



3.68

**UNIVERSIDADE NOVA DE LISBOA**  
**DEPARTAMENTO DE INFORMÁTICA**

ORBI - AN EXPERT SYSTEM IN PROLOG

Luis Moniz Pereira  
Eugénio Oliveira

UNL - 14/83

JULHO 1983

**SEMINÁRIO DOS OLIVAIS**  
**QUINTA DO CABEÇO**  
**OLIVAIS - LISBOA**

ORBI - AN EXPERT SYSTEM IN PROLOG

Luis Moniz Pereira  
Eugénio Oliveira

UNL - 14/83

JULHO 1983

# URBI - AN EXPERT SYSTEM IN PROLOG

Luis Moniz Pereira  
Eusebio Oliveira

Universidade Nova de Lisboa  
2825 Monte de Caparica, Portugal

## ABSTRACT

We describe URBI [PEI] a computer system, interrogatable in a flexible subset of Portuguese and implemented in Prolog on a small machine, which embodies and assimilates expert knowledge on environmental biophysical resource evaluation. It is capable of explaining the application of that knowledge to a territorial data base, and also of answering questions about its linguistic abilities. The system was developed in 2-man year, under contract with the Portuguese Department of the Environment.

## INTRODUCTION

Our expert system comprises a data base of facts and a knowledge base of inference rules, and is capable of answering queries by using the rules to reason about the facts. It also explains its reasoning. Rules and facts can be updated by the user. Queries are in natural language (Portuguese). They can be about the domain of expertise, including how the knowledge is organized, or about the system's natural language. To our knowledge, the principal innovations are:

- It is the first expert system written in Prolog which includes natural language, explanations and metaknowledge. Furthermore it deals with a real-world, real-sized domain of application.
- It is the first among expert systems, with a comparable combination of features, which is implemented on a small machine (PDP11/23). Use of Prolog was crucial in this respect.
- It interactively acquires very complex rules.
- It includes the first natural language interface with a meta-linguistic facility, thus rendering it capable of answering questions regarding its linguistic competence. Moreover the grammar simultaneously comprises treatment of ellipsis, extraposition and coordination of sentences and relatives. Also the semantic formula generated is optimized to some extent. A planning stage solves deterministic goals first and generates deterministic explanation parts.

## DOMAIN OF EXPERTISE

The domain of expertise concerns the evaluation of the biological and physical environmental offer [SEA]. The typical region size is that of the county, described by means of 23 descriptors. To each descriptor there corresponds a map of the region, which is covered by a uniform grid forming squares 200 meters side. On each square the value of descriptors is digitized to integers ranging from 1 to 5. Furthermore, to each value is associated a representativity, an integer ranging

from 1 to 5, describing the extent to which that value is homogeneous in the square. A county typically comprises some 3,000 squares. Examples of descriptors are:

d1 -- microclimate quality d9 -- risks of soil erosion.

Given the observed data base of ground facts as expressed by descriptors, more abstract derived information can be obtained by defining factors and aptitudes. The value of a factor is computed from the values of the underlying descriptors through a multiple possibility rule provided by experts.

The representativity of factor values is always a weighed sum of the representativities of the intervening descriptors. There are 16 factors. An example is :

f5b -- cinematic and landscape riches.

Aptitudes are likewise defined in terms of factors, as these are in terms of descriptors, through inference rules supplied by a team of experts. There exist 5 aptitudes. An example is :

a3 -- intensive agriculture.

Finally a bio-physical synthesis rule relies on the aptitude values and representativities to extract the more promising areas according to development objectives, but tacking into account an environment protection rule. The knowledge of the experts is further made explicit by a large collection of explanatory texts, one for each possible descriptor value.

PROTOCOL (translated)

> What is the value of the aptitude 3 at the point 103,709 ?  
- At the point with coordinates 103,709, aptitude 3 has value 5 (vers apt) (with representativity 2) because :

5 was computed as  
 $(25*(6-1) + 10*4 + 5*1 + 60*5 + 50)/100$  by substitution in  
 $(25*(6-F6d)+10*F3+5*F2b+60*F4+50)/100$

\*\*\*factor 2b has value 4 (with representativity 5) because :  
descriptor 9 has value 4 (with representativity 5) because :  
The more importantly occurring aquiferous present a HIGH protection since they belongs to the captive meso-ziocenic layer.

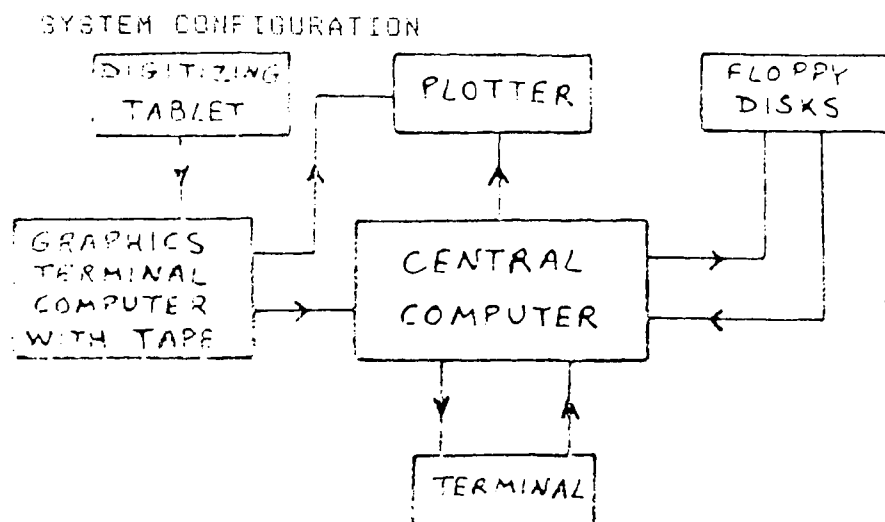
\*\*\*factor 3 has value 4 (with representativity 4) because :  
4 was computed as  
 $(30*(6-1) + 70*4 + 50)/100$  by substituting in :  
 $(30*(6-D7)+70*D10+50)/100$

Descriptor 7 has value 1 (with representativity 5) because :  
The superficial drainage of rain is VERY LOW given that the characteristics of the hydrographic basin to where the point belongs and medium annual pluviosity determine a corrected roughness number LESS THAN 150.

Descriptor 10 has value 4 (with representativity 3) because :  
The soil is considered to have a LOW risk of erosion since it presents as limitations those associated to its reduced effective thickness.

\*\*\*factor 4 has value 5 (with representativity 1) because :  
descriptor 10a has value 5 (with representativity 1) because :  
The soil is susceptible of highly intensive agricultural use. The pedological aptness for agriculture can be said VERY HIGH. This soil is very likely totally or partially protected by article 1 of decree 308/79.

\*\*\*factor 6a has value 1 (with representativity 1) because :  
descriptor 11 has value 1 (with representativity 1) because :  
The vegetation balance is considered VERY LOW from the fact that natural phytocenoses are in a very degraded state and with an extremely difficult return to the potential climate.



System configuration

Digitizing is performed off-line. The central computer, a PDP 11/23 has 2K 64 KB of memory, runs a Prolog interpreter [CLO] under the RT11 operating system. With foreground-background capabilities we define two Jobs, one for each 64k partition. It communicates with two double density floppies. We are content with the Prolog system, which externally is almost the same as DECsystem10 Prolog [CLO]. The full interpreter has a size of 17 KB, but can be trimmed down to 12 KB by removing the debugging package, and even more if necessary. The RT11 has 5 KB. We added a few improvements to Prolog, like the 'all' predicate for obtaining sets of solutions [PE2], extended memory usage, and message queue communication between Jobs.

#### PROGRAM ORGANIZATION

There are four separate programs : one in Basic for digitizing, referred before; another, in Pascal, for receiving the digitized points and storing them on disk; one for knowledge updating; and a fourth comprising two basic modules, in Prolog, a natural language interface, and an evaluation and explanation module, each running as a separate Job.

Questions are typed to the evaluation Job for morphological analysis, and the resulting list of tokens is passed on to the natural language Job, though the message queue, for syntactic/semantic analysis. The first Job backtracks and awaits the second one to write the query on the message queue, which it does and fails to recover space and wait for the next token string. Evaluation starts by reading the query from the message queue, and then evaluating it (see below). After the answer is produced, the evaluation Job backtracks to recover space and waits for the user's next question. The knowledge update program is used by itself. It reads the knowledge base in force, interacts with the user to accept any updates, checks them for well-formedness and consistency, and produces a file with updating information. The user may then later instruct the main program to incorporate it.

#### COGNITIVE EXPERTISE

Edinburgh Prolog notation will be used henceforth. The knowledge base is comprised by a set of rules for computing the values and representativities of facts and attitudes, and also to evaluate synthesis of best use and environment protection advice. The data base consists in a set of unit clauses, one per digitized point, of the form :

point(X,Y,d(v1-r1,v2-r2,...)).

where X and Y are the coordinates and d is the descriptors functor containing as arguments all value-representativity pairs, one for each descriptor.

A typical rule for an aptitude is :

```

a(1,D,V-R) :-
    V=0 if (F4=4 or F4=5) where f(4,D,F4-R)
           or (F6A=5 or F6A=3) where f('6a',D,F6A-R)
           or (F6B\=0 and F6B\=e and F6B\=f)
                                   where f('6b',D,F6B-R)
           or F6C=5                 where f('6c',D,F6C-R)
           or F6F=5                 where f('6f',D,F6F-R)
    else
(expr(6,V) is (40*F1+25*F2+20*F2A+5*F2B+5*F3+5*F5+50)/100),
R is (40*R1+25*R2+20*R3+5*R4+5*R5+5*R6+50)/100
where (f(1,D,F1-R1),f(2,D,F2-R2),f('2a',D,F2A-R3),
      f('2b',D,F2B-R4),f('3',D,F3-R5),f(5,D,F5-R6) ).

```

where D receives the descriptors' functor of any point where a1 is to be evaluated, and V and R the value and representativity of a1 at the point.

### DEDUCTION, EVALUATION AND EXPLANATION

Given a query, three processes take place in sequence : deduction, evaluation and explanation. In the deduction phase, all deterministic goals and subgoals of the query are executed, and the deterministic part of the explanation tree is also generated. The objective is to achieve efficiency in queries involving a set of solutions differing only in their nondeterministic portions. Next, in the evaluation phase, the nondeterministic parts of the query and of the explanation structure are evaluated. Finally the results are presented and explained. Schematically, we have :

```

execution(Query) :-
    deduction(Query,Nondeterminate_part,Explanation),
    evaluation(Nondeterminate_part),
    output(Explanation).

```

For a more precise understanding consider the clause :  
deduction(all(X,P,S),all(explain(X,E),NDF,S),E):-

```

    !, deduction(P,NDF,E).

```

takes a query of the form 'all' and transforms it into a new query, where P is replaced by its nondeterminate parts NDF, and the explanation E that results from executing the deterministic parts of P is attached to each object X. The clause below picks up a clause for aptitude A (where A is specified), executes the determinate part of its body B, and returns its nondeterminate part and the determinate explanation part B, to which it adds the head explanation part :

```

deduction(a(A,D,V), NDB, a(A,V)&E ) :-
    nonvar(A), clause(a(A,D,V), B), !, deduction(B,NDB,E).

```

The nondeterminate explanation parts are obtained during the evaluation phase.

```

deduction(A else B,solve(NDA,E=EA;NDB,E=EB), E) :- !,
    deduction(A,NDA,EA),
    deduction(B,NDB,EB).

```

where solve(G) :- G, !.

The tips of the explanation structure are nondeterminate. They are generated by the descriptor clauses as calls to predicate 'table', which produces the relevant explanatory messages as we have seen during the evaluation phase:

```

deduction(d(N,D,V), Body, d(N,V)&E ) :-
    nonvar(N), clause(d(N,D,V), Body),
    !, Body = (arg(A,D,V), table(N,V,E)).

```

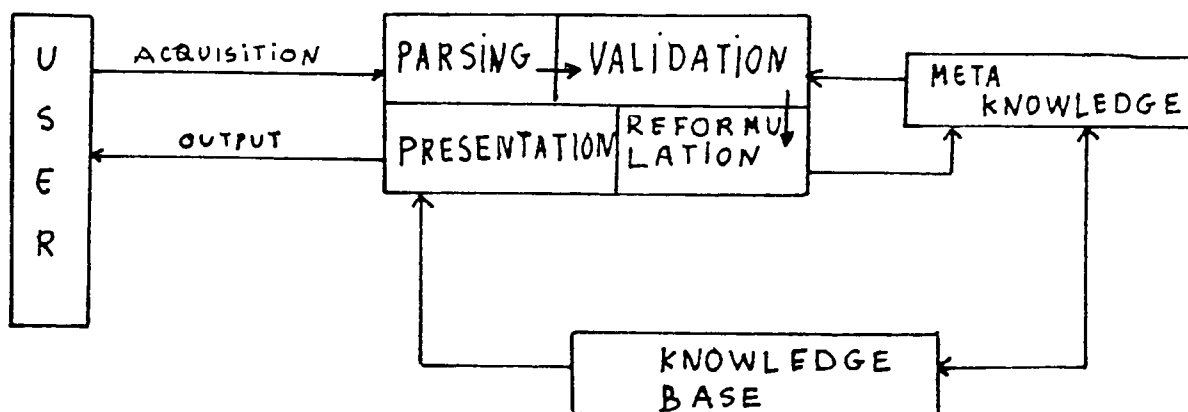
For nondeterministic goals that do not contribute with any explanation part there is a final catch-all clause :  
deduction(G,G,nil).

#### DATA BASE ACCESS

The data base of points is not in memory but as unit clauses on disk. Because we assume that each users has at most one call  $p(X,Y,D)$ , we can use the file pointer to keep track of the last clause read. To avoid exhaustive search into the complete data base this is divided into many smaller files whose names are meaningful about point's coordinates they contain. A search procedure is able to access the right files where are the known points.

#### KNOWLEDGE ACQUISITION AND META-KNOWLEDGE.

The knowledge acquisition program, aims at helping the authorized user to reformulate the knowledge base, in particular through the use of a conserial notation, by providing consistency checks, and by not requiring any familiarity whatsoever with implementation issues. Meta-knowledge is needed for these purposes, but also for the system to answer questions about itself.



Orbi's rules can embody complex chunks of knowledge with alternative conclusions where necessary, and allows for boolean connectives. A simple acquisition example follows :

```

>Rule acquisition
-About what concept : synthesis; aptitude; factor; descriptor?
>aptitude
-Is it a new rule or an existing one ? If you are in doubt
type metaknowledge
>new
-What is its number and name ? If you are in doubt type
metaknowledge
>1 lodgings
(here the screen splits into a fixed window with the BNF of a
rule and a scrolling region to enter the rule)
-name the new aptitude a1. Finish it with "."
>a1=0 if
    f4=0 or f4=5 or f6a=s or f6a=3
    or f6b\=0 and f6b\=e and f6b\=f
    or f6c=5 or f6f=5
    else a1=40*f1+25*f2+20*f2a+5*f2b+5*f2b+5*f3+5*f5.
  
```

The computing of the representativity is implicit, and the Prolog clause compiled from this rule includes it. When the user requests to see a rule it is decompiled into the original surface notation. The compiled rules retain the surface structure so that explanations are provided using it.

We have strived not to duplicate any information in the

knowledge base, to facilitate automatic updating of the relevant structures, and strived as well to have data structures that can be both easily read and used by other program parts, but are also easily executable program parts on their own. In a nutshell, knowledge base clauses should be both used as active processing agents, and as passive data for meta-knowledge agents. Ideally, the meta-knowledge agents should be describable to themselves. We find Prolog amenable also in this respect, and set again for writing specialized interpreters [PE4] to achieve the control regimes required by knowledge processing.

## NATURAL LANGUAGE COMPETENCE

### LINGUISTIC COMPETENCE

ORBI's linguistic competence is achieved by means of a lexical and a syntactic-semantic analysis, transforming a natural language sentence into an almost optimally directly evaluable list of Prolog goals. For example, the query :  
 ? Which are the points having the same value for the aptitude intensive agriculture? is transformed into the Prolog goal :  
 all( V-points:S ,  
     all( (X,Y), (a(X,Y,D),a(3,D,V-R)) same V,S ), L )  
 where 3 is the code for 'intensive agriculture', and L is the answer in the form of a list of pairs V-points:S, such that for each value V of aptitude 3, S is the list of points with coordinates X,Y that have the value V.

### THE LEXICAL ANALYSIS

Like MICROSIAL [PIQ], our lexical analysis replaces each word of the input sentence by its corresponding lexical category.

### THE SYNTACTIC ANALYSIS

The syntactic-semantic analysis is realized by means of a core grammar containing context-free and sensitive rules (expressed in the definite clauses grammar formalism, see [PE1]) with syntactic and semantic checks. These rules handle the fundamental structures of Portuguese, namely :

- yes/no questions; wh questions; commands;
- affirmative, negative, relative, prepositional, coordinate, extraposed and elliptic clauses;
- complex noun complementation, adjunction and abbreviations;
- universal, existential, numeral, definite and indefinite determiners;
- adjectives, verbs and adverbs;
- metalinguistic terminology (allowing the asking of questions about ORBI's linguistic competence itself).

This core grammar is independent of its application and portable to other domains. This has indeed been done [PE2]. The core grammar is completed by a domain dependent module (10 times smaller than the core) containing structures, vocabulary (noun and verb phrases essentially) and semantic controls. We have incorporated elliptic and extraposed structures in ORBI's linguistic competence because they are essential to a good interaction. Syntactic and semantic controls verify number and gender agreements, designations of complex entities, and compatibility between nouns and verbs and their complements, pointing out any faults.

The syntactic and semantic analysis are not separate as in CHAT-80 [WAR] but blended as in MICROSIAL [PIQ]. This solution is best to stop the parsing short as soon as a semantic error



is detected. It also provides for a more compact grammar.

FULL SENTENCES

Full sentence structures are yes/no questions, wh-questions and commands, with possible extrapositions. Yes/no questions have the structure : (It is the case that) NP\_SUBJECT VP ? where (is it the case that) is a facultative yes/no operator ( facultative elements are bracketed). Example :

> Some points that have the same value for the aptitude industry have the factor riches of the subsoil between 1 and 3? Wh-questions may have the following structures :

WH\_NP\_COMPLEMENT NP\_SUBJECT (NEGATION) VERB ?  
WH\_NP\_COMPLEMENT (NEGATION) VERB (DOUBLE NEGATION) NP\_SUBJECT ?

> What value has each aptitude of the points whose descriptor risks of erosion is 5?

> What is the region that does not have any point with the value 1 for the industry aptitude ?

Or the structure : WH\_NP\_SUBJECT VP ?

> How many points having the same value for each aptitude exist?

Commands have the structure :

(IMPERATIVE\_TRANSITIVE\_VERB) NP\_OBJECT !

> The points of region 105,206;118,219 that have 2 aptitudes with value 1?

Extrapositions are acceptable in yes/no, wh-questions and commands. ORBI understands left extraposition of complements of any NP\_SUBJECT. The extraposed complements may occur at the beginning of the sentence and/or just before the NP subject. Each extraposition is first syntactically analysed and later concatenated to the NP\_SUBJECT. Examples :

> At each point, which factors of each aptitude are greater than 2?

> How much is the value, for point 28,106, of the permeability of subsoil?

ELLIPTIC SENTENCES

Ellipsis may occur inside the same sentence or from sentence to sentence in the dialogue. (In our examples, we shall note elliptic strings by [], and bracket the full strings they refer to ). In the first case, ORBI deals with ellipsis of subjects and/or verbs in coordinate sentential clauses. For example :

> [Has point 69,103] the value apt for the industry aptitude and [] the value of factor resistance of the soil for erosion greater than factor 4?

The interpretation of such ellipsis is not complex. The different types of gap are expected in the grammatical rules for coordinate sentential clauses.

Sentence to sentence ellipsis are rather more complex and differ according to the types of sentence : yes/no, wh-questions and commands. In our examples we give first the full sentence and then the elliptic one introduced by 'And'. Ellipsis between yes/no questions - ellipsis of the whole of the NP subject.

> [Is it the case that the aptitude concentrated lodgings of the points

whose the descriptor DB is equal to 3] is between 1 e 4? > And [] is equal to 3? - ellipsis of the whole of the VP. Example :

> Is it the case that the aptitude concentrated

losses of the points whose descriptor vesetal resistance to fire is equal to 3 [is between 1 and 4]?

> And the factors of the point whose aptitude 3 is equal to 5 [ ]?

- ellipsis of a part of the NP subject and the whole of the VP. Examples :

> [The descriptor D10] of the points in the region 91,707;103,900 that is between 3 and 5 [is lower than the descriptor surface pluvial drainage of point 26,40]?

>And [ ] of the points whose the aptitude 2 is greater than 3 [ ] ?

> [The descriptor D8] of the points in the region 91,707;103,900 [that is between 3 and 5, is lower than the descriptor vesetal resistance to fire of the point 26,40]?

> And [ ] in the region 100,109;140,150 [ ]?

> The descriptor D6 [of the points of the region 91,707;103,900 that is between 3 and 5 is lower than descriptor D2 of the point 26,40]?

> And the factor riches of the subsoil [ ]?

Ellipsis between wh-questions

- ellipsis of the whole of the WH\_NP. Example :

>[Which points of the region 23,78;84,98] have the value apt for the A1?

> And have the value 5 for the aptitude industry?

- ellipsis of the whole of VP. Example :

> Which points of the region 54,57;78,80 [have the value not set for the aptitude intensive agriculture]?

>And which points have the descriptor surface pluvial drainage with the value 5 [ ]?

- ellipsis of a part of the WH\_NP and the whole of the VP. Examples :

>[Which descriptors] of aptitude 3 in the point 84,12 that are greater than 2 [are equal to the factors of point 32,10] ?

>And [ ] of aptitude 4 of the points of the region 45,67;12,34 [ ] ?

> [Which descriptors] of factor 5 [at point 14,36 that are between 2 and 4 are equal to factor F1 of point 45,56] ?

> And [ ] of the aptitude recreation [ ]?

> Which factors [ of the aptitude industry at point 67,54 that are lower than 4 are greater than the factor geologic resources of the point 45,78]?

> And which descriptors [ ]?

Ellipsis between commands

- ellipsis of the verb. Example :

> [Give me] the value of the factor F8 of the point 23,175?

> And [ ] the points whose factor F8 is between 3 and 5?

- ellipsis of the part of the NP object. Examples :

> [The descriptors] of the aptitude 6 [of each point] that are greater than 3?

> And [ ] of each factor [ ] ?

How sentence to sentence ellipsis are solved :

Ellipsis occurring sentence to sentence in ORBI are solved according to the structure of the previous sentence in the dialogue.

In order to do that, for each grammatical sentence accepted by ORBI one records the morphological sequences that could be an ellipsis in the next sentence of the dialogue. For example, regarding the sentence :

> Which descriptors of the aptitude intensive agriculture at point 56,78 that are greater than 2, are equal to the factor F1 of point 12,95?

ORBI will record the morphological strings corresponding to the structures [which descriptors] [of the aptitude intensive agriculture] [at point 56,78] [that are greater than 2] [are equal to the factor F1 of point 12,95], which may be

an ellipsis in the next sentence. If the next sentence is not elliptical, ORBI forgets these informations. But, if it is elliptical, those beginnings with 'And', ORBI tries to fill-in the presumed gaps of the sentence with the different strings recorded in certain pre-specified ways, in an attempt to find a grammatical analysis.

In short, the technique consists in transforming the elliptical sentence into a full (complete) sentence and to analyse it as a full sentence. The new full (ex-elliptical) sentence being morphologically built from parts of a different sentence, some syntactic controls as gender and number agreements being suspended during the analysis. Of course, an elliptical sentence may be followed by another one.

#### PHRASE STRUCTURES

We give here some examples about phrase structures of noun (NP) and verb (VP) that ORBI understands. Noun phrases have the following structure :

(DETERMINER) (ADJECTIVE) NOUN (COMPLEMENTS)  
(RELATIVE-PARTICIPLE CLAUSES)

... the same value for the descriptor vesetal balance at the points of the region 23,41;67,90 that is between 2 and 5 ...

... the points whose the aptitude concentrated loddings is greater than 4, whose the aptitude A2 is lower than 3, or having the factor climatic comfort different from the factor D8 ...

... the average of the values of factor F12 of the points that have the value less apt for each aptitude ...

Verb phrases have the following structures :

(NEGATION) VERB

(NEGATION) VERB (PREPOSITION) (DOUBLE NEGATION) NP

... not have the aptitude 3 lower than 3 ...

... is equal to the factor f5 ...

ORBI also understands their conjunction or disjunction, as in ... are equal to 5 or have the factor 2 greater than the descriptor D9 ...

...have the aptitude non-intensive agriculture with the value apt, the aptitude 5 equal to 1 and no factor of the aptitude 4 between 1 and 3 ...

(Note the ellipsis of the verb, which is permitted in coordinations)

#### THE VOCABULARY

ORBI's vocabulary may be divided in two parts : a core vocabulary and a specific one. THE CORE VOCABULARY (as the core grammar) is independent of the application, and transportable to other domains. It may be divided into two subparts : a metalinguistic part and a non-metalinguistic one. The LINGUISTIC part contains : determiners, prepositions, contractions, common verbs, relative and interrogative pronouns, phrase and sentence connectors, negation, prelocutory expressions. The METALINGUISTIC vocabulary contains words used for asking ORBI about its linguistic competence. THE SPECIFIC VOCABULARY, especially nouns, abbreviations, names of entities verbs, adjectives and abverbs, refers to ORBI's domain of application.

#### THE SEMANTIC ANALYSIS

For each grammatical sentence a logical expression is produced, ready for evaluation, which describes its semantics, plus a term expressing the focus of attention used for output. The semantics of a sentence is obtained from that of the subject noun phrase (SNP) and from that of the quantifier freed (as shown below) verb phrase (VP). If the SNP is

universally quantified, the final evaluation form of the logical expression differs according to whether the sentence is a yes/no one or not, example :

yes/no : each( [X,Y] , (SNP,VP) ) becomes not( SNP, not VP )  
 others : each( [X,Y] , (SNP,VP) ) becomes all( [X,Y] ,  
 (SNP,VP) , S )

Noun phrases may feature embedded quantifiers, whose score is reversed relative to order of appearance. Furthermore, they are reduced to a single one, where the order of quantified variables expresses scorings :

each X of each Y P becomes each ( [Y,X] , P )

This makes all subsequent treatment of quantification easier, without the need to recurse into semantic structures.

'some', 'averse' and 'how many' pose special scoring problems which we will not go into here. There is no treatment of presuppositions (the definite and indefinite articles are similarly treated), since they are unnecessary in the ORBI domain. Existential quantification only needs to be made explicit in noun phrases with 'mesmo' ('some'), by using the 'some' construction of the 'all' system predicate [PE3]. Care is taken to produce the right order of evaluation when relative clauses(R) are present (and on all other occasions). When the relative refers to a noun (N) although it appears after its complements (C), the order (N, R, C) is used in the logical expression matrix of the noun phrase. If the relative refers to a complement, it appears immediately after it : (N, C, R). Any universal quantification in relative clauses is always reduced to the evaluation form not( \_ , not \_ ) before their semantic embedding in the noun phrase takes place. The semantics of verb phrases is obtained from the verb semantics V (with possible double negation) plus that of the possible ensuing noun phrase NP, where any eventual universal quantification in NP is first reduced to the form not( \_ , not\_ ) :

V + each( X , NP ) becomes not( NP , not V )  
 not V + each( X , NP ) becomes not( NP , V )  
 V + NP becomes ( NP , V )  
 not V + NP becomes ( NP , not V )

The dictionary does not contain the semantics of individual words : they are included in two tables. In one, the entries are 5-tuples, each corresponding to a noun/complement pair. Given their morphologies, their individual predicates with appropriately linked variables can be obtained, plus an extra predicate condition if necessary (as the fifth element in the tuple). Example :

complements\_table( factor:F-FV, aptidao:A-AV,  
 f(F,D,FV), a(A,D,AV),  
 part\_of( f(F,D,DV), a(A,D,AV) ) ).

Individual semantics of nouns can be obtained from this table as well, when they occur without complements, by simply ignoring the irrelevant arguments. A table similar to this one is used for verbs and their complements.

#### METALINGUISTIC COMPETENCE

Due to the wealth and variety of structures, vocabularies and uses involved in natural language, and interface for it will always fail to understand a part of the sentences produced by a user, casual or otherwise. Therefore the interface must be really informative showing explicitly why it rejects a sentence, and what its capabilities are. As in MICROSIAL [PIQ], ORBI points out unknown words, grammatical disagreements, erroneous designations of complex entities, and some false presuppositions; but also incorrect complementations of nouns and incompatible subjects and complements of verbs. All these types of diagnosis are not in fact sufficient to ensure a good

natural language interaction. A better solution is to offer the user the possibility for asking the system questions about its linguistic competence, as we have done, to some extent, in our expert system. For example :

> Give me the vocabularies! > Does the word "factor" exist?  
 > Give me the prepositions, that exist !  
 > What are the possible complements of the noun "point" ?  
 > What are the possible subjects and complements of each verb?

The knowledge necessary to answer these metalinguistic questions is grouped in a specific database. Future developments will consider allowing ORBI's grammar to be consulted as a linguistic database.

## CONCLUSIONS

We have found Prolog an excellent language for expert system implementation. Several reasons can be adduced :

- it integrates expediently into one same simple and powerful formalism all the various desirable components of expert systems : natural language, knowledge base, explanation facility, relational database, meta-knowledge, and interpreters for specialized control.
- its conciseness of expression, coupled with a good implementation, allow a program with the complexity of an expert system to be realized on a small machine, for practical use.
- the dual semantics, declarative and operational, of the knowledge base clauses facilitates the development of meta-knowledge features.

## ACKNOWLEDGEMENTS

We are grateful to JNICT for their financial support, to Antonio Porto and Miguel Filgueiras for improving the Prolog system, and to Paul Sabatier for his help with the natural language interface.

## REFERENCES

- [CLO] Clocksin,W. ; Mellish,C.,  
Programming in Prolog, Springer-Verlag, 1981.
- [PE1] Pereira,L.M. ; Sabatier,P. ; Oliveira,E.,  
ORBI- an expert system for environmental resource evaluation through natural language, Proc. of 1st Int. Logic Programming Conf. ,Marseille, 1982.
- [PE2] Pereira,L.M. ; Porto,A.,  
A Prolog implementation of a large system on a small machine, 1st Int. Logic Programming Conf. ,Marseille.
- [PE3] Pereira,L.M. ; Porto,A.,  
All solutions, Logic Programming Newsletter n.2, Autumn 1981.
- [PE4] Pereira,L.M., Logic Control with logic, 1st Int. Logic Programming Conf. , Marseille,1982.
- [PER] Pereira,F. ; Warren,D.H.D.,  
Definite clause grammars for language analysis - a survey of the formalism and its comparison with augmented transition networks, Artificial Intelligence 13, 1980, 231-278.
- [PIQ] Pique,J.F. ; Sabatier,P. An informative, adaptable and efficient natural language consultable database system Proc. ECAI-82 Orsay, France, 1982.
- [SEA] Servico de Estudos do Ambiente, Ordenamento biofisico do concelho de Sesimbra, Tratamento automatico, Relatorio preliminar, 1981.
- [WAR] Warren,D. ; Pereira,F., An efficient easily adaptable system for interpreting natural language queries, Dept. of Artificial Intelligence, University of Edinburgh, 1981.