SAMOVAR - Setting up and Exploitation of Ontologies for Capitalising on Vehicle Project Knowledge

Joanna Golebiowska

Equipe Acacia, Inria Sophia Antipolis, 2004, Route des Lucioles BP 93, 06 902 Sophia Antipolis,
Joanna.Golebiowska@Inria.fr,
& TPZ D12 138, sce 12113, 860, quai de Stalingrad,
92109 Boulogne, Joanna.Golebiowska.Inria@Renault.fr

Abstract This article presents SAMOVAR (Systems Analysis of Modelling and Validation of Renault Automobiles) SAMOVAR is a tool for capitalising knowledge in the field of automobile and is based on the development and exploitation of corporate memory. It is based on several ontologies which structure knowledge and it uses the search mechanisms of CORESE (Conceptual Resource Search Engine) to exploit such ontologies.
After describing the origin and the objectives aimed by SAMOVAR, we explain how the core of the tool - the set of diverse ontologies- was developed. Finally we present how the different actors of the validation process will be able to use it.
Key-words Ontology, knowledge extraction, terminology, information retrieval, XML, vehicle validation

1 Introduction

The field of SAMOVAR is the process of prototype validation during a vehicle project. This process is intrinsically complex and raises many problems. These problems frequently slow down the shortening of the cycle, due to the necessity of repeating validations, with the resulting delay and costs.

A close observation of validation shows that part of the failure is due to loss of information and of experience gained. The objective of SAMOVAR is to improve the exploitation of this information and to make it available for future projects. Useful data exists in the form of text. It is therefore necessary to find suitable techniques and tools, in this case linguistic techniques.

1.1 Context

The product development cycle of automobile is made of numerous repetitive sub cycles (design/development/validation - of short and/or long duration). The whole cycle is punctuated by milestones and vague prototypes which mark the production of more or less complex successive models and prototypes. In the course of a vehicle project, validations are carried out, during which the testing department checks that parts or functions come up to the requirements of the product specifications.

Thus, the quality of smoothness of the dashboard, the noise of a car door being shut the behaviour of the car on cobble stones, or even its resistance to high or low temperatures are tested. These validations are spread throughout the vehicle project and done successively by the testing department right from the most elementary functions to the final synthesis test. The project begins with the tests linked to the engineering centre according to the parts validated and finishes with the tests on performance, speed and crash.

Validation reveals discrepancies with specifications. Such problems are documented in a unique data management system (Problem Management System, hereafter PMS)[1], from detection of a problem till its resolution. This system uses a data base with the information necessary to the process of problem management: notably information on the actors present and above all the description and comments on the problems which have arisen.

1.2 Problems processing

The appearance of problems engenders supplementary costs and lengthens duration of projects. Solutions are therefore envisaged. One, aims at exploiting information contained in the PMS in order to use the PMS not only as a problem management system but also as a source of information.

The PMS can be considered as a mine of information, thanks to the textual fields of the base which are particularly rich and under-exploited. The players express themselves freely describing problems, the various solutions proposed, or the constraints in carrying out such solutions. This base can therefore be considered as a record or even the memory of a project.

Furthermore, there are other sources of information, such as the official company records or the numerous local bases of the testing department. It would be useful to cross reference this information with the contents of the PMS.

The idea is therefore to propose a means of retrieving, structuring and rendering re-usable this large quantity of information for projects. Current projects have expressed needs linked to information search done during validations. The needs were, in particular, for similarity in incidents, detection of any correlation or dependence with other incidents and so use existing solutions within the same or even different projects. « Manual » spot reading done by

[1] *Problem Management System hereafter PMS*

certain departments have shown that in view of the volume of information available, it is preferable to use automatic means.

Some information is relatively simple to retrieve. However, this is not the case for the textual data of PMS. The vocabulary used in such comments is broad and varied – A given term (existing in official references) frequently has different names according to the department or even the phase reached in the project. Therefore, our objective will be to detect a suitable semantic term, to file same according to the process of the validation and to link it with all the variations encountered. The problem deals with retrieving terms (and the relationships between them if possible) and their structure. In the first phase, we dealt with the tools to retrieve candidate terms: Lexter (Bourigault 1994), Nomino, Ana (Enguehard 1993).

With regard to the acquisition of semantic relationships, there exists several approaches for acquiring semantic information (based on the exploitation of syntactical contexts : Gerfenstette 1994, or the use of the lexical-syntactical patterns : Hearts 1992, Désclès & Jouis 1993) but few tools (Coatis 1997 for causality relationships, Cameleon 1997 for hypnonyms and meronyms). Once the terms have been retrieved they must be structured while managing their diversity. For the setting up of ontologies Terminae (Biébow 1997, 1999) seems to be an interesting approach. Terminae proposes a methodology and an environment for the constitution of ontologies thanks to text analysis. The method is based on a study of the occurrences of terms in a corpus in order to extract the conceptual definition and the environment helps the user in his modelling task by checking the characteristics of a new concept and by proposing potential family knot.

Lexiclass [ASSADI 98] offers an interesting approach for building a regional ontology from technical documents. This tool enables the classification of syntagms extracted from a corpus, so as to help the knowledge engineer to discover important conceptual fields in the domain. Lexiclass coupled with Lexter, carries out a syntagm classification from Lexter according to the terminological context of the terms (more specially, syntactic dependencies which give information on the semantic proximity of syntagms.

1.3 SAMOVAR's contribution

The heart of SAMOVAR relies on the PMS textual fields.

We propose a means to facilitate and increase current exploitation of PMS information in order to be able to effectively make use of the knowledge acquired on projects.

In concrete terms with SAMOVAR we propose to structure this knowledge allowing «intelligent» search to be done. Taking directly exploitable sources as a starting point, the different databases of the company, we have built up several ontologies offering different viewpoints on the validation process: problems, projects, services, parts. After having primed our base, it will be completed

progressively, with the elements from the PMS textual data using NLP tools (Nomino[2] and others). After, we propose to note the problems with the ontological terms. Finally we propose a means to facilitate access to the base of the problems, and thus carry out guided search through the ontology set up by using the Corese[3] platform.

2 SAMOVAR ontologies

The SAMOVAR base is composed of 4 ontologies, each dedicated to the description of a precise field :

- *Component* Ontology - This is based on the official company references, corresponding to the functional segmentation of a vehicle by sub components. It is enriched by additional information from the textual fields, enabling for example management of vocabulary in all its diversity (frequently departments use their own vocabulary) ;
- *Problem* Ontology - This contains problem types and is built up semi-automatically from a manually activated core from field texts taken from the problem management system. It reflects the different types of problems observed in the problem management system ;
- *Service* Ontology - This corresponds to the services cross referenced with those of the company organization (Management and profession) and is completed by PMS information. This ontology gives an added overall point of view on the problems ;
- *Project* Ontology - This reflects the structure of a project and is made up of knowledge acquired during a project vehicle, following interviews carried out with different actors on the project.

An ontology is the hierarchy of specialization of n levels.

We explicitly kept the links between the elements of an ontology and its sources (that is to say the textual fields of the PMS) in order to be able to retrieve the whole information at any given time.

3 Example of ontology construction

The ontologies were built in two phases. If the data which we needed already exist in an external data base, the ontology is primed with the contents of this base. If not, the core is primed manually. After the ontology is enriched progressively with information from texts.

---

[2] *See Linguistic Techniques set up for ontology construction*

[3] *Conceptual Resource Search Engine, see Use of the Corese platform*

Thus for the Parts ontology, we built the core ontology with elements retrieved from local vehicle project bases. The ontology is presented as a hierarchy of n levels linked by specialisation relationships. It is currently structured in three sub-sets corresponding to three (among other existing) possible segmentations of a vehicle project. Each presents a point of view according to which it is possible to retrieve the parts. These points of view correspond to possible regrouping of components :

- by section (segment) that is to say as per progress on the assembly line,
- by architectural area - that is to say as per the volume dealt with in a vehicle,
- by perimeter of a parts manager.

Such segmentation is hardly generic and can vary according to projects.

These elements were retrieved directly from bases, *via* the LotusDomino tool, towards the (RDFS) format which is the Corese format (see Use of Corese) The ontological elements thereafter served to note down problems from the chosen project vehicle base. After the first successful tests on a sample of problems, the process will be applied to the whole of the base content.

The primed ontology thus obtained was then completed with information from texts on problems. For that we generally start by using existing two linguistic tools (Nomino, Cameleon), in order to obtain a »raw» corpus, which we refined later with the help of our own tools (see Linguistic Techniques carried out).

4 Linguistic techniques used to build an ontology

The second phase of the building of an ontology required the use of tools and techniques adapted to textual material. The ontology was started off with data which were directly exploitable taken from various references of the company. It is further enriched by information from PMS texts. The proceedings are further amplified for the Problem ontology: after studying the textual fields we manually set up a core ontology, which we completed after with additional elements obtained thanks to interviews. Later on, this was completed with elements from PMS texts.

4.1 Ontology building

We used the Nomino results as a starting point for our ontology.

Nomino takes a textual corpus as input and produces in output a lexical group (nouns, complex noun units - CNU, additional complex noun units - ACNU, verbs, adjectives, adverbs). The (A)CNU are series of structured terms : nominal groups or prepositional groups (manipulation effort, cyclical whining sounds, door seals, non closing, lack of stiffness, difficulty in assembly etc.)

Furthermore the (A)CNU contain problem 'indicator' terms : bad, problem of , difficulty, fault, impossible ...

We exploited the structural regularity of (A)CNU as well as the problem indicator terms to build heuristic rules which will allow a semi automatic feeding of the ontology.

The objective is to correctly link each relevant CNU to one (or several) primed ontologies. In concrete terms it is necessary to find the family knot to which the CNU (or one of the component terms) can be linked.

4.2 Cinematics of the process

The input of the system is the Nomino output, the Problem and Components ontology, and the heuristic base. To find the family knot the system analyses the CNU to see with which rule it can be matched. The rules represent the possible combinations between the elements of the Components and Problems ontologies stated in the texts. A rule is presented as a series of categories. To each category information in the form of characteristics (for example Problem type to indicate that the element is part of the Problem ontology, Components type for an element of Components ontology etc.) The term type is determined by the ontology to which it belongs.

The rules were established by analysis of texts and with the help of the tools used for text processing.

Example of a Nominal Group and the corresponding rule :

BRUIT DE FROTTEMENT DU VOLANT PENDANT SON REGLAGE EN HAUTEUR

Nom[type=Problème,n=i] Prep[lemme=« de »] Nom[type=Problème,n=i+1] ;[4]

The rule matches the NG, recognises the first term as a noise (also figures in the Problem Ontology) and proposes to link the second noun in The problem ontology, as a son of the term Noise. In the following case the rule matches the name of the part and proposes to link the first term as a Problem :

BROUTEMENT DU BRAS-BALAI AR SUR PPP3

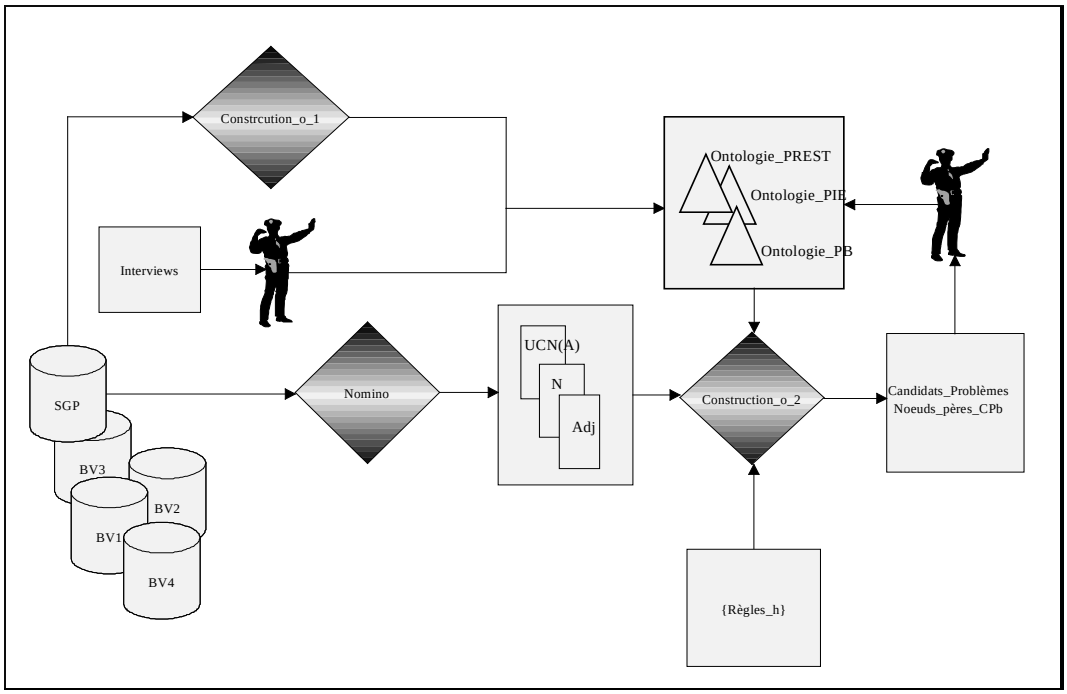
Nom[type=Problème] Prep[lemme=« de »||lemme=« sur »||lemme=« sous »] **Nom[type=Pièce]** ;[5]

The output gives candidate terms to put in the ontology, and proposals for places where they may be attached. The user validates each candidate and decides if the place for insertion in existing hierarchy is correct.

---

[4] *RUBBING NOISE ON THE WHEEL DURING ITS HEIGHT ADJUSTMENT*
Noun[type=Problem, n=i] Prep[lemme= »of »] Noun [type=Problem,n=i+1]

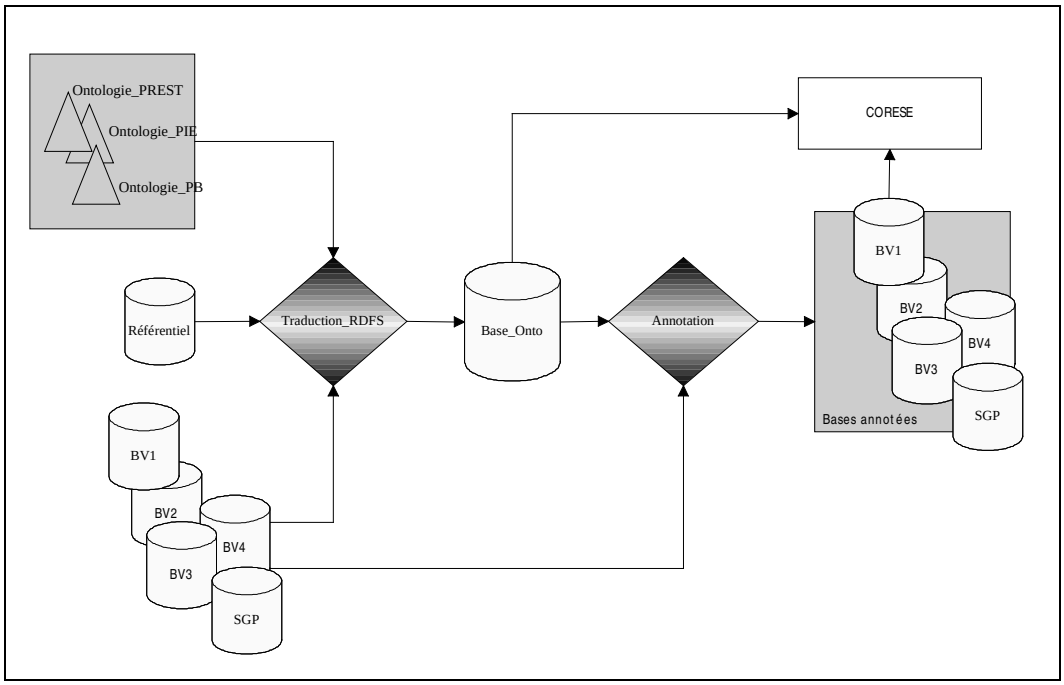
[5] *JUDDERING OF THE REAR SWEEP ARM ON PPP3*
Noun[type=Problem] Prop[lemme=»of »]||lemme= »on »]||lemme= »under »] Noun[type=Part]



Setting up of ontologies of Samovar 1

At the end of the process, a parser generates a RDF version of the ontology for Corese[6], and the notes of the PMS problems are updated. This latter phase is being worked out.

[6] *See Use of Corese platform*



Preparation of the Samovar base for Corese

4.3 Retrieval of the relationships between elements

We plan to use Cameleon to obtain additional information concerning the relationship between the ontological elements.

Cameleon provides a validation environment for semantic relationships. It makes explicit such relationships by relying on the terms obtained from Nomino and by exploiting its own predefined marker base. The relationships thus made explicit are then proposed to a human operator for validation.

The first tests that we carried out with our corpus showed that the Camelon predefined base of markers is too general - Cameleon provides a generic marker base associated with classic hyperonymy and meronymy relationships. It is likely we will need to enlarge this marker base and implement specific relationships for our corpus.

5 Use of the Corese platform

The ontologies set up are used to make notes on documents (in our case : problems from the PMS) for Corese[7] (Corby 2000).

[7] *Conceptual Resource Search Engine*

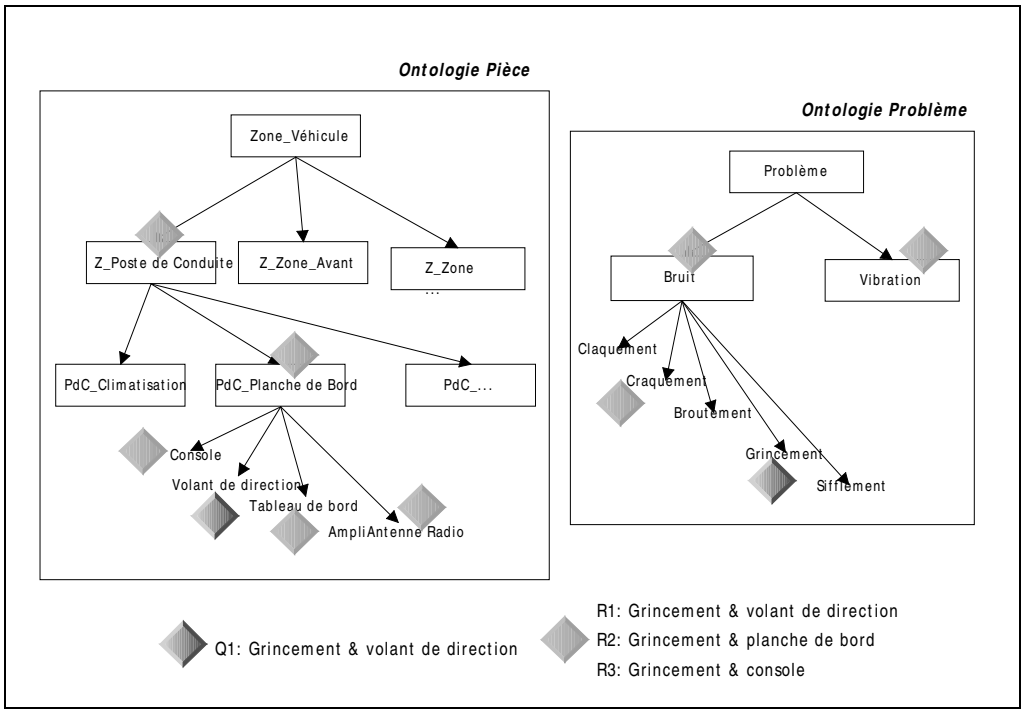
The Corese platform proposed by the Acacia team (at Inria Sophia-Antipolis), implements an RDF/RDFS processor based on the conceptual graph (CG) formalism (Sowa 1984). Corese uses RDF and RDFS to express and exchange document meta-data. It proposes a mechanism of querying and inference based on the formalism of conceptual graphs. It may be compared to a search engine which enables inferences on the RDF statements by translating them into CGs and the CGs to RDF.

Corese relies on 3 languages : RDF, RDF Schema and CG. The general idea of RDF is to enable the description of the content of documents through semantic annotations and to use these RDF statements to search for information. The annotations are based on an ontology which can be described and shared thanks to RDF Schema. Thus, inside a community, it is possible to specify concepts and their relationship in the ontologies, to note the community documents using these ontologies and to use the annotations for search and navigation.

Corese translates the class and characteristics of RDFS towards concepts and the relationships with CGs. This enables us to make queries in the RDF/CG base. A query is presented in the form of an RDF statement which is translated towards a graph which in turn is then projected on the CG base in order to isolate the graph which matches. The graph results are then translated back to the RDF. The mechanism of the projection takes into account the hierarchy and the specialised relationship mentioned in the RDF Schema and translated into CG vocabulary.

To exploit Corese we translate the ontologies in the form of RDF(S). After that, we index the problems of the PMS base with these ontologies, respecting the XML syntax . After these two stages, it is possible to carry out information retrieval. The results of the user's query take into account not only the initial terms of the query but the links modelled in the different ontologies.

In the example, the user is looking for cases involving *steering & wheel squeaks* (Q1 - *Grincement & volant de direction*). Following a successive route through the ontologies from generalisation to specialisation, the user can expand the request to subsumant (see the elements of the request's fathers) and « brother « concepts.



Use of Corese : pathway for the ontologies to retrieve information

In the example, he can explore level by level, the squeaks on the father term of *steering wheel* (*Pdc_Planche de bord*[8]), on the sons (*Console*, *Tableau de bord*[9], *AmpliAntenneRadio*[10]), then on the father of this last term, *Z_Poste de Conduite*[11]. It is therefore possible to move around in the *Problem* ontology (*Creak --> Noise -->Problem -->Vibration*). The generalisation of the request is configurable.

It is possible to move around within an ontology in this way. Nevertheless, it would be more interesting to couple the research mechanism with an ontology browser.

There, we can imagine the user freely navigating in the RDFS ontology, and as soon as a term catches his eye, he switches to the QRDF module to launch a request on the annotated texts. The request's result places him in a precise area of the QRDFS ontology that he can then explore in detail.

---

[8] *Dashboard*
[9] *Control panel*
[10] Amplifier
[11] Cockpit

6 Preliminary results

The first tests were carried out on the *Component* and *Problem* ontologies, for an SGP base extract concerning a vehicle project with a specific milestone. We created the *Components* ontology, taking the different information sources into account (official references cross-checked with items from the problem base. In its present state it contains 2207 structured parts within 6 architectural zones, 12 sections and 39 parts managers as well as 3 detail levels reflecting the official reference system. The problem ontology has been initiated with problems coming directly from the SGP on one side (an approximate classification already exists proposed by the actors) and on the other side, problems extracted from interviews. It contains about 100 problem types, respectively 55 and 47 for the first and second families. The Service ontology comprises 36 services extracted automatically from the base.

These ontologies have been used to annotate around 2200 problems. The first tests have proved interesting and illustrate well the development of the process. Nevertheless, problems originating from texts and acquired by the pattern-matching mechanism will reveal the real importance of knowledge organisation in an ontological form. This phase is in the process of being created, a first parser was tested on a basis of 2 heuristic rules, which now contains 57. We also observed that before the semi-automatic construction phase of the *Problem* ontology, it was necessary to treat vocabulary diversity. In fact, the *Component* ontology contains official terms for the moment, but users frequently don't use these terms, but their own vocabulary (we can surmise that each profession has its own specific vocabulary). It is these terms and expression that can often be found in SGP texts.

For example, considering the term *Planche de bord* (dashboard), various abbreviations and synonyms of this term can be found : *PDB, planche, TDB, tableau, tableau de bord* (DB, board, .control panel) etc. or for an expression such as *Mise au point* (tuning), we can find *MEP* or *montage* (assembling). These various terminologies present a real obstacle to information access. Before launching the heuristic rules for ontology construction, and above all to insure the success of these rules, the diversity of terminologies must be reduced (or they have to be increased considerably in order to cover all variations). In other respects this information will be interesting to use in the request phase. The user can therefore have access to parts, not only via precise terms, but also using other terms semantically similar.

This is part of what we are working on at this time. We are gathering various « semantically close » families, constructed around « normalised » representative terms, the ontologies will then be completed with these variations.

7 Discussion and Conclusions

Inside the company there are several heterogeneous sources of information: different data bases, official references, problem management systems and other specific bases in departments. In addition to basic data which can be processed by traditional means, some bases contain important textual data. These texts may be considered as being a mine of information. Early «manual» readings have shown the value of an automatic process of exploitation of this data.

With SAMOVAR we propose to exploit all the different heterogeneous sources of data to build ontologies. Textual information is retrieved through linguistic tools and is structured in order to present the different points of view possible in the field in question. These ontologies are used afterwards to index the PMS base and with the help of the Corese platform we can carry out search, while being guided by these ontologies.

This method can be used in other projects in the company. To do this, representation structures have to be worked out (in our case - the ontologies) common to the project population. Firstly, the structures are developed using the information retrieved from the traditional data base (see interviews). After, they are added to gradually with the information from the texts, having taken care beforehand to work out rules in accordance with the corpus studied.

After having translated the totality into the Corese format, annotations are made on the documents using the elements structured in ontologies. Finally Corese furnishes a means of search in documents annotated in this way.

Even though the idea appears to be relatively simple, many problems can be encountered (we have already met them at our level). From the company's point of view, the multiplicity and the heterogeneous nature of the information sources seems tricky to deal with - the different bases need specific treatments adapted to it, and co-operation from the departments managing these data. The « ownership » view, which is frequent, is an obstacle to a process founded on sharing and co-operation between the players. The problems, therefore, also exist at a human level.

There is a certain risk linked to the use of TALN tools (Nomino, Cameleon) - the results often contain unnecessary « noise » that has to be refined - the question of human validation costs can be asked.

8 Acknowledgements

We wish to thank Rose Dieng for her patient and judicious advices, in helping us draw up this article, as well as Olivier Corby for his support on using Corese. We also wish to thank our colleagues at IPIA-Renault for their contribution in going over this paper.

9 References

[ASSADI 98] Houssem Assadi, Construction d'ontologies à partir de textes techniques, Application aux systèmes documentaires, Thèse de doctorat, Université Paris 6, 1998

[BIEBOW & SZULMAN 99] BIEBOW B & SZULMAN S, Terminae : a linguistics-based tool for building of a domain ontology, In D. Fensel and R. Studer, editors, Proc. of the 11th European Workshop(EKAW'99), LNAI 1621, pages 49--66. Springer-Verlag, 1999.

[BOURIGAULT 94] BOURIGAULT D, Lexter, un Logiciel d'Extraction de TERminologie, Application à l'acquisition des connaissances à partir de textes, PhD thesis, Ecole des Hautes Etudes en Sciences Sociales, Paris, France, 1994

[CORBY, DIENG, HEBERT 00] CORBY O, DIENG R , HEBERT C, A Conceptual Graph Model for W3C Resource Description Framework, To appear in Proc. of the 8th International Conference on Conceptual Structures Logical, Linguistic, and Computational Issues, Springer-Verlag, Darmstadt, Germany, August 2000, Springer-Verlag

[ENGUEHARD 92] Enguehard C, ANA, Apprentissage Naturel Automatique d'un réseau sémantique, thèse de doctorat, Université de Technologie de Compiègne, 1992

[GARCIA 98] GARCIA D, Analyse automatique des textes pour l'organisation causale des actions. Réalisation du système informatique COATIS, PhD thesis, Université de Paris-Sorbonne (PARIS IV), Paris 1998

[GREFENSTETTE 94] Grefenstette G, Explorations in automatic thesaurus discovery, Kluwer Academic Publishers, Boston, 1994

[HEARST 92] HEARST M, Automatic Acquisition of Hyponyms from Large Text Corpora, proc to the ICCL, COLING 92, Nantes July 25-28, 1992, p 539-545

[JOUIS 93] Jouis C, Contribution à la conceptualisation et à la Modélisation des connaissances à partir d'un analyse linguistique de textes. Réalisation d'un prototype : le système SEEK. Thèse de doctorat, 1993, EHESS de Paris

[SEGUELA & AUSSENAC 99] P. Séguéla and N. Aussenac. Extraction de relations sémantiques entre termes et enrichissement de modèles du domaine. In Proc. of Ingénierie des Connaissances (IC'99), pages 79-88, Paris, 1999.

[SEGUELA 99] SEGUELA P, Adaptation semi-automatique d'une base de marqueurs de relations sémantiques sur des corpus spécialisés. Terminologies Nouvelles}, 19:52-60, 1999.

Bulletin de l'AFIA, Dossier Terminologie et Intelligence Artificielle, N° 32, janvier 1998