Customization of Natural Language Interfaces to Databases: Beyond Domain Portability

Jose A. Zarate M.[1], Rodolfo A. Pazos
[1]*Centro Nacional de Investigación y Desarrollo Tecnológico (Cenidet)*
Interior Internado Palmira S/N
{jazarate,pazos}@cenidet.edu.mx

Alexander Gelbukh[2]
[2] *CIC-IPN, National Polytechnic Institute of Mexico*
gelbukh@cic.ipn.mx

Abstract

The first Natural Language Interfaces to Databases were built and designed for specific domains, and their customization processes implied source code manipulation. Open systems and database inter-operability enabled these interfaces to be independent of the operating system and database management system, and the separation of the knowledge base from the translation process allowed for domain portability. Although commercial interfaces incorporate semi-automatic configuration wizards that help configure the interface without knowledge of its inner workings or its source code, it is still difficult to customize these interfaces for a given database, due to confusion on the information that is necessary to provide to the knowledge base of the interface in order to make it able to answer some query category. For solving this problem, we propose an ontology whose design is simple and flexible enough to assist the customizer's work. This paper describes the design of the ontology, as well as an empirical evaluation of this approach versus the customization process of a commercial interface. The evaluation was useful to detect problems with different types of queries used to retrieve information from a specific database. In spite of the difficulties found to make the evaluations and some unquestionable advantages offered by commercial interfaces.

1. Introduction

Although there are some Natural Language Interfaces to Databases (NLIDBs) that offer portability to multiple types of systems (expert systems, e-mail etc.), multiple operating systems and multiple databases management systems (DBMS) [7] and that were designed using a modular architecture that allows them certain versatility [1], the portability from a domain of knowledge to another (i.e., the capacity to work with different databases) has not been completely attained. This is due to the fact that those NLIDBs that claim to be easily portable from one domain to another, base their claim on the assumption that their knowledge database can be easily configured for a new domain, which is not always true. This approach does not take into account that to facilitate the configuration of the knowledge base, it should be built in such a way that it would only be necessary to make a few changes to adapt it to a new context and it would be able to reuse other knowledge for interface customization.

The above-mentioned capability is very important, since someone (knowledge engineer or database administrator) has to configure the interface, by changing grammatical rules according the database context, adding words to the dictionary, and defining semantic relations among these words.

A poll of forty-plus MS students shows that just 5% is acquainted with NLIDBs or any other natural language interface. This poll is an example of the insufficient publicizing of the existence of this type of interfaces, and it shows the difficulty for assessing the use of natural language interfaces. Another factor that contributes to its limited use is the complexity to customize the interface to the final user needs. We propose as an improvement for the NLIDB customization process the use of ontology as knowledge base, designed for achieving simplicity and flexibility, which will render a more accessible interface in its use and acceptance.

We propose using an ontology as knowledge base (in addition to the default customization process and tools), which offers as novelties the incorporation of principles of reuse, explicit knowledge base structure, classification of queries, generality, and simplicity. An empirical evaluation was carried out for comparing the ontology-based customization vs. the most available commercial

NLIDB (English Query, a component of SQL server), using MS students.

This paper is organized as follows: Section 2 describes the evolution over time of the customization process of NLIDBs, Section 3 describes the ontology proposed as knowledge base and the customization methodology, Section 4 presents the empirical evaluation process, Section 5 shows the evaluations results, Section 6 discusses the results obtained, and Section 7 presents some final remarks.

2. Related Work

In the 70s the first interface appears: Lunar [14], a system for searching chemical analysis of lunar rocks. Another similar system is Rendezvous [13], developed by IBM, which incorporated help to the final users for querying the database.

Ladder [8] allowed to access large databases on different DBMSs, and included facilities such as spelling correction and elliptic reasoning. This system was based on semantic grammars, a mechanism that mixes syntactic and semantic parsing and enhances its understanding capacities. One problem was that the semantic grammar that allowed it to be adjusted to a certain domain, was an obstacle for its portability, since it required rewriting the grammar for another database.

In spite of the great effort devoted in the 80s, this type of interfaces did not become popular probably because of untreated language phenomena, the perception of NLIDBs as "exotic" systems, and the emergence of friendlier graphic and form-based interfaces. Despite the release of the first commercial prototypes, their use was quite reduced [1].

In the 90s, although research was no longer as intense as in the 80s, the general advance in NLIs contributed to the appearance of general-purpose products that translated a natural language query into a logical form, which was used to construct Structured Query Language (SQL) queries to DBMSs.

Nowadays, the most important commercial NLIDBs, English Query [10] and English Language Front-end (ELF) [4] carry out an automatic analysis of the database data and metadata to setup the NLIDB for a specific database. This analysis uses a lexicon, a dictionary, attributes descriptions, and predefined properties. Due to limitations of the customization process, ELF allows modifying the lexicon, which contains information gathered during the analysis, and English Query allows adding synonyms and other auxiliary words to the lexicon. Besides this information, both interfaces allow to define relations among database entities using verbs and nouns and to add functionality to the interface by links between sentences and external function calls. English Query is integrated with Visual Studio 6.0, which permits defining relations among concepts that represent entities using a graphic editor, similar to the entity-relationship diagrams. It provides the information EQ uses to answer a query (useful when EQ fails to answer the query appropriately), and it includes a wizard that guides the user to provide feedback to the interface with the information required to generate the correct answer. This feedback consists of some forms for providing additional information: data not set up in the dictionary, user-defined relations and metadata.

3. Customization Methodology

The customization methodology proposed for an NLIDB [15] comprises the following stages: analysis of the database semantic, obtaining a query corpus from potential users, classification of this corpus in categories (similar to the ones defined in [6]) whose definition is linked with a relation, and defining classes linked by relations to define the knowledge base of the NLIDB.

The basic idea is that the problems found in the query corpus are solved through relations between classes that constitute the ontology. The semantic parser of the NLIDB would use each relation as a solution to each problem of the translation form natural language to SQL. The set of solutions defined inside the ontology would constitute a problem solving library (PSL). Two key features for the acceptance of an ontology as a configuration process for NLIDBs are reuse and resource sharing, and consequently, it is necessary to design a more generic and more reusable organization of concepts.

To achieve the most generic ontology possible, linguistics [11] and grammar were used as design guides to define categories for organizing concepts and relations among them. Additionally, the relational database theory was employed to categorize database elements. The translation of a database query expressed in natural language involves the search of relations that link words of the query (nouns, adjectives, etc.) with elements of the database (tables, columns, etc.), which allow to translate the query to SQL. Additional elements were added to the ontology, such as classes and relations that allow relating concepts of the database, Parts of Speech (POS) and new properties with external function calls, an extension mechanism for the NLIDB, similar to those in ELF and English Query.

To make sure that the ontology was more reusable, it was formalized using the Web Ontology Language (OWL), which allows compatibility with other

ontologies formalized in OWL for reuse and sharing the ontology developed with other users and applications through the Web.

3.1 Classes (Categories), Concepts (Synsets) and Words

The ontology defines categories or classes for organizing concepts that explain the database context. The definition of the top-level classes is explained hereupon:

ElementosBD (ElementsDB).- They define categories where the main relational database elements are classified [3]; for example: primary key, foreign key, etc. Some subcategories were omitted such as indexes or triggers, because they are not part of a query.

Palabra (Word).- Subcategories are POSs (noun, adjective, verb, adverb and other). We borrowed concepts from WordNet [12], such as *word form* for referring to physical pronunciation or writing of a word and *word meaning* for referring to the lexical concept that a word form can use to express something.

Synset.- It is a representation of a word meaning constituted by synonyms. Synset subcategories are based on POSs, except for category *other*, since this POS almost has not synonyms.

Funciones (Functions).- They are classified in three subcategories: aggregation functions (part of SQL), user-defined functions and link-call functions. The first one allows defining groups of words or synsets equivalent semantically to SQL functions such as AVG, MAX, etc. The second one allows to associate words or sentences with user-defined programs through synsets. The last one permits to define a label used as a bridge between a user-defined relation and an external program that implements a new semantic relation.

3.2 Relations (Properties)

Relations or properties link classes (categories), concepts (synsets) and words, so that they define all together the database context for an NLIDB. The top-level relations defined in the ontology are the following:

Lexical relation.- It is a culturally recognized pattern of association that exists between lexical units in a language. Its subcategories are syntagmatic and paradigmatic. The lexical-syntagmatic relations defined are: perception, sound, instrument, degradation, and benefactor. The lexical-paradigmatic relations defined are: synonymy, hiponymy-hiperonymy (sub-relations: class inclusion, scalar, lineal, and troponymy), opposition (sub-relations: antonymy, relational and directional converses, and complement), and meronymy (sub-relations: substance, place, component, action, portion, and member).

Relaciones_elementosBD (Relations_elementsDB).- Represents relations between elements of the relational database model and synsets, and through transitivity establishes a connection of database elements with words.

Relaciones_funciones (Relations_functions).- Connects instances of the user-defined functions class to synsets and to program names (including their absolute path). Through transitivity, synsets allow to connect these functions with database elements. Its sub-relations are:

Relación_programa (Relation_program).- Links an instance of the user-defined relations class with an external program name.

Palabra_función (Word_function).- Links an instance of the user-defined functions class with an instance of noun class, subclass of palabra (word).

Función_synset (Function_synset).- Links an instance of the user-defined functions class with a synset.

3.3. Instances

The instances of the pre-filled ontology are words (word forms), synsets (named after WordNet'synets), terms identifying databases, tables and columns, and names of the functions used to increase the interface capacity. The population of the ontology was carried out in a previous work [16]. The last stage of the proposed methodology (i.e., the description of concepts and connections defining relations among words), consists just of the definition of instances and their relations.

4. Description of the Experiment

The evaluation objective was to observe the behavior of potential NLIDB customizers, in the presence of queries of different difficulty, as well as different databases to customize. Therefore, the empirical evaluation has not tried to validate the answers provided by NLIDBs, since there exist many factors involved.

The experimental plan consisted of three empirical evaluations. A group of MS students was used to measure their preference for using an ontology to customize an NLIDB using Protégé [13], versus the English Query's customization process. In each of the three evaluation experiments, crossed evaluations were carried out: first a team evaluated the proposed approach using Protégé and the other team evaluated English Query, and afterwards, the roles of the teams were inverted. Since the evaluation teams were small, we had

to resort to this trick in order to cancel out the biasing resulting from the learning process; i.e., the customization using the second approach will become easier after the customization using the first one.

4.1 Description of the Evaluation Teams

The participants of the evaluations were MS students, which did not received formal training, just an informal briefing to explain them the experiment (they did not receive training proper in order to avoid the instructor's possible biases). The participants received a document that explains the proposed ontology approach, the English Query documentation provided by Microsoft, and a document with customization examples was added for both approaches (EQ and the ontology approach). Additional information of each evaluation team is shown in Table 1.

Table No.1. Information of the evaluation teams

	Group
Kind of student	Freshman MS student
Query corpus (difficulty level low/medium/high)	a) 10(4/3/3) b) 10(1/4/5) c) 10(4/3/3)
Number of participants	6

4.2 Description of the Evaluation Task

The participants were asked to carry out the customization using Protégé for the ontology approach and the English Query customization interface for ten queries from the ELF corpus [5] for evaluations No. 1, No. 2 and No. 3. The difficulty of the queries for each evaluation is shown in Table 1. The students received feedback between the first and second evaluations and between the second and third evaluations. This was useful for understanding both customization processes, but trying not to bias the evaluation.

4.3 Description of the Evaluation

The evaluation form questions were grouped according to the main factors affecting the customization process of an NLIDB: configuration interface, customization methodology and other features, such as motivation, background, and analysis skills of the evaluation participants.

The metric used was the Likert scale (one to seven). The values presented in the section "Summary of Results" are average values and they are normalized on a 0-100 scale. The time spent on customization was not measured, since it was not possible to gather the participants at the same time. The quality metric of the customization was excluded because we did not have a group of experts in ontology design to assess the quality of the ontology resulting from the customization, nor the configurations generated were tested because the semantic analyzer [6] of our ILNBD only exploits the synonymy relation and we wanted to allow the participants to express other relations in the definition of the configuration of the interface knowledge base.

5 Summary of Results

This group of evaluations compares the ontology approach proposal vs. English Query's customization process. The results for Evaluations No. 1, No. 2 and No. 3 are shown in figures 1, 2 and 3 and Tables No. 2 and No. 3. Those figures show the differences between the averages of the evaluations of questions related to the customization interface, customization methodology and diverse features of English Query and the ontology approach. In these figures a positive difference indicates that the ontology approach was better and a negative difference indicates the opposite.

Table No. 2. Evaluation for questions related to the customization methodology of English Query

Question	English Query 1	English Query 2	English Query 3
1. The training process allowed me to understand the configuration methodology built into the tool	45.83 (13.82)	58.33 (18.63)	58.33 (18.63)
2. The documentation of the configuration process is easy to understand	54.17 (13.82)	66.67 (11.79)	66.67 (16.67)
3. The terminology used in the configuration process is strange or confusing	70.83 (13.82)	83.33 (11.79)	66.67 (16.67)
4. The necessary steps to carry out the configuration process were clear	50.00 (58.33)	58.33 (18.63)	70.83 (13.82)

Table No. 3 Evaluation for questions related to the customization methodology of the ontology approach

Question	Proposal 1	Proposal 2	Proposal 3
1. The training process allowed me to understand the configuration methodology built into the tool	41.67 (8.33)	66.67 (11.79)	70.83 (21.65)
2. The documentation of the configuration process is easy to understand	45.83 (13.82)	54.17 (13.82)	54.17 (21.65)
3. The terminology used in the configuration process is strange or confusing	45.83 (7.22)	58.33 (8.33)	70.83 (13.82)
4. The necessary steps to carry out the configuration process were clear	45.83 (13.82)	62.50 (13.82)	66.67 (26.35)

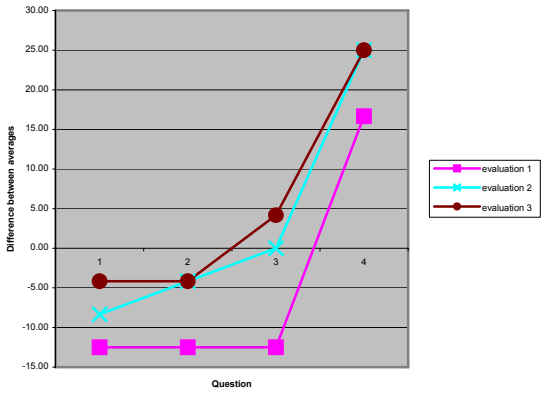

Fig. No. 1. Evolution of the differences between average evaluations of questions related to the interface of English Query and Protégé for evaluations No. 1, No. 2 and No.3

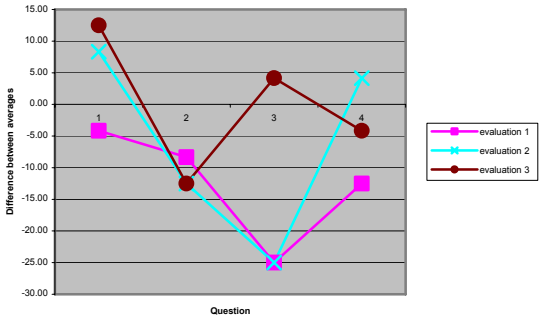

Fig. No. 2. Evolution of the differences between average evaluations related to the customization methodology of English Query and the ontology approach for evaluations No. 1, No. 2 and No. 3

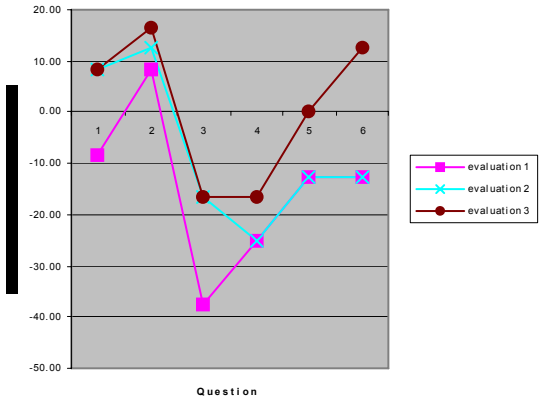

Fig. No. 3. Evolution of the differences between average evaluations related with diverse features of English Query and the ontology approach for evaluations No. 1, No. 2 and No. 3

6 Discussion

The difference in some aspects of the participants (freshmen vs. senior, voluntary vs. compulsory participation, degree of experience with open source and commercial software) and in individual capacities, produce –despite the obtained configurations were very similar– very different results, since the evaluations are perceptions conditioned by many factors. An interesting detail is that, although the proposed approach starts at the first evaluation with almost all evaluations against, in the subsequent evaluations its advance is notable. However, it is important to remark that flexibility is the

only feature of the ontology approach that always remains better evaluated than that of EQ.

7 Conclusions

Off-the-record talks with some evaluation participants revealed that, more than the customization for different databases, the largest problem is to configure the interface for different types of questions. The participants in evaluation No. 2 preferred EQ because its user interface and configuration methodology allows configuring more easily simple questions (those that involve synonyms) and because it spared them a lot of work, although they did not have any idea of how it worked internally. Despite the support tool of EQ, the configuration process is very confusing, especially when the query involves some complexity. The general feeling was that the use of the ontology offers many additional possibilities and a more natural way of representing the necessary knowledge so that the interface can answer any type of queries.

Although English Query undoubtedly prevails concerning its support tools (wizard, graphic editor of relations, transparency in the translation process, etc.), it was more desirable for the participants to know all the terms and its relations, i.e., an explicit knowledge base (ontology).

The most important contributions of the ontology approach are: a general-purpose ontology that incorporates elements from a relational database, and a methodology that allows connecting, through the ontology, query elements with the database elements, which can be useful to the semantic analyzer to understand the query and translate it correctly to SQL. The methodology incorporates the idea of establishing patterns to classify the queries issued to the NLIDB and, in this way, to simplify the customization work, since it would essentially be the same customization task for each pattern or category of queries.

10. References

[1] J. Androutsopoulos, G. Ritchie and P. Thanish, *MASQUE/SQL, an Efficient and Portable Language Query Interface for Relational Databases*, Department of Artificial Intelligence, University of Edinburgh, 1993.

[2] E.F. Codd, *Seven Steps to RENDEZVOUZ with the Casual User*. North-Holland Publishers, 1974.

[3] C.J. Date, *An Introduction to Database Systems*, 7th Edition, Addison Wesley Longman, 2000.

[4] ElfSoft, "English Language Front-End Software", http://www.elf-software.com

[5] Elfsoft, " ELF vs. English Query vs. English Wizard", http://www.elf-software.com/FaceOff.htm.

[6] J.J. González B., Interfaz en lenguaje natural independiente del dominio capaz de procesar dominios complejos, P.h. disserattion, Cenidet, Dec. 2005.

[7] H. Jung and G. Geunbae Lee, "Multilingual Question Answering with High Portability on Relational Databases", International Conference On Computational Linguistics, proceeding of the 2002 conference on multilingual summarization and question answering - Volume 19 Department of Computer Science and Engineering, Pohang University of Science and Technology, Korea, pp. 1-8, 2002 .

[8] G. Hendrix, E. Sacerdoti, D. Sagalowicz, and J. Slocum. "Developing a Natural Language Interface to Complex Data", *ACM Transactions on Database System*, Vol. 3, No. 2, pp. 105 – 147, Jun.1978.

[9] Russian Institute of Artificial Intelligence, "Inbase", http:// www.inbase.artint.ru/english/default-eng.asp, Jun. 07

[10] Microsoft Corp., "Chapter 32 - English Query Best Practices",http://www.microsoft.com/technet/prod technol sql/2000/reskit/part9/c3261.mspx, jun. 07

[11] E. E. Loos (general editor), Susan Anderson (editor), Dwight H., Day, Jr. (editor), Paul C. Jordan (editor), and J. Douglas Wingate (editor), *Glossary of Linguistic Terms*,http://www.sil.org/linguistics/GlossaryOfLinguistic Terms/, Jun. 2007.

[12] G. Miller, "Wordnet, a Lexical Database, Cognitive Science Laboratory", Princeton University, last access June 18, 2007, http://www.cogsci .princeton.edu/~wn/.

[13] Stanford Medical Informatics, Stanford University, "Protégé Ontology Editor", http://protege.stanford.edu/ index.html, Jun. 2007.

[14] W. Woods, R. Kaplan, and B. Webber, *The Lunar Sciences Natural Language Information System: Final Report*, BBN Report 2378, Bolt Beranek and Newman Inc., Cambridge, Massachusetts, 1972.

[15] A. Zarate, R. Pazos, A. Gelbukh, and I. Padrón, "A Portable Natural Interface for Diverse Databases Using Ontologies", *Proc. Computational Linguistics and Intelligent Text Processing*, pp. 494-505, Mexico, Feb. 2003.

[16] J.A. Zarate M., R.A. Pazos R., R. Toledo, "Aquisition of Lexical-sintactic Relatioships from a Dictionary", 11th International Congress on Computer Science Research, Tlanepantla, pp. 45-52, Mexico, Sept. 2004.