SESAME

A PORTABLE DATA BASE INTERFACE GENERATOR

Simon Sabbagh
and the InterLN Development Team(*)

CEDIAG
BULL A.I. Corporate Center
68 Route de Versailles
78430 Louveciennes
France
Simon.Sabbagh@cediag.bull.fr

ABSTRACT

SESAME is being developed to provide an easy access to the content of relational data bases to users without a specific computer training. Queries are typed in natural language either freely or with a guided mode. The system dynamically proposes through menus the different words and phrases that can make up a query. Users are able to exploit the results of their queries with standard electronic office tools or specialized applications.
The SESAME system is a user interface generator. To develop a particular application, different knowledge bases have to be built: lexicon, conceptual schema of the data base... Knowledge base editors and design methodologies provide help for the developement of applications.
SESAME is a good example of techniques created in research laboratories and applied to the development of an industrial product.

INTRODUCTION

The purpose of the SESAME project is to provide users with the possibility of extracting data from an information system in a language as close as possible to natural language. Users will be allowed to express queries with concepts closer to the external model of an application than to the logical schemas of data bases. It is thus necessary to use a model of the content of the information system at the conceptual level. The semantics of the application will be represented by a conceptual model. Mapping modules will be used to translate conceptual representations into relational schemas.

The natural language query expressed by the user is translated into a logical form by a parser. This parser uses a grammar and a lexicon specific to a particular application. It has access to the conceptual schema to validate the semantics of queries. The logical form is then translated into a SQL query, using the information provided by the mapping between the conceptual schema and the relational schema.

SESAME is an industrial product. It only uses fairly standard techniques. It is an interface generator, portable, intended for a large diffusion. Its most important feature is that it allows people who are not computational linguistics specialists to develop an application fairly easily. The whole design of SESAME is intended to respect this very strong constraint, even when it meant to use less sophisticated techniques than what is available. Great care has been also put in the design of the user environment for the querying steps as well as for the manipulation of the results provided by the

(*) Many people are taking part in the development of SESAME. For the Linguistic aspects: Blandine Gelain, Stéphane Guez, Jean-Michel Liaunet, Fariba Ommani, and Zhengce Peng, and for the Information System and User Interface aspects: Pascal Fischer, Elie Kerbaje, Laurent Lacote, and Arnaud Villemin. Olivier Deguine and Pierre Alain Vast were in charge of the integration.

data base SESAME is interfaced with (Bates & Weischedel 1987).

The SESAME system is made of four environments:

The information system environment used to build the conceptual schema of a given application and to generate mapping rules between the conceptual level and the relational level.

The linguistic environment to generate the linguistic knowledge bases from the conceptual schema of the application and a corpus of the domain.

The query environment provides the user with the tools to query the data base.

The results management environment provides the user with the tools to manipulate the results of the query provided by the D.B.M.S. and to exploit them through standard office systems or specialized applications.

The first two environments are used by the designers of a particular application to generate the knowledge bases necessary for this application. The two other environments are intended for the regular use of SESAME, once a particular application has been implemented.

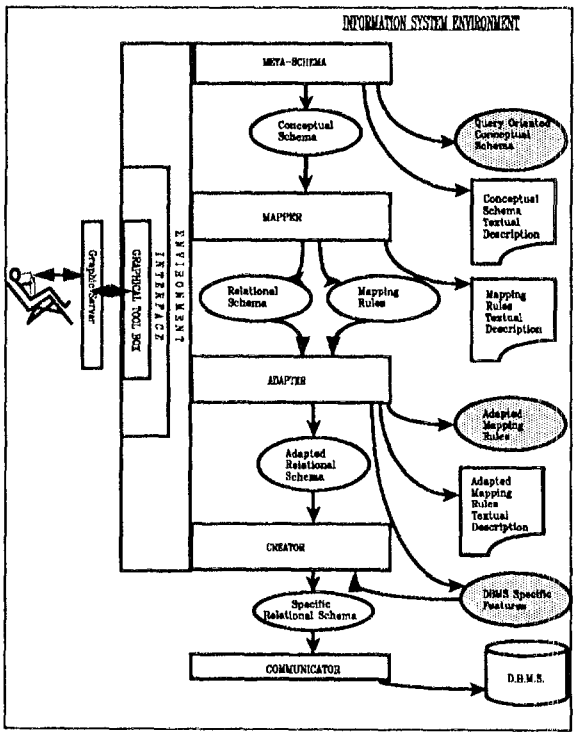
THE INFORMATION SYSTEM ENVIRONMENT

The information system environment is used to design the information system parts of a particular application, which consist essentially of the conceptual schema of the application and mapping rules between the conceptual level and the data base relational level.

The **conceptual schema** is a set of specifications which describe the semantic structure of the data base. It is specified in an entity relationship (ER) model. This model contains the traditional concepts (entity, relationship, property) used in standard design methods (Merise, Yourdon, IDA...). We have chosen to use this type of model instead of a knowledge representation language or a semantic network, in order to ease the implementation of applications by people used to standard data base tools. To respect the purpose of SESAME, it was essential not to fall in the trap where only the designers of SESAME would be able to develop applications. See (Grosz *et al.* 1987) for a review of existing systems, regarding the problem of portability.

We have extended the ER model to include multivalued properties, structured value domains (e.g. the domain *date* will be built from the domains *day*, *month* and *year*), and the generalization/specialization of entity types through the definition of inheritance relations between entity types. It is possible to specify on a schema the dependencies between entities, which makes the generation of a normalized schema easier for the mapper. It is also on this schema, at the conceptual level, that access rights for confidential data can be specified. They have their counterpart at the linguistic level: only words expressing authorized concepts will be accessible for a particular user.

The **mapper** produces a set of mapping rules which are rewriting rules which link the conceptual schema with the relational schema of the data base. An additional module contains a description of the specific features of the D.B.M.S. used, in order to fill the gap between standard SQL and the actual SQL of the D.B.M.S.



Information system environment

THE LINGUISTIC ENVIRONMENT

The linguistic environment is used to build the linguistic knowledge bases of a particular application. The unification grammar formalism is used to describe the lexicon and the grammar (Shiebert 1986).

A lexicon editor is used to generate the **lexicon** in the unification grammar formalism. The first source of information used is the conceptual model of the data base: all the concepts have to be associated with words and the semantics of the natural language interface is the semantics of the conceptual modelling of the application information system. But a natural language query may also contain semantic relations which are not directly expressed in the conceptual schema. These "virtual" semantic relations are defined in the **linguistic conceptual schema** as rewriting rules on "real" relations from the conceptual schema. These extensions of the lexicon are made possible by the analysis of a domain corpus.

The **grammar** is described in the unification grammar formalism: each grammatical category can be associated with a features structure represented as a tree. Syntactic as well as semantic constraints are expressed as contraints on the trees (features equations) and operated through unification.

A grammar rule is made of a rewriting rule and a set of equations which specify the syntactic constraints. Semantic constraints (selection restrictions) are also expressed as features equations. The values of these features specify the semantic types of the application. The lexicon is described in the same formalism. Each lexical entry is associated with a set of equations which specify the category of the word as well as the value of certain features. The semantics of a natural language query is represented with a logical formalism. The construction principle of semantic representation is compositionality. Each syntactic rule is associated with equations which express the rules of semantic composition (Moore 1989).

The grammar and the lexicon are compiled into a Prolog program. Unification which is a basic Prolog operation is thus directly and efficiently used.

```
np :- det, noun, n_pp   &
     [np,agr,number] = [noun,agr,number],
     [noun,compl,concept]
              = [n_pp,concept],
     [noun,compl,preposition]
              = [n_pp,preposition].

salary :- noun   &
     [concept] = salary,
     [agr,number] = sing,
     { compl :
          [concept] = employee,
          [type] = real,
          [preposition] = of,
          [presence] = non_obl,
          [semantic_relation] = salary }.
```

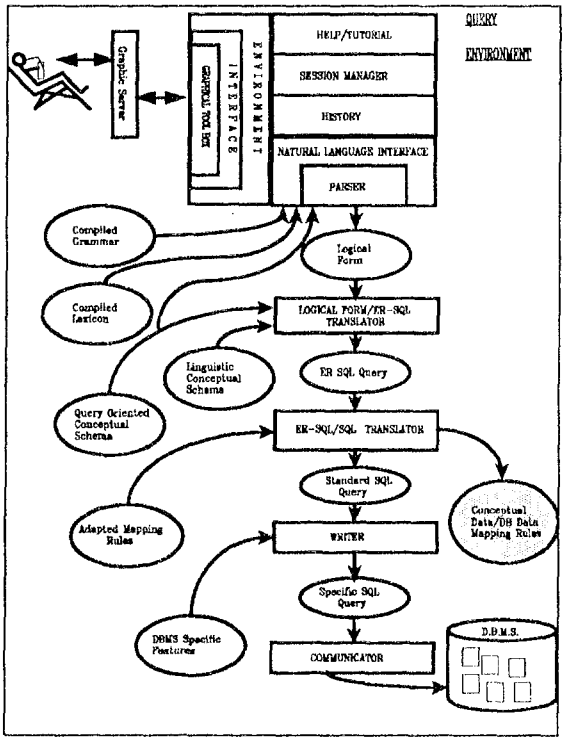
Simplified descriptions of a grammar rule and a lexicon entry

The linguistic covering of the grammar and the lexicon is the sub-language of data base query, which include the processing of expressions concerning the sorting of information, comparisons, etc. The grammar also processes coordination, pronominal reference and it detects ambiguities. The covering is large enough so that the present grammar should fit any standard application without any major addition. Only very specialized applications will require important changes, mainly at the level of noun phrases. This is a difference with NaturalLink which only provides a formalism: the semantic grammar and the semantic representation building rules have to be written by the application developer (Texas Instruments 1985).

THE QUERY ENVIRONMENT

In the SESAME project, we have taken great care of the user interface which is the only way to have a friendly access to the data base. The query environment provide the user with several powerful functionalities. If the analysis of a freely typed query succeeds, then the SQL translation is completed after a dialogue with the user in order to specify the form of the answer. If the analysis fails, the list of possible continuations after the failure point is proposed. The user can select a word from this list or type it directly. The remaining part of the sentence which has not been parsed is also displayed so that the user can use it directly or edit it

to complete the query. The user can also choose to complete the query in guided mode with the help of dynamically synthesized menus. For more information on these techniques, see (Rincel & Sabatier 1989). A graphic query interface has been specified, but not yet implemented.



Query environment

An history module provides the user with tools to memorize, organize and manage the queries of a working session and their results. Usually 70% of the queries belong to a small fixed set, SESAME include tools to manage a library of queries with parameters to be specified by the user, with the advantage that queries in this library are expressed in natural language. A help system, implemented with hypertext tools, can be called at any point in the user interface. This help facility is implemented through a hypertext system integrated to the project graphic toolbox. The tutorial provides the user with a demo of the product and a learning session, including sketches with choice points so that the user can control the demo.

The logical form produced by the natural language interface is translated into a query expressed in ER-SQL. ER-SQL is the query language of the conceptual schema. It is a SQL like language where joins are replaced by semantic paths. The translation into ER-SQL is completed in two steps. In a first step, the logical form which contains virtual relations is translated into an equivalent logical form which only contains real predicates, using the linguistic conceptual schema. The second step transforms a query expressed in a logical language with quantified variables into a query expressed in an algebraic language (ER-SQL) operating on the conceptual model.

The ER-SQL query is translated into a standard SQL query, using the mapping rules generated by the information system environment. The translation process keeps trace of the direct link which is set between the conceptual data of the ER-SQL query and the data which will make up the answer when the query is sent to the D.B.M.S. This information is kept in the form of conceptual data / data base data mapping rules. These rules define the semantics of the content of the relational table the result of the query is made of. The results management environment will use these mapping rules to present and display the results of a query. A writer module refines the query expressed in the standard SQL language to fit the specific features of the particular D.B.M.S. used for the application.

THE RESULTS MANAGEMENT ENVIRONMENT

The results management environment must be able to take the answers of the D.B.M.S. and to present them in such a form that they can be exploited by the user. A broad choice of possibilities is given to the user for the presentation of the results: environment in which the results are to be exploited and type of tool used for this exploitation. This module uses the conceptual data / data base data mapping rules to retrieve the information specified at the conceptual level, in the relational table which is returned as the result of the query. The results can then be presented in the terms the user choose to express the query, and not with the logical names provided by the data base. This module will make possible a presentation of the structured domains and the multivalued properties which do not exist at the relational level.

CONCLUSION

SESAME is an insdustrial project designed from the beginning with a strict life cycle and strong quality criteria. The product will be announced before the end of 1990. There is a French and an English version, and other European languages are planned for the near future. The software architecture of SESAME is designed to fit various hardware and software environments and network protocols.

We are also taking part in a European Eureka Project strongly connected with SESAME. This project addresses the multi-lingual aspects of data base interfaces and the integration between graphics and natural language. Its results will be integrated in future enhanced versions of SESAME.

REFERENCES

Texas Instruments (1985). Explorer Natural Language Menu System, Data Systems Group, Technical Report No 2533593-0001, Austin, Texas.

Bates, M. and Weischedel, R. (1987). Evaluating natural language interfaces - Tutorial at the 25th ACL - Stanford University.

Grosz, B.J. *et al.* (1987). TEAM: An experiment in the Design of Transportable Natural Language Interfaces, AI Journal, Vol.32, No.2, May 87, p.173-243.

Moore, R.C. (1989). Unification based semantic interpretation, Proceedings of the 27th ACL meeting, p.33-41, Vancouver, Canada.

Rincel, Ph. and Sabatier, P. (1989). Leader: un générateur d'interfaces en langage naturel pour bases de données relationnelles, Congrès AFCET, RFIA 89, Paris.

Shiebert, S.M.(1986). An introduction to Unification Based Approaches to Grammar, CSLI lectures notes, number 4, CSLI, Stanford University, Stanford, California.