

# Conceptual Graphs for Semantic Web Applications

Rose Dieng-Kuntz, Olivier Corby

INRIA Sophia-Antipolis, ACACIA Project  
2004 route des Lucioles, BP 93, 06902 Sophia-Antipolis Cedex, France  
{Rose.Dieng, Olivier.Corby}@sophia.inria.fr

**Abstract.** In this paper, we aim at showing the advantages of Conceptual Graph formalism for the Semantic Web through several real-world applications in the framework of Corporate Semantic Webs. We describe the RDF(S)-dedicated semantic search engine, CORESE, based on a correspondence between RDF(S) and Conceptual Graphs, and we illustrate the interest of Conceptual Graphs through the analysis of several real-world applications based on CORESE.

## 1 Introduction

*"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."* [5]. The W3C works for *"defining standards and technologies that allow data on the Web to be defined and linked in a way that it can be used for more effective discovery, automation, integration, and reuse across applications. The Web will reach its full potential when it becomes an environment where data can be shared and processed by automated tools as well as by people."* [6].

When Tim Berners-Lee presented his vision of the Semantic Web [4], several research communities studied thoroughly how their research field results could contribute to reach this ambitious goal. In particular, researchers working in Knowledge Representation (KR) recognised the potential important role that their KR formalisms could play for representing the ontologies needed in Semantic Web. Object-oriented (OO) representation formalisms, Description Logics (DL) and Conceptual Graphs (CG) were the main candidates to achieve this purpose. The DL community was strongly involved in the definition of the Ontology Web Language (OWL)<sup>1</sup> [14] [33], that W3C recommended in 2004 for representing ontologies. However some researchers of the CG community had also brought their contributions very early, with various strategies. Some researchers adopted CG directly as formalism for representing ontologies and annotations in Semantic Web context: e.g. WebKB [32]. Others preferred to rely on a correspondence between CG and RDF(S)<sup>2</sup> – the language recommended by W3C for describing Web resources [29]: the ACACIA team thus proposed and implemented a translation of RDF(S) into CG and built a semantic search engine, CORESE, based on this correspondence [11] while in [24], the authors sug-

---

<sup>1</sup> <http://www.w3.org/sw/WebOnt>.

<sup>2</sup> Resource Description Framework, <http://www.w3.org/RDF>

gested to use CG as the Ontolingua for allowing the automatic translation of knowledge structures between different KR formalisms, and they described RDF(S) meta-model in CG.

In this paper, we summarise the ACACIA team approach for the Semantic Web and emphasise the role of CG in this approach. Our objective is to show that in the framework of Semantic Web, Conceptual Graphs have enough expressivity for KR and enough reasoning capabilities for real-world applications.

## 2 Corporate Semantic Web Approach

### 2.1 ACACIA Project Evolution

The ACACIA project<sup>3</sup> is a multidisciplinary team that aims at offering methodological and software support for knowledge management (KM), and in particular for building, managing, evaluating and evolving corporate memories.

Historically, in 1992, we were focusing on corporate memories materialised in documents and knowledge bases (KBs). Our main topics were: (1) multi-expertise, (2) “intelligent” information retrieval and (3) management of links between documents and KBs. Therefore we chose CG as privileged KR formalism since it offered reasoning capabilities interesting for intelligent information retrieval, and it seemed a good basis for tackling multi-expertise or for representing a KB associated to texts. The main results of our previous research in CG formalism were the following:

- The CGKAT system integrated a CG-represented ontology extending WordNet and enabled to associate a base of CGs to a structured document. The user could ask queries about either the base of CGs or the document contents and CGKAT could retrieve relevant document elements through a projection of the user’s query on the CG base associated to the document [31].
- The MULTIKAT system offered an algorithm for comparing KBs modelling knowledge of two experts in a given domain. MULTIKAT enabled comparison and merging of two CG ontologies [19] as well as comparison and integration of two CG bases, the integration being guided by different integration strategies [18];
- The C-VISTA and CG-VISTA models enabled to represent viewpoints in CG formalism (both in a CG support and in a CG base) [39].

In 1998, Tim Berners-Lee proposed his vision of the Semantic Web [4]. The main approach suggested was to use an ontology for making explicit semantic annotations on Web resources. According to us, CG was clearly a relevant formalism for representing such ontology and annotations: there was an analogy between on the one hand, a Web document annotated by semantic annotations w.r.t. an ontology and on the other hand, a structured document associated to a base w.r.t. the CGKAT ontology. But W3C had already started to define RDF as the future language for describing Web resources. So, as there was also an analogy between RDF and CG, and as a W3C-recommended language was more likely to be widely adopted by the different

---

<sup>3</sup> <http://www-sop.inria.fr/acacia/>

research communities or industrial companies than CG, our strategy was to rely on this RDF – CG correspondence, so as to take advantage of both the standard feature of RDF and our competence in CG. Therefore, instead of building an RDF-dedicated tool from scratch, we preferred to rely on this RDF – CG correspondence for developing CORESE, an RDF-dedicated search engine based on CG: thus, the first version of CORESE was quickly implemented and tested using the API of Notio [44]. Moreover, our KM approach evolved towards our so-called “Corporate Semantic Web” approach. From research viewpoint, using CORESE as the kernel of our research and applications enables us to check the validity of our hypothesis that CG is a good internal KR formalism for “Corporate Semantic Web” applications.

### 2.2 Corporate Semantic Webs

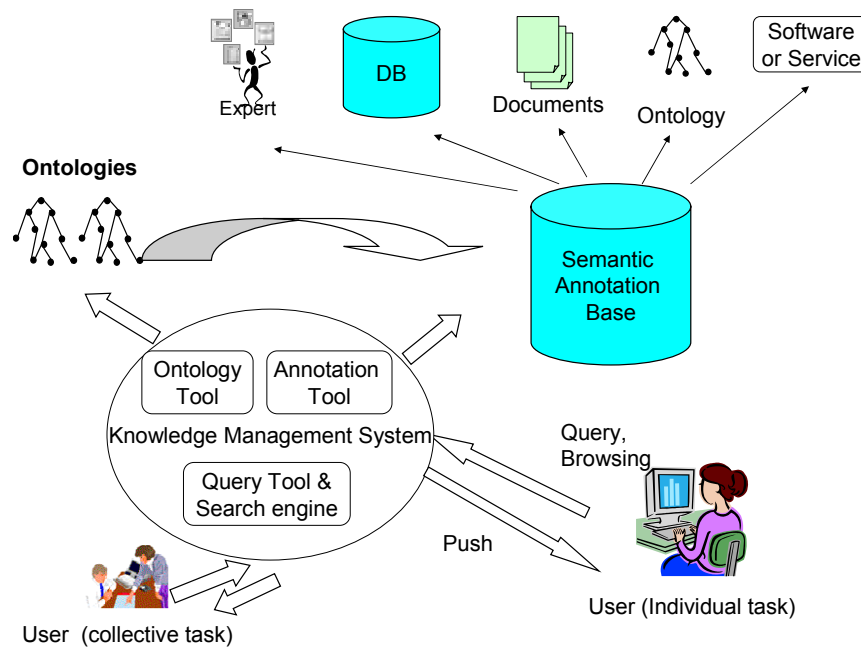


Fig. 1. Architecture of a corporate semantic Web

In 1999, we proposed an approach called « Corporate Semantic Web » approach, that relies on a natural analogy between Web resources and corporate memory resources, and on a generalisation of CGKAT approach. We thus proposed to materialise a corporate memory through:

- *resources*: they can be documents (in various formats such as XML, HTML, or even classic formats), but they can also correspond to people, services, software or programs,

- *ontologies*, describing the conceptual vocabulary shared by one or several communities in the company,
  - *semantic annotations on these resources* (i.e. contents of documents, skills of persons or characteristics of services / software / programs), based on these ontologies.
- with diffusion on the Intranet or the corporate Web.

A Corporate Semantic Web has some specificities w.r.t. the Semantic Web: the fact that an organisation is bounded should allow an easier agreement on a corporate policy, an easier creation of ontologies and annotations, an easier verification of validity and reliability of information sources, a description of more precise user profiles, a smaller scale for the document corpora and for the ontologies.

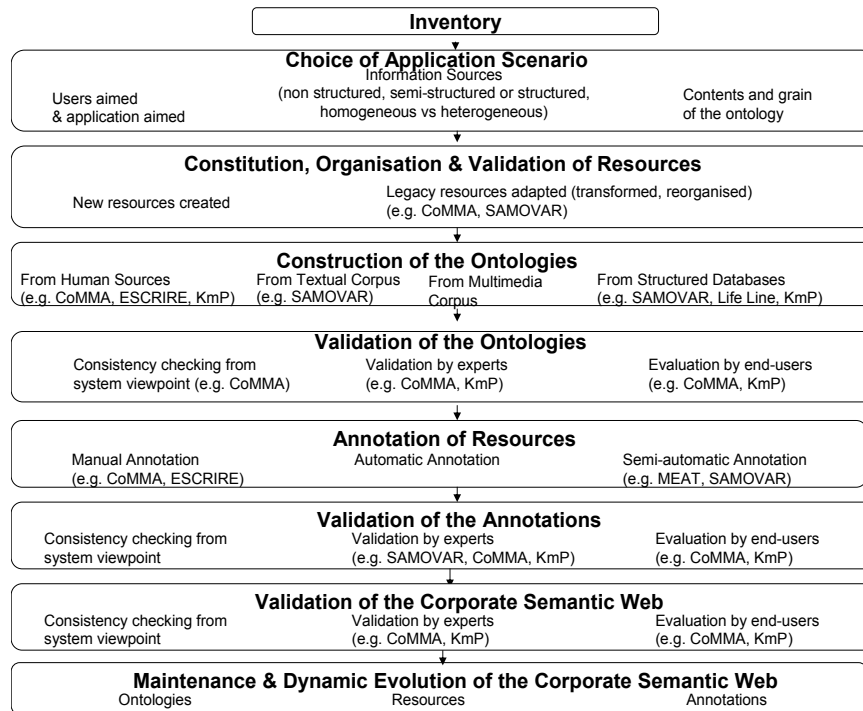


Fig. 2. « Corporate Semantic Web » Approach

Figure 1 shows the architecture of a corporate semantic Web and figure 2 summarises our method for building it, method stemming from our synthesis and abstraction after analysis of all our semantic web applications. In [17], we study thoroughly the components of a corporate semantic Web (resources, ontologies, annotations).

In this paper, we will illustrate this “Corporate Semantic Web” approach by several ACACIA team research results: the CORESE semantic search engine, project memory (SAMOVAR system), distributed memory (CoMMA system), information retrieval from Medline (ESCRIRE system), support to interpretation or validation of DNA micro-array experiments (MEAT project), support to collaborative reasoning in

a healthcare network (Life Line project), support to skills cartography (KmP project). Table 1 summarises their contributions to the ACACIA team research program.

**Table 1.** Contribution of the different projects to ACACIA research program

<b>Research questions</b>	<b>Contributing projects</b>
How to build or enrich an ontology from textual sources?	SAMOVAR
How to build or enrich an ontology from a structured database?	SAMOVAR, Life Line, KmP
What existing KR formalisms are the most relevant for the semantic Web?	ESCRIRE
How to create semantic annotations manually through an editor?	CoMMA
How to create semantic annotations (concepts or relations) semi-automatically from texts?	SAMOVAR, MEAT
How to offer ontology-guided information retrieval?	CORESE, ESCRIRE, SAMOVAR, CoMMA
How to offer approximate reasoning?	CORESE, KmP
How to distribute annotations in a memory? How to distribute query processing among agents? How to use agents and ontologies for information retrieval from a distributed memory?	CoMMA
How to offer scenario-guided, user-centred evaluation of a corporate memory?	CoMMA, KmP
How to offer friendly interfaces (for ontology browsing, querying or result presentation)?	KmP

**Table 2.** Scenarios studied in the different applications

<b>Application Scenario</b>	<b>Contributing project</b>
Project memory	SAMOVAR, Aprobation
Integration of a new employee	CoMMA
Support to technological watch	CoMMA
Experiment memory for a research community	MEAT
Support to cooperative work	Life Line
Skills cartography	KmP

### 3 CORESE Semantic Search Engine

As detailed in [11][12][13], CORESE<sup>4</sup> is a semantic search engine, i.e. an ontology-based search engine for the semantic web: it enables to retrieve web resources anno-

<sup>4</sup> <http://www-sop.inria.fr/acacia/soft/corese>

tated in RDF. CORESE ontology representation language is built upon RDFS, that enables to represent an ontology by a class hierarchy and a property hierarchy. CORESE thus takes into account subsumption links between concepts and between relations when it needs to match a query with an annotation. CORESE ontology representation language also enables to represent domain axioms on which CORESE relies when matching a query with an annotation.

Annotations are represented in RDF and related to the RDF Schema representing the ontology they are built upon. The CORESE query language is also based on RDF; for each query, an RDF graph is generated, related to the same RDF Schema as the one of the annotations to which the query will be matched.

The CORESE engine works on CG internally. When matching a query with an annotation according to their common ontology, both the RDF graphs and their schema are translated into the CG model [45] [10]. Figure 3 summarises the principle of CORESE. Through this translation, CORESE takes advantage of previous work of the KR community on reasoning with CG [41] [42] [37] [3].

### 3.1 RDF(S) and Conceptual Graphs

As stressed in [11][12][13], the RDF(S) and CG models share many common features and a mapping can easily be established between RDFS and a large subset of the CG model. An in-depth comparison of both models was the starting point of the development of CORESE. Both models distinguish ontological knowledge and assertional knowledge. In both models, the assertional knowledge is positive, conjunctive and existential and it is represented by directed labelled graphs. In CORESE, an RDF graph representing an annotation or a query is thus translated into a CG.

Concerning the ontological knowledge, the class (resp. property) hierarchy in an RDF Schema is translated into a concept (resp. relation) type hierarchy in a CG support. RDF properties are declared as first class entities as RDFS classes, just as relation types are declared independently of concept types in a CG support. This common handling of properties makes the mapping of RDFS and CG models relevant, contrarily to OO formalisms where properties must be defined as attributes inside classes.

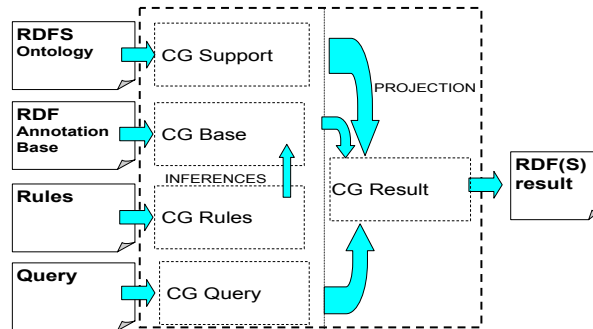


Fig. 3. Principle of CORESE

There are a few differences between the RDF(S) and CG models in their handling of classes and properties but such differences can be easily dealt with:

- In RDF(S), a resource can be instance of several classes while in CG, an individual marker has a unique concept type corresponding to the lowest concept type associated to the instance referred by this individual marker (i.e. a concept is exact instance of only one concept type). The declaration of a resource as an instance of several classes in RDF is translated in CG model by generating the concept type corresponding to the highest common subtype of the concept types translating these classes.
- Similarly, an RDF property can have several domains (resp. ranges), while in CG, a relation type signature is unique. Moreover in RDF, properties are binary while in CG, relations are n-ary. The multiple domains (resp. ranges) constraint of an RDF property is translated into a single domain (resp. range) constraint in CG by generating the concept type corresponding to the highest common subtype of the concept types constraining the domain (resp. range).

The projection operation is the basis of reasoning in the CG model. A query is thus processed in the CORESE engine by projecting the corresponding CG into the CGs obtained by translation of the RDF annotations. The retrieved web resources are those for which there exists a projection of the query graph into their annotation graph.

For example, for the ontology shown in figure 4, the query graph G in figure 5 can be projected on the two annotation graphs G<sub>1</sub> and G<sub>2</sub>. The web resources annotated by these graphs will be found as answers of the G query and will be retrieved by CORESE when processing this query.

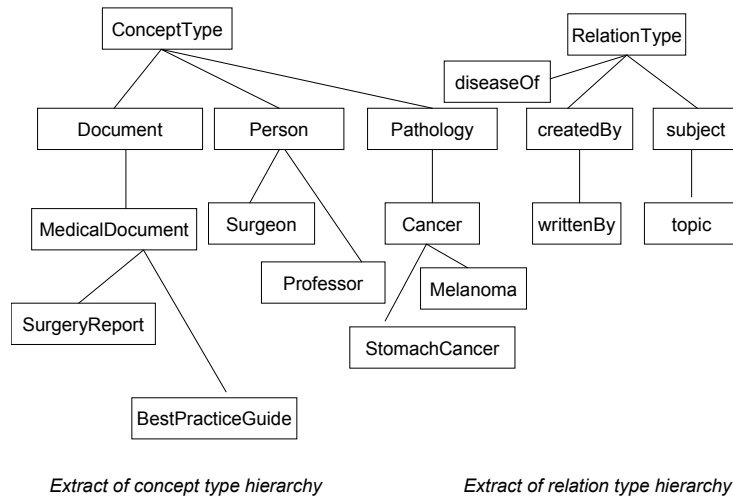


Fig. 4. Example of CG support

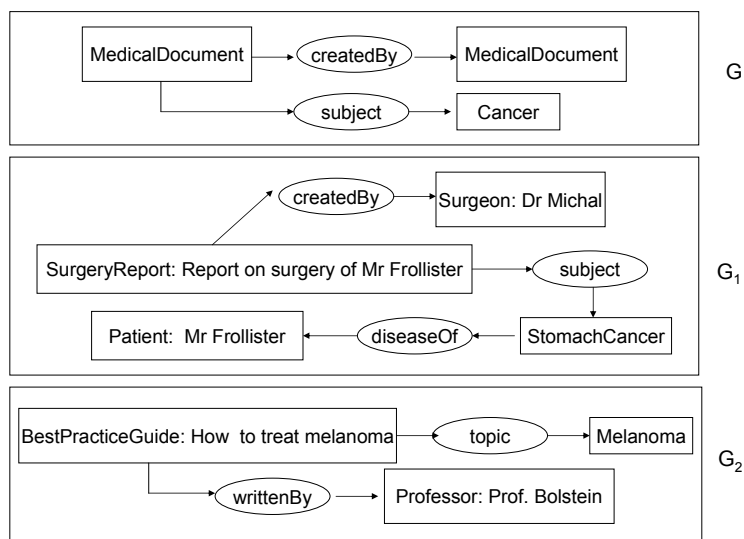


Fig. 5. Examples of CG graphs based on the previous CG support

### 3.2 Domain Axioms

In addition to a concept type hierarchy and a relation type hierarchy, a more expressive ontology can contain domain axioms enabling to deduce new knowledge from existing one. However RDF Schema does not offer such a feature. Therefore an RDF Rule extension to RDF was proposed in [13] and CORESE integrates an inference engine based on forward chaining. The rules are applied once the annotations are loaded in CORESE and before the query processing. Hence, the annotation graphs are augmented by rule conclusions before the query graph is projected on them.

CORESE production rules implement CG rules [41] [42] [37] [3]: a rule  $G_1 \Rightarrow G_2$  (also noted IF  $G_1$  THEN  $G_2$ ) 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 ?p suffers from an allergy to a molecule ?m and if a drug ?d contains this molecule, then this drug must not be prescribed to this patient:

```
IF [Person: ?p] - (allergic_to) - [Molecule: ?m] - (con-
tained_in) - [Drug: ?d]
THEN [Drug: ?d] - (forbidden_to) - [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 specialisation 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 specialise the types of some concepts, to create relations between concepts and to create new individual concepts (i.e. concepts without variable).



*Remark:* In the first implementation of CORESE rule language and rule engine, in order to avoid possible loops during the execution of the forward-chaining engine, we restricted the rule language by the following constraint: no new generic concept could be created in a rule conclusion. However, later applications required the need to use RDF anonymous resources in rule conclusions. So we decided to suppress this restriction, so as to allow the creation of generic nodes for expressing such graphs as:  
 IF [Patient: ?p] - (taken-care-in) - [Hospital: ?h]  
 THEN [Patient: ?p] - (attended-by) - [Doctor: ?d] -  
 (working-in) - [Hospital: ?h].

But the rule engine keeps track of the variable values that led to the triggering of such a rule, so as to avoid to apply twice the same rule with the same variable values.

### 3.3 CORESE Query Language

CORESE had several successive query languages, that evolved through the various applications of CORESE. The last version of the CORESE query language enables to express queries in the form of RDF triples or of a Boolean combination of RDF triples. It is an SQL-like query language for RDF, compatible with the W3C SPARQL RDF query language [38].

The query processor is the CG projection. It relies on the RDF Schema by using subsumption links (*rdf:subClassOf* and *rdfs:subPropertyOf*) and it processes datatype values. The query language can also query the RDF Schema itself (i.e. the CG support). Last, it is able to return the best approximate answers by relaxing types.

For example, the following query enables to retrieve any doctor author of a medical document; the name of this doctor as well as the title of the medical document must be returned in the answer:

```
select ?n ?t where
?p rdf:type lv:Doctor
?p lv:name ?n
?p lv:author ?doc
?doc rdf:type lv:MedicalDocument
?doc lv>Title ?t
```

(lv is the namespace where the classes and properties are defined; as a notation, the variables are characterised by a name starting by an interrogation point.)

This query is translated into the following CG:

```
[Doctor: ?p] - {
- (name) - [Literal: ?n]
- (author) - [MedicalDocument: ?doc] - (Title) -
[Literal: ?t]}
```

### 3.4 Approximate Reasoning

In [12], CORESE approximate reasoning capabilities are detailed. The principle is to calculate the semantic distance of concept types or of relation types in the ontology

hierarchies, so that two brothers are closer at the ontology deepest levels. When the user asks an approximate search about a query, the query projection on the annotation base is performed independently of the subsumption link between concept types (resp. relation types). For each retrieved resource, the distance to the query is computed and only the resources having a semantic distance to the query lower than a given threshold are presented to the user; these results are sorted by increasing distance.

### 3.5 Conclusions

CORESE works with domain ontologies represented through RDF(S) extended with domain axioms (which corresponds to simple CG extended by graph rules). It can process queries expressed in a query language close to SPARQL, and it offers approximate reasoning in case of need. CORESE has been tested with several existing RDF Schemas such as the Gene ontology, IEEE LOM, W3C CC/PP, etc.

Let us now present several approaches illustrating corporate semantic web applications developed in the ACACIA team and based on CORESE.

## 4 Memory of a Vehicle Project: SAMOVAR Project

The first application of CORESE was the SAMOVAR system. The objective of the SAMOVAR project was to capitalise knowledge on the problems encountered during a vehicle design project at Renault. The capitalisation was already initiated at Renault through a database (DB) called “Problem Management System” and describing the problems detected during the validation phases in a vehicle project: in the textual fields of this DB, the participants discussed about these problems, about the possible solutions for solving them, with their constraints and their advantages, as well as the solution finally chosen. But due to the volume of this DB, these textual fields of the Problem Management System constituted an unused information mine.

### 4.1 Methodological Approach

The SAMOVAR ontology was made up by using both (1) a corpus of texts constituted by the textual fields of the Problem Management System and (2) a structured database describing the nomenclature of all parts that can be used in a vehicle project at Renault. The SAMOVAR approach [25] consisted of the following steps:

- Make an inventory, through discussions with the experts, for analysing the organisation structure, a vehicle project development, and the corporate data available.
- Apply a *natural language processing tool* (more precisely, the *term extractor* Nomino) on the textual fields of the Problem Management System.
- Analyse the *structure of the obtained candidate terms* for determining their *linguistic regularities* to be used in *heuristic rules* allowing to enrich the ontology.

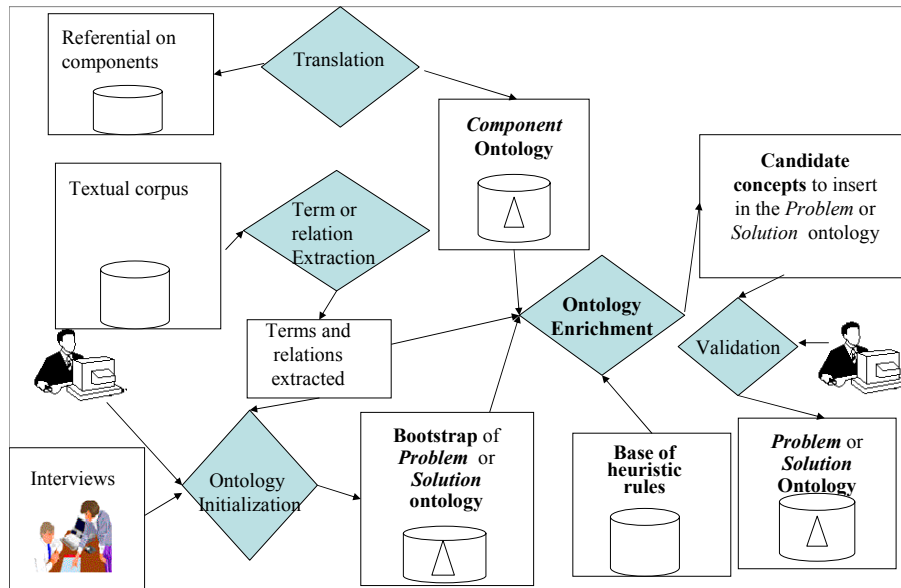


Fig. 6. Method of construction of the SAMOVAR ontology [25]

- Build an *ontology with several components: Problem, Part, Project, Service*. The *Part* sub-ontology was built automatically from a part nomenclature, available in the company. The *Problem* sub-ontology, manually initialised from the analysis of the discussions with the experts and of the Nomino-obtained candidate terms, was then enriched semi-automatically using the previous heuristic rules (cf. figure 6).
- *Annotate semi-automatically descriptions of problems with this ontology.*
- Use the *semantic search engine CORESE* for information retrieval (once the ontology represented in RDFS and the annotations in RDF): in particular, CORESE enables to retrieve problem descriptions satisfying given features.

#### 4.2 Example of Query

The user searches all the problems of Geometry encountered on the *Driver\_cockpit\_crossmember*:

```
[GeometryProblem: ?pb] - (concerning) - [Part:
Driver_cockpit_crossmember]
```

which corresponds in CORESE query language to:

```
?pb rdf:type GeometryProblem
?pb concerning Driver_cockpit_crossmember
Driver_cockpit_crossmember rdf:type Part
```

Using the projection, the SAMOVAR system is able to find not only descriptions of problems that are annotated by « *GeometryProblem* concerning *Driver\_cockpit\_crossmember* » but also those annotated by « *CenteringProblem* (resp. *InterferenceProblem* or *ClearanceProblem*) concerning *Driver\_cockpit\_crossmember* », since *CenteringProblem*, *InterferenceProblem* and *ClearanceProblem* are subtypes of *GeometryProblem*.

The SAMOVAR system is an illustration of a corporate semantic Web, implementing a project memory through resources constituted by problem descriptions annotated semantically w.r.t. the ontology. This ontology represented in RDFS then allows CORESE to guide information retrieval in this memory.

**Table 3.** Summary of SAMOVAR project

System	SAMOVAR: vehicle project memory system
Context or scenarios	Memory of problems encountered during a vehicle project
Domain	Automotive sector
Company	Renault car manufacturer
Scope of Semantic Web (SW) Approach	Corporate semantic web on an internal web
Resources	Database of problem descriptions
Information sources	Human experts + Structured DB on part nomenclature describing all the parts used for vehicle manufacturing + Textual corpus constituted by textual comments of the database of problem descriptions
Ontology	792 concepts and 4 relations in the SAMOVAR ontology.
Annotations	4483 problem descriptions annotated
Expert validation	Validation by some experts at Renault
Typical user's query	"Find all problems of a given (resp. any) type that occurred on a given (resp. any) part of the vehicle in a given (resp. any) project"
Used reasoning	Classic projection + Browsing of the concept type hierarchy
CORESE functions used	CORESE past query language (consisting of RDF + variables and operators)
End-user evaluation	Evaluation by some experts at Renault
Research progress	Use of linguistic tools for semi-automatic construction / enrichment of an ontology and of semantic annotations

### 4.3 Conclusions

SAMOVAR was the first application of CORESE in a real world concrete problem. It proved the interest of ontology-guided information retrieval. It was based on CORESE first query language (that was similar to RDF with variables). SAMOVAR illustrated how legacy systems can be used as sources of the memory. It offered an example of knowledge acquisition from heterogeneous information sources: texts, databases, experts. It enabled us to study thoroughly the problem of creation of an ontology and of semantic annotations from textual sources, such ontology and annotations being then represented in RDF(S) in order to serve as inputs of CORESE.

The corpus-based approach for building the ontology and the annotations is generic. It relies on the two following stages: (1) Analyse the structures of the candidate terms obtained from a term extractor, in order to determine their regularities that will be used to create heuristic rules; (2) Use these rules in order to enrich the ontology.

The SAMOVAR method was generalised in a method for construction of a memory of problems encountered during a project of design of a complex system [25].

After SAMOVAR, several applications enabled to improve CORESE query language, CORESE Graphical User Interface (both for browsing the ontology, for querying and for presenting the answers).

## 5 Distributed Memory: COMMA Project

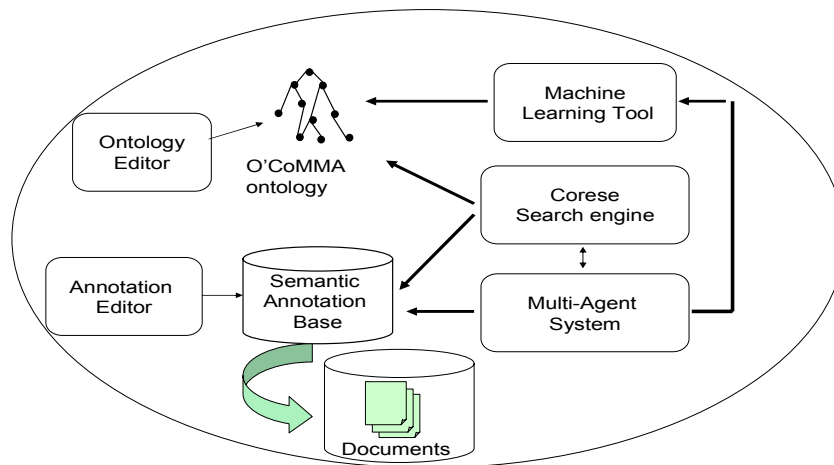


Fig. 7. Architecture of the CoMMA system

The management of a distributed memory was studied thoroughly in the CoMMA IST project that enabled to develop a multi-agent system in order to manage a distributed memory [23]. This memory was constituted by a corporate semantic Web made of:

- O’CoMMA corporate ontology and user models,
- resources constituted by the corporate members or by documents useful for a new employee or handled for technological watch,
- semantic annotations on these resources.

The CoMMA system [23] includes five main components: (1) the O’CoMMA ontology, (2) the multi-agent knowledge management system, (3) the CORESE semantic search engine, (4) machine learning algorithms enabling to improve relevance of retrieved documents according to the users’ interest centres, (5) graphical interfaces.

### 5.1 The O’CoMMA Ontology

The method of construction of the O’CoMMA ontology [23] relied on the following phases: knowledge elicitation from human sources, manual terminological analysis phase, ontology structuration, validation by experts, ontology formalisation in RDFS. The O’CoMMA ontology is structured in three levels [23]:

- A high level comprising abstract types of concepts, very reusable but not very usable by end-users in their daily work, and having to be hidden during the ontology browsing by the end-user: e.g. *Entity* and *Situation* types of concepts.
- An intermediate level comprising concept types useful for the processed scenarios of corporate memory and for the considered domains and reusable for other scenarios and similar domains: e.g. concepts related to the aspects *Document*, *Organisation*, *Person*, *Telecommunications* and *Building*, useful for support to integration of new employees and technological watch in telecommunications (cf T-NOVA and CSELT) and in building industry (CSTB) but generic enough to be reusable for other similar scenarios and domains.
- A specific level comprising concepts typical to the considered enterprise and therefore very useful for the end-users but not very reusable outside this enterprise: e.g. concepts also related to the aspects *Document*, *Organisation*, *Person*, *Telecommunications* and *Building*, but typical of the company considered (for example, *guide of the route of the new employee*, *technological watch file* or *thematic referent*).

The ontology construction method and the ontology structure are reusable. The reusability of the O’CoMMA ontology was proven through the Aprobatiom project where the O’CoMMA ontology was extended by concepts useful for building a project memory in building sector for CSTB.

Let us note that this O’CoMMA ontology was not only aimed at annotating documents and people, but also aimed at being browsed by human users and used by software agents (helping in the search of information in a distributed memory).

## 5.2 Examples

Here is an example of query in CoMMA for integration of a new employee:

*“Find the titles and authors of all documents aimed at newcomers”.*

In the new CORESE query language:

```
select ?t ?auth
?doc rdf:type comma:Document
?doc comma:title ?t
?auth comma:author ?doc
?doc comma:target ?targ
?targ rdf:type comma:Newcomer
```

An example of query in CoMMA for technological watch is:

*“Find all the persons for which the documents found on fire detection systems are interesting”.*

```
select ?pers
?doc rdf:type comma:Document
?doc comma:Subject comma:FireDetectionSystem
?pers rdf:type comma:Person
?pers comma:HasForWorkInterest
comma:FireDetectionSystem
```

An example of use of rule is:

*“If a team includes a person professionally interested in a subject, the team can be considered as professionally interested in the subject and the documents on the subject are relevant for being sent to the team.”*

```
IF [Team:?team] - (Includes) - [Person:?pers] - (Has-
ForWorkInterest) - [Topic:?theme]
[Document:?doc] - (Subject) - [Topic:?theme]
THEN [Team:?team] - (HasForWorkInterest) -
[Topic:?theme]
[Document:?doc] - (RelevantFor) - [Team:?team]
```

In CORESE new rule language, it is expressed as:

```
IF ?team rdf:type comma:Team
?team comma:Includes ?pers
?pers rdf:type comma:Person
?pers comma:HasForWorkInterest ?theme
?theme rdf:type comma:Topic
?doc rdf:type comma:Document
?doc comma:Subject ?theme
THEN ?team comma:HasForWorkInterest ?theme
?doc comma:RelevantFor ?team
```

**Table 4.** Summary of CoMMA project

System	CoMMA, a multi-agent system for corporate memory management
Context or scenario	Support to integration of a new employee and to technological watch
Domain	Telecommunications and Building sector
Companies	T-NOVA (Deutsche Telecom), CSELT (Italian Telecom) and CSTB (French Centre for Science and Technique of Building)
Scope of Semantic Web Approach	Distributed corporate semantic Web, both on internal and external web, with agents used for search and for push towards the end-user
Resources	Documents describing the organisation, the corporate members and the domain
Information sources	Human experts + Document manual analysis
Ontology	The O'CoMMA ontology comprises 472 concepts, 80 relations and 13 levels of depth
Annotations	Annotations on documents or on people in an organisation
Expert validation	Validation by T-NOVA, CSELT, CSTB experts
Typical User query	“Find the document of this type speaking about this subject or interesting for this kind of user” “Find the persons of this type that have these characteristics”
Used reasoning	Classic projection + Use of global objects + Graph rules + Ontology browsing
CORESE functions used	CORESE past query language + Use of rules
End-user evaluation	Evaluation by end-users at T-NOVA, CSELT and CSTB
Research progress	O'CoMMA ontology represented in RDFS + DRDF(S) a CG-based extension of RDF(S) for expressing contextual knowledge + Concept learning from RDF annotations + Annotation and query distribution + Multi-agent system for distributed information retrieval

### 5.3 CORESE Extensions Designed for CoMMA.

In the CoMMA project, CORESE was extended with:

- a global graph representing global knowledge true for all annotations, so that this global graph would be joined to each annotation during query processing,
- the expression and processing of reflexive, symmetric or transitive properties,



- extensions of the query language with type operators enabling to specify more precisely the type of the requested resources,
- rule graphs and a forward-chaining rule engine.

#### **5.4 DRDF(S): Extensions of RDFS for Expressing Contextual Knowledge**

One lesson of the CoMMA project was that RDF(S) was insufficient to express contextual knowledge such as definitions or axioms: therefore we took inspiration of CG for proposing DRDFS (Defined Resource Description Framework), an extension of RDF(S) with class, property and axiom definitions [16]. DRDFS more generally enables to express contextual knowledge on the Web. The RDF philosophy consists of letting anybody free to declare anything about any resource. Therefore the knowledge of by whom and in which context an annotation is stated is crucial. DRDF(S) enables to assign a context to any cluster of annotations. The representation of axioms or class and property definitions is just a particular use of DRDFS contexts. DRDFS is a refinement of the core RDFS which remains totally compliant with the RDF triple model. More precisely, DRDFS is an RDF Schema extending RDFS with new primitives. This extension of RDFS is inspired of CG features. DRDFS is built upon the CORESE mapping established between RDF(S) and the Simple CG model. A DRDFS context corresponds to a CG; a DRDFS class (resp. property) definition corresponds to a concept (relation) type definition; a DRDFS axiom corresponds to a graph rule.

#### **5.5 Learning Concepts from RDF Annotations**

We offered a method for classifying documents and capturing knowledge by learning concepts from the RDF annotations of documents [15]. Our approach consists of extracting descriptions of the documents from the RDF graph gathering all the document annotations and then forming concepts by generating the most specific generalisation of these descriptions for each possible set of documents. In order to deal with the intrinsic exponential complexity of such a task, the concept hierarchy is built incrementally by increasing at each step the maximum size of the RDF document descriptions extracted from the whole RDF graph gathering all the annotations.

#### **5.6 Annotation Distribution and Query Distribution**

In [22], algorithms enabling the CoMMA multi-agent system to allocate and retrieve semantic annotations in a distributed corporate semantic web were proposed. The agents used the underlying graph model when they needed to allocate a new annotation and when solving distributed queries. As the agents dedicated to ontology or to the annotations used the CORESE API, they used CG reasoning.

### 5.7 Conclusions

CoMMA project was a very important step for improving CORESE. This application also showed the interest of RDF(S) but also its insufficiencies. From CG viewpoint, it enabled to propose multiple improvements of RDF(S) inspired of CG: graph rules, expression of contexts, concept learning.

## 6 Comparison of KR Formalisms: The ESCRIRE Project

**Table 5.** Summary of ESCRIRE project

System	EsCorServer, system for querying annotated abstracts of Medline database
Context or scenarios	ESCRIRE project, cooperative project among INRIA teams
Domain	Biology
Company	None
Scope of SW Approach	External, open Web
Resources	Abstracts of Medline database on genetics
Information sources	A researcher (computer scientist) that had taken part in projects representing knowledge on biology and genetics
Ontology	ESCRIRE ontology has 28 classes (enabling to describe 163 genes as instances of these classes) and 17 relations
Expert validation	No expert available for validation
Typical User query	“Find the articles describing interactions where a given gene acts as target and where the instigator is either this other gene or that other gene”
Used reasoning	Classic projection + Processing of “OR queries” + Use of global objects (for representing global knowledge)
CORESE functions used	Another CORESE query language a la <i>select from where</i> + Use of “OR queries” + Use of rules for handling reflection, symmetry and transitivity properties and inverse relations of relations
End-user evaluation	No evaluation by end-users
Research progress	Techniques of translation between a pivot language and conceptual graphs via RDFS + Virtual document composition

The ESCRIRE project [1] [34] was a cooperative project launched in 1999 - before the existence of OWL - among three INRIA teams, in order to compare DL, CG and OO languages for representing the contents of documents and for querying about them. It relied on the annotation of abstracts of Medline DB on genetics. It enabled to answer “OR 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?*”

For comparing the three languages, chosen test articles were annotated in order to constitute a test base to be queried. An XML-based pivot language was defined, for describing the ESCRIRE ontology, expressing queries and describing the answers. So each team had to translate this pivot language into the target language (DL, CG, OO KR). ACACIA developed the translator from ESCRIRE language to CG.

**Example of ESCRIRE rule:**

```

IF [Interaction: ?int1] - {
  - (promoter) - [Gene: ?x]
  - (target) - [Gene: ?y]
  - (effect) - [Inhibition] }
  [Interaction: ?int2] - {
  - (promoter) - [Gene: ?y]
  - (target) - [Gene: ?x]
  - (effect) - [Inhibition] }
THEN [Gene: ?x] - (mutually-repressive-with) - [Gene: ?y]

```

As a conclusion, the ESCRIRE project illustrates how using a translation from different KR formalisms to this pivot language enables then to work with CGs.

## 7 Experiment Memory in Biology: The MEAT Project

**Table 6.** Summary of MEAT project

System	MEAT, a memory of experiments of biologists on DNA micro-arrays
Context	Memory of DNA-micro-array experiments
Domain	Biology
Company	IPMC
Scope of SW Approach	External, open web
Re-sources	Scientific articles useful for interpretation or validation of results of DNA micro-array experiments
Information sources	Human experts
Ontology	UMLS semantic network that contains 134 concept types and 54 relations, and is linked to millions of terms via UMLS metathesaurus.
Expert validation	Validation of extracted relations and of generated annotations, by biologists of IPMC

Typical user query	<p>“ Find all the articles asserting a given (resp. any) relation between a given biological entity (gene, protein...) and another biological entity”</p> <p>“ Find all the articles asserting any relation between a given gene and any disease”</p>
Used reasoning	Classic projection
CORESE functions used	CORESE new query language + Use of rules + Use of approximate reasoning
End-user evaluation	Evaluation by biologists of IPMC
Research progress	Automatic extraction of relations from texts + Automatic generation of RDF annotations

The MEAT project aims at building a memory of experiments in DNA micro-array, and at supporting biologists in their interpretation and validation of the results of their experiments, through analysis of semantically annotated Medline scientific articles. We consider the UMLS semantic network as a general ontology for the biomedical domain: the UMLS hierarchy of semantic types can be regarded as a hierarchy of concept types and the terms of the metathesaurus can be considered as instances of these concept types.

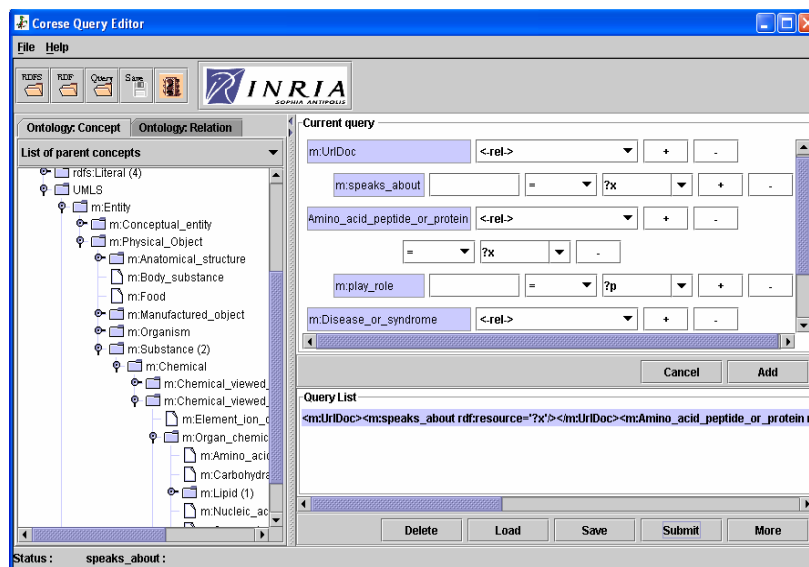


Fig. 8. CORESE interface showing the ontology and the query structure [28]

The MEAT-Annot system [28] relies on analysis of scientific articles through linguistic tools in order to generate automatically RDF annotations (not only concepts as in SAMOVAR but also relations among concepts). In the text, MEAT-Annot recognises terms corresponding to UMLS concepts and then it uses a relation extraction grammar for extracting automatically relations linking terms denoting UMLS concepts. It generates an RDF annotation that is stored, once validated by the biologist.

The annotation base is then used by CORESE for retrieving the articles possibly relevant for answering the biologist’s query and supporting him/her in the interpretation of a DNA micro-array experiment.

Figure 8 shows the CORESE interface for asking a query “Find the URL of documents speaking about an amino-acid-peptide-or-protein playing a role in a disease” and figure 9 shows the answers to this query.

As a conclusion, the MEAT project illustrates the reuse of an existing ontology and the use of linguistic tools for generating RDF annotations.

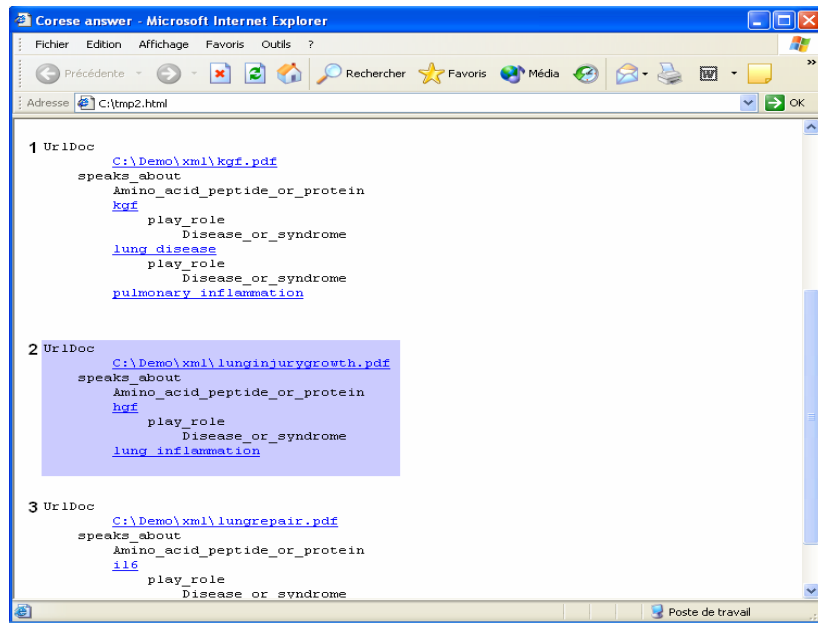


Fig. 9. CORESE answers for the previous query [28]

## 8 Support to Cooperative Work in a Healthcare Network: The Life Line Project

Table 7. Summary of Life Line project

System	A Virtual staff in the framework of the “Life Line” project
Context or scenario	Support to cooperative reasoning of members of a health care network
Domain	Medicine
Company	Nautilus (specialised in marketing medical software)
SW Scope	Medical semantic Web among distributed health partners
Resources	Medical documents: patient records, guide of best practices
Info. sources	A medical database translated automatically into RDF(S)

Ontology	Nautilus ontology with 26432 concepts and 13 relations
Expert validation	Validation by our industrial partner Nautilus
Typical User query	“Find the past sessions of virtual staff where a given therapy was chosen for the patient and indicate the arguments that were given in favour of his solution” “Find a past session of virtual staff where the patient suffered from a given symptom and indicate the disease diagnosed and the therapy protocol decided”
Used reasoning	Classic projection
CORESE functions used	CORESE past query language
End-user evaluation	Evaluation by our industrial partner
Research progress	Translation of a database into an RDF(S) ontology + Integration of an ontology with SOAP and QOC graphs

The Life Line project [20] [40] aims at developing a knowledge management tool for a care network.

### 8.1 Nautilus Ontology

We developed a translator of the Nautilus medical DB from its internal format towards RDF(S), by using an approach of "reverse engineering" relying on the analysis of the principle of coding of this DB. The obtained Nautilus ontology was then extended by a classification of medical professions and by classes relevant for a health-care network, and represented in RDFS. It was used in a tool called "Virtual Staff", that allows the members of a healthcare network to visualise their reasoning when formulating diagnosis assumptions or when making decisions of therapeutic procedures [20]. This application corresponds to an organisational semantic Web dedicated to a medical community cooperating in the context of a health care network.

In the Virtual Staff, the dependencies between the various diagnostic and therapeutic hypotheses are represented through a graph using the concepts defined in the Nautilus ontology. The doctor reasons by linking the health problems to the symptoms, the clinical signs and the observations in order to propose care procedures.

The Virtual Staff relies on the SOAP model (Subjective, Objective, Assessment, Plan) used by the medical community [46]. In this model, the S nodes describe current symptoms and clinical signs of the patient, the O nodes describe analyses or observations of the physician, the A nodes correspond to the diseases or health problems of the patient, and the P nodes correspond to the procedures or action plans set up in order to solve the health problems.

Sometimes, the doctor may need to visualise all the possible solutions and the arguments in their favour or against them. The QOC model (Question Options Criteria) [30] is useful for support to decision-making. In this model, a question Q corresponds to a problem to solve. To solve the question Q, several Options are thought out, with, for each option, the criteria in its favour and the criteria against it. Two types of questions are possible for the Virtual Staff: (1) *Diagnosis of a pathology*:

Which pathology explains the clinical signs of the patient? (2) *Search of a prescription*: Which action plan will enable to treat the diagnosed pathology?

In the Virtual Staff, the SOAP graph enables to visualise the patient’s record and in phase of decision, QOC graphs enable to choose between pathologies or between action plans. Using the Nautilus ontology, the system can propose a list of possible concept types to help the users to build SOAP and QOC graphs [20]. Table 8 indicates the concept types among the subtypes of which each category of node must be chosen.

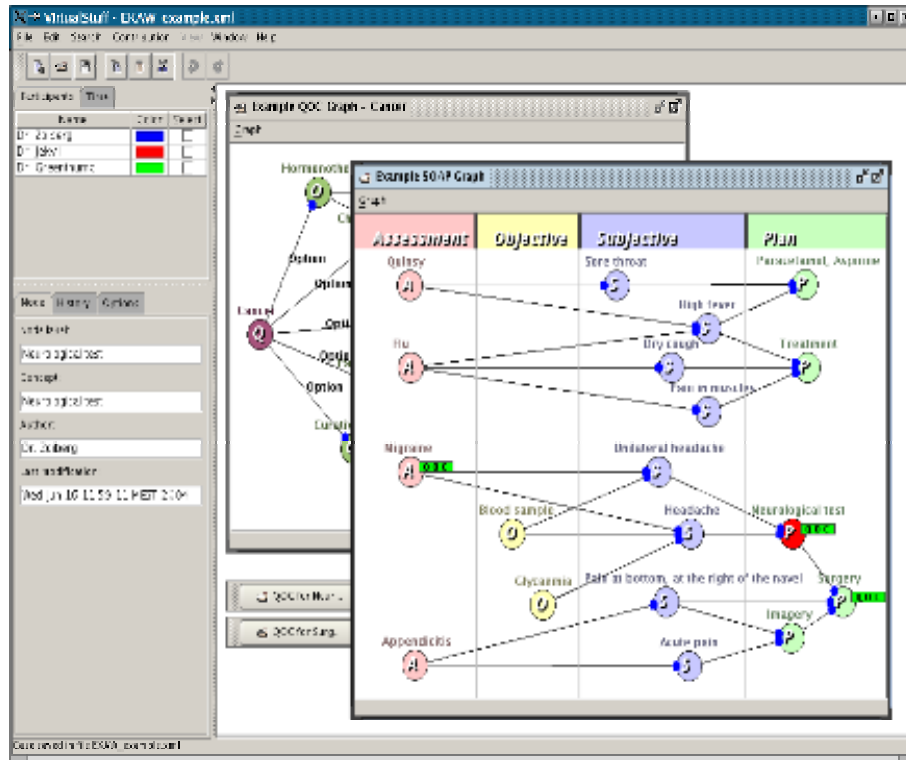


Fig. 10. Interface of the Virtual Staff [40]

Table 8. Nodes of Virtual Staff graphs and Nautilus ontology concept types

Node Category	Possible concept types
S node in a SOAP graph	Symptom
O node in a SOAP graph	LaboratoryTest, PathologicalAgent or ForeignBody
A node in a SOAP graph	Malformation, Pathology or PsychologicalProblem
P node in a SOAP graph	Treatment or DiagnosticGesture
Option in a QOC graph	Pathology or Treatment
Criterion in a QOC graph	Symptom, LaboratoryTest, Pathology or Treatment

The arcs between the nodes correspond to relations among concepts:

[Symptom] - (has\_for\_cause) - [Pathology]  
 [Pathology] - (has\_for\_consequence) - [Symptom]  
 [Pathology] - (confirmed\_by) - [LaboratoryTest]  
 [Pathology] - (treated\_by) - [Treatment]

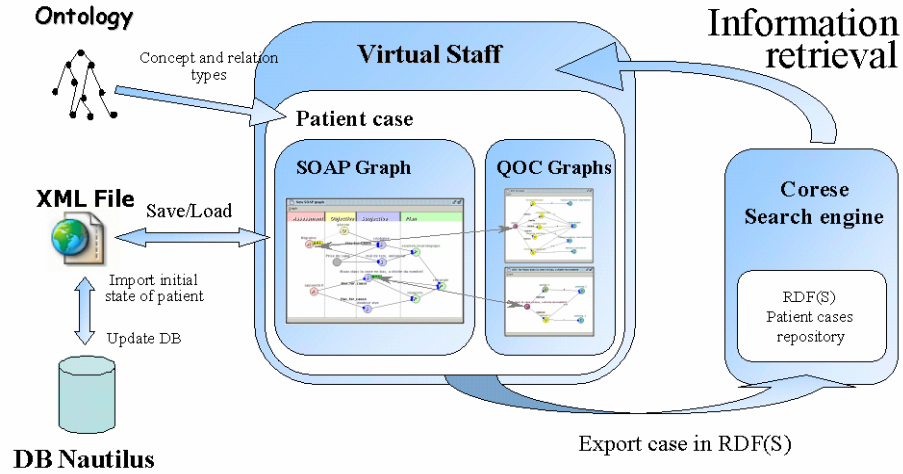


Fig. 11. Architecture of the Virtual Staff [40]

The arcs between the nodes of a QOC graph can be interpreted by « Question has-solution Option » or by « Option has-positive-criterion Criterion » or by « Option has-negative-criterion Criterion ».

## 8.2 Examples of Queries

*“Find the past sessions of virtual staff where a given therapy was chosen for the patient and indicate the arguments that were given in favour of this solution”*

```
[VirtualStaff Session: ?session] - (has-QOC) -
[QOCGraph: ?graph] - (chosen-therapy) - [Therapy:
?therap] - (positivecriteria) - [ConceptNautilus: ?cri-
terion]
```

*“Find a past session of virtual staff where the patient suffered from a given symptom and indicate the disease diagnosed and the therapy protocol decided”*

```
[VirtualStaff Session: ?session] - (has-SOAP) - [SOAP-
Graph: ?graph] - (has_symptom) - [Symptom: ?symp] -
(has_for_cause) - [Pathology: ?patho] - (treated_by) -
[Therapy: ?therap]
```



### 8.3 Conclusions

The Virtual Staff illustrates an application where CORESE helps to retrieve the relevant subclasses for editing the SOAP and QOC graphs and enables to retrieve information on past virtual staff sessions stored in RDF(S).

## 9 KMP (Knowledge Management Platform) Project

The KmP project<sup>5</sup> is an RNRT project for skills cartography of Sophia Antipolis firms in telecommunications. It is so far the largest application using CORESE and its new features such as new query language, new rule language, approximate reasoning. CORESE is the kernel of the semantic web server developed for KmP. Design of the KmP system was characterised by a user-centred, participative approach with a special care for GUI interfaces developed in SVG (see figure 12).

**Table 8.** Summary of KmP project

System	KmP system for cartography of skills in telecommunications for Sophia Antipolis firms
Context or scenarios	1) Increase visibility of Telecom Valley 2) Cartography of skills in order to enhance inter-firms cooperation 3) Support to cooperation between industrial companies and research laboratories
Domain	Telecommunications
Companies	Companies of Telecom Valley: Amadeus, ATOS Origin, Coframi, Elan IT, Eurecom, France Télécom R&D, HP, IBM, INRIA, Philips Semiconductors, Qwan System, Transiciel, UNSA-CNRS
Semantic Web Approach	Semantic Web server
Resources	Documents describing collective competencies of firms or of research laboratories
Information sources	Human experts
Ontology	The KMP ontology comprises 1136 concepts and 83 relations and 15 levels of depth
Expert validation	User-centred participative design and user-centred validation
User query	“Find one company having this type of skills” “Show me the poles of complementary competencies in this field” “Show me the clusters of similar competencies in this field”

<sup>5</sup> <http://www-sop.inria.fr/acacia/soft/kmp>

Used reasoning	Classic projection + Approximate reasoning
CORESE functions used	CORESE past and new query language + Use of rules + Use of approximate reasoning
End-user evaluation	User-centred evaluation
Research Progress	Competence ontology represented in RDF(S) + New ontological distance for approximate reasoning + SVG graphical user interfaces

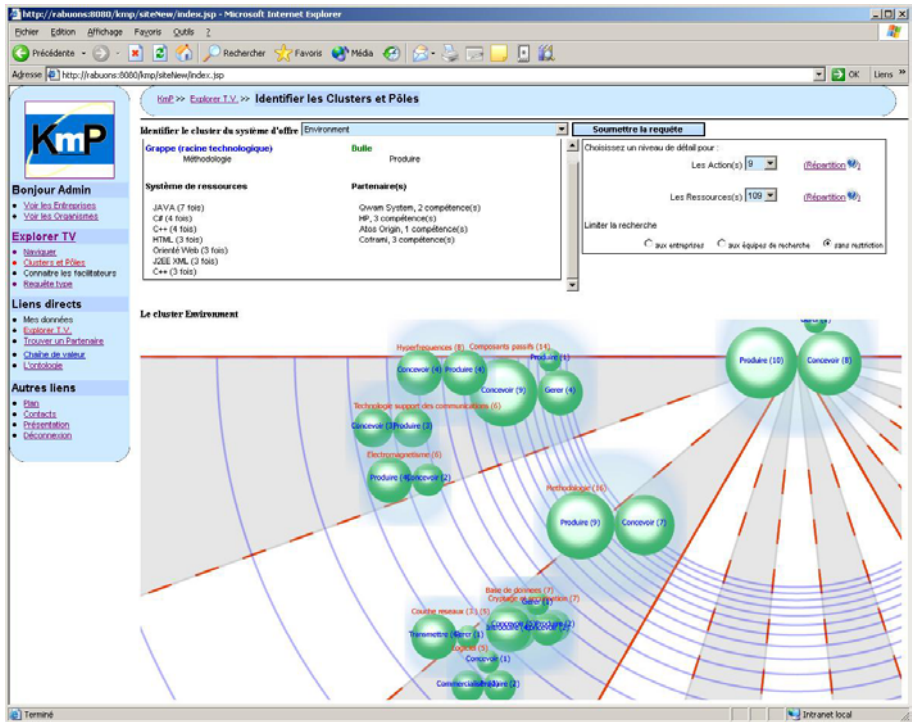


Fig. 12. Interface of KmP showing clusters of complementary competences (credit: Gronnier)

## 10 Conclusions

### 10.1 Evolution of CORESE

Let us summarise the evolution of CORESE through all these applications:

- In SAMOVAR, we used CORESE first query language and we used a linguistic tool for generating both the ontology concept types and the annotations from texts.
- In CoMMA, CORESE was enhanced by representation of graph rules enabling to express property relations such as reflexivity, symmetry, transitivity, inverse rela-

- tion and to express domain rules; a forward-chaining engine was developed for using such rules in order to enlarge the annotation base.
- In the ESCRIRE project, a new *select ... from ... where* query language was introduced, and the “or queries” were processed, as well as reflexive, symmetric, transitive and inverse relations.
  - In Life Line and MEAT projects, the new query language was used.
  - In KmP, a semantic web server was developed, as well as new interfaces with SVG and the new CORESE query language and approximate reasoning were used.

CORESE can be compared to query languages or tools dedicated to RDF such as RQL [27], Triple [43], SquishQL [35], Sesame [7], or tools such as [21]. But CORESE is the only RDF(S)-dedicated engine that offers both inference rules and approximate search and the only RDF(S)-dedicated engine relying on CG. A few researchers use CG in semantic Web context: a link between CG and RDF(S) meta-model is proposed in [24]; CG is also the KR formalism used by Ontoseek (a content-based information retrieval system for online yellow page and product catalogs) [26] and by WebKB (an ontology server and Web robot) [32]. With respect to the synthesis on CG applications [9], CORESE is a prototype that has been used in ten applications and by several dozens of users. It proves a real usefulness of CG reasoning for corporate semantic webs. Moreover, the applications MEAT, Life Line and KmP are not restricted to a single organisation and are relevant for the (open) semantic Web.

## 10.2 Evaluation Issues

In the previous applications, evaluation was carried out from three viewpoints [17]:

- *Checking from system viewpoint*: we can check whether the system actually realises the functions it was intended for, and whether it satisfies classic criteria of software quality (performance, robustness, etc). We performed such a verification partially in most of our applications.
- *Validation from expert viewpoint*: the experts must validate the correctness, quality and relevance of the knowledge included in the system (mainly the ontology and the semantic annotations): we performed such validation for all our applications except ESCRIRE project where no expert was available.
- *Evaluation from end-user viewpoint*: the end-users evaluate whether the system is useful and usable according to them. We performed a very detailed, scenario-guided, end-user evaluation in two projects: CoMMA and KmP.

In our past applications such as SAMOVAR or CoMMA, we did not organise experiments for calculating systematically information retrieval recall and precision. However, during SAMOVAR evaluation, the Renault experts appreciated the ontology-guided retrieval offered by CORESE: for example, CORESE enabled them to discover similarity between different problems described in the description problem base, similarity that could not have been found automatically using classic SQL queries since it depended on the semantic contents of textual fields and not on structured fields of the DB. As the SAMOVAR ontology had been created from these textual fields, it enabled to capture this similarity between different problems (in the same project or in different projects). Likewise, during the CoMMA project, T-NOVA

considered CORESE as retrieving far more relevant documents than its previous search engine and in a more efficient way. More generally, in most of our past applications, the evaluation was ergonomics-based, qualitative and based on human assessment. However, we performed a quantitative evaluation in some projects not detailed here (e.g. ontology alignment [2]). we are now preparing quantitative evaluation experiments in the MEAT project and in a technological watch application.

### 10.3 Discussion

The main advantages of CG in the applications previously described are:

- *The expressiveness at the level of KR*: Sowa’s CGs “a la Pierce” [45] enable to represent first-order logic; in CORESE, we translated RDF(S) into simple CG [10]; SG family [3] is more expressive than RDF(S): for example, it enables to represent n-ary relations. Even though some expressions of OWL cannot be represented in SG family, however, in most of our applications, RDF(S) extended with rules was sufficient for expressing what was needed, which means that simple CGs with graph rules – as studied in the SG family – were sufficient.
- *The power at the level of reasoning*: CORESE takes advantage of the classic CG projection (that we optimised for dealing with large ontologies), and offers forward-chaining on graph rules. The approximate search based on the calculation of an ontological distance requires the implementation of an “approximate projection” [12]. It would be interesting to study its complexity as for the SG family [3].
- *The power at the level of queries*: CORESE query language is close to SPARQL specification [38] – that should become a recommendation of W3C.

As a further work, we intend to study systematically (1) what can be expressed with CGs (KR and queries) that could not be expressed with the other languages and (2) what cannot be expressed in CGs while it could be expressed with other languages.

One frequent argument in favour of CG formalism is its greater readability thanks to its graphical notation. However in CORESE, it is RDF(S) that is handled and visible externally and not CGs, since RDF(S) is the W3C-recommended language, that plays the role of exchange language between CORESE and external world (e.g. the user or other applications). However, adequate graphical user interfaces can represent RDF(S) through graphs (as CGs): in the MEAT project, our validation interface for the biologists presents the generated RDF annotations through user-friendly graphs.

We distinguish the internal language handled by the semantic web tools (i.e. RDF(S) and OWL), the language handled by the developer (in our case, RDF(S) and CG), and last, what can be seen by the end-user through user interfaces.

Our choice of RDFS is mainly historical since the first implementations of CORESE preceded the emergence of OWL. But as, in 2004, W3C recommended OWL for representing more complex ontologies, our strategy can be either to continue to focus on RDF(S) – i.e. privilege applications needing only simple ontologies since most existing Web-available ontologies are still in RDF(S) – or to evolve towards OWL. The ACACIA team is studying CORESE extensions to handle annotations represented in OWL Lite. Handling simple ontologies should be an advantage in

the framework of the open semantic Web, where heterogeneity and scalability issues are more crucial than for corporate semantic Webs.

#### 10.4 Towards the Semantic Web

In [17], we emphasise the most important research topics needed to be performed on construction, management and evolution of the different elements of a corporate semantic Web: automation, heterogeneity, evolution, evaluation and scalability.

In addition to the topics described in this paper, the ACACIA team also studies thoroughly RDF(S) ontology alignment ontology [2], multi-viewpoint ontologies [39] support to technological watch with ontology-guided search on the external Web [8] and support to e-Learning (a new KM scenario) using semantic Web technologies. As the results of this research will be used to extend CORESE search engine, this research on RDF(S) indirectly works with CGs. We hope to have shown through this paper that more than 10 years of research in CG are useful for contributing to reach Tim Berners-Lee's vision of the Semantic Web.

#### Acknowledgements.

We deeply thank all our colleagues of the ACACIA team for their enthusiasm in our collective work on corporate memories and on corporate semantic Webs. We are specially grateful to Khaled Khelif, Nicolas Gronnier, and Marek Ruzicka for the hard-copies of MEAT-Annot, KmP and Virtual Staff integrated in this paper. CORESE was mainly designed and developed by Olivier Corby, with a support of Olivier Savoie for implementation. Rules were added to CORESE by Alexandre Delteil and Catherine Faron-Zucker, that also proposed the DRDF(S) language and the algorithm for concept learning from RDF annotations. Joanna Golebiowska developed the SAMOVAR system, Fabien Gandon the O'CoMMA ontology and the annotation and query distribution algorithms in the CoMMA multi-agent system, Carolina Medina-Ramirez developed the EsCorServer server of biological knowledge, Khaled Khelif the MEAT-Annot system for generation of annotations in the framework of the MEAT project initiated by Leila Alem on biochip experiment memory, Frédéric Corby and David Minier reconstituted the Nautilus ontology from the medical database, David Minier and Marek Ruzicka implemented the Virtual Staff, Nicolas Gronnier and Cécile Guigard developed the KmP system specified by Alain Giboin, Fabien Gandon and Olivier Corby. Bach Thanh-Le developed the Asco algorithm for RDF(S) ontology alignment, Cao Tuan-Dung an algorithm for supporting technological watch guided by an ontology, Laurent Alamarguy an algorithm for relation extraction from texts, Sylvain Dehors semantic Web techniques for eLearning, and Luong Hiep-Phuc studies the evolution of a corporate semantic Web. Many thanks also to the previous members of the ACACIA team, since their past research was an important step towards corporate semantic Webs: Philippe Martin developed the CGKAT system linking structured documents, WordNet ontology and CG, Sofiane Labidi, Krystel Amergé, Laurence Alpay contributed to knowledge acquisition from multiple experts, Stéphane Lapalut to reasoning on CGs, Christophe Cointe, Nada Matta, Norbert Glaser and Roberto Sacile to research on CommonKADS, Kalina Yacef to research on eLearning, Stefan Hug developed the MultiKat system for

CG ontology comparison and Myriam Ribi re proposed models for integrating viewpoints in a CG ontology.

## References

1. Al-Hulou, R., Corby, O., Dieng-Kuntz, R., Euzenat, J., Medina Ramirez, C., Napoli, A. and Troncy, R. Three knowledge representation formalisms for content-based manipulation of documents, KR'2002 Workshop on "Formal Ontology, Knowledge Representation and Intelligent Systems for the World Wide Web". Toulouse, France, April 2002.
2. Bach, T.-L., Dieng-Kuntz, R., Gandon, F. On Ontology Matching Problems (for building a corporate Semantic Web in a multi-communities organization). ICEIS 2004. Porto, 2004.
3. Baget, J.-F., Mugnier, M.-L. Extensions of Simple Conceptual Graphs: The Complexity of Rules and Constraints, JAIR, 16:425-465, 2002
4. Berners-Lee, T. Semantic Web Road Map. <http://www.w3.org/DesignIssues/Semantic.html>, September 1998.
5. Berners-Lee, T., Hendler, J., Lassila, O. The Semantic Web. Scientific American, May 2002.
6. Berners-Lee T., Miller E. The Semantic Web lifts off. ERCIM News No. 51, Oct. 2002.
7. Broekstra, J., Kampman, A., van Harmelen, F. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. Proc. of ISWC'2002, pp. 54-68, Sardinia, Italy, 2002.
8. Cao, T.-D., Dieng-Kuntz, R., Fi s, B. An Ontology-Guided Annotation System For Technology Monitoring, IADIS Int. Conf. WWW/Internet 2004, Madrid, October 2004.
9. Chein, M., Genest, D. CGs Applications: Where are we 7 years after the first ICCS. In ICCS'2000, Darmstadt, Germany, August 14-18, Springer LNAI 1867, p. 127-139.
10. Chein, M., Mugnier, M.-L. Conceptual graphs: fundamental notions. RIA, 6(4): 365-406. 1992.
11. Corby, O., Dieng, R., H bert, C. A Conceptual Graph Model for W3C Resource Description Framework, In Proc. of ICCS'2000, Darmstadt, 2000, LNAI 1867, p. 468-482.
12. Corby, O., Dieng-Kuntz, R., Faron-Zucker, C. Querying the Semantic Web with the CORESE Search Engine. ECAI'2004, Valencia, August 2004, IOS Press, p. 705-709.
13. Corby, O., Faron, C., CORESE: A Corporate Semantic Web Engine. WWW'2002 Workshop on Real World RDF and Semantic Web Appl., Hawaii, USA, May 2002. <http://paul.rutgers.edu/~kashyap/workshop.html>
14. Dean, M., Schreiber, G. (eds). OWL Web Ontology Language Reference. W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/owl-ref/>
15. Delteil A., Faron C., Dieng R. Learning Ontologies from RDF Annotations. In Proc. of IJCAI'01 Workshop on Ontology Learning, Seattle, USA, August 2001.
16. Delteil A., Faron C., Dieng R. Extensions of RDFS Based on the Conceptual Graph Model. ICCS'2001, LNAI 2120, Springer-Verlag, pp. 275-289, Stanford, CA, USA, 2001.
17. Dieng-Kuntz, R. Corporate Semantic Webs. To appear in D. Schwartz ed, Encyclopaedia of Knowledge Management, Idea Publishing, July 2005.
18. Dieng, R., Hug, S. MULTIKAT, a Tool for Comparing Knowledge from Multiple Experts. Proc. of ICCS'98, Montpellier, 1998, Springer-Verlag, LNAI 1453
19. Dieng, R., Hug, S. Comparison of "personal ontologies" represented through conceptual graphs. Proc. of ECAI'98, Wiley & Sons, p. 341-345, Brighton, UK, 1998.
20. Dieng-Kuntz, R., Minier, D., Corby, F., Ruzicka, M., Corby, O., Alamarguy, L. & Luong, P.-H. Medical Ontology and Virtual Staff for a Health Network, Proc. of EKAW'2004, Whittlebury Hall, UK, October 2004, p. 187-202.
21. Eberhart, A. Automatic Generation of Java/SQL Based Inference Engines from RDF Schema and RuleML. Proc. of ISWC'2002, pp. 102-116, Sardinia, Italy, 2002.

22. Gandon, F., Berthelot, L., Dieng-Kuntz, R., A Multi-Agent Platform for a Corporate Semantic Web, AAMAS'2002, p. 1025-1032, July 15-19, 2002, Bologna, Italy.
23. Gandon, F., Dieng-Kuntz, R., Corby, O., Giboin, A. Semantic Web and Multi-Agents Approach to Corporate Memory Management, Proc. of the 17th IFIP World Computer Congress IIP Track, p. 103-115, August 25-30, 2002, Montréal, Canada.
24. Gerbé O., Mineau G. W. The CG Formalism as an Ontolingua for Web-Oriented Representation Languages. ICCS'2002, Borovetz, July 2002, Springer, p. 205-219.
25. Golebiowska, J., Dieng, R., Corby, O., Mousseau, D. Building and Exploiting Ontologies for an Automobile Project Memory, K-CAP, Victoria, Oct. 2001, p. 52-59.
26. Guarino, N., Masolo, C., Vetere, G. Ontoseek: Content-based access to the Web. In IEEE Intelligent Systems, vol. 14(3), pp. 70-80, 1999.
27. Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., Scholl M. RQL: a declarative query language for RDF. In Proc. of WWW'2002, Honolulu, pp. 592-603.
28. Khelif, K., Dieng-Kuntz, R. Ontology-Based Semantic Annotations for Biochip Domain, EKAW'2004 Workshop on Application of Language and Semantic Technologies to support Knowledge Management Processes, UK, October 2004, <http://CEUR-WS.org/Vol-121/>
29. Lassila, O., Swick, R. R. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recomm., February 22, <http://www.w3.org/tr/rec-rdf-syntax/>, 1999.
30. Maclean, A., Young, R., Bellotti, V., Moran T.: Questions, Options, and Criteria: Elements of a Design Rationale for User Interfaces. IJHCI, 6(3/4):201-250. 1991.
31. Martin, P. CGKAT: A Knowledge Acquisition and Retrieval Tool Using Structured Documents and Ontologies. ICCS'97, 1997, Springer, LNAI 1257, p. 581-584.
32. Martin, P., Eklund, P. Knowledge Retrieval and the World Wide Web. IEEE Intelligent Systems, 15(3):18-25, 2000.
33. McGuinness, D. L., van Harmelen, F. (eds) OWL Web Ontology Language 57. Overview, W3C Recommendation, February 10, 2004. <http://www.w3.org/TR/owl-features/>
34. Medina Ramirez, R. C., Dieng-Kuntz, R., Corby, O. Querying a heterogeneous corporate semantic web: a translation approach, Proc. of the EKAW'2002 Workshop on KM through Corporate Semantic Webs. Sigüenza, Spain, October 2002.
35. Miller, L., Seaborne, A., Reggiori, A. Three Implementations of SquishQL, a Simple RDF Query Language. In Proc. of ISWC'2002, 2002, pp. 423-435, Sardinia, Italy.
36. Mineau, G. W. A First Step toward the Knowledge Web: Interoperability Issues among Conceptual Graph Based Software Agents, Part I. Proc. of ICCS'2002, p. 250-260.
37. Mugnier, M.-L. Knowledge Representation and Reasonings Based on Graph Homomorphism. ICCS'2000, Darmstadt, August 2000, Springer 1867, p. 172-192.
38. Prud'hommeaux E., Seaborne A., SPARQL Query Language for RDF, W3C Working Draft, 17 February 2005, <http://www.w3.org/TR/2005/WD-rdf-sparql-query-20050217/>
39. Ribière, M., Dieng-Kuntz, R. A Viewpoint Model for Cooperative Building of an Ontology, ICCS'2002, Borovets, July 2002, Springer LNAI 2393, p. 220-234.
40. Ruzicka, M., Dieng-Kuntz, R., Minier, D. Virtual Staff - Software Tool for Cooperative Work in a Health Care Network INRIA Research Report RR-5390, November 2004.
41. Salvat E. Theorem Proving Using Graph Operations in the Conceptual Graph Formalism, In Proc. of ECAI'98, pp. 356-360, Brighton, UK, 1998.
42. Salvat É., Mugnier M.-L. Sound and Complete Forward and Backward Chaining of Graph Rules. ICCS '96, Sydney, 1996, Springer, LNAI 1115, p. 248-262.
43. Sintek, M., Decker, S. Triple: A Query, Inference and Transformation Language for the Semantic Web. Proc. of ISWC'2002, pp. 364-378, Sardinia, 2002.
44. Southey, F. and Linders, J. G., Notio - A Java API for developing CG tools, ICCS'99, 1999.
45. Sowa, J. Conceptual Graphs: Information Processing in Mind and Machine. Reading, Addison Wesley, 1984.
46. Weed, L. D. The Problem Oriented Record as a Basic Tool in Medical Education, Patient Care and Clinical Research. Ann Clin Res 3(3):131-134. 1971.

47. Wielemaker J., Schreiber G., Wielinga B. Prolog-Based Infrastructure for RDF: Scalability and Performance. ISWC'2003, pp. 644-658, 2003.
48. Zhong J., Zhu, H., Li, J., Yu, Y. Conceptual Graph Matching for Semantic Search, ICCS'2002, pp. 92-106, Borovets, July 2002.