Interpretation of Description Logics for Natural Language and for Databases

François de Beuvron, François Rousselot, David Rudloff
ERIC-LIIA, Ensais / Neurocim, Strasbourg
e-mail: {beuvron, rousse, rudloff}@eric.u-strasbg.fr

Abstract
This paper describes some functionalities of the CALIS project. This project intends to offer to end-user a natural language interface for database. We study the interactions between the natural language layer and the conceptual layer. Our conceptual layer is based on description logics, with some use of the Generalized Upper Model. We propose a dual interpretation of a conceptual model to query a database or to translate into natural language. In this manner, same terminological algorithms should be usable for natural language processing and for query translation and optimization.

CALIS: a natural language interface for database
CALIS (Conceptual with Advanced Language Information System) is a datawarehouse project involving the integration of natural language for database interrogation.

This multilayered system has the following structure:
- A Database Layer, currently restricted to relational database.
- A Conceptual Layer based on Description Logics, containing the database model and allowing query representation. Every conceptual expression is unambiguous and complete.
- A Semantic Layer representing partial and possibly ambiguous elements of information. At this level, anaphoric references and ellipses can be done.
- An Interface Layer that manipulates textual and graphical elements that can be converted into semantic objects.
- An Application Layer that integrates the interface and handles the dialog with the user. This application layer could either take place in a Web browser applet or in a reporting tool.

At each level, an expression is formed in a particular language.

Indeed, a natural language is translated into a potentially ambiguous semantic expression. These semantic expressions are transformed, if possible by consulting the terminology, in unambiguous conceptual forms. Then the concepts represent SQL queries for the database access.

When an inconsistency is detected, the system has to inform the user and explain (by the mean of a natural language message) the position and the reason of the inconsistency. Consequently, the semantic layer must be defined in such a manner that the part of the concept that causes the inconsistency can be extracted, defined as a semantic expression and translated in natural language.

The conceptual layer is based on description logics. A conceptual expression can be translated to a semantic expression and then to natural language or, in the reverse direction, in SQL.

The design of the conceptual layer was made with the following requirements:
- To make strong connections with natural language.
- To represent the relational database structure.

It will then be used both for natural language processing and for database
access. We propose to use the same taxonomy but with two possible concept interpretations.

As presented below, the first one is the interpretation of a concept as a database

query result. The second one considers a concept as the set of all correct natural language sentences corresponding to the concept meaning.

The Database interpretation

The conceptual layer contains the database model. In our system, this model will be mapped on the concept hierarchy.

When we define a query, we distinguish two parts:
  • the qualification of the concerned object by a description logics based concept.
  • the qualification of the elements of the concerned object that we want in the result. This second qualification is done outside the concept hierarchy in order to simplify the semantic of our description logic.

A query then points to a concept and defines a projection of this concept. More precisely, a query-concept interpretation is the projection of a cartesian product of related concepts.

A query associates a concept to a projection function to produce a partial view of a concept. It will then be interpreted in a particular way because of this projection function. The query interpretation process consists of translating the query definition in SQL and accessing the database. Several methods were already discussed in [2] and [3].

Moreover, the CALIS project involves simplification and optimization of database queries. Optimization methods range from value set reduction to caching methods by query indexation.
The Natural Language interpretation

In the CALIS project, we would like to provide an interactive natural language interface where the user could be helped to construct his query. By instance, if the user begin a sentence by:

\[ I \text{ want the articles sold by } \]

The system has to recognize the noun 'article' and the verb 'to buy' and then look in the conceptual model for the definition of the concept 'article'. Indeed, it has to match the verb to the relation and then get the constraint on the codomain of the relation. Actually, we want to associate to one relation many natural language representation, that is, many verbs. The user could indifferently use one of them and the system would be able to understand and retranslate with the different possible verbs.

Thus, in a linguistic context the interpretation of a concept will be a set of natural language expression. Each expression gives a comprehensible natural textual representation of the concept definition. Our idea is to create a linguistic database containing verbal forms. This linguistic database will be viewed as instances of reified concept-relations. It will be used to translate natural language in concept and to rewrite conceptual expression in natural language.

In order to make the connection between the verbs and the role, we decided to name the roles in a standardized form. Each role has a corresponding special concept-relation. Furthermore, these concept-relations are classified in a hierarchy inspired by the General Upper Model (GUM) [1].

As a matter of fact, we consider two terminological hierarchies.

- The first one is based on concept name like, in our study example, 'client', 'article', 'supplier'. Roughly, it represents noun expression in natural language.

- The role names are chosen among the concept-relation organized in the second terminological hierarchy. This relation hierarchy usually corresponds to verbal expressions. Each concept will be associated to several 'verb-instance'. The classification of this hierarchy is simply based on an explicit inheritance given by the user. The description logics inferences will be used to explore this hierarchy during the natural language translation.

When an user describes a relation between two concepts, he may not quote the relation type. The following sentences describe the same relation:

"the supplier of article"
"the article of supplier"
"the supplier who sell an article"
"the article sold by the supplier"
"the article made by the supplier"

They don't have exactly the same meaning but the same relation is regarded: the relation between a supplier and an article. These linguistic variations are instances of the same relation-concept. The best translation will depend on how the user previously wrote his query in natural language.

Let's detail our idea by a little example:

\[
\text{(defconcept ARTICLE} \hfill \\
\text{(AND} \hfill \\
\text{(SOME (inv r-supply) supplier)} \hfill \\
\text{(SOME r-cost price)} \hfill \\
\text{(SOME r-named name))})
\]

The roles beginning with 'r-' correspond to a concept-relation in the first hierarchy.

We create in the relation type hierarchy, a concept-relation "r-supply" that has two roles "domain" and "range".
Every concept-relation is based on the same structure with the roles 'domain' and 'range'.

Suppose the concept-relation R-SELL defined as:

(defprimconcept R-SELL
  (AND R-SUPPLY))

We have to examine the interpretation of R-SUPPLY in the natural language point of view. An instance of this concept can be the verb "to supply". To "sell" is instance of R-SELL and thus also of R-SUPPLY.

Indeed, from the information "article sold by", the system will search what concept does "sell" instanciate. It will find that "sell" instanciate R-SELL and then look for a role in article named R-SELL.

If the role is not present, the system will use description logics services to get all the superconcept of R-SELL. By this way, it will find R-SUPPLY as the only possible role of article that instanciates "to sell".
The system, by retrieving the properties of the role, will then be able to predict that the object following of "article sold by" has to be a SUPPLIER.

This kind of linguistic interpretation service may be used during the query construction for syntactic prediction and for concept textual representation.

Conclusion

In a natural language interface for database, a concept can be considered under different points of view.
On the one hand, a query will be interpreted as a set of database instances.
On this side, we are working on optimization method and caching method.
On the other hand, query concept can be viewed as the set of all natural language sentences that have the corresponding meaning. Different translations can be done because of different possible verbs to describe relation types. We are studying interactive translation processes that will guide the user to build queries.

This dual interpretation methodology may simplify the structure of our CALIS system and its reusability and portability.

Bibliography


