightarrow \min_{\{t \geq t_1, t \geq t_2\}} (l_H(\langle t_1, t \rangle) + l_H(\langle t_2, t \rangle)) = 0 \Rightarrow l_H(\langle t_1, t \rangle) = l_H(\langle t_2, t \rangle) = 0 \Rightarrow t_1 = t_2$ ,
- $\forall (t_1, t_2) \in H^2, D_H(t_1, t_2) = \min_{\{t \geq t_1, t \geq t_2\}} (l_H(\langle t_1, t \rangle) + l_H(\langle t_2, t \rangle)) = \min_{\{t \geq t_2, t \geq t_1\}} (l_H(\langle t_2, t \rangle) + l_H(\langle t_1, t \rangle)) = D_H(t_2, t_1)$ ,
- the triangle inequality  $D_H(t_1, t_2) \leq D_H(t_1, t) + D_H(t, t_2)$  does not hold for any random third type  $t$ . However, by construction, it does hold for any third type  $t$  chosen among the super types. This weak notion of the principle of parsimony is enough in our case as we are only interested in paths going through the super types.

#### 4.1.2 Contextual Closeness

The ontological distance between two classes is not always sufficient to render the closeness of some concepts. We have often encountered in the experiments of Corese some concepts which are somehow distant from each other in the ontology but which share some features that make them closer from the search point of view. For instance, in the O'CoMMA ontology, *KnowledgeDissemination* which is in the Activity viewpoint and *KnowledgeEngineering* which is in the Topic viewpoint share some semantics that is not expressed by the `rdfs:subClassOf` link. When querying for *KnowledgeDissemination*, one may want to retrieve *KnowledgeEngineering* resources in case of failure.

Hence, the Corese ontology representation language has been provided with the ability to express approximation by means of the standard `rdfs:seeAlso` property. `rdfs:seeAlso` properties can be added to any existing RDF Schema, so that a given ontology can be parameterized to better fit a specific Web search task or a particular user class. This addition does not only improve browsing capabilities, it also shorten the semantic distance and tunes approximate matching.

For instance, shortening the semantic distance between *KnowledgeDissemination* and *KnowledgeEngineering* is simply achieved by instanciating a `rdfs:seeAlso` property between these two classes, as shown below:

```

<rdfs:Class rdf:ID='KnowledgeDissemination'>
  <rdfs:seeAlso
    rdf:resource='#KnowledgeEngineering'/>
</rdfs:Class>
<rdfs:Class rdf:ID='KnowledgeEngineering'/>

```

As for classes, some properties may share a semantic proximity from the Web search point of view. For instance *isInterestedIn*, *graduatedIn* and *hasForPersonalInterest* are obviously close properties. A `rdfs:seeAlso` property can be set between close properties which authorizes the occurrence of one of them instead of the other when matching a query with an annotation.

It is worth considering the `seeAlso` property be inherited by subclasses and subproperties. Hence any Corese ontology has the following rule for classes (and the equivalent one for properties):

```

?x rdfs:seeAlso ?y
?z rdfs:subClassOf ?x
=> ?z rdfs:seeAlso ?y

```

In addition to ontological distance, `rdfs:seeAlso` is available to tune approximations on concept types and it is the only mean to approximate properties.

#### 4.1.3 Approximate Projection

Based on the ontological distance defined above, Corese supports an approximate search process. It distinguishes between *exact answers* for which there exists a projection of the query upon their annotations and *approximate answers* for which there exists an *approximate projection* of the query upon their annotations. These annotations have a structure upon which the query can be projected but whose concept and relation types are not necessarily subsumed by those of the query: they are just close enough to them in the ontology. Formally, we define the approximate projection as follows.

**Definition 3** *An approximate projection from a CG  $G = (C_G, R_G, E_G, l_G)$  to a CG  $H = (C_H, R_H, E_H, l_H)$  is a mapping  $\Pi$  from  $C_G$  to  $C_H$  and from  $R_G$  to  $R_H$  which:*

- *preserves adjacency and order on edges:  $\forall rc \in E_G, \Pi(r)\Pi(c) \in E_H$  and if  $c$  is the  $i^{th}$  neighbor of  $r$  in  $G$  then  $\Pi(c)$  is the  $i^{th}$  neighbor of  $\Pi(r)$  in  $H$ ;*
- *may change the labels of concept nodes to ontologically close ones:*  
 $\forall c \in C_G, D_{\mathcal{T}_c}(\text{type}(c), \text{type}(\Pi(c))) < \varepsilon$ , *where  $D_{\mathcal{T}_c}$  is the ontological distance in the concept type hierarchy and  $\varepsilon$  is a threshold chosen as the maximal distance allowed.*
- *may decrease the labels of relation nodes or change them to contextually close ones:*  
 $\forall r \in R_G, l_G(r) \leq l_H(\Pi(r))$  *or a seeAlso property stands between  $l_G(r)$  and  $l_H(\Pi(r))$ .*

Corese authorizes the approximation of a class by potentially any other class of the ontology whereas for combinatorial constraints, the approximation of a property is limited to contextual closeness. Ontological distances are thus computed between concept types (and not between relation types) and the similarity between a resource annotation and a query depends on the ontological distances between the types of their concept nodes, contextual closenesses being translated in terms of ontological distances. Setting a **rdfs:seeAlso** property between two concept types  $c_1$  and  $c_2$  has for effect to shorten the ontological distance between them to a brotherhood distance and consequently increase the similarity between two graphs for which there exists an approximate projection mapping a node of type  $c_1$  in one graph to a node of type  $c_2$  in the other graph. Setting a **rdfs:seeAlso** property between two relation types  $r_1$  and  $r_2$  is also taken into account in the computation of the similarity between two graphs for which there exists an approximate projection mapping a node of type  $r_1$  in one graph with a node of type  $r_2$  in the other graph. The cost of this approximation is proportional to the ontological distances of the types  $c_1$  and  $c'_1$  of the neighbors concept nodes of  $r_1$  (RDF properties being binary relations).

Formally, we define the similarity between two graphs for which there exists an approximate projection from one to the other as follows.

**Definition 4** *The similarity between a CG  $G = (C_G, R_G, E_G, l_G)$  and a CG  $H = (C_H, R_H, E_H, l_H)$  for which there exists an approximate projection  $\Pi$  from  $G$  into  $H$  is equal to:*

$$\Delta_{\Pi}(G, H) = \frac{1}{1 + k * \frac{\sum_{c \in C_G} \delta(c, \Pi(c)) + \sum_{r \in R_G} \delta(r, \Pi(r))}{D_{max} * (\text{card}(C_G) + \text{card}(R_G))}}$$

where, for concept:

- $\delta(c, \Pi(c)) = 0$  if  $\text{type}(\Pi(c)) \leq \text{type}(c)$  or else
- $\delta(c, \Pi(c)) = 1/2^{d_{\mathcal{T}_c}(c)-2}$  if there is a **seeAlso** property set between  $\text{type}(c)$  and  $\Pi(\text{type}(c))$  or else
- $\delta(c, \Pi(c)) = D_{\mathcal{T}_c}(\text{type}(c), \text{type}(\Pi(c)))$ ;

and for relations:

- $\delta(r, \Pi(r)) = 0$  if  $\text{type}(\Pi(r)) \leq \text{type}(r)$  or else
- $\delta(r, \Pi(r)) = 1/2^{\sup(d_{\mathcal{T}_c}(\text{type}(c_r)), d_{\mathcal{T}_c}(\text{type}(c'_r)))}$  with  $c_r$  and  $c'_r$  the neighbor concept nodes of  $r$  in  $G$ , if there is a **seeAlso** property set between  $\text{type}(r)$  and  $\Pi(\text{type}(r))$ .

$D_{max} = 2 * \sum_{i=1}^{d_{\mathcal{T}_c}(\perp)} 1/2^i$  is bounding the distances on  $\mathcal{T}_c$ .  $\frac{\sum_{c \in C_G} \delta(c, \Pi(c)) + \sum_{r \in R_G} \delta(r, \Pi(r))}{D_{max} * (\text{card}(C_G) + \text{card}(R_G))}$  varies between 0 and 1: it is the average of the ontological distances between the types of the nodes of  $G$  and the types of the corresponding nodes of  $H$ , normalized by the maximal distance on  $\mathcal{T}_c$ . The constant  $k$  makes this norm vary between 1 and 100 percent. It depends on both the ontology depth and the user queries. In the applications where most of the queries are built with concept types which are leaves of the type hierarchy, we fix  $k$



equal to  $2^{d\tau_c(\perp)}/100$ . Therefore, the similarity between  $G$  and  $H$  is contained between 1 and 100 percent.

The relative relevance of the retrieved annotations is measured by their similarity to the query. Those whose similarity does not overpass a given threshold are presented to the user, sorted by decreasing similarity and their approximate concepts and relations identified and enhanced with a special color or font. This threshold is relative to the best found approximation of the query: the annotations whose similarity to the query is more than 10 times smaller than the similarity of the best found approximation are put aside.

Syntactically, the **more** keyword in the **select** clause of a Corese query asks for approximate answers. In this case, Corese basically approximates every concepts of the query. However, its query language allows to require the specialization of some concepts while approximating the others by using type comparators. For instance, by using the `<=:` operator, Corese is able to retrieve the persons interested in Knowledge Engineering (or something close) and member of a project (or something close) by processing the following query :

```
select more where
?person c:interestedBy ?k
?person <=: c:Person
?k rdf:type c:KnowledgeEngineering
?person c:member ?project
?project rdf:type c:Project
```

In this query, the class **Person** or one of its subclasses is required by using the `<=:` specialization operator, while **KnowledgeEngineering** and **Project** may be approximated.

## 4.2 Structural Approximation

The ontology-based approximations described above make up for the possible divergencies between the *vocabularies* of ontology designers, annotation designers and end-users, i.e. query designers. Another kind of approximation supported by the Corese query language makes up for the possible divergencies between the annotation design and the query design. In some cases, the user will search for conceptually related resources while ignoring how to express their relationship, i.e. how the annotator has described it. For instance, he or she may search for organizations related to Human Science, whatever the relationship. This kind of approximation concerns the *structure* of the annotations but still remains semantic. It can be viewed as the approximation of a complex relationship that cannot be represented by a single property and requires a graph to define it (which is a subgraph of the annotation graphs).

The Corese query language supports such approximations through the *path graph* feature. It allows to search for resources related by a relation path graph (made of successive binary relations between a series of intermediate concepts). Let us note  $\mathcal{P}_G$  the set of the relation path graphs in a CG  $G$ . We extend the definition of the projection as follows in order to allow the mapping of a relation node with a path graph.

**Definition 5** A projection from a CG  $G = (C_G, R_G, E_G, l_G)$  to a CG  $H = (C_H, R_H, E_H, l_H)$  is a mapping  $\Pi$  from  $C_G$  to  $C_H$  and from  $R_G$  to  $\mathcal{P}_H$  which:

- preserves adjacency and order on edges:

- $\forall rc \in E_G$ , if  $\Pi(r) \in R_H$  then  $\Pi(r)\Pi(c) \in E_H$  and if  $c$  is the  $i^{th}$  neighbor of  $r$  in  $G$  then  $\Pi(c)$  is the  $i^{th}$  neighbor of  $\Pi(r)$  in  $H$ ;
- $\forall rc \in E_G$ , if  $\Pi(r) \notin R_H$  then  $\Pi(r)$  is a path graph defining a relation type  $t$  and considering the contraction in  $H$  of  $\Pi(r)$  to a relation node  $r'$  of type  $t$ ,  $r'\Pi(c) \in E_H$  and if  $c$  is the  $i^{th}$  neighbor of  $r$  in  $G$  then  $\Pi(c)$  is the  $i^{th}$  neighbor of  $r'$  in  $H$ ;

- may decrease labels:

- $\forall c \in C_G$ ,  $l_G(c) \leq l_H(\Pi(c))$ ;
- $\forall r \in R_G$ , if  $\Pi(r) \in R_H$  then  $l_G(r) \leq l_H(\Pi(r))$ ;
- $\forall r \in R_G$ , if  $\Pi(r) \notin R_H$  then  $\forall x \in R_{\Pi(r)}$ ,  $l_{\Pi(r)}(x) \leq l_G(r)$

The definition of relation types with CGs is formally studied in [37]: a n-ary relation type  $t$  is defined by a n-ary lambda-abstraction *type*  $t(x_1, \dots, x_n)$  is  $\lambda x_1, \dots, x_n G$  where  $x_1, \dots, x_n$  correspond to generic concepts of  $G$  among which a relation of type  $t$  thus stands. A type expansion denotes the replacement of a type  $t$  by the graph  $G$  of its definition; symmetrically, a graph contraction denotes the replacement of a graph  $G$  by the type  $t$  it defines.

Finally, a combination of our two definitions of a projection (Definition 3 and Definition 5) provides Corese with the ability of asking for both ontological and structural approximations of the queries in the retrieval process.

Syntactically, in the Corese query language, the relation stated between the resources for which a complex relationship is searched for must be suffixed by the maximal length of the path graphs to search for. This value is put between curly brackets to search for any paths, or between square brackets to restrict the search to directed paths. By default, Corese stops after retrieving one path (with the shortest length). It computes all the possible paths when the relation is prefixed by the **all** qualifier.

For instance, let us consider the following query asking for the organizations related to Human Science by a (non directed) relation path of length smaller or equal to two:

```
?org all::c:relation{2} ?topic
?org rdf:type c:Organization
?topic rdf:type c:HumanScience
```

The two following annotations answer this query: the CNRS institute is interested in Human Science and a member of the INRIA institute is graduated in Human Science.

```
[Institute:#CNRS]
- (interestedIn) - [HumanScience:*
```

```
[Person:#Alain]
  -(memberOf)-[Institute:#INRIA]
  -(graduatedIn)-[HumanScience:*)
```

Of course this kind of structural approximation only makes sense in case there are criteria on the searched related resources to limit the number of relation paths to search for, otherwise such queries would lead to combinatorial explosion.

## 5 Software, Applications and Evaluation

### 5.1 Architecture

Corese is developed in Java and publicly available under the INRIA licence at <http://www.inria.fr/acacia/corese> including Java packages, documentation and Swing GUI. A Corese semantic web server has also been developed according to a 3-tier architecture (figure 2) as described in the following subsections

#### 5.1.1 Presentation Layer

generating the content to be presented in the users' browser (ontology views and browsing controls, query edition interfaces, annotation forms, answers, etc.), this part relies on a model-view-controller architecture to handle HTTP requests and generate responses fed by the appropriate Corese services of the Application Logic Layer and formatted using XSLT or JSP templates. It is implemented by servlets and provides the front-end of what we call a Semantic Web Server, i.e. an HTTP server able to: solve semantic web queries submitted through HTTP requests; provide JSP tags to include semantic web processing and rendered results in web pages; provide XSLT extensions to perform semantic web functions related to XPath expressions, thus improving RDF/XML transformation capabilities; provide a form description language to dynamically build forms using queries for instance to populate the different choices of a drop-down box.

#### 5.1.2 Application Layer

a platform that implements three main services accessible through an API: a Conceptual Graph server (using the Notio API<sup>2</sup>), a Query engine and a rule engine. Parsers transform RDF to CG, Rules to CG Rules and Queries to CG graphs to be projected. The core CG server implements the management of the CG base, the projection and join operators and type inferences on the type hierarchies. A CG-to-RDF pretty-printer produces results in RDF/XML syntax. This layer is an independent package and provides an API that can be used by developers to add semantic web capabilities to their applications.

---

<sup>2</sup><http://www.cs.ualberta.ca/~finnegan/notio/>

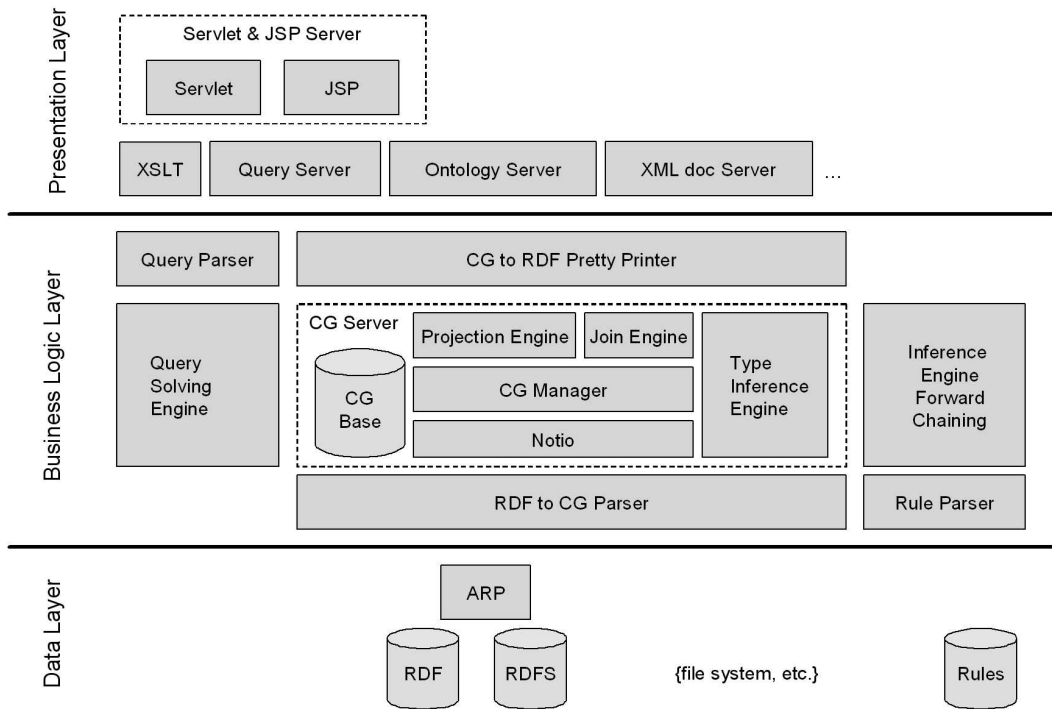


Figure 4: Corese 3-tier architecture

### 5.1.3 Persistent Layer

RDF(S) data accessed by means of the ARP<sup>3</sup> parser and translated by the RDF-to-CG Parser. Rules are saved in separate files and parsed by the Rule Parser.

## 5.2 Real World Applications

Corese has been tested on several real world applications with ontologies of large sizes: these applications are detailed in [17].

### 5.2.1 SAMOVAR

Samovar is a system supporting a vehicle project memory for Renault car manufacturer. The ontology has 792 concept types and 4 relation types, and annotates 4483 problem descriptions. Corese answers queries such as: *"Find all fixing problems that occurred on the dashboard in a past project"*.

<sup>3</sup>ARP parser from HP : <http://www.hpl.hp.com/semweb/arp.htm>

### 5.2.2 CoMMA IST project

CoMMA is a multi-agent system for corporate memory management (integration of a new employee and technological watch). The O'CoMMA ontology comprises 472 concepts types and 80 relation types used for annotating documents or people in an organization. Corese answers distributed queries over several annotation bases such as *"Find the users who may be interested in the technological news that was just submitted about GSM v3"*.

### 5.2.3 ESCRIRE

ESCRIRE was oriented towards the annotation and search of abstracts of Medline database on genetics. Corese answers queries such as: *"Find the articles describing interactions where the Ubx gene acts as target and where the instigator is either en gene or dpp gene"*.

### 5.2.4 KMP

KMP is a knowledge management platform for cartography of skills in telecommunications for Sophia Antipolis firms. The KMP ontology comprises 1136 concept types with a maximal depth of 15 and 83 relation types. Corese answers queries such as: *"Who are the possible industrial partners knowing how to design integrated circuits within the GSM field for cellular/mobile phone manufacturers?"*.

### 5.2.5 Ligne-de-Vie

Ligne-de-Vie (Life Line) is a virtual staff for a health network relying on an ontology comprising 26432 concept types and 13 relation types. It guides physicians discussing the possible diagnoses and the alternative therapies for a given pathology, according to the patient's features. It enables to answer queries such as: *"Find the past sessions of virtual staff where a given therapy was chosen for the patient and indicate what were the arguments in favour of this therapy"*.

### 5.2.6 MEAT

MEAT is a memory of experiments of biologists on DNA microarray relying on annotations on scientific articles, and using UMLS as an ontology. Corese answers queries such as *"Find all the articles asserting that HGF gene plays a role in lung disease"*.

Corese has also been tested with other ontologies such as the Gene ontology (represented by an RDF graph with 13700 concept types and 950000 relations), IEEE LOM, W3C CC/PP, Dublin Core, etc.

## 5.3 Evaluation

We will illustrate the evaluation from system viewpoint (performance) and from end-user viewpoint (scenario-based evaluation).

### 5.3.1 Corese Performance

Corese engine performance has been measured on an RDF(S) base which comprises 19000 properties and 8000 resources. The Corese standard test base of 260 queries covering all the features of the query language and with a rule base of 11 rules runs in 18 seconds on a laptop (P4 1,7Ghz, 512Mb). The average answer time is 0.07 second per query.

### 5.3.2 Scenario-based evaluation

Once provided with domain axioms, approximate queries and presentation capabilities (features that were really required by the users), Corese received a very positive evaluation by its users and [24] details the scenario-based evaluation used for several applications among which CoMMA and KMP.

The evaluation conducted on the KMP application of Corese involved 10 mediators and about 30 users from 17 organizations. While users appreciate the technical features of Corese, they criticize some useability aspects of the application.

The following positive points were emphasized:

- the users found the Corese query language power effective,
- the users appreciated the ontology-driven user interface forms,
- The users appreciated the approximate search feature of Corese and considered it as unique and very useful since it enabled them to find the best match for any query with the ontology.

Several useful improvements on the KMP system were suggested by the users:

- Some users would like more dynamic interactions in the query answer cycle. They would like to be able to easily refine a query from the answer. When a query is refined, they would like the differences in the answer to be enhanced. They would also appreciate the system to manage a history of queries.
- Some users estimated that the ordering of approximate answers could be improved. They would also like the system to justify the proposed approximations. Some users would also like the ability to tune the approximation: e.g. which concept can be approximated and how. In the result, it should be possible to document the distance of each approximate concept to its query concept. Specific style sheets are necessary for approximate results, which is already the case in the KMP application.
- Some experiments showed that the generic distance was not always completely accurate: sometimes a class is closer to its brother class than to its direct ancestor. This incites us to some more work on distance modelling in ontologies.

To sum up, ontology-driven tools are powerful and useful. However, the *interaction* with users should not be directly driven by the ontology but by user, task and domain models.

## 5.4 Related Work

## 5.5 Query Languages for RDF

The Corese RDF Query Language is close to RDQL[44] , SeRQL <sup>4</sup> [31] (Sesame language [5]) and SPARQL<sup>5</sup>.

RDQL is the query module of the Jena RDF toolkit. It is an implementation of the SquishQL language. SquishQL is a simple RDF Query language based on SQL-like constructs. In its model, an RDF graph is represented by a set of triples (in conjunction); a query is a set of triples where any resource, property or literal can be replaced by a variable; the answer to a query is a pair made of a set of triples subset of the target knowledge base which matches the query and a table of sets of legal values for the variables. The semantics of RDFS is not represented in SquishQL and the triples retrieved are those explicitly present in the store: those that would be implied by the semantics of RDFS (e.g. by the `subClassOf` semantics) are not looked for.

Sesame is a generic architecture for persistent storing of RDF(S) data into Data Based Management Systems (DBMS) and querying of RDF(S) data with the SeRQL language. SeRQL is an RDF query language defined by means of a set of core queries, a set of basic filters and a way to build new queries through functional composition and iterators. The core queries support the retrieval of all classes, the retrieval of all properties and the retrieval of all instances of a given class. More complex queries are composed with the SeRQL **select-from-where** constructor whose **from** clause expects path expressions to query both RDF schemas and RDF data. When parsing an SeRQL query, the query module of Sesame builds an optimized query tree model for it which is evaluated through a set of calls to the Storage and Inference Layer SAIL API of Sesame which are following this tree structure into which the query has been broken down.

SPARQL may become a W3C recommendation to query RDF. Like SPARQL, Corese Query Language is based on a boolean combination of triples that can be constrained by evaluable expressions. Corese also processes datatyped RDF literals, optional properties, alternatives and the named graph scheme of SPARQL using a source statement. Corese returns an RDF/XML graph or an XML binding format. The bindings are available through an API. Corese also provides the **select**, **distinct**, **sort** and an equivalent of **limit** statements but not the **describe** and **ask** SPARQL statements. The two last ones can be simulated by Corese.

In addition to SPARQL statements, Corese provides : approximate search and structural path graph. Corese enables to group and count results. It also enables to merge all results into one graph or provide the results as a list of graphs. Corese can also, at user option, generate the result using the vocabulary (the classes) used in the query instead of the possibly specialized vocabulary of the target RDF graph.

<sup>4</sup><http://www.openrdf.org/doc/users/ch06.html>

<sup>5</sup><http://www.w3.org/TR/rdf-sparql-query/>

## 5.6 Ontology-Based Web Search Applications

Among the most famous ontology-based Web search applications, let us cite DAMLJessKB [35] and OntoBroker [14] in which ontologies and queries are expressed in Frame Logic and translated into Horn Logic.

DAMLJessKB and its successor OWLJessKB and the e-Wallet [22] are tools for reasoning with OWL-Lite. They both integrate the Jess production system to carry out the semantics of RDF, RDFS, XSD and OWL-Lite. They map the RDF triples in a given set of annotations and ontologies into facts in the CLIPS-like language of Jess and then apply rules implementing the relevant Semantic Web languages. By using Jess, both systems can perform class instance reasoning and terminological reasoning about the relationships among classes. In addition, the e-Wallet is able to run rules to complete the knowledge base, to invoke external services to obtain new knowledge, to answer queries and to control the precision and truthfulness of answers to preserve privacy.

OntoBroker and its successor On2broker [19] are early ontology-based systems based on Frame Logic. OntoBroker handles metadata embedded in HTML documents with special tags while On2Broker handles RDF annotations. In both systems, ontologies and queries are expressed in Frame Logic that allows the representation of a concept hierarchy, a relation hierarchy and rules. The query engine translates these Frame Logic data into Horn Logic to answer a query.

Beside these general-purpose reasoners, WebKB [42] and OntoSeek [27] are search-oriented applications based on CG. WebKB interprets statements expressed in a CG linear notation and embedded in HTML documents; it allows to query lexical or structural properties of HTML documents. OntoSeek focuses on lexical and semantic constraints in the encoding of resources into CG and the building of queries. WebKB, OntoSeek and Corese all build upon CG and consequently use the same core principle of matching a query graph against annotation graphs with respect to subsumption relations between concepts or relations. However neither WebKB nor OntoSeek handle RDF(S) data as Corese, and they do not handle rules in their ontology representation language. Moreover they both focus on the annotation activity and ontological problematics and they are not provided with an expressive query language as Corese.

Above all, when compared to these applications, Corese is the only ontology-based system to provide approximate search features. To the best of our knowledge, Corese is the only web search application addressing the problem of structural approximation of queries. There are some few recent works addressing the problem of ontological approximation for searching the web. Among them, let us cite [28] that approximates overlap between RDFS concepts based on Bayesian networks and proposes to apply their approximation to define a semantic distance between concepts and sort the answers to an ontology-based search. But this method has not actually been applied to web search and it focuses on overlap rather than subsumption. The PASS system [59] searches abstracts of research papers from IEEE Transactions; the search uses a fuzzy ontology of term associations for query refinement. When compared to Corese, PASS searches for documents tagged with domain-specific keywords while Corese searches for documents annotated by more expressive descriptions



(RDF graphs), based on ontologies; the PASS fuzzy ontology of term associations is similar to the Corese *see-Also* network of concepts and the measure of the so-called *narrower* and *broadener relations* between terms would correspond to our semantic distances between the only concepts related by *see-Also* relations - and not between any two concepts in the ontology.

## 5.7 Ontology Alignment or Versioning

Last, an analogy could be made with the mapping between classes of two ontologies to be aligned or with the comparison of two versions of the same ontology. The various approaches for ontology alignment (see the state of the art on current alignment techniques provided by the Knowledge Web network <sup>6</sup>) or the PromptDiff algorithm heuristic matchers [46] for finding the differences of two versions of the same ontology could be useful if Corese aimed at finding an alignment between the ontology and the user's (implicit) personal ontology. But Corese does not aim at tackling such a case: it rather focuses on finding the RDF annotations the closest "semantically" (i.e. w.r.t. the ontology and our ontological distance) with the user's query.

### 5.7.1 Other Domain-Specific Web Search Approaches

Other Web search techniques draw from the fields of data mining, machine learning, statistics, databases, classical information retrieval. Current approaches of domain specific and personalized Web search based on these techniques are presented in [38]: the scoring of Web pages, calling for probabilistic techniques; the indexation of Web pages, calling for classification techniques; the learning of domain specific keywords used as contexts of the queries; the learning of user profiles used to constrain the search. All these approaches based on numerical techniques cannot be compared to ontology based approaches of Web search, based on symbolic techniques. However they may be complementary and the KIM platform [33] is an example of such a combination of techniques. It is dedicated to the automatic ontology-based annotation of Web pages and the Retrieval of Information based on the indexing of Web pages by concepts and classes in the ontology and measures relevance according to them. When compared to CORESE, KIM only handles light-weight upper level ontologies without axioms.

## Conclusion

We have presented the Corese ontology-based Web search system whose query language handles RDF annotations, RDFS and some major features of OWL Lite. We have stressed the need for approximation in querying the semantic web and detailed the mechanism Corese integrates to provide a generic scheme for approximate search: a semantic approximation based on (1) the definition of an ontological distance which enables us to sort approximate

<sup>6</sup><http://knowledgeweb.semanticweb.org/semanticportal/home.jsp>

answers by decreasing similarity from the query, and on (2) the definition of relation paths which enables us to approximate the structure of the searched annotations.

Beside ontology-based Web search, Corese definition of semantic distances between concepts could be integrated in existing alignment techniques. On the other hand, we could benefit from such alignment techniques for integrating other aspects than simple structural distance and ontology depth in the Corese semantic distance.

We plan to investigate how to specify semantic distances or semantic heaps between classes in the ontology depending on viewpoints to take into account different user profiles in the query processing. With this very same goal, we aim at contextualizing the distance of the `seeAlso` property and make it depend on user profiles or user tasks. This will enable us to integrate user profile features into the Corese query language.

## Acknowledgements

We deeply thank Olivier Savoie for its participation to Corese implementation, INRIA that funded him and Alain Giboin, Cécile Guigard, Nicolas Gronnier and Karine Delêtre for KMP evaluation.

## References

- [1] ARP from HP. <http://www.hpl.hp.com/semweb/arp.htm>
- [2] J.C. Arpirez, O. Corcho, M. Fernandez-Lopez, A. Gómez-Pérez. WebODE in a Nutshell. In *AI Magazine*, vol 24(3), 2003
- [3] S. Bechhofer, R. Volz, P. Lord. Cooking the Semantic Web with the OWL API. In *Proc. of the 2nd International Semantic Web Conference, ISWC2003*, LNCS 2870, pp. 659-675, Sanibel Island, Florida, USA, 2003
- [4] T. Berners-Lee, J. Handler, O. Lassila. *The Semantic Web*, Scientific American, 2001
- [5] J. Broekstra, A. Kampman, F. van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In *Proc. of the 1st International Semantic Web Conference ISWC2002*, LNCS 2342, pp. 54-68, Sardinia, Italy, 2002
- [6] A. Budanitsky, G. Hirst Semantic distance in WordNet: An Experimental, Application-oriented Evaluation of five Measures, In *Workshop on WordNet and Other Lexical Resources*, Second meeting of the North American Chapter of the Association for Computational Linguistics. Pittsburgh, PA, 2001
- [7] M. Chein, M.L. Mugnier, G. Simonet. Nested Graphs: A Graph-based Knowledge Representation Model with FOL Semantics, In *Proc. of the 6th International Conference on Principles of Knowledge Representation and Reasoning, KR'98*, pp. 524-534, Trento, Italy, 1998

- [8] A. Collins, E. Loftus. A Spreading Activation Theory of Semantic Processing. *Psychological Review*, vol. 82, pp. 407-428, 1975
- [9] Comma Consortium 2001. Corporate Memory Management through Agents IST project <http://www.si.fr.atosorigin.com/sophia/comma/Htm/HomePage.htm>
- [10] O. Corby, R. Dieng, C. Hébert. 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
- [11] O. Corby, C. Faron-Zucker. Corese: A Corporate Semantic Web Engine, In *Proc. of the Workshop on Real World RDF and Semantic Web Applications, 11th International World Wide Web Conference, WWW2002, Honolulu, Hawaii, USA, 2002*
- [12] O. Corby, R. Dieng-Kuntz, C. Faron-Zucker. Querying the Semantic Web with the CORESE search engine. In R. Lopez de Mantaras and L. Saitta eds, *Proc. of the 16th European Conference on Artificial Intelligence (ECAI'2004), Valencia, 22-27 August 2004, IOS Press*, p. 705-709.
- [13] P. Coupey, C. Faron. Towards Correspondences between Conceptual Graphs and Description Logics. In *Proc. of the 6th International Conference on Conceptual Structures, ICCS'98, LNAI 1453, Springer Verlag*, pp. 165-178, Montpellier, France, 1998
- [14] S. Decker, M. Erdmann, D. Fensel, R. Studer. *OntoBroker: Ontology based Access to Distributed and Semi-structured Information*. In *Semantic Issue in Multimedia Systems*, Kluwer Academic Publisher, Boston, 1999
- [15] A. Delteil, C. Faron. A Graph-Based Knowledge Representation Language. In *Proc. of the 15th European Conference on Artificial Intelligence, ECAI2002, IOS Press*, pp. 297-301, Lyon, France, 2002
- [16] A. Delteil, C. Faron, R. Dieng. Extensions of RDFS Based on the Conceptual Graph Model. In *Proc. of the 9th International Conference on Conceptual Structure, ICCS2001, LNAI 2120, Springer-Verlag*, pp. 275-289, Stanford, CA, USA, 2001
- [17] R. Dieng-Kuntz, O. Corby. Conceptual Graphs for Semantic Web Applications, To appear in *Proc. of the 13th Int. Conference on Conceptual Structures (ICCS'2005)*, F. Dau, M. L. Mugnier, G. Stumme (eds), Kassel, Germany, July 17-23, 2005, Springer-Verlag, LNAI 3596, pp. 19-50, 2005.
- [18] R. Dieng-Kuntz, D. Minier, F. Corby, M. Ruzicka, O. Corby, L. Alamarguy, P.H. Luong. Medical Ontology and Virtual Staff for a Health Network, in *Proc. of the 14th Int. Conf. on Knowledge Engineering and Knowledge Management, EKAW2004, UK, October 2004*.

- [19] D. Fensel, J. Angele, S. Decker, M. Erdmann, H.P. Schnurr, S. Staab, R. Studer, A. Witt. On2broker. Semantic-Based Access to Information Sources at the WWW. In Proc. of Webnet'99, Volume 1, pp. 366-371, Honolulu, Hawaii, USA, 1999
- [20] F. Gandon, A. Poggi, G. Rimassa, P. Turci. Multi-Agent Corporate Memory Management System, In Journal of Applied Artificial Intelligence, vol 16(9-10), pp. 699-720, 2000
- [21] F. Gandon, A. Giboin, J. Hackstein. Report on Enterprise Modelling and User Modelling, CoMMA Project (IST-1999-12217), Deliverable COMMA/WP2/D11, 2001
- [22] F. Gandon, M. Sadeh. Semantic Web Technologies to Reconcile Privacy and Context Awareness, In Web Semantics: Science, Services and Agents on the World Wide Web, Elsevier Science, vol 1(3), pp. 241-260, 2004. [http://mycampus.sadehlab.cs.cmu.edu/public\\_pages/OWLEngine.html](http://mycampus.sadehlab.cs.cmu.edu/public_pages/OWLEngine.html)
- [23] S. Handschuh, S. Staab, F. Ciravegna. S-CREAM - Semi-automatic CREAtion of Metadata. In Proc. of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW2002, LNAI 2473, pp. 358-372, Siguenza, Spain, 2002
- [24] A. Giboin, F. Gandon, O. Corby, R. Dieng. Assessment of Ontology-based Tools: Systemizing the Scenario Approach, In Proc. of the EON Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW 2002, pp. 63-73, Siguenza, Spain, 2002
- [25] J. Golebiowska, R. Dieng, O. Corby, D. Mousseau. Building and Exploiting Ontologies for an Automobile Project Memory, In Proc. of the International Conference on Knowledge Capture (K-CAP), ACM Press, pp. 52-59, Victoria, Canada, 2001
- [26] A. Gómez-Pérez, O. Corcho. Ontology Languages for the Semantic Web. In IEEE Intelligent Systems, vol 17(1), pp. 54-60, 2002
- [27] N. Guarino, C. Masolo, G. Vetere. Ontoseek: Content-based access to the Web. In IEEE Intelligent Systems, vol. 14(3), pp. 70-80, 1999
- [28] M. Holi, E. Hyvönen. A Method for Modeling Uncertainty in Semantic Web Ontologies. In Proc. of WWW'2004, 2004.
- [29] A.K. Iyengar, D. De Roure (Eds). Special Section on WWW2002. In IEEE Transactions on Knowledge and Data Engineering, vol 15(4), pp. 769-870, 2003
- [30] J. Jiang, D. Conrath. Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. In Proc. of International Conference on Research in Computational Linguistics, Taiwan, 1997

- [31] G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, M. Scholl. RQL: a declarative query language for RDF. In Proc. of the 11th International World Wide Web Conference, WWW2002, pp. 592-603, Honolulu, Hawaii, USA, 2002
- [32] K. Khelif, R. Dieng-Kuntz. Ontology-Based Semantic Annotations for Biochip Domain, ECAI2004 Workshop on Knowledge Management and Organizational Memories, Valencia, Spain, August 22, 2004.
- [33] A. Kiryakov, B. Popov, D. Ognyanoff, D. Manov, A. Kirilov, M. Goranov. Semantic Annotation, Indexing and Retrieval. In Proc. of the 2nd International Semantic Web Conference, ISWC2003, LNCS 2870, pp. 484-499, Sanibel Island, Florida, USA, 2003
- [34] KMP. Knowledge Management Platform RNRT Project [http://www.telecom.gouv.fr/rnrt/projets/res\\_02\\_88.htm](http://www.telecom.gouv.fr/rnrt/projets/res_02_88.htm)
- [35] J. Kopena, W. Regli. DAMLJessKB: A Tool for Reasoning with the Semantic Web. In IEEE Intelligent Systems, vol 18(3), pp. 74-77, 2003
- [36] C. Leacock, M. Chodorow. Filling in a Sparse Training Space for Word Sense Identification. ms., 1994.
- [37] M. Leclerc. Reasoning with Type Definitions, In Proc. of the 5th International Conference on Conceptual Structures, ICCS'97, LNAI 1257, Springer Verlag, pp. 401-415, Seattle, USA, 1997
- [38] B. Liu, S. Chakrabarti (Eds) Special Section on Mining and Searching the Web. In IEEE Transactions on Knowledge and Data Engineering, vol 16(1), pp. 2-96, 2004
- [39] D.L. McGuinness, R. Fikes, J. Hendler, L.A. Stein. DAML+OIL: An Ontology Language for the Semantic Web. In IEEE Intelligent Systems, vol 17(5), pp. 77-80, 2002
- [40] A. Maedche, B. Motik, L. Stojanovic, R. Studer, R. Volz. An Infrastructure for Searching, Reusing and Evolving Distributed Ontologies. In Proc. of the 12th International World Wide Web Conference, WWW2003, pp. 439-448, Budapest, Hungary, 2003
- [41] A. Maedche, B. Motik, L. Stojanovic, R. Studer, R. Volz. Ontologies for Enterprise Knowledge Management. In IEEE Intelligent Systems, vol 18(2), pp. 26-33, 2003
- [42] P. Martin, P. Eklund. Knowledge Retrieval and the World Wide Web. In IEEE Intelligent Systems, vol 15(3), pp. 18-25, 2000
- [43] C. Medina-Ramirez, R. Dieng-Kuntz, O. Corby. Querying a heterogeneous corporate semantic web: a translation approach, In Proc. of the EKAW'2002 Workshop on KM through Corporate Semantic Webs, Siguenza, Spain, October 2002
- [44] L. Miller, A. Seaborne, A. Reggiori. Three Implementations of SquishQL, a Simple RDF Query Language. In Proc. of the 1st International Semantic Web Conference, ISWC2002, LNCS 2342, pp. 423-435, Sardinia, Italy, 2002

- [45] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Ferguson, M. A. Musen. Creating Semantic Web Contents with Protege-2000. In IEEE Intelligent Systems, vol 16(2), pp. 60-71, 2001
- [46] N. F. Noy, M. A. Musen. Ontology Versioning in an Ontology Management Framework. In IEEE Intelligent Systems, 19(4), pp. 6-13, 2004
- [47] W3C, Web Ontology Language <http://www.w3.org/sw/WebOnt>
- [48] M. Quillian. Semantic Memory, in M. Minsky (ed.), Semantic Information Processing, pp 227-270, MIT Press; reprinted in Collins & Smith (eds.), Readings in Cognitive Science, section 2.1
- [49] R. Rada, H. Mili, E. Bicknell, M. Blettner. Development and Application of a Metric on Semantic Nets. In IEEE Transaction on Systems, Man, and Cybernetics, vol. 19(1), pp. 17-30, 1989.
- [50] A. L. Ralescu, A. Faddalla. The Issue of Semantic Distance in Knowledge Representation with Conceptual Graphs, In Proc. of AWOCS 90, pp. 141-142, 1990
- [51] P. Resnik. 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
- [52] C.J. Rijsbergen. A new theoretical framework for information retrieval, In Proc. of the ACM Conference on Research and Development in Information Retrieval, pp. 194-200, Pisa, Italy, 1986
- [53] 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
- [54] M. Sintek, S. Decker. Triple: A Query, Inference and Transformation Language for the Semantic Web. Proc. of the 1st International Semantic Web Conference, ISWC 2002, LNCS 2342, pp. 364-378, Sardinia, Italy, 2002
- [55] 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
- [56] J.F. Sowa. Conceptual structures: Information Processing in Mind and Machine. Addison-Wesley, Reading, Massachusetts, 1984
- [57] W3C, Resource Description Framework, <http://www.w3.org/RDF>
- [58] W3C, RDF Vocabulary Description Language 1.0: RDF Schema, <http://www.w3.org/TR/rdf-schema/>

- [59] D.H. Widyantoro, J. Yen. A Fuzzy Ontology-based Abstract Search Engine and its User Studies. In Proc. of the 10th IEEE Int. Conf. on Fuzzy Systems, pp. 1291-1294, 2001.
- [60] Z. Wu, M. Palmer. Verb Semantics and Lexical Selection. In Proc. of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, 1994
- [61] M. Vargas-Vera, E. Motta, J. Domingue, M. Lanzoni, A. Stutt, F. Ciravegna. MnM: Ontology Driven Semi-automatic and Automatic Support for Semantic Markup. In Proc. of the 13th International Conference on Knowledge Engineering and Management, EKAW2002, LNCS 2473, pp. 379-391, Siguenza, Spain, 2002
- [62] H. Zargayouna, S. Salotti. Mesure de Similarité dans une Ontologie pour l'Indexation Sémantique de Documents XML, In Proc. of Conférence Ingénierie des Connaissances, IC'2004, Lyon, 2004.
- [63] 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



---

Unité de recherche INRIA Sophia Antipolis  
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes  
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique  
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

---

Éditeur  
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)  
<http://www.inria.fr>  
ISSN 0249-6399