Improving the Customization of Natural Language Interface to Databases Using an Ontology

M. Jose A. Zarate[1], R. Rodolfo A. Pazos[1], Alexander Gelbukh[2], and O. Joaquin Perez [1]

[1] Centro Nacional de Investigación y Desarrollo Tecnológico (Cenidet)
[2] CIC-IPN, National Polytechnic Institute of Mexico
{jazarate, pazos,jperez}@cenidet.edu.mx, gelukh@cic.ipn.mx

Abstract. Natural language interfaces to databases are considered one of the best alternatives for final users who wish to make complex, uncommon and frequent queries, which is a very common need in organizations. The use of this type of interfaces has been very limited, due to their limited publicizing and the complexity to adapt them to users' needs, and because their precision varies widely. We propose as a solution to the problem of customizing this type of interfaces, the use of an ontology as a knowledge base whose design is simple and flexible enough to make the use and acceptance of these interfaces more accessible. This paper describes the design of the ontology, as well as a series of comparative evaluations of this approach versus the customization process of a commercial interface. This evaluation aims at assessing the acceptance of this approach by of those that will potentially customize the interface to a database, in contrast to the precision tests that are commonly applied to this type of interfaces. In spite of the difficulties found to carry out the evaluations, the results show that the use of our approach is preferred as a natural language interface customization means to the process of the most popular commercial interface. The estimations indicate that the potential people on charge of the process of customization of this type of interfaces considers that using the ontology as interface knowledge base would allow to answer a wider diversity of types of queries than those that would allow to answer a commercial interface.

1 Introduction

In a study carried out at Pittsburg University [11], it was found out that Natural Language Interfaces to Databases (NLIDBs) are one of the best options for users who look for information located in more than one table and formulate nontrivial and infrequent queries. The assumption that this type of queries is more common is based on the emphasis toward a larger flexibility of database reporting tools.

A poll of MS students of a private university and a research center showed that just 5% knows NLIDBs or any other natural language interface. This poll is an example of the insufficient diffusion 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

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4705, Part I, pp. 424–435, 2007.

user needs. We propose as an improvement for the NLIDB customization process the use of an ontology as knowledge base, designed for achieving simplicity and flexibility, which will render a more accessible interface in its use and acceptance.

NLIDBs evaluations [3], [4], [10] refer to interface precision to answer a query corpus using an automatically generated configuration. This default configuration process uses information from the database metadata and linguistic knowledge embedded in the interface. Although the results on precision from those evaluations are very high (over 90%) assuming that the corpus used is representative; in practice, the interfaces provide several tools (dictionary editor, wizards, etc.) that allow making adjustments for situations not considered by the automatic customization process.

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. Comparative empirical evaluations were carried out on the customization of the most available commercial NLIDB (English Query, a component of SQL server) versus an ontology-based customization, using MS students.

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

2 Related Work

English Language Front-end (ELF) [2] carries out an automatic analysis of data and database metadata, to setup ELF for a specific database. This analysis uses a lexicon and a dictionary (Moby dictionary). Besides this information, ELF allows to define relations among database entities using verbs and nouns. Due to limitations of the customization process, ELF allows modifying the lexicon, which contains information gathered during the analysis. ELF permits to revise the Moby dictionary, an embedded dictionary of 17,000 entries which includes synonyms.

Although ELF is considered one of the best available NLIDBs [3], and according to its documentation, it needs a minimum extra effort to tune its default-configuration, some problems were found with its configuring process: the categories of its knowledge base are not organized, the categories attributes mix elements related to syntactic parsing with semantic parsing, and ELF does not allow to add new attributes. The ELF documentation mentions that the automatic analysis detects synonymy relations, but it does not clarify if the interface can deal with another type of relations (antonymy, meronymy, etc.) or it provides a mechanism to define new relations.

English Query (EQ) [6] carries out an analysis very similar to that of ELF, but in this case the dictionary is not accessible, neither are the categories used to classify the database tables and table columns. Its analysis is restricted to linking database columns with words and defining relations such as "has" (very generic, because it just establishes "an entity has columns") and “unique" (column identifiying a table or entity). Additionally, it has a modifiable dictionary of synonyms and it allows to define temporal relations among concepts of the database, heteronymy-hyponymy relations among tables, and to add functionality to the interface by links between sentences and external function calls (feature similar to ELF's).

The last version of English Query is integrated with Visual Studio 6.0, which allows defining relations among concepts that represent entities using a graphic editor, similar to entity relationship diagrams. It provides the information EQ uses to answer a query (useful when EQ fails answer the query appropriately) and it has a wizard that guides the user to feedback the interface with the information required to generate the correct answer. This feedback consists of some forms that have to be filled with additional information not set up in the dictionary, user-defined relations and metadata.

Some of the problems found for English Query are the following: the process for adding new words to the dictionary is confusing as well as the use of the new words by EQ; the mechanism to define new relations is inflexible, because it is restricted to a few sentence patterns (trait, verb, adjective, adverb, command and preposition phrasing); the difference when defining a relation using one or another pattern is not clear; the default relations defined by English Query are very generic and not very useful; and the feedback wizard is not very intuitive, because similar queries that are not correctly answered by the interface may need different information so they can be answered correctly.

Inbase [5], an NLIDB developed at the Russian Research Institute of Artificial Intelligence, bases its operation on the separation of knowledge about semantic patterns which are used in querying the database and knowledge of the problem domain of a particular database. Inbase allows to quickly adjust the capacities of the component of the natural language analysis to the database to be queried. To answer the queries, a model of the domain is needed (DM), which is obtained partly by an analysis of the database, and partly from information that a customizer provides. The database domain is formalized in System with Networks and Objects-Oriented Productions (SNOOP) [12].

Unfortunately, an English on-line demo cannot be configured and is not very reliable, because Inbase does not distinguish between variants of the same query, (for example "which is the employee with highest salary" and "which is the employee’s age with highest salary"). A description of customization process could not be found; however, a project reference [5] mentions that Inbase uses KL-ONE [14], one of the most stable languages for knowledge representation. Unfortunately, it was not possible to evaluate the process of customization of this NLIDB (and other ones, such as PRECISE [8], for the same reasons).

3 Customization Methodology

The customization methodology proposed for an NLIDB [16] is composed of 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 [4]) whose definition is linked with a relationship; definition of useful concepts to answer queries; identification of relations and concepts in the knowledge base; and connecting query elements with concepts and relations that explain the database semantics.

Concepts and relations have to be organized, because the lack of order complicates their use. In order solve this problem we propose the use ontologies as organization model. Important principles of ontologies are reuse and resource sharing. For this reason it is necessary that the organization of concepts be the most generic possible, so that several tools can share it, and besides, that the relation should be based on generally accepted principles such that it can be understood and reused. This is very useful, because knowledge contained in an ontology can be used by some applications, which in turn can increase the number of users to justify the ontology costs incurred by its creation, customization, operation and maintenance.

To achieve the most generic ontology possible, linguistics [7] 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 formalize the query in Structured Query Language (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 in Web Ontology Language (OWL) [12], 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 define the database context. The definition of top-level classes is explained hereupon:

ElementosBD (ElementsDB). - They define categories where main relational database elements are classified [1]; for example: primary key, foreign key, etc. Some subcategories were omitted such as indexes or triggers, because they are not part of one query.

Palabra (Word). - Subcategories are POSs (noun, adjective, verb, adverb and other). We borrowed concepts from WordNet [15], 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 that "contains" synonyms. Synset subcategories are based on POSs, excepting 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 establish 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 (Synset_function). - 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 (which are identified with the most representative word form with a serial number, similar to WordNet [15]), 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 [17]. 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

Empirical evaluations have not tried to validate all the components of an NLIDB, neither to validate the answers that it provides, since there exist many involved factors: completeness of the knowledge base, syntactic and semantic parsing, and the type of queries of the test corpus (defined in [4]).

The experimental plan consists of three empirical evaluations for comparing the English Query's customization process, and the use of an ontology to customize an NLIDB using Protégé [9], one of the most popular ontology editors. In each of 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. Between the first one and the second evaluation, a small tuning experiment of ontology design was performed using five students, to improve the ontology design and the evaluation process.

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 the English Query documentation provided by Microsoft and a document that explains the proposed ontology approach. For evaluation No. 3 a document with customization examples was added for both approaches (EQ and the ontology approach). None of the participants had previous experience in English Query neither they had heard about ontology concepts. The participants for evaluation No. 3 were recruited from a university without a rigorous admission process; while those for evaluation No. 2 were recruited from Cenidet, a research institute with a rigorous admission process. Additional information of each evaluation team is showed in Table 1.

Table 1. Information of the evaluation teams

	Evaluation No. 2	Evaluation No. 3
Source	Research center	Private university
Query corpus (difficulty level low/medium/high)	7 (2/3/2)	8 (3/3/2)
Available documentation	English Query documentation and documentation of the ontology approach	Same documentation + examples.
Number of questions of the evaluation form	14	14
Participants' number	18	10

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's customization process, for eight queries from a corpus for evaluations No. 2 and No. 3.

Several NLIDBs define their own evaluation corpus [4], [6], [8]. We decided to use queries from the ELF corpus [3] because it is the most used, and selected a set of queries such that four queries were answered with the English Query default configuration and the other four were not. An interesting detail was found when comparing the ELF corpus with one created by ourselves, and another used in some other experiment [4]: although the three referred to the same database (NorthWind), the types of queries found in each corpus were very different. The first one has a majority of complex queries, the second one contains queries of little difficulty, and the third one consists of queries of different difficulty. Afterwards, we gathered a fourth corpus with queries from real database users that formulate queries to their operation databases; in this case again, the query types found were different from those of the previous three corpuses.

4.3 Description of the Evaluation

Questions of an evaluation form were grouped according to the main factors affecting the customization process of an NLIDB: configuration interface, customization methodology and other features, such as motivation 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 in a 0-100 scale. Two metrics used in other experiments (but not used here), were time spent on customization and quality of the resulting configuration. The time metric was excluded because the time invested in the customization was not possible to measure, since it was not possible to gather participants at the same time. The quality metric was excluded because we did not have a group of experts in ontology design to asses the quality of the ontology resulting from the customization.

5 Summary of Results

Evaluations No. 2 and No. 3 have the same evaluation procedure, the only difference consists of the evaluation teams' characteristics and the documentation handed out. The results for Evaluations No. 2 and No. 3 are shown, together with their standard deviations (within parenthesis), in Tables 2, 3 and 4 according to the three types of factors affecting the customization of an NLIDB, mentioned the previous section.

Figure 1 shows the differences between the averages of the evaluations of questions related to the customization interface of English Query and Protégé. In this figure a positive difference indicates that the ontology approach was better and a negative difference indicates the opposite.

Figure 2 shows the differences between the average evaluations of questions related to the customization methodology of English Query and the ontology approach.

Figure 3 shows the differences between average evaluations of questions related with diverse features of English Query and the ontology approach.

Table 2. Evaluation for questions related to the customization interface of English Query and the ontology approach

Question	English Query 2	Ont. App. 2	English Query 3	Ont. App. 3
1. I was comfortable with the configuration process tool after the training session.	51.04 (19.07)	72.92 (9.99)	62.50 (18.16)	70.83 (11.02)
2. The configuration process tool was easy to learn	47.92 (23.00)	71.88 (21.91)	60.42 (16.54)	58.33 (16.67)
3. The interface configuration process is manageable	56.25 (19.43)	76.04 (14.40)	62.50 (19.98)	70.83 (24.65)
4. The interface elements that are not in your native language affect the configuration process	44.79 (24.80)	68.75 (21.95)	50.00 (16.67)	75.00 (22.05)

Table 3. Evaluation for questions related to the customization methodology of English Query and the ontology approach

Question	English Query 2	Ont. App. 2	English Query 3	Ont. App. 3
1. The training process allowed me to understand the configuration methodology built into the tool	52.08 (20.31)	70.83 (13.82)	64.58 (21.14)	68.75 (15.45)
2. The documentation of configuration process is easy to understand	50.00 (22.05)	71.88 (12.80)	64.58 (17.55)	72.92 (11.60)
3. The terminology used in configuration process is strange or confusing	47.92 (24.91)	66.67 (14.43)	54.17 (19.98)	60.42 (8.07)
4. The necessary steps to carry out the configuration process were clear	40.63 (16.63)	69.79 (14.69)	72.92 (18.52)	68.75 (19.43)

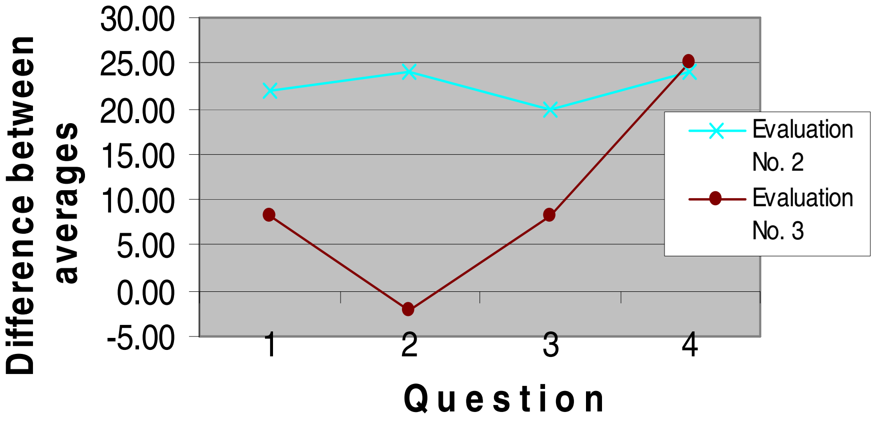

Fig. 1. Evolution of the differences between average evaluations of questions related to the interface of English Query and Protégé for evaluations No. 2 and No. 3

Table 4. Evaluation for questions related to diverse features of English Query and the ontology approach

Question	English Query 2	Ont. App. 2	English Query 3	Ont. App. 3
1. The training was adequate to make the configuration task	51.04 (19.96)	65.63 (14.99)	66.67 (18.63)	66.67 (16.67)
2. The configuration process is flexible	60.42 (15.45)	77.08 (11.60)	60.42 (16.54)	75.00 (18.63)
3. The configuration process is intelligible	55.21 (22.61)	76.04 (10.15)	62.50 (19.98)	72.92 (14.28)
4. Do you consider that the configuration hints at how the NLDIB works	53.13 (20.60)	79.17 (13.82)	60.42 (16.54)	72.92 (20.31)
5. I felt comfortable analyzing and filling concepts for the configuration process	57.29 (22.80)	76.04 (11.74)	62.50 (16.14)	68.75 (15.45)
6. I felt comfortable analyzing and filling relations for the configuration process	51.04 (19.07)	72.92 (16.54)	64.58 (17.55)	72.92 (8.07)

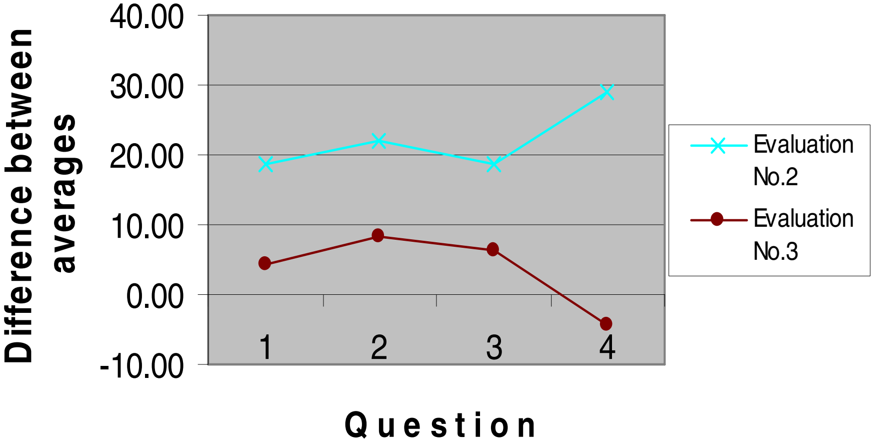

Fig. 2. Evolution of the differences between average evaluations related to the customization methodology of English Query and the ontology approach for evaluations No. 2 and No. 3

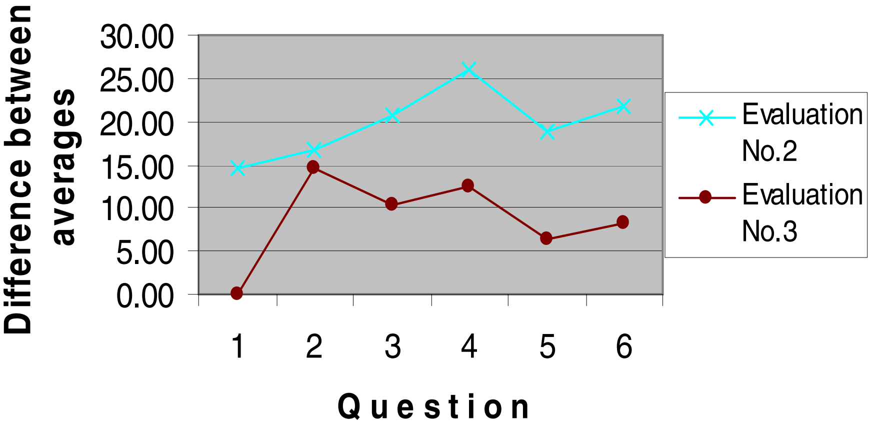

Fig. 3. Evolution of the differences between average evaluations related with diverse features of English Query and the ontology approach for evaluations No. 2 and No. 3

6 Discussion

The differences found between average evaluations from evaluations No. 2 and No. 3 favor our proposal in most of the aspects, since out of twenty-eight differences (fourteen for each evaluation), twenty-five are positive, two are negative and one is a tie (figures 1, 2 and 3).

An interesting detail is that the values from evaluation No. 2 are superior to those of evaluation No. 3, although this one had a more polished and complete documentation, and its participants had more time to learn the user-interfaces. A possible explanation is the difference between the selection processes of the students' institutions for each evaluation, since the students for evaluation No. 2 have to go through a rigorous selection process and as opposed to students for evaluation No. 3; consequently, the first ones must have a larger analysis capacity than the second ones.

The results for evaluations No. 2 and No. 3 favor our proposal, except in “The configuration process tool was easy to learn” (evaluation No. 3) and in “The necessary steps to carry out the configuration process were clear” (evaluation No. 3). The first exception can be explained because the participants of evaluation No. 3 have less experience using non commercial software, and the second exception can be accounted for by the a difference in autodidactic capacity, a skill more developed in the participants of evaluation No. 2.

7 Conclusions

Evaluations of the customization process of NLIDBs have not been found in the specialized literature; therefore, this work is pioneer in its field. Although there exists a great deal of work and interest in usability aspects for the design of user's interfaces, the customization process of knowledge bases is different, since it implies, besides certain repetitive tasks, activities that involve certain knowledge of the internal operation of the application and, for NLIDBs, linguistics knowledge.

Although English Query is a complete NLIDB and our proposed approach not, it was more desirable for the evaluation participants to know all the terms and its relationships, i.e., an explicit knowledge base (ontology), instead of the support elements (wizard, graphic editor of relations, transparency in the translation process, etc.).

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, that will be useful to the a 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.

References

1. Date, C.J.: An introduction to Database Systems, 7a edn. Addison Wesley Longman (2000)
2. English Language Front—end, http://www.elf-software.com/other.htm
3. English Language Front—end corpus, http://www.elf-software.com/FaceOff.htm
4. González, J.J., Pazos, B.R.A., Pérez, J.: A Domain Independent Natural Language Interface to Databases Capable of Processing Complex Queries, P.h. thesis, Cenidet, dic. (2005)
5. Inbase (last consult, June 18, 2007), http://www.inbase.artint.ru/english/default-eng.asp
6. Microsoft English Query documentation (last consult, June 18, 2007), http://www. microsoft.com/technet/prodtechnol/ sql/2000/reskit /part9/c3261.mspx

7. Miller, G.: Wordnet, a lexical database. Cognitive Science Laboratory, Princeton University (last consult, June 18, 2007), http://www.cogsci .princeton.edu/~wn/
8. Loos, E.E., Anderson, S., Day Jr., D.H., Jordan, P.C., Douglas Wingate, J. (eds.): Modular book: Glossary of linguistic terms (last consult, June 18, 2007), http://www.sil.org/ linguistics/ GlossaryOfLinguisticTerms/
9. Precise, http://cognews.com/1062409630/index_html
10. Protégé ontology editor: Stanford Medical Informatics at the Stanford University, School of Medicine (last consult, June 18, 2007), http://protege.stanford.edu/index.html
11. Richa, A.B.: Natural Language Interfaces: Comparing English Language Front End and English Query, Master of Science thesis, Virginia Commonwealth University, Richmond, Virginia (December 2004)
12. Sethi, V.: Natural Language Interfaces to Databases: MIS Impact, and a Survey of Their Use and Importance. Graduate School of business, Univ. Of Pittsburg, Pittsburgh, PA 15260
13. Sharoff, S.: SNOOP a system for development of linguistic processors. In: Proceedings of EWAIC93, Moscow (1993)
14. Web Ontology Language (OWL): w3c recommendation, http://www.w3.org/2004/ OWL/
15. Woods, W.A., Schmolze, J.: The KLONE family. Computers and mathematics with applications 23, 2–5 (1993)
16. Zarate, M.J.A., Pazos, R.R.A., Gelbukh, A., Padrón, C.J.I.: A Portable Natural Language Interface for Diverse Databases Using Ontologies. In: Gelbukh, A. (ed.) CICLing 2003. LNCS, vol. 2588, Springer, Heidelberg (2003)
17. Zarate, M.J.A., Pazos, R.R.A., Toledo, R.: Acquisition of lexical-syntactic relationships from a dictionary. In: 13tth international Congress on Computer Science Research, Tlanepantla, Mexico (September 2004)