Edite - A Natural Language Interface to Databases
A new dimension for an old approach

Paulo Reis João Matias Nuno Mamede
INESC
Av. Duque d'Ávila 23, 1000 Lisboa, Portugal
{Paulo.Reis, Joao.Matias, Nuno.Mamede @inesc.pt}

Abstract

This article presents the Edite system, a Natural Language Interface for Databases (NLIDB), that tries to explore the advantages of joining natural language processing with the expressiveness of graphical interfaces. In order to guarantee a permanent adaptation of this type of solution to a dynamic domain one should consider two critical fundamental factors: extensibility and portability.
An overview of the system architecture is presented, emphasising those choices that were imposed by the demands of portability and extensibility. Several general problems of natural language processing that were faced in constructing the system are discussed. Future work is highlighted.

Keywords: tourism, natural language processing, intermediate representation languages, database querying.

1. Introduction

The importance of the tourist industry for the Portuguese economy has significantly increased in the last 10 years. It accounts nowadays for approximately 8% of the GDP and equals that of the financial sector. The main reasons for the success of tourism are: the climate, the historical and cultural heritage, the tourist infrastructure, the hospitality of the Portuguese people and the relative close emissary markets.

It is important to create conditions to consolidate the registered growth, which can be achieved through a firm development strategy focusing on the quality of supply and human resources and on diversification of markets and products.

National tourism has adopted new instruments in order to reinforce their performance focusing on the new information and communication technologies.

The Inventory of Tourist Resources (IRT) emerges as the largest R&D Portuguese project in this area, actually exercising fundamental support on tourist ordering and planning and on forming a global reference point our tourism. The IRT initially emerged to eliminate the shortage of institutional information, positioning itself as the largest data repository of tourist resources on Portuguese tourism.

IRT's vast variety of purposes determined the adoption of an architecture able to support the integration of a large informational universe on a stage of multiple user segments and on the various applicational objectives (e.g., planning, promotion, distribution, management …). Nowadays, IRT's informational domain uses Internet, multimedia "kiosks", GIS, graphic user interfaces and "natural language interfaces".

The supporting technologies on the diverse functions of IRT are in well marked, distinct stages of development, from a scientific and technological point of view as well as from a commercial point of view.

This article intends to approach and deal with the problem of integrating an advanced technology, the so-called plain language interfaces for databases, into systems with a complex architecture.

2. Motivation

One of the main characteristics of multimedia kiosks is their familiar visual appearance, reducing the complexity of communication between man and machine to a minimum. The anthroponomical synchronisation of Image, Video, Audio and Text is one of the crucial factors to "seduce" the user into wanting to experiment the system. Another fundamental characteristic is the emergence of the utility. The user must feel, when using the system for the first time, that it can be useful. This is only possible with a well designed interface where the information can be easily accessed without having to learn another vast and complex communication language (the one used by the system).

In spite of the large variety of existing systems there is not yet a standard for these interfaces. This causes the user to fully understand what the system does, only after a certain amount of time. Another critic one could have on this method of interaction has to do with the fact that in a traditional system, with navigation through several successive windows, one can not always get the needed information. This occurs either because it does not exist (and the system is not able to inform the user on this) or because the user does not know the system language well enough to extract the information in question (it may take too many steps to get there).

What could be the best browsing alternative that passes beyond the above mentioned limitation? The answer could be a NLIDB.

The evolution of technology has caused a continous development of NLIDBs, especially in the area of natural language processing, exploring architectures that transform the NLIDBs into relational agents, and integrating languages and graphics that explore the advantages of both modalities [1,6].

In order to guarantee a permanent adaptation of this type of solution to a dynamic domain one should consider two critical fundamental factors: extensibility and portability [9]. On the one hand, a product like a NLIDB has to guarantee the ease to increase the linguistic coverage related with the domain, without requiring a vast technical knowledge of the natural language. On the other hand, the system has to guarantee the portability to other databases, minimising the number of changes.

If the above mentioned factors are indispensable for the development of the system whilst a product, we will have to guarantee the emergence of the utility. We believe that in order to achieve this, three factors are indispensable: a good linguistic coverage, the comprehension of the domain (ideally an interface of this type can use all of the existing information in the database) and great interaction with the user.

Bearing this factors in mind, INESC has been developing Edite, a multimodal interface, since 1995. This interface tries to combine the advantages of the natural language with the expressiveness of the graphical interfaces.

Chapter 3 presents the NLIDB architecture of the Edite system, describing the principle modules. Chapter 4 will present some of the options for a future development. Chapter 5 will present the main conclusions of this work.

3. An overview of Edite

3.1. Introduction

Edite is a *multi-lingual* (Portuguese, French, English, Spanish) natural language front-end for relational databases. It answers written questions about tourism resources by transforming them into SQL queries. The answer depends on the type of question. It can be a nominal list of resources, text, images or graphics. At present, the database contains 53000 tourism resources, arranged on 253 distinct types, corresponding to 209 tables.

The main goal of a NLIDB is to provide users with the capability of, in an efficient alternative way, obtaining information stored in the database [5]. The user is not required to learn an artificial communication language being possible to formulate the question in his own native language. The system building up was driven by two main objectives: (a) the exploration of a new technology in an existing context, *i.e.* by exploring how this technology can be used to increase the efficiency of current processes. The technology should also work as a new motive of attraction. We could integrate this into multimedia kiosks, but we can foresee the adequacy to other information supports, like *e.g.* Internet; (b) the advantages NLIDBs have, compared to others interfaces like formal query languages, form-based interfaces and graphical interfaces: (1) the user is not required to learn an artificial communication language to use the system's potentiality. This does not mean that there is no need for some information (training) about the system's functionality (linguistic coverage, language's domain); (2) there are kinds of questions (*e.g.* questions involving negation or quantification) that can be easily expressed in natural language, but that seem difficult (or at least tedious) to express using graphical or form-based interfaces. For example, "*What Lisbon's hotels are rated over 3 stars?*" (numerical quantification), or "*Which are the Algarve's golf courses without a driving range?*" (negation), can be easy expressed in natural language, but they would be difficult to express in most graphical or form-based interfaces; (3) the system will support anaphoric and elliptical expressions. NLIDBs of this kind allow the use of very brief, under-specified questions, where the meaning of each question is complemented by the discourse context. In other interfaces this notion of discourse context is usually not supported.

3.2. System Architecture

The system architecture (see Figure 1) is based on *intermediate representation languages*, where the natural language question is transformed into an intermediate logical query (LIL), before the final translation to a SQL query. This language expresses the meaning of the sentence in terms of high-level concepts, independent of the database structure [3,4].

The system architecture can be seen as being made of two big modules. The first module controls the natural language processing (linguistic component), where a question is submitted and successively transformed (morphological, syntactic, and semantic analysis), obtaining at the end of this process one or more LIL expressions. These expressions correspond to the possible interpretations of the initial question. Given the dimension of the domain and the flexibility of the natural language, there exist usually, several interpretations for the same question (the same happens with the spoken language).

The second component is in charge of the connection with the database, translating the LIL expressions to Structured Query Language (SQL) expressions (using mapping

tables, see 3.3.4) and sending them to the Data Base Management System (DBMS) to produce the answers.

The main advantage of this architecture is the complete separation between the linguistic component and the database knowledge. The portability of the system to another relational database is guaranteed by the configuration of the mapping module. Of course, there will always be other modules to configure, because of the linguistic coverage but the main work remains on the mapping tables.

3.3. System Components

Morphological Analysis

The morphological analysis module is an extension of Ispell (dictionary support usually employed by UNIX users) and allows one to obtain information about the words that form the sentence. For this purpose, it uses a dictionary with the root form of words and one affix file which contains the construction rules. Thus, it is possible to get the morphologic and semantic information of all the words of a family; however, this information does not have an explicit representation on the dictionary [2,10].

Each entry of the dictionary holds, besides the root, the construction rules that can be applied, syntactic and semantic information. *E.g.* the words `hotel` and `lodge` have the following entries:

```
hotel/CAT=nc,SEM=hotel(X),TYPE=HOTEL/p/
lodge/CAT=v,SEM=lodge(X,Y),TYPE=IND,T=inf,TR=t/XYPLSM/
```

About the word `hotel`, we have the information that it is an ordinary name that could be conjugated in the plural form. Its semantics is `hotel(X)` where `X` is a variable of the type HOTEL.

About the `lodge` word we have the information that it is a transitive verb, and one can apply the construction rules XYPLSM; the semantics of the word is `lodge(X,Y)` where `X` and `Y` are variables of an indeterminate type.

The syntactic and semantic information are used in the correspondent analysis. It is important to remark that there may exist equal words in the dictionary that have different semantics. This fact is extremely important for the semantic analysis.

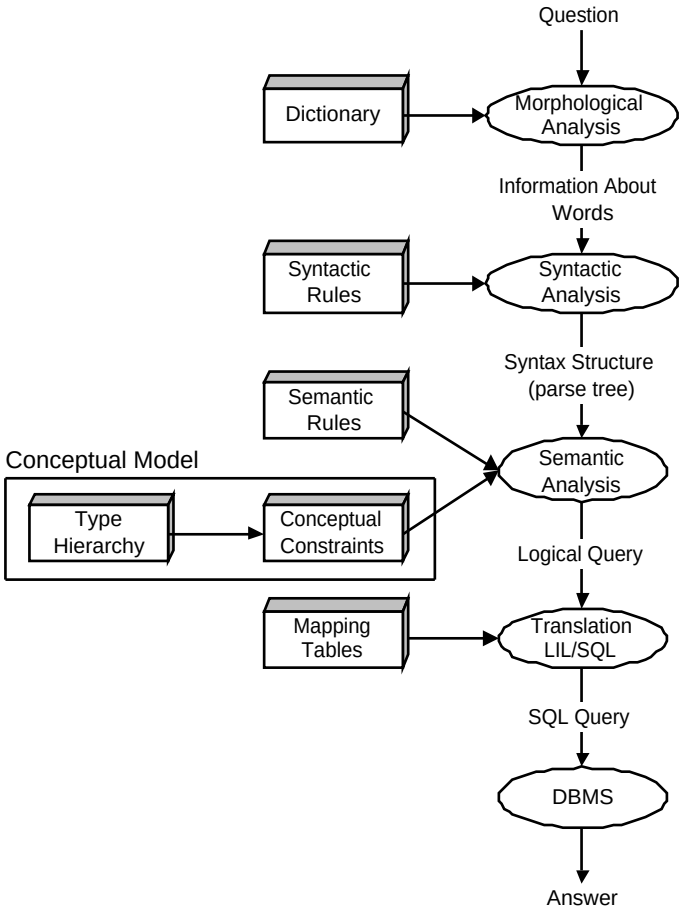

Figure 1 - The Edite Architecture

Syntactic Analysis

The syntactic analysis is based on the chart-parser technique, using a context free grammar and the Earley algorithm [1,7]. The result of this module is a set of parse trees, corresponding to the possible syntactical interpretations of the sentence.

One remark about this phase is the relaxation of some grammatical rules. This means that certain "less correct" sentences are also accepted and interpreted by the system. This was motivated by the evidence that those are currently used sentences.

Semantic Analysis

The final goal of NLIDB is the determination of the meaning of a sentence. In our case, and in the context of a database, that meaning is given by the SQL expression obtained after processing the initial statement (a question).

To obtain that expression, it is necessary to determine its meaning, usually represented in a logical form. In our case, we use the LIL language [14], similar to the First Order Predicate Calculus.

The sentence "*Which is the telephone number of the Ritz hotel*?"[1] is represented in LIL as:

```
hotel(Ritz-hotel), telephone (_11), of(_11, Ritz-hotel).
```

We will describe, very briefly, the LIL generation process (the space available being insufficient for a full description), concentrating on the interaction with the conceptual model. [10] The translation to LIL is similar to the syntactic analysis, using a dictionary and rules.

The semantic dictionary contains the words sense, called unit of meaning. When the meaning of a word is domain dependent it has to be defined using a logical predicate [14]. As an example, the meaning of the word "*city*" can be described by the predicate *is_city*(X), and consequently the question "*Is Lisbon a city?*" should be translated into: *is_city*(*Lisbon*).

Besides the predicates related with the words that are domain dependent , there exist another class of predicates, domain independent, such as the logical operators (*AND*, *OR*, and *NOT*) and the predicates *min*, *max*, *inf*, *sup*, and *exact*.

The semantic rules define the relations between the meanings of the words in a sentence. Each syntactic rule has a correspondent semantic rule, and for that reason this process is called rule-by-rule style.

Like syntactic analysis, semantic analysis is compositional, which means that the meaning of a constituent is derived solely from the meanings of its subconstituents. Compositional theories have some attractive properties. In particular, interpretations can be built incrementally from the interpretations of subphrases [1].

The semantic analysis process generates several LIL expressions, however, not all relations are conceptually acceptable in a given domain. The **conceptual module** is used to help ruling out the invalid LIL expressions. We say that a sentence is conceptually well defined when there are no violations of the constraints of the *domain of discourse.* If a sentence does not respect the conceptual constraints, then it is conceptually wrong (nonsense). In [13] we present a formalism for modelling conceptual constraints. The conceptual module is extremely important because it invalidates the wrong translations performed by the semantic analysis, reducing the number of LIL expressions that are generated.

The following example shows the importance of the conceptual module. Suppose that the dictionary contains three resources named Lisbon, a city, a hotel, and a dancing:

```
Lisbon/CAT=np,SEM=city(X),TIPO=city/p/
Lisbon/CAT=np,SEM=hotel(X),TIPO=hotel/p/
Lisbon/CAT=np,SEM=dancing(X),TIPO=dancing/p/
```

If asked "*Is Lisbon a city?*", during the semantic analysis the system would try to generate the following expressions: *is_city*(*Lisbon-city*), *is_city*(*Lisbon-city*), *is_city*(*Lisbon-city*). However, when the conceptual module is requested to validate the previous expressions, two of them would be eliminated due to the constraint *is_city*(*city*) (city represents the type of the argument).[2]

---

[1] In this paper, all the english questions and vocabulary were adapted from the portuguese system module.

[2] The arguments of a logical predicate correspond to a given semantic. In the previous example, *is_city* would have to be declared, in the conceptual module, as the constraint *is_city*(*type_city*), denoting that the argument of the predicate has to be of *type_city* [11,13].

In the Edite system types are organised in a hierarchy: a forest of trees, where each ascendant subsumes its descendants types.

The conceptual model is also used to solve other problems related with the scope of the modifiers (adverbial phrases and prepositional phrases). The question "*Which are the hotels of Lisbon with air-conditioning?*" only has one interpretation, because "*Lisbon does not have air-conditioning*". During the syntactic analysis, and due to the second modifier, two syntactic trees are generated. In one, "*with air-conditioningr*" is a modifier of the noun "*hotel*" (syntactic rules NP⇇ Det N NP PP, PP⇇Det N), in the other, as a modifier of the noun "Lisbon" (syntactic rules NP⇇ Det N PP, PP⇇Det N PP, PP⇇ Det N). However, during the semantic analysis, when applying the correspondent semantic rules, the second interpretation fails, since the predicate *with(Lisbon, air-conditioning)*, is not validated by the conceptual model. Consequently, a unique LIL expression is generated at the end of the interpretation process. Note that the question "*Which are the hotels with a bar in Lisbon?*" has two possible interpretations.

LIL-SQL Translation

The LIL-SQL translator [12] is based upon several mapping tables, which are highly dependent on the database organisation. This process is very efficient and, more importantly, it allows Edite to be used with any other relational database, being only necessary to define new mapping tables.

The question "*Has hotel Berna a swimming pool?*" is translated to the following LIL expression: `hotel(Berna-hotel), have(Berna-hotel,swimming-pool)`. This last expression is then transformed into SQL:

```
SELECT FROM hotel B
        WHERE  B.cod_type_abord=16      AND     B.cod_hotel_gr=1 AND
               B.cod_rec_turist=12234   AND     B.swimmpool > 0"
```

The answer is produced by another system, in which the natural language package is included. It is this system that presents the answers using nominal lists of tourism resources, texts, and graphics.

The relations used in LIL cannot be directly translated into database relations. In the systems that use a similar architecture, the database is defined taking into account the natural language processing. Since the possibility of expressing questions in natural language was only considered when the database was defined and full loaded, we were forced to define meta-relations to map the LIL relations into the database organisation. This approach does not degrade the response time, because the time spent during this last translation step is very small when compared with the previous modules.

4. Future Work

At the present moment the project is in the final stage of conception and prototyping.

In fact the Portuguese prototype is ready to be integrated in the experimental version of the entire system. On the other hand, the dictionaries necessary for the multi-language processing are being embedded in the application. Therefore, in the beginning of 1997 the product will be available for testing in the WEB and in the Portuguese Tourism Information Network.

In the future, we intend to extend Edite in order to achieve the following goals:

- Resolution of several language phenomenons such as anaphora and ellipse with the purpose of increasing the communication speed. *E.g. "Which are the country clubs in Albufeira area? And which are the hotels?*".
- Treatment of negation. *E.g. "Which campings do not require a camping licence?*"
- Handling temporal questions. In this type of questions, the numbers, dates and times have to be carefully treated. *E.g. "Is the Jeronimos monastery open on Saturdays?*" , *"In Lisbon, what are the churches open between 7 am. and 10 am.?"*
- Semi-automatization of the fulfilment data. The goal is construct a software tool, Domains Editor, that partially automates the information augmenting process; by information we mean dictionaries, conceptual model and mapping tables. This acquisition component is crucial to the success of a portable system. This tool allows a better management with fewer demands on (i) the knowledge of the system's internal workings; (ii) the intricacies of the grammar; (iii) computational linguistics in general.
- The development of techniques for generating co-operative responses. Nowadays, if the user asks "*What are the hotels in Marinhais?*", the system will return an empty list. In the future, we expect that the system answers with: *"There are no hotels in Marinhais.*"; next it can present information related to other types of lodging. The idea is to progress from the actual answers to co-operative ones.
- Natural language generation.
- Due to the evolution of signal processing technology, namely digital speech recognition, it will be possible to integrate, in the future, a voice analyser in this system. This will allow the establishment of a direct oral dialogue between the final user and the system.

5. Conclusions

In spite of the linguistic limitations, the system is powerful, portable and user-friendly enough to be used in multimedia kiosks and the Internet. The main conclusions are the followings:

- After being duly tested, one can conclude that Edite can be used for various purposes. Its advantages were laid out along this paper: the user communicates in his own language and the system embraces the whole domain.
- One has to fit this type of interface in a class that still has limitations because of the computational demands and the complexity of the natural language. However, we think that with perfectly user defined functionalities and linguistic coverage, the NLIDBs can play a complementary role in current systems.
- This type of systems are important for the evolution of its informational domains as the characteristics of the natural language do not impose any restrictions on the information that the user might want to request.
- The system architecture allows this software to be ported to other databases. This has been one of the main guidelines during the software design phase and has resulted in an independence between natural language processing and linkage to databases. This feature makes possible to apply this module to other databases or Universes of Discourse with low development overheads and costs and high efficiency.

- The system has some limitations that we address here. First of all, regarding the linguistic coverage, it only accepts questions. Imperative or declarative statements are not allowed. With this approach it was possible to perfectly identify the subset of the natural language universe of the system. This is also an indication to the user, that the system only responds concerning the information of the database domain. It is not useful to have a chat about football games (typical expectations generated by the natural language systems). Another limitation is the set of restrictions imposed on the design and conception of the database.
- Despite the fact that the system has a good linguistic coverage, it is important to foresee the future extensions of the database domain. Thus, there should exist mechanisms to log the failures, so that the system could be properly extended. This mechanism should support vocabulary enrichment and the extension of the conceptual model and mapping tables.
- This project does not expect to demonstrate that NLIDBs are the future, but that they can perfectly complement the other types of interfaces.

6. Acknowledgements

Ministério da Economia de Portugal, Secretaria de Estado do Comércio e do Turismo - for project financial support and for the wise decision of linking National Information Technologies & Telecommunications to Portuguese Tourism Strategic Development Plan.

Dr. José Sancho Silva - Director of Gabinete de Estudos e Planeamento da Direcção Geral do Turismo and IRT Chief Executive Project - for highly professional competence, availability, persistence and courage for sustaining the project unreservedly, beyond all the adversities met.

Dra. Vanda Boavida (ICEP), Dra. Elisa Almeida (DGT), Dr. Luís Costa (ICEP) for the precious ideas, encouragement and assistance.

INESC, Instituto de Engenharia de Sistemas e Computadores - Director Prof. José Alves Marques - for skilful management and project belief. INESC, Centro de Sistemas Computacionais, Grupo de Sistemas e Serviços Telemáticos - for teamwork and for some sleepless nights.

Finally we would like to show our special gratitude to Ana Matos, Luisa Marques, Eduardo Augusto and Nuno Santos.

7. Bibliography

[1] Allen, J. 1995. "*Natural Language Understanding*". The Benjamin/Cummings Publishing Company, Inc.

[2] Almeida, J. J. D., Pinto, U. 1994. "*Jspell - um módulo para análise léxica genérica de linguagem natural*". Technical Paper, Dept. de Informática, Universidade do Minho.

[3] Androutsopoulos I., Ritchie G., Thanisch, P. 1993. "*An Efficient and Portable Natural Language Query Interface for Relational Databases*". Proceedings of the 6th International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems, Edinburgh, U.K., pages 327-330. Gordon and Breach Publishers Inc., Langhorne, PA, U.S.A., June 1993.

[4] Androutsopoulos, I. 1993. "*Interfacing a Natural Language Front-End to a Relational Database (MSc thesis)*". Technical paper 11, Dept. of AI, Univ. of Edinburgh.

[5] Androutsopoulos, I. 1994. "*Natural Language Interfaces - An Introduction*". Journal of Natural Language Engineering, Cambridge University Press.

[6] Cohen, P.R. 1991. "*The Role of Natural Language in a Multimodal Interface*". Technical Note 514, Computer Dialogue Laboratory, SRI International, 1991.

[7] Earley, J. 1970. "*An efficient context-free parsing algorithm*". Commun. of the ACM 13, 2:94-102. Reprinted in "*Readings in Natural Language Processing*".
[8] Godbert, E. 1994. "*Modelling domain and connectivity constraints in natural language processing*". Current Issues in Mathematical Linguistics, C. Martin-Vide (Ed.), Elsevier Science B. V.
[9] Grosz, B. J., Appelt, D. E., Martin, P. A., Pereira, C. N. 1987. "*TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces*". Artificial Intelligence 32, pages 173-243. Elsevier Science Publishers B.V. (North-Holland).
[10] Marques, Luisa 1996. *M.Sc. Dissertation*. Instituto Superior Técnico, Universidade Técnica de Lisboa.
[11] Milhaud, G. 1994. *Ph.D. Dissertation*. Université d'Aix-Marseille II.
[12] Reis, P., Mamede N. 1996."*LIL-SQL. Processamento de Interrogações LIL por Tradução para SQL*". Technical Report. Grupo de Sistemas e Serviços Telemáticos, INESC. (in preparation)
[13] Reis, P., Mamede N. 1996."*Modelo Conceptual. A Hierarquia de Tipos*". Technical Report. Grupo de Sistemas e Serviços Telemáticos, INESC.
[14] Reis, P., Mamede N. 1996. "*LIL - Linguagem de Interrogação Lógica*". Technical Report. Grupo de Sistemas e Serviços Telemáticos, INESC. (in preparation)