NATURAL-LANGUAJE ACCESS TO THE DIANEGUIDE DATABASE

José VEGA & Antoine OGONOWSKI

GSI-ERLI
1, Place des Marseillais
94220 CHARENTON (FRANCE)
tel : (1)48 93 81 21; Fax : (1)43 75 79 79
E-mail : jose.vega@erli.gsi.fr

This application has been developed by GSI-ERLI in collaboration with ECHO (provider of the database Dianeguide) within the frame of a contract CEC.

ABSTRACT

This paper describes an information retrieval system called NLA which analyses natural language queries. The languages used here are : English, Italian & French. The query analysis is performed in two stages : the comprehension stage, where the user's request is translated into an unambiguous internal form using linguistic analysers, and the search stage where a documentary machine uses the internal representation in order to map it into facts (predicates) recorded in a knowledge base. The documentary machine then tries to broaden the relevant concepts (Normalised Terms [NT]) of the query in order to find other NT's semantically close to the same domain. Finally, replies are worked out by calculating answers to each constraint (predicate-NT pair) through applicative rules and cross-referencing search strategies using Boolean equations.

This system is also used to classify and index the records (files) in the Dianeguide database. The internal representation of indexed records is a dependency tree where the leaves are predicates similar to those produced by the natural language interface.

INTRODUCTION

The aim of this system is to give fast and easy access to the users of Dianeguide. Dianeguide is a European directory of on-line information in the form of a database. It gives details of European databases, all those involved in the information market and about the organizations responsible for hosting and producing these databases.

Natural language access must solve the problems inherent in document retrieval - namely the need to be familiar with the structure of the data, with the control values used to formulate interrogation constraints (especially in mixed databases [textual, factual] like this one), with the syntax of the command language, and even with the approximate content of the database. The Natural Language interface allows the user to formulate in his own language a multi-criteria request of average complexity, and to obtain answers in the form of entries which best meet those criteria, by means of a strategy which is guided by the system.

---

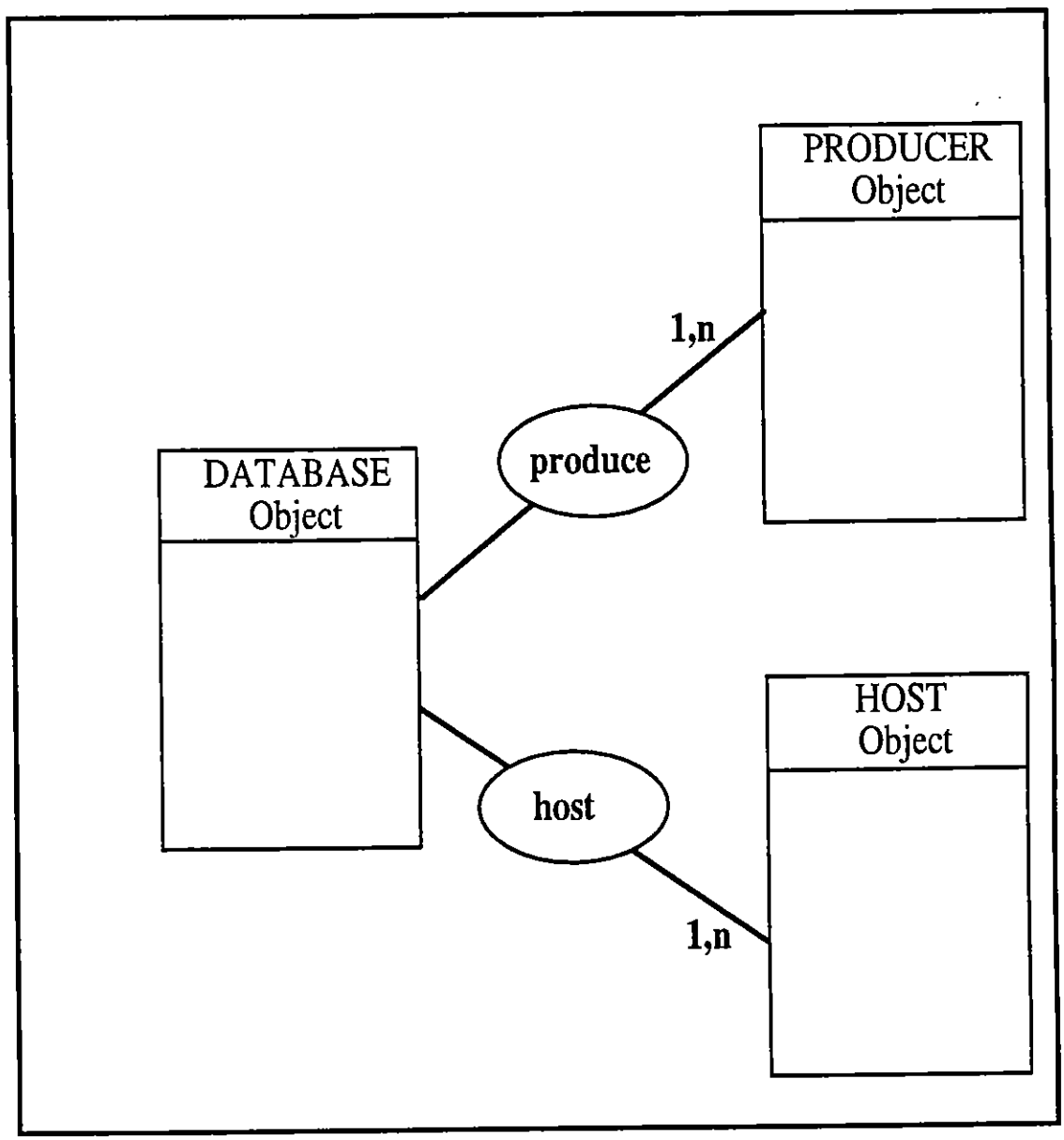
[1] For information : the French dictionary contains about 70.000 words (simple or compound), the English dictionary about 40.000 words and the Italian dictionary about 60.000 words.

SYSTEM DESCRIPTION

THE DATA

DIANEGUIDE DATABASE STRUCTURE

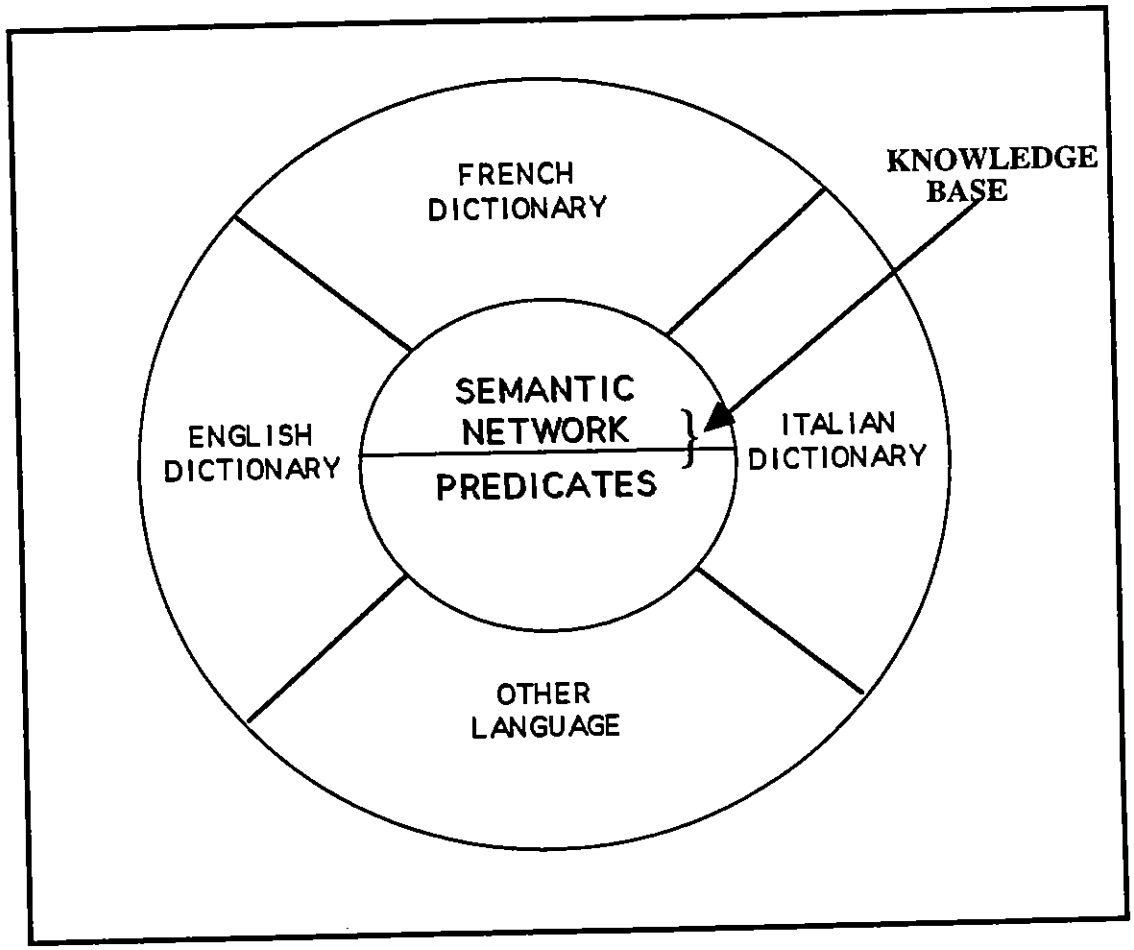
The database conceptual model contains three objects: «databases», «hosts» and «producers». Each of these objects is endowed with determining attributes: «name», «type», «language», «activity», etc. The values of the attributes are either factual (number of records, dates, ...) or textual (an abstract about the subject of a database). Interactions between the conceptual objects are represented by relationships («produce» and «host»).



DICTIONARIES & KNOWLEDGE BASE

It is possible to access the system in three languages : Italian, French or English.

The system uses two kinds of data : a general dictionary of the language[1] for the linguistic aspects and an applicative knowledge base (see figure below).



The knowledge base includes the Normalised Terms (NT : descriptors) of the application and the applicative semantic network. This information is represented in a surface language independent form called «pivot representation».

In concrete terms, the pivot representation is a notion materialised by a number differentiates between a surface word and its semantic representation. For example : Spanish will have 2 pivot numbers (in our context) —> Spanish [language] & Spanish [Nationality].

The knowledge represented by the conceptual model is incorporated into the knowledge base (KB) by using a formalism that employs logical proposition (predicates) expressing elementary semantic knowledge of the application. For example :

produce producer database (x, y);

contain database kind_of_data (y, z)

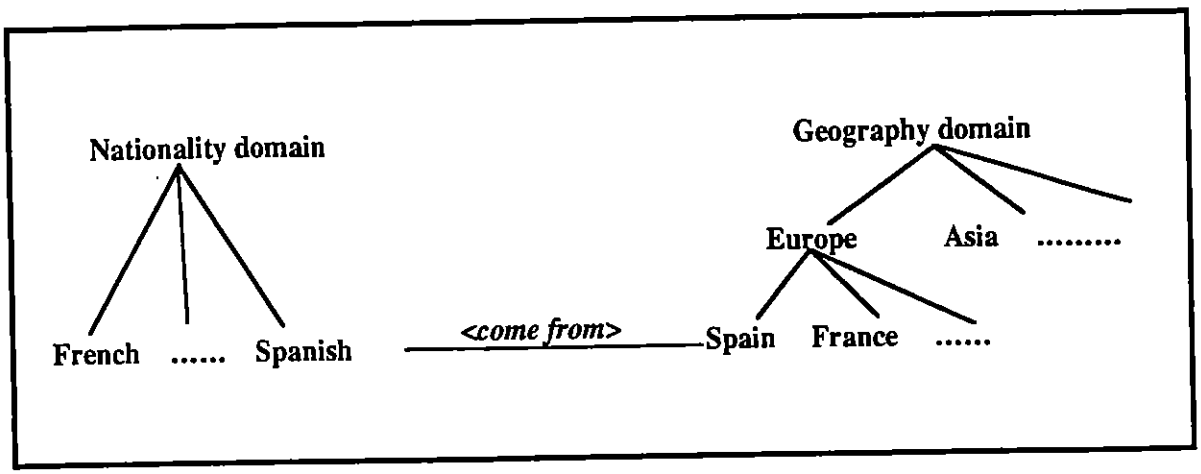
the first predicate is a conceptual interaction (through the verb 'produce') between the objects DATABASE and PRODUCER.

in the second predicate, **kind_of_data** is an attribute of the object database.

With these 2 predicates, the system can analyse the following user's query : «Who are the producers of bibliographical databases ?» Where bibliographical is an instance of **kind_of_data**.

Beside this information, we find in the KB a thesaurus where the leaves are the NT's described above.

This thesaurus is a network organised in sub-networks where each of them is attached to a semantic domain. For example :



THE LINGUISTIC ANALYSIS OF THE REQUEST

The linguistic analysis includes three main aspects : the morpho-lexical analysis, the syntactical analysis and the semantic analysis.

The morpho-lexical analysis

The aim of the morpho-lexical analysis is to identify unequivocally all the words of the request.

Morphological analysis of a request calls for a dictionary containing most of the words likely to be typed in by the user. The words in this lexicon must be classified (grammatical categories and sub-categories) in simple or compound words. They are stored in simple typeface (unaccented capitals), as the system must be able to analyse requests regardless of typography. The words are recorded in a standardised form : in French, for example, nouns are in the singular, adjectives in the masculine singular and verbs in the infinitive.

Besides the dictionary, this phase of analysis uses rules for morpho-lexical analysis. These recognise words by their graphic variants, inflections and near synonyms. Other rules serve to determine the category of words that are not recorded in the dictionary by analysing their graphic forms (recognition of numbers, dates, etc.). A final set of rules helps resolve ambiguities of grammatical categorisation : these are used to assign the correct category to words that might belong to several categories.

The syntactical analysis

The analyser in this system uses a mixed grammar : a syntactical grammar and a role grammar.

The first one is a deterministic analyser where no temporary structures are built or information deleted during the course of a parse.

This option has been taken because speed is crucial in this kind of application where thousands of people can interrogate the system at the same time.

Syntactical rules are used to test the syntactical validity of each language and also some trasformations like a distribution, passive structure, ...

When a predicate-argument structure is identified the parser sends it to the role grammar.

The meaning of role grammar (or case grammar) is different here to the meaning given in Fillmore and other literatures. The aim of the role grammar here is to map the syntactical description of each word of the string into the correct roles that it plays in the query. This is performed using the information in the knowledge base.

When this parser has verified the semantic coherence information, it generates a surface predicate.

The surface predicates will be used by the semantic analysis.

At this level, the analysis is done independently of all languages; the system works only with the notions (pivot representation) of the knowledge base.

In this application there are two main roles : Object which is a conceptual class defining the «actors» of the application and Attribute which is a set of characterisations of the Objects. These Attributes take their values in different semantic domains (see knowledge base section above) which can be open, closed or semiopen (for example : date domain, colour domain, proper nouns).

Example of surface predicates :

«Who produces bibliographical databases on landscape management and biotechnology in English ?»

The analyser produces the following surface predicates :

1- ((CNN = to produce) (ARG1 = who) (ARG2 = database))

2- ((CNN = to be) (ARG1 = database) (ARG2 = bibliographical))

3- ((CNN = in) (ARG1 = database) (ARG2 = English))

4- ((CNN = on) (ARG1 = database) (ARG2 = landscape management))

5- ((CNN = on) (ARG1 = database) (ARG2 = biotechnology))

Where : CNN = Connector (is a semantically class of verbs or prepositions) ; ARG1 = is an object; ARG2 = is an : object, attribute or value

The semantic analysis

This phase compares the surface predicates found above with the «deep predicates» recorded in the knowledge base. The aim of this comparison is to filter the pertinent forms and to build a logical representation of the user's query .

This comparison is carried out using the transformational rules. These rules are activated when the conceptual features (object, attribute, value) and syntactical categories of each surface predicate element match with each element of the transformational rule.

When the match is made, the transformation formulae take each element of the surface predicate and try to match these elements with the deep predicates after exploring a set of relationships in the knowledge base.

The types of relationship which we can find at this level are : <is a>; <kind of>; <interrogate by>; <domain of>; ...

Example :

The predicate **1-** is validated (after transformations [who becomes producer, in the context of to produce]) via the deep predicate :

6- produce producer database (x, y)

The predicate **2-** is validated (after transformations [the adjective 'bibliographical' becomes 'bibliographic' which is the Normalised Term (NT)]) via the deep predicate :

7- contain database kind_of_data (y, = bibliographic)

The predicate **3-** is validated (after transformations) via the deep predicate :

8- to_have database language (y, = English)

The predicate **4-** is validated (after transformations [the compound name 'landscape management' becomes 'landscape planning' which is the Normalised Term (NT)]) via the deep predicate :

9- to_deal database subject (y, = landscape planning)

The predicate **5-** is validated (after transformations) via the deep predicate :

10- to_deal database subject (y, = biotechnology)

The set of deep predicates is a 'predicate tree'.

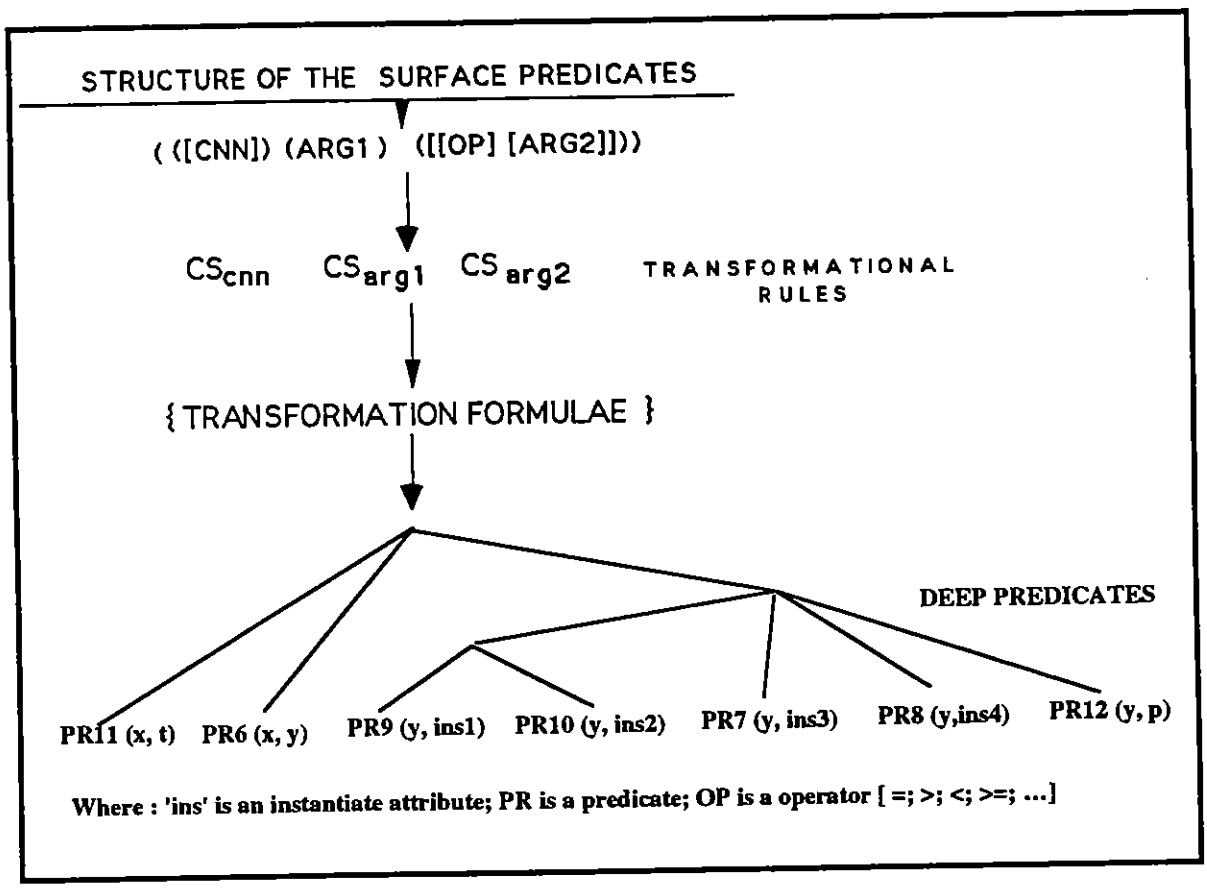
After the production of deep predicates, some complementary predicates will be generated automatically.

Indeed, when the user asks about 'producers' and 'databases', the system will understand that the user also wants the names of these objects. The system will then produce two other predicates :

11- to_have producer name (x, t)

12- to_have database name (y, p)

The schema below shows each step of the analysis starting from surface predicates



RETRIEVING INFORMATION FROM THE DATABASE

After the analysis above, the user's question becomes a combination of basic constraints. Replies are worked out by calculating answers to each of these basic constraints through documentary strategies and then cross-referencing partial results. These strategies are the following:

Automatic broadening of search

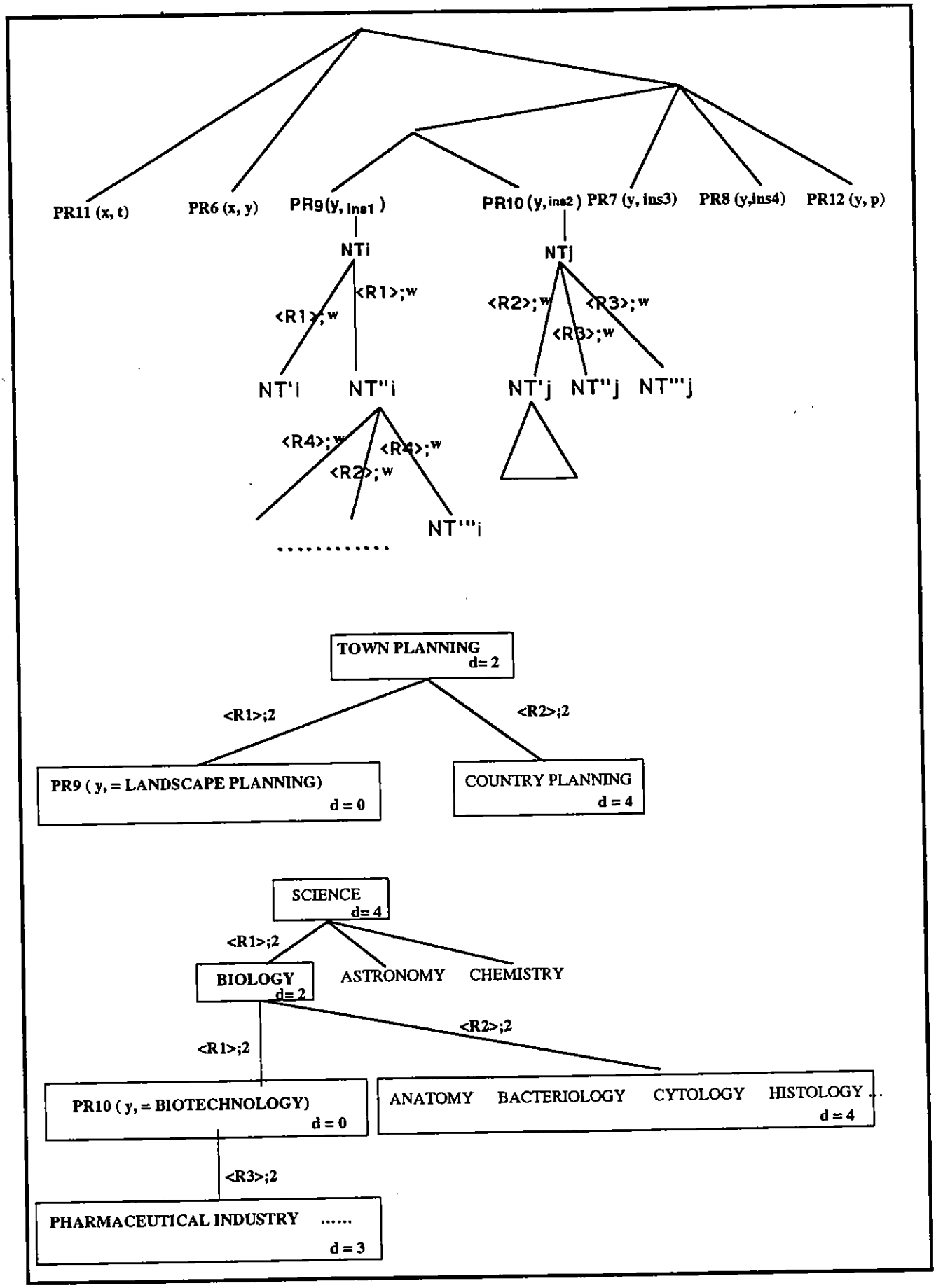
The aim here is to give a precise answer (if one exists) to a precise question, but also to allow the user to consult closely related answers as well.

Once the internal representation is completed, the documentary machine goes on to perform a set of operations upon the instantiated predicates (PR_i See deep predicates 7,8,9 and 10).

These operations consist in passing from one NT to another, starting at an initial NT, in a tree-walk (the arcs $<R_j>$) through the semantic and pragmatic network of the knowledge base. This is performed in an outward broadening manner and is determined by the strategy associated with the predicates (defined by the Administrator of the system). Since the arcs between the nodes of the network are weighted (w_t) as a function of the deep predicate, the result of a broadening process is a new NT' at a distance (**d**) from a starting NT.

NT are then classified according to the 'semantic or pragmatic' distance (d) that separates a NT identified at the start from another NT (i.e. d of NT'_i from $NT_i = \Sigma .<R_x>;w_t)$.

This distance is one of the parameters necessary to calculate the final distance or the pertinence of the answer.


PR11 (x, t)
PR6 (x, y)
PR9(y, ins1)
PR10 (y, ins2)
PR7 (y, ins3)
PR8 (y,ins4)
PR12 (y, p)
NTi
NTj
<R1>;w
<R1>;w
<R2>;w
<R3>;w
<R3>;w
NT'i
NT"i
NT'j
NT"j
NT'"j
<R4>;w
<R4>;w
<R2>;w
NT'"i
.....
TOWN PLANNING
d= 2
<R1>;2
<R2>;2
PR9 (y, = LANDSCAPE PLANNING)
d = 0
COUNTRY PLANNING
d = 4
SCIENCE
d= 4
<R1>;2
BIOLOGY
d= 2
ASTRONOMY
CHEMISTRY
<R2>;2
<R1>;2
PR10 (y, = BIOTECHNOLOGY)
d = 0
ANATOMY BACTERIOLOGY CYTOLOGY HISTOLOGY ..
d = 4
<R3>;2
PHARMACEUTICAL INDUSTRY .....
d = 3

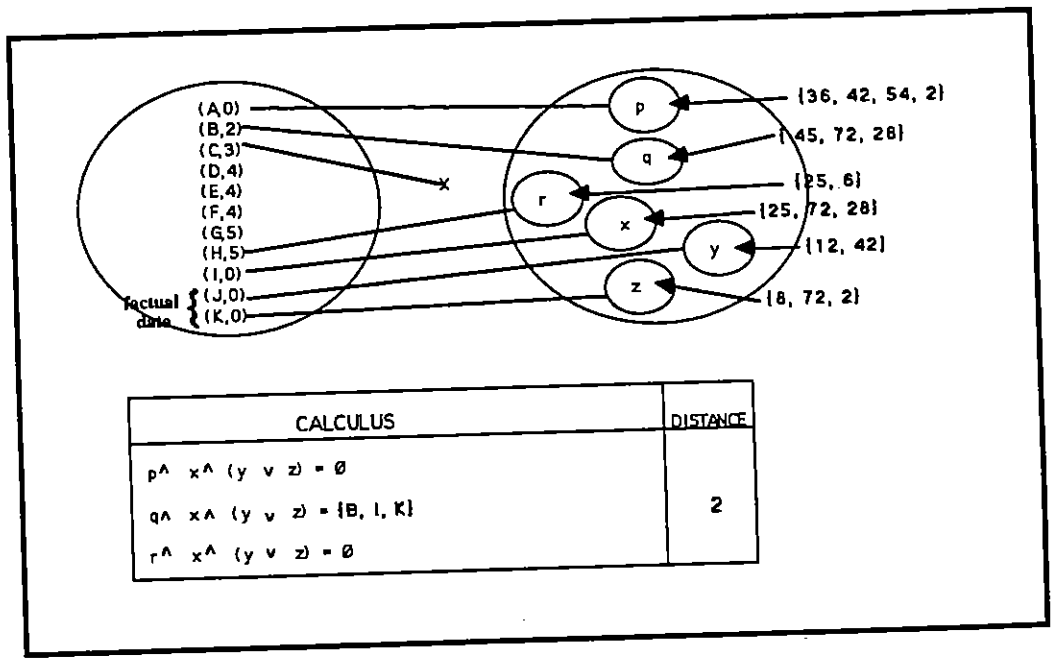
Cross-referencing of criteria

If the logical request identified at the end of the comprehension phase contains several criteria, these must be combined using set-theoretic operators. The Boolean equations are thus constructed using formal rules.

In order to know if there are answers to the user's questions, these functions use a limited inverse list identifying the documents of DIANEGUIDE database.

In this stage, the documenary machine can abandon or mark some instantied predicates are being less pertinents (in the context of the query) when no answers are found at the time of the cross-referencing phase. The documentary machine then re-start the same process without these predicates.

This mechanism allows the system to establish very early on if there is an answer or not. The number of answers may possibly trigger a dialogue allowing the user to broaden or narrow the initial request. For example, in the schema below, the circle at the left represents the NT classified by distance (A at distance 0; B at distance 2; ...). The circle at the right shows the subsets of documents which are indexed by the NT. Immediately below, we can see the Boolean equations generated by the system. The result of these operations will give the answers which satisfy the criteria of the user's query. In this example it is the second formula which will give an answer. The distance (the pertinence) of the answers is a function of the total sum of weight distance NT .



CALCULUS	DISTANCE
$p \wedge x \wedge (y \vee z) = \emptyset$	
$q \wedge x \wedge (y \vee z) = \{B, I, K\}$	2
$r \wedge x \wedge (y \vee z) = \emptyset$	

Generation of the requests in CCL language

Once the request has been analysed, the system generates a formal query in CCL[2] which is the GRIPS DBMS interrogation language. This generation is performed mapping each deep predicate into mapping formulae which describe the physical implementation of Dianeguide data (name of «relational» tables, name of field table, ...).

The search criteria contained in the CCL request are the identification of the documents (in the following example : F NR = 72, 45, ...) which the system checked in the previous phase, plus the

[2] Common Command Language.

name of fields that the system will get from the database (in the example : NADB is the name of the database, NAP is the name of producers, AB is an abstract about the contents of the selected databases) :

F NR = 72, 45, 86, ...

S F = NADB, NAP, AB

The CCL request will be transmitted to the Front-end processor to be treated by GRIPS (see : Technical architecture).

TECHNICAL ARCHITECTURE

The technical architecture of the system is based on a division of functions between two processors : a so called «FRONT-END processor» and a «BACK-END processor».

The «FRONT-END processor» is a SIEMENS mainframe computer, running under the BS 2000 system which hosts the DIANEGUIDE database managed by the GRIPS DBMS. This processor also handles all the communication with users.

The «BACK-END processor» is a SIEMENS UNIX work station, which runs the linguistic analysers and hosts applicative knowledge. This computer also calculates answers (broadening of searches and cross-referencing) using a locally stored image of the database and of its indexes .

A special application level communication protocol has been developed for information transfer between the two processors.

FUTURE DIRECTIONS

Future directions will be : to work towards the semantic representation of dictionaries in order to link a general multilingual vocabulary through corresponding notions, as opposed to this application where the link is made only between NT and certain other concepts. The GENELEX project works in this direction. At present, the morphological and syntactical models of French language have been defined. The semantic model is in progress.

The short term effort will focus on developing powerful grammars and their associated parsers in order to perform efficient linguistic analysis. The aim of the GRAAL project will be to work in this direction : definition of a standard representation of grammars; definition of parsers. Their functional characteristics will be : high re-usability of these grammars and ease of evolution and maintenance.

CONCLUSION

Our first conclusion will be in relation to the ergonomic aspects of this system. This interface has demonstrated that a natural language interface allows users to learn to formulate queries successfully even with minimal training. The trilingual aspect of this interface gives more opportunities to people wishing to interrogate this European database in their mother tongue. The extension to other languages could be possible thanks to the data conceptual architecture which allows other languages (vocabulary) to be «plugged» into the core of application knowledge .

The representation of the application knowledge has proved to be a crucial point in this system. It is clear that a straight-forward («flat») indexing of data would have been a bad option in this application because this representation only retrieves the significant information in record texts, but doesn't say how this information is semantically organised. For example, in database text : «German database on Italian agriculture». A straight-forward indexing will give : database - German - Agriculture - Italian. At the time of the interrogation phase a similar query will produce the same NT. So, some answers will be correct and others wrong («noises», in documentary terminology) because the system could be propose also Italian database, database of german agriculture, ...

The tree-predicates representation provides a powerful way to represent the meaning of surface queries and database text.

The broadening strategy rules which are applied trough the knowledge base (predicates and semantic network) are the key to intelligent information retrieval system because they give the possibility of finding both precise answers and, above all, closely related answers to a question. That is to say that there is no «silence» (absence of answers). There is «noise», but it is controlled noise.

ACKNOWLEDGEMENTS

Bernard Euzenat, Bob Kuhns, Mike Lunt made comments on an earlier draft of this paper. The members of the development team were : A. Chaouachi, M.G. Cuccu, S. Flores, P. Le Loerer, M. Macary et al. We wish to thank the ECHO team for their collaboration during the specification phase.

REFERENCES

KUHNS, .R.J., (1990). News Analysis : A Natural Language Application to Text Processing. AAAI Spring Symposium Series, Text-Based Intellingent Systems. Stanford University, Palo Alto, California.

HARMAN D., CANDELA G., (1990). Bringing Natural Language Information Retrieval Out of Closet. SIGCHI Bulletin.

HAYES, P.J., L.E. KNECHT, M.J. CELLIO, (1988). A News Story Categorization System. Proceedings of the Second Conference on Applied Natural Language Processing. Austin, Texas.

VEGA, J., (1990). Semantic Matching Between Job Offers and Job Search Requests. International Conference on Computational Linguistics. Proceedings of COLING'90. Helsinky.