



UNIVERSIDADE NOVA DE LISBOA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA

ORBI-AN EXPERT SYSTEM FOR ENVIRONMENTAL RESC
RCE EVALUATION THROUGH NATURAL LANGUAGE

Luís Moniz Pereira
Paul Sabatier
Eugénio de Oliveira

FCT/DI - 3/82

Marco 82

ORBI-AN EXPERT SYSTEM FOR ENVIRONMENTAL RESOUU
RCE EVALUATION THROUGH NATURAL LANGUAGE

Luís Moniz Pereira
Paul Sabatier
Eugénio de Oliveira

FCT/DI - 3/82

Março 82

ORBI - AN EXPERT SYSTEM FOR ENVIRONMENTAL RESOURCE
EVALUATION THROUGH NATURAL LANGUAGE

Luis Moniz Pereira
Paul Sabatier
Eugenio Oliveira

Departamento de Informatica
Universidade Nova de Lisboa
Quinta da Torre
2825 Monte da Caparica
Portugal

March 1982

Abstract

We describe 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, is capable of explaining the application of that knowledge to a territorial data base, and also of answering questions about its linguistics abilities. The system was developed in one year, under contract with the Portuguese Department of the Environment.

Keywords: Expert System, Natural language, Prolog, Environment.

Contents

Introduction
Domain of expertise
Protocol
System configuration
Program organization
Cognitive expertise
 Knowledge base
 Deduction, evaluation and explanation
 Data base access
 Knowledge acquisition and meta-knowledge
Natural language competence
 Linguistic competence
 The lexical analysis
 The syntactic analysis
 Full sentences
 Elliptic sentences
 Phrase structures
 The vocabulary
 The semantic analysis
 Metalinguistic competence
 Future natural language developments
Conclusions
Acknowledgements
References

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, in more or less detail, according to the needs of the user. Rules and facts can be updated by the user,

new items being automatically incorporated without any further programming.

Queries can either be in natural language (Portuguese), or result from following the options in a menu. They can be about the domain of expertise, including how the knowledge is organized, or about the system's natural language competence, thereby alleviating the user from the need to consult the manual.

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 copes 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 (PDP11/02). Use of Prolog was crucial in this respect.
- 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. Additionally, it performs extensive syntactic and semantic checks backed by appropriate error messages. Also the semantic formula is generated optimized for execution, with no need for extra planning.

DOMAIN OF EXPERTISE

=====

The domain of expertise concerns the evaluation of the biological and physical offer of the environment [SEA]. The typical region size is that of the county, which is 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 with 200 meters side. On each square the value of descriptors is digitized to integers ranging from 1 to 5, with some exceptions. 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 6.000 squares. Examples of descriptors are:

- d1 -- microclimate quality
- d9 -- risks of soil erosion
- d11 -- vegetation equilibrium

Given the observed data base of ground facts as expressed by descriptors, more abstract derived information can be obtained by defining factors and aptitudes. A factor is the result of a combination of descriptors. It has an integer value ranging from 1 to 5, or the value s, meaning it is a selective factor. Again, to each value is associated a representativity ranging from 1 to 5. The value of a factor is computed from the values of the underlying descriptors through a rule provided by experts. A rule may have several alternative cases, depending on the actual values of the descriptors. For example, a descriptor may have a value that overrides all other descriptors and then completely defines the factor by itself, or the value of the factor may be a weighed sum of all or some of the underlying descriptors, according to the actual values present. The representativity of the factor value however, is always a weighed sum of the representativities of the intervening descriptors. There are 16 factors. Examples are :

- f1 -- climatic comfort
- f3 -- resistance of soil to erosion
- f6b -- cinegetic 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 :

- a1 -- lodging
- a2 -- non-intensive agriculture
- a3 -- intensive agriculture
- a4 -- industry
- a5 -- recreation

Finally, a biophysical synthesis rule relies on the aptitude values and representativities to extract the more promising areas according to development objectives.

The knowledge of the experts is further made explicit by a large collection of explanatory texts, one for each possible value of each descriptor.

PROTOCOL (translated)

> What is the value of the aptitude 3 at the point with coordinates 103,709 ?

- At the point with coordinates 103,709,
aptitude 3 has the value very apt (with representativity 2)

Do you wish explanations about the reasons that led me to conclude that the point considered has the indicated value for the aptitude in question ?

> Yes.

- Do you wish a very detailed explanation ?

> Yes.

- At the point with coordinates 103,709,
aptitude 3 has the value 5 (very apt) (with representativity 2) because :
5 was computed as

$$\frac{(25*(6-1)+10*4+5+60*5+50)}{100} \quad \text{by substituting in}$$
$$\frac{(25*(6-F6d)+10*F3+5*F2+60*F4+50)}{100}$$

*** factor 2b has the value 4 (with representativity 5) because :

descriptor 9 has the value 4 (with representativity 5)
because according to the specialist :

The more importantly occurring aquiferous present a HIGH protection since they belong to the captive meso-pliocenic layer.

*** factor 3 has the value 4 (with representativity 4) because :
4 was computed as

$$\frac{(30*(6-1)+70*4+50)}{100} \quad \text{by substituting in}$$
$$\frac{(30*(6-D7)+70*D10+50)}{100}$$

descriptor 7 has the value 1 (with representativity 5)
because according to the specialist :

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 the value 4 (with representativity 3)
because according to the specialist :

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 the value 5 (with representativity 1) because :

descriptor 10a has the value 5 (with representativity 1)
because according to the specialist :

The soil is susceptible of highly intensive agricultural usage. The pedo-logical aptness for agriculture can be said VERY HIGH. This soil is very likely totally or partially protected by article 1 of decree 308/79.

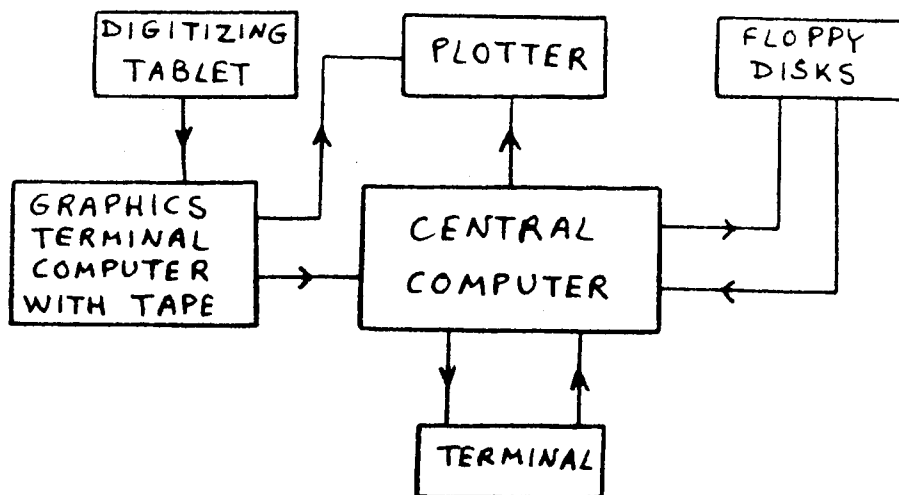
*** factor 6d has the value 1 (with representativity 1) because

descriptor 11 has the value 1 (with representativity 1)
because according to the specialist :

The vegetation balance is considered VERY LOW from the fact that natural phitocenoses are in a very degradade state and with an extremely difficult return to the potential climate.

SYSTEM CONFIGURATION

=====



System configuration

Digitizing is performed off line, using the tablet Tektronix 4956 and the graphics terminal computer (with tape) Tektronix 4052. The language Basic is used for this purpose. The interested reader is referred to [DIA] for details. Suffice it to say the user can indicate that an enclosed area has the same pair value-representativity throughout by giving its contour alone. All points inside the contour on the grid are assigned the same pair by the program. Any conflicts between contours are spotted and a decision requested from the user.

The plotter, a Tektronix 4662, can be used to monitor and furnish a hard copy of the digitizing, as well as an output device where answers to queries

can be drawn on maps.

The digitized points and their descriptors are sent from the graphics terminal computer to the central computer, which stores them on a disk in unit clauses.

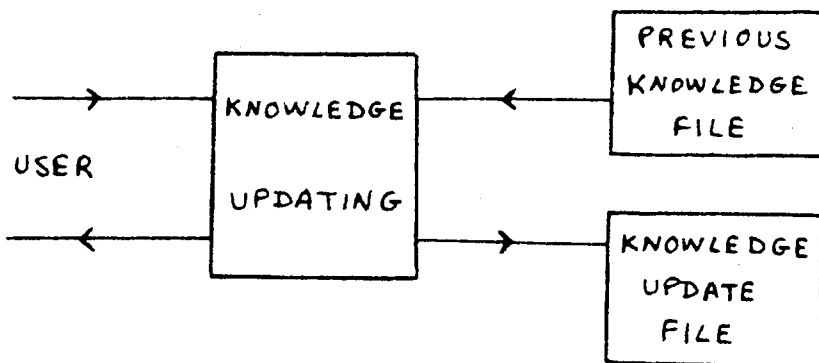
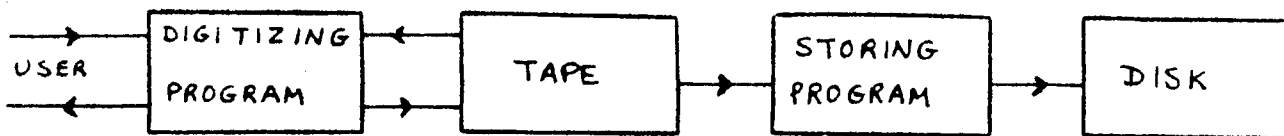
The central computer, a Heathkit 11/02 with a LSI 11 CPU, addresses 64 KB of memory, runs a Prolog interpreter [CLO] under the RT11 operating system. It communicates with a dual drive of single density floppy disks, Heathkit H27. The terminal is a DEC VT100. There are immediate plans to replace the central computer by a PDP11/23 with 256 KB, which is 3 times faster, and the floppy disks by a Winchester hard disk of 80 MB. This will allow use of the Unix operating system and the corresponding Prolog [CLO].

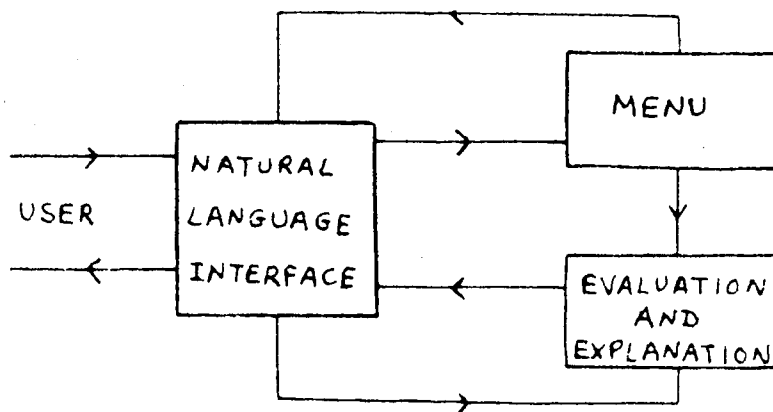
We are quite happy with the Prolog system, which externally is almost the same as DECsystem10 Prolog [CLO]. The full interpreter has a size of 17 KB, but can be trimmed down to 12 KB by removing the debugging package, and even more if necessary. The RT11 has 5 KB, roughly. We did add a few improvements to Prolog, like the 'all' predicate for obtaining sets of solutions [PE2], and a facility for loading a Prolog program module from another. This facility is indispensable because our program modules for this application do not fit together in 64 KB of memory. Communication between program modules is through files. Presently the loading of a module takes 4 secs. With PDP11/23, rather than having to load the next program from disk, it can be stored in another memory bank for immediate use.

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 three basic modules, all in Prolog, a natural language interface, a menu, and an evaluation and explanation module, linked as shown.





Each basic module itself, decomposes into further modules. They will be described in detail in the appropriate sections. For our present purpose, it is enough to point out that the three modules do not coexist in memory. Initially the user communicates with the natural language interface, either to express a question a command or a fact, or to give the initiative to the program by typing 'speak'. In this case, the menu module is loaded and the user is guided through a fixed menu of possible questions, or aided in the construction of his query. At any time, though, he can obtain the initiative back, by typing 'listen' as an answer to any program question. If this is done, then the natural language interface is reloaded; otherwise, after a query has been identified, the evaluation and explanation module is loaded, and the query read from a file.

If a question, command or statement is typed to the natural language interface, it produces the corresponding semantics, writes it on a file, and loads the evaluation and explanation module. During the evaluation, which accesses the external data base on disk of points and their descriptors, an explanation structure is generated in case the user requires it later. After the answer is produced on the terminal and/or drawn on the plotter, and an explanation eventually demanded and produced, the natural language module is reloaded, ready for the next request, or to end the session should the user type 'goodbye'.

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 aptitudes, and also to evaluate a synthesis of best use. The data base consists in a set of unit clauses, one per digitized point, of the form :

$$\text{point}(X,Y,d(v1-r1,v2-r2,\dots)).$$

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 aptitude rule and a typical factor rule are expressed by the user as terms as shown next, with appropriate assumed infix operator definitions. These rules are transformed into clauses as exemplified below for the factor rule.

$a3=Va3$ with $Ra3$ such that

$Va3=0$ and $Ra3=R$

if $Vf4=0$ where $f4=Vf4$ with R
 or $Vf6a=s$ where $f6a=Vf6a$ with R
 or $Vf6b=s$ where $f6b=Vf6b$ with R

or $Vf6c=s$ where $f6c=Vf6c$ with R

else $Va3= (25*(6-f6d)+10*Vf3+5*Vf2b+60*Vf4+50)/100$
and $Ra3= (25*Rf6d+10*Rf3+5*Rf2b+60*Rf4+50)/100$
where $f6d=Vf6d$ with $Rf6d$
and $f2b=Vf2b$ with $Rf2b$
and $f3=Vf3$ with $Rf3$
and $f4=Vf4$ with $Rf4$.

$f6c=Vf6c$ with $Rf6c$ such that

$d6a=Vd6a$ with $Rd6a$
and $d23a=Vd23a$ with $Rd23a$
and ($Vf6c=0$ and $Rf6c=R$
if $Vd6a=0$ and $R=Rd6a$
or $Vd23a=0$ and $R=Rd23a$
else $Vf6c=s$ and $R=\text{maximum}(Rd6a,Rd23a)$).

This rule is transformed into the clause :

```
f(6c, D, Vf6c-Rf6c) :- d('6a', D, Vd6a-Rd6a)
and d('23a', D, Vd23a-Rd23a)
and ( Vf6c=0 and Rf6c=R
if Vd6a=0 and R=Rd6a
or Vd23a=0 and R=Rd23a
else Vf6c=s and maximum(Rd6a,Rd23a,r f6c) ).
```

where D receives the descriptors functor of any point where $f6c$ is to be evaluated.

An example of a descriptor clause is :

```
d(9,D,D9) :- arg(13,D,D9), table(9,D9,E9).
```

```
table(9,V-R,E9) :- t9(V,E9), !.
```

```
t9(I,msg(m)).
```

stating that the pair value-representativity of descriptor 9 is bound to $D9$ by taking the 13th argument from the descriptors functor to which D is bound. Predicate `table` then binds the explanation variable $E9$ to a term of the form `msg(m)`, according to the value of V . Where 'm' is a file name and 'msg' a system predicate developed by us used later to output on the terminal the explanatory message stored in file 'm', elucidating what does it mean for descriptor 9 to have value V . Note, however, that $E9$ has but one occurrence. This is because the knowledge base clauses are not called directly. They are used under control of a special interpreter. Hence the special format the factor and aptitude rules have, which make it easier for the program to exhibit them to the user, by translating them back to the original notation they were entered in, more congenial to the user.

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 sub-goals 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),  
    explanation(Answer, Explanation).
```

where ^ is a garbage collection predicate that recovers the space used in the deduction phase :

```
^G:- G, assert(G), fail.  
^G:- retract(G), trimcore.
```

For a more precise understanding consider the clause :

```
deduction(all(X,P,S), all(explain(X,E),NDP,S), E):- !, deduction(P,NDP,E).
```

takes a query of the form 'all' and transforms it into a new query, where P is replaced by its nondeterminate parts NDP, and the explanation E that results from executing the deterministic parts of P is attached to each object X.

The next clause 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 E, 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. Hence :

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

where solve(G) :- G.

is necessary for the '!' in the 'or' not to affect other parts of the query.

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 query has at most one call p(X,Y,D), we can use the file pointer to keep track of the last clause read :

```
point(X,Y,D) :- see(data), seen, /* make sure the file is closed */  
                see(data), /* open it */  
                repeat, /* create a backtrack point */
```

```

        read(P),                /* read next unit clause      */
(P=end_of_file, !, seen, fail ; /* no more solutions        */
(var(X);var(Y)), P=point(X,Y,D) ; /* pick up a solution        */
nonvar(X), nonvar(Y), P=point(X,Y,D),!, /* pick up single solution  */
seen ) . /* so close file      */

```

KNOWLEDGE ACQUISITION AND META-KNOWLEDGE

=====
The knowledge acquisition program, still under development, aims at helping the authorized user to reformulate the knowledge base, in particular through the use of a congenial 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.

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 yet again for writing specialized interpreters [PE3] to achieve the control regimes required by knowledge processing. Reporting on these features is left for future publications.

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 optimally ordered directly evaluable list of Prolog goals. For example, the query :

>Quais sao os pontos tendo o mesmo valor para a aptidao agricultura intensiva?
(Which are the points having the same value for the aptitude intensive agriculture?)

will be transformed into the Prolog expression (cf. [PE2] for a definition of 'all') :

```

all ( V-S ,
      all ( [ponto:(X,Y)] , (p(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-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. That is done by indexing on the word entries of ORBI's dictionary. For example, associated with the word

pontos (points)

we shall have the following lexical information, in the form of the unit clause :

pontos(common_noun(ponto,pontos,masculine,plural)).

according to the scheme :

<word>(<lex.cat.><deep morph.>,<surface morph.>,<gender>,<number>)).

If there is no entry for a word it is reported unknown, unless it is detected as part of a compound name :

compound_name([intensive,agriculture!L],L,number(3)).

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 [PER]) 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 transportable to other domains. It has indeed been done [PE1].

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 pertaining to ORBI's domain of expertise.

We have devoted a lot of energy to incorporate elliptic and extraposed structures in ORBI's linguistic competence, because they are essential to a good language interaction. A natural language interface unable to understand sentences involving these structures is not truly natural.

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. Examples :

- > Em cada ponto, quais sao a factor de cada aptidoes ?
(In each point, what are the value of the factor of each aptitudes?)
- * Falta de acordo entre 'a' e 'factor', entre 'sao' e 'factor', entre 'cada' e 'aptidoes' !
(Lack of agreement between 'the' and 'factor', between 'are' and 'factor', between 'each' and 'aptitudes'!)
- > Da-me o valor da aptidao resistencia dos solos 'a erosao dos pontos cujo descriptor potencial do recreio e' superior a 2 !

(Give me the value of the aptitude tolerance of soils to erosion of the points whose the descriptor potential of recreation is greater than 2!)

- * 'resistencia dos solos 'a erosao' e' um factor, nao uma aptidao !
('tolerance of soils to erosion' is a factor, not an aptitude!)

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 is the case, for example, when a wrong complement for a verb, is encountered. In that case, OREI always informs the user. It also provides for a more compact grammar. Examples :

- > Sera' que a regioao de cada ponto tem o valor muito apto para a aptidao 3 ?
(Is it the case that the region of each point has the value very apt for the aptitude 3?)
- * 'ponto' nao poder ser complemento de 'regiao' neste contexto !
('point' can't be a complement of 'region' in this context!)
- > Quantos factores sao superiores a cada ponto ?
(How many factors are greater than each point?)
- * 'factores' nao pode ser sujeito do verbo 'ser' com o complemento 'ponto' !
('factors' can't be the subject of the verb 'to be' with the complement 'point'!)

Yes/no questions have the structure :

(sera' que) NP_SUBJECT VP ?

where 'sera' que' ('is it the case') is a facultative (the facultative elements are bracketed) yes/no operator in Portuguese. Examples of yes/no questions :

- > Sera' que cada ponto cujo factor resistencia ao incendio e' igual a nao tem nenhuma aptidao superior a 3 ?
(Is it the case that each point whose the factor resistance for fire is equal to 2 has no aptitude greater than 3?)
(Note the double negation in Portuguese)
- > Alguns pontos que tem o mesmo valor para a aptidao industria tem o factor riquezas do subsolo entre 1 e 3 ?
(Some points that have the same value for the aptitude industry have the factor riches of the subsoil between 1 and 3?)
- > A media dos valores da aptidao 2 dos pontos da regioao 101,106;120,140 e' inferior 'a media da aptidao 3 dos pontos da regioao 202,215;240,260 ?
(The average of the values of the aptitude 2 of the points of the region 101,106;120,140 is lower than the average of the aptitude 3 of the points of the region 202,215;240,260?)

Wh questions may have the following structures :

WH_NP_COMPLEMENT NP_SUBJECT (NEGATION) VERB ?

WH_NP_COMPLEMENT (NEGATION) VERB (DOUBLE NEGATION) NP_SUBJECT ?

Examples :

- > Que valor tem cada aptidao dos pontos cujo descriptor riscos de

erosao e' 5 ?

(Which value has each aptitude of the points whose the descriptor risks of erosion is 5?)

- > A que factores cada aptidao de cada ponto que existe e' igual ?
(To what factors is each aptitude of each point that exists equal to?)
- > Qual e' a regioao que nao tem nenhum ponto com o valor 1 para a aptidao industria ?
(What is the region that do not have any point with the value 1 for the aptitude industry?)
(Note the double negation in Portuguese)
- > Que valor e' que tem cada descriptor da aptidao 2 dos pontos tendo 2 factores iguais a 1 ?
(What value does have each descriptor of the aptitude 2 of the points having 2 factors equal to 1?)
- > O que e' a aptidao A2 ?
(What's the aptitude A2?)

Or the structure :

WH_NP_SUBJECT VP ?

Examples :

- > Que pontos tem a aptidao agricultura intensiva com valor muito apto ?
(Which points have the aptitude intensive agriculture with value very apt?)
- > Que factores sao superiores ao valor da aptidao 2 do ponto 11,40 ?
(Which factors are greater than the value of the aptitude 2 of point 11,40?)
- > Quantos pontos tendo o mesmo valor para cada aptidao existem ?
(How many points having the same value for each aptitude exist?)
- > Porque tem a aptidao industria para o ponto 121,57 o valor nao apto ?
(Why has the aptitude industry for the point 121,57 the value not apt?)

Commands have the structure :

(IMPERATIVE_TRANSITIVE_VERB) NP_OBJECT !

Examples :

- > Da-me a media dos valores do factor conforto climatico dos pontos que tem o mesmo valor para a aptidao 4 !
(Give me the average of the values of the factor climatic comfort of the points that have the same value for the aptitude 4!)
- > O valor de cada factor do ponto 205,102 !
(The value of each factor of point 205,102!)
- > Os pontos da regioao 105,206;118,219 que tem 2 aptidoes com o valor 1 ?
(The points of the region 105,206;118,219 that have 2 aptitudes with the 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. Examples :

- > Em cada ponto, quais sao os factores de cada aptidao que sao superiores a 2 ?
(In each point, what are the factors of each aptitude which are greater than 2?)
- > Quanto vale, para o ponto 28,105, o descriptor permeabilidade do substracto ?
(How much is the value, for point 28,106, of the descriptor permeability of the subsoil?)
- > Sera' que, em alguns pontos, a aptidao A3 e' igual ao factor F6 ?
(Is it the case that, in some points, the aptitude A3 is equal to the factor F6?)

Each extraposition is first syntactically analysed and concatenated to the NP_SUBJECT.

ELLIPSIS SENTENCES

Ellipsis may occur inside the same sentence or from sentence to sentence in the dialogue. (We shall note elliptic strings between [and] and the point where they are missing with [...]).

In the first case, ORBI deals with ellipsis of subjects and/or verbs in coordinate sentential clauses. For example :

- > [O ponto 69,103 tem] o valor apto para a aptidao industria e [...] o valor do factor resistencia do solo 'a erosao inferior ao F4 ?
([Has the point 69,103] the value apt for the aptitude industry and [...] the value of the factor resistance of the soil for erosion greater than the factor 4?)
- > [Cada ponto] tendo o F4 inferior a 3 ou [...] nao tendo o F12 inferior a 2 tem o valor muito apto para a aptidao habitat concentrado ?
([Each point] having the factor 4 lower than 3 or [...] not having the factor 12 lower than 2 has the value very apt for the aptitude concentrated lodging?)

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, preceded by #, and introduced by 'E' (and).

Ellipsis between yes/no questions

- ellipsis of the whole of the NP subject.

- > [Sera' que a aptidao habitat concentrado dos pontos cujo descriptor D8 e' igual a 3], e' entre 1 e 4 ?
([Is it the case that the aptitude concentrated lodging of the points whose the descriptor D8 is equal to 3], is between 1 e 4?)

E [...] e' igual a 3 ?
(And [...] is equal to 3?)

- ellipsis of the whole of the VP. Example :

> Sera' que a aptidao habitat concentrado dos pontos cujo descriptor piroresistencia vegetal e' igual a 3 [e' entre 1 e 4] ?
(Is it the case that the aptitude concentrated lodging of the points whose descriptor vegetal resistance for fire is equal to 3 [is between 1 and 4]?)

E os factores do ponto cuja aptidao industria e' igual a 5 [...] ?
(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 :

> [O descriptor D10] dos pontos na regioao 91,707;103,900 que e' entre 3 e 5, [e inferior ao descriptor escoamento pluvial superficial do ponto 26,40] ?

([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 the point 26,40]?)

E [...] dos pontos cuja aptidao 2 e' superior a 3 [...] ?
(And [...] of the points whose the aptitude 2 is greater than 3 [...] ?)

> [O descriptor D8] dos pontos na regioao 91,707;103,900 [que e' entre 3 e 5, e' inferior ao descriptor piroresistencia vegetal do ponto 26,40] ?
([The descriptor D8] of the points in the region 91,707;103,900 [that is between 3 and 5, is lower than the descriptor vegetal resistance for fire of the point 26,40]?)

E [...] na regioao 100,109;140,150 [...] ?
(And [...] in the region 100,109;140,150 [...] ?)

> O descriptor D6 [dos pontos da regioao 91,707;103,900 que e' entre 3 e 5 e inferior ao descriptor D2 do ponto 26,40] ?
(The descriptor D6 [of the points of the region 91,707;103,900 that is between 3 and 5 is lower than the descriptor D2 of the point 26,40] ?)

E o factor riquezas do subsolo [...] ?
(And the factor riches of the subsoil [...]?)

Ellipsis between wh questions

- ellipsis of the whole of the WH_NP. Example :

> Que [pontos da regioao 23,78;84,98] tem o valor apto para a aptidao A1 ?
([Which points of the region 23,78;84,98] have the value apt for the aptitude A1?)

E [...] tem o valor 5 para a aptidao industria ?
(And [...] have the value 5 for the aptitude industry?)

- ellipsis of the whole of VP. Example :

> Que pontos da regioao 54,57;78,80 [tem o valor nao apto para a aptidao agricultura intensiva] ?

(Which points of the regioao 54,57;78,80 [have the value not apt for the aptitude intensive agriculture]?)

E que pontos tendo o descriptor escoamento pluvial superficial com o valor 5 [...] ?
(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 :

> [Que descriptors] da aptidao 3 no ponto 84,12 que sao superiores a 2 [sao iguais aos factores do ponto 32,10] ?
([Which descriptors] of the aptitude 3 in the point 84,12 that are greater than 2 [are equal to the factors of the point 32,10] ?)

E [...] da aptidao 4 dos pontos da regioao 45,67;12,34 [...] ?
(And [...] of the aptitude 4 of the points of the region 45,67;12,34 [...] ?)

> [Que descriptors] do factor 5 [no ponto 14,36 que sao entre 2 e 4, sao iguais aos factores do ponto 45,56] ?
([Which descriptors] of the factor 5 [in the point 14,36 that are between 2 and 4 are equal to the factors of the point 45,56] ?)

E [...] da aptidao recreio [...] ?
(And [...] of the aptitude recreation [...] ?)

> Que factores [da aptidao industria no ponto 67,54 que sao inferiores a 4 sao superiores ao factor riquezas geograficas do ponto 45,78] ?
(Which factors [of the aptitude industry in the point 67,54 that are lower than 4 are greater than the factor geologic resources of the point 45,78]?)

E que descriptors [...] ?
(And which descriptors [...]?)

Ellipsis between commands

- ellipsis of the verb. Example :

> [Da-me] o valor do factor F8 do ponto 23,175 ?
([Give me] the value of the factor F8 of the point 23,175?)

E [...] os pontos cujo factor F8 e' entre 3 e 5 ?
(and [...] the points whose the factor F8 is between 3 and 5?)

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

> [Os descriptors] da aptidao 4 do ponto 184,78 que sao superiores a 2 ?
([The descriptors] of the aptitude 4 of the point 184,78 that are greater than 2?)

E [...] dos pontos da regioao 91,45;76,62 ?
(And [...] of the points of the region 91,45;76,62?)

> [Os descriptors] da aptidao 6 [de cada ponto] que sao maiores que 3 ?
([The descriptors] of the aptitude 6 [of each point] that are greater than 3?)

E [...] de cada factor [...] ?
(And [...] of each factor [...] ?)

> Os descriptors [da aptidao industria do ponto 64,81 que sao menores que 2] ?
(The descriptors [of the aptitude industry of the point 64,81 that are lower than 2]?)

E os factores [...] ?
(And the factors [...]?)

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,

> Que descriptors da aptidao agricultura intensiva no ponto 56,78 que sao superiores a 2, sao iguais aos factores do ponto 12,95 ?
(Which descriptors of the aptitude intensive agriculture in the point 56,78 that are greater than 2, are equal to the factors of the point 12,95?)

ORBI will record the morphological strings corresponding to the structures

[que descriptors]
([which descriptors])
[da aptidao agricultura intensiva]
([of the aptitude intensive agriculture])
[no ponto 56,78]
([in the point 56,78])
[que sao superiores a 2]
([that are greater than 2])
[sao iguais aos factores do ponto 12,95]
([are equal to the factors of the point 12,95])

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 beginning with 'e' ('and') as in

E no ponto 65,78 ?
(And at the point 65,78?)

ORBI tries to fill in the gaps of the sentence with the different strings recorded in certain pre-specified ways, in an attempt to find a grammatical analysis. In the above example, the good interpretation will be

> Que descriptors da aptidao agricultura intensiva no ponto 65,78 que sao superiores a 2, sao iguais aos factores do ponto 12,95 ?
(Which descriptors of the aptitude intensive agriculture in the point 65,78 that are greater than 2, are equal to the factors of point 12,95?)

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 are suspended during the analysis.

Of course, an elliptical sentence may be followed by another one. For example :

> [Que factores] do ponto 21,66 [que sao diferentes de 1 sao iguais ao descriptor riscos de erosao] ?
([Which factors] of the point 21,66 [that are different from 1 are equal to the descriptor risks of erosion] ?)

E [...] do ponto 45,67 [...] ?
(And [...] of the point 45,67 [...] ?)

E [...] dos pontos da regioao 68,23;89,20 [...] ?
(And [...] of the points of the region 68,23;89,20 [...] ?)

E [...] do ponto com o valor pouco apto para a aptidao recreio [...] ?
(And [...] of the point with the value less apt for the aptitude recreation [...] ?)

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)

Examples :

[o mesmo valor para o descriptor equilibrio da vegetacao nos pontos da regioao 23,41;67,90 que e' entre 2 e 5]
([the same value for the descriptor vegetal balance in the points of the region 23,41;67,90 that is between 2 and 5])

[os pontos cuja aptidao habitat concentrado e' superior a 4, cuja aptidao A2 e' inferior a 3, ou tendo o factor conforto climatico diferente do factor D8]
([the points whose the aptitude concentrated lodging is greater than 4, whose the aptitude A2 is lower than 3, or having the factor climatic comfort different from the factor D8])

[a media dos valores do factor F12 dos pontos que tem o valor pouco apto para cada aptidao]
([the average of the values of the 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

Examples :

[nao tem a aptidao 3 inferior a 3]
([not have the aptitude 3 lower than 3])

[e igual ao factor f5]
([is equal to the factor f5])

ORBI also understands their conjunction or disjunction, as in

[sao iguais a 5 ou tem o factor 2 superior ao descriptor D9]
([are equal to 5 or have the factor 2 greater than the descriptor D9])

[tem a aptidao agricultura nao intensiva com o valor apto, a aptidao 5 igual a 1 e nenhum factor da aptidao 4 entre 1 e 2]
([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 are 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 in two subparts : a metalinguistic part and a non-metalinguistic one.

The LINGUISTIC part contains :

Determiners

o, os, a, as (the) ;
um, uns, uma, umas (a/an) ;
nenhum, nenhuns, nenhuma, nenhuma (no) ;
algum, alguns, alguma, algumas (some) ;
todo, todos, toda, todas (all/every) ;
qualquer, quaisquer (any) ;
cada (each) ;
0, 1, 2, 3, ...

Prepositions

de (of/from) ; a (to) ; em (in/on) ; com (with) ; sem (without) ;
para (for).

Contractions

ao, aos, as (to+the) ; no, nos, na, nas (in/on/at + the) ; num, nuns, numa, numas (em + the) ; do, dos, da, das (of + the) ; dum, duns, numa, dumas (of + a/an).

Common verbs

ser (to be), e' (is), sao (are) ;
ter (to have/possess), tem (has), tem (have), tendo (having) ;
existir (to exist), existe (exists), existem (exist) ;
ha' (there is).

Relative and interrogative pronouns

que (that/which) ; porque (why) ; qual, quais (which) ;
cujo, cuja (whose) ; quanto, quantos, quanta, quantas (how much/many).

Phrase and sentence connectors and negation

e (and) ; ou (or) ; mas (but) ; nao (not).

Prelocutory expressions

sera' que (is it the case that) ; e' que (do(emphasis)) ;
da-me, de-me (give-me) ;
diga-me (tell me) ;
desenhe (draw) ;
adeus (good-bye).

The METALINGUISTIC vocabulary contains words used to ask ORBI about its

linguistic competence. In particular :

vocabulario[s] (vocabulary[ies]) ; palavra[s] (word[s]) ; verbo[s] (verb[s]) ; nome[s] (noun[s]) ; artigo[s] (article[s]) ; adjetivo[s] (adjective[s]) ; preposicao[oes] (preposition[s]) ; pronomes[s] (pronoun[s]) ; conjuncao[oes] (connector[s]) ; sujeito[s] (subject[s]) ; complemento[s] (complement[s]) ; ...

THE SPECIFIC VOCABULARY, especially nouns, abbreviations, names of entities verbs, adjectives and adverbs, refers to ORBI's domain of application. We give here only a subset of this vocabulary.

Nouns, abbreviations, expressions

regiao[oes], rg (region[s]) ; ponto[s], p (point[s]) ; aptidao[oes], A (aptitude[s]) ; factor[s], F (factor[s]) ; descriptor[es], D (descriptor[es]) ; representatividade[s], rp (representativity[ies]) ; valor[es] (value[s]) ; media[s] (average[s]) ; agricultura intensiva (intensive agriculture) ; recreio (recreation) ; industria (industry) ; riscos de erosao (risks of erosion) ; habitat concentrado (concentrated lodging) ; riquezas geologicas (geologic riches) ;

Adjectives

mesmo, mesmos, mesma, mesmas (same) ;

Verbs and adverbs

valer (to be worth), vale, valem ;
ser/e'/sao igual[ais]/superior[es]/inferior[es]/maior[es]/menor[es]/
diferente[s]/entre .. e .. (to be/is/are equal/greater/lower/different/
between .. and ..)

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)) ---> not(SNP , not VP)
others : each([X,Y] , (SNP,VP)) ---> all([X,Y] , (SNP,VP) , S)

Noun phrases may feature embeded quantifiers, whose scope is reversed relative to order of appearance. Futhermore, they are reduced to a single one, where the order of quantified variables expresses scoping :

each X of each Y P ---> each([Y,X] , P)

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

'same', 'average' and 'how many' pose special scoping problems which we will not go into.

There is no treatment of presuppositions (the definite and indefinite

articles are similarly treated), since they are unnecessary in the QRBI domain.

Existential quantification only needs to be made explicit in noun phrases with 'mesmo' ('same'), by using the 'same' construction of the 'all' system predicate [PE2].

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 $\text{not}(_ , \text{not } _)$ 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 $\text{not}(_ , \text{not } _)$:

V + each(X , NP)	---->	not(NP , not V)
not V + each(X , NP)	---->	not(NP , V)
V + NP	---->	(NP , V)
not V + NP	---->	(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.

FULL SENTENCES

Full sentence structures are yes/no questions, wh questions and commands, with possible extrapositions.

METALINGUISTIC COMPETENCE

Due to the wealth and variety of structures, vocabularies and uses involved in natural language, an 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 false presuppositions; but also incorrect complementations of nouns and incompatible subjects and complements of verbs.

In fact all these types of diagnosis are not sufficient to ensure a good natural language interaction. In some systems, the user endeavours to consult a reference manual, generally not very successful because overly complex.

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, ORBI can answer to the following questions :

> Da-me o vocabulario !
(Give me the vocabulary!)

> A palavra 'factor' existe ?
(Does the word 'factor' exist?)

as preposicoes
> Da-me os nomes que existem !
os verbos

(Give me the prepositions, the nouns, the verbs, ... that exist !)

> Quais sao os possiveis complementos do substantivo 'ponto' ?
(What are the possible complements of the noun 'point'?)

> Quais sao os possiveis sujeitos e complementos de cada verbo ?
(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 (and any grammar of a natural language interface) to be consulted as a linguistic database for giving examples of queries with certain required words, and for expliciting the accepted structures [SAB].

FUTURE NATURAL LANGUAGE DEVELOPMENTS

Concerning ORBI's linguistic competence, future developments will deal with

- implicit typing of an entity (aptitude, factor, descriptor, ...) through the mention of its specific designation only. For example :

a permeabilidade do substracto
(the permeability of the subsoil)

instead of

o descriptor permeabilidade do substracto
(the descriptor permeability of the subsoil)

- conjunction and disjunction of designations. Examples :

os factores 9 e 12
(the factors 9 and 12)
a aptidao industria ou recreio
(the aptitude industry or recreation)

- conjunction of the NP subject . Example :

a aptidao 5 e o factor 12 do ponto 21,66 tem o mesmo valor
(the aptitude 5 and the factor 12 of point 21,66 have the same value)

- interesting expressions and words as :
 'o maior' ('the greatest'), 'o menor' ('the least')
 'so', 'apenas' ('only', 'just')
- pronominalization within the same sentence and sentence to sentence.
- enlargement of extrapositions and ellipsis.

In the matter of ORBI's metalinguistic competence, we expect to provide

- definition of synonymous (and abbreviations) through a declarative sentence. For example :
 'erosao' e' um sinonimo de 'riscos de erosao'
 ('erosion' is a synonym of 'risks of erosion')
 'RPMA' e' uma abreviatura de 'resistencia 'a poluicao dos
 mantos aquiferos'
 ('RPMA' is an abbreviation for 'resistance of water-tables
 topolution')
- sentence exemplification, such as
 Da-me 2 perguntas sim/nao com o verbo 'valer' e com o
 substantivo 'aptidao' como sujeito !
 (Give me 2 yes/no questions with the verb 'to be worth'
 and the noun 'aptitude' as subject!)
- explicit diagnosis about incomplete syntactic parsing, exhibiting the
 largest analysable substrings, thanks to an optional sentence debugging
 package consultable by the user.

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 lan-
 guage, knowledge base, explanation facility, relational database, me-
 ta-knowledge, and interpreters for specialized control.
- its compactness of expression, coupled with a good implementation, al-
 low 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 the Junta Nacional de Investigacao Cientifica for their
 financial support in addition to our contract with the Servico de Estudos do
 Ambiente of the Secretaria de Estado do Ambiente e Qualidade de Vida.

Our gratitude goes also to Antonio Porto and Miguel Filgueiras for their
 help with improving the Prolog system.

In no way are we thankful to the Instituto Nacional de Investigacao Cienti-
 fica for not supporting this project.

REFERENCES

=====

- [CLO] Clocksin,W. ; Mellish,C., Programming in Prolog, Springer-Verlag, 1981
- [DIA] Dias,V. ; Pereira,L.M., Digitalizacao de dados territoriais, Departamento de Informatica, Universidade Nova de Lisboa, Lisbon, 1982.
- [OLI] Oliveira,E., A representacao do conhecimento num sistema pericial visto como uma base de dados logica, 2 Congresso Portugues de Informatica, Lisbon, 1982.
- [PE1] Pereira,L.M. ; Porto,A., A Prolog implementation of a large system with natural language on a small machine, Submitted to the First International Conference on Logic Programming, Marseille, 1982.
- [PE2] Pereira,L.M. ; Porto,A., All solutions, Logic Programming Newsletter n.2, Autumn 1981.
- [PE3] Pereira,L.M., Logic control with logic, Submitted to the First International Conference on Logic Programming, 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, Proceedings of ECAI-82 Orsay, France, 1982
- [SAB] Sabatier,P., Transparent natural language interfaces, in preparation.
- [SEA] Servicos 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.