User query understanding by the InBASE system as a source for a multilingual NLG module(first step)

Michael V. Boldasov[1], Elena G. Sokolova[2], and Michael G. Malkovsky[1]

[1] Moscow State University (MSU),
Computational Mathematic and Cybernetic department,
Moscow, Russia,
malk@cs.msu.su

[2] Russian Research Institute for Artificial Intelligence (RRIAI),
Moscow, Russia,
sokolova@aha.ru

Abstract. In the paper we consider the NL generation component of InBASE system - the system for understanding of NL queries to Data Bases. This component generates new NL-query from the internal InBASE Q-representation of the user query. During the planning phase a linear positioned query representation is constructed, positions bearing at first conceptual, then syntactic information. Realization phase deals with the NL means to express the concepts (objects, attributes, values, relations between objects and attributes). The NL generation component is conceived as the first step in the direction from one way question - answering system, as InBASE is now, to a larger scale information system able to communicate with user in different areas.

1 Introduction

Our first goal is practical - constructing a NLG module for a commercially oriented InBASE system (http://www.inbase.artint.ru/nl/nllist-eng.asp) which is used for the NL queries understanding and answering. InBASE deliver the information in the form of DB fragments. Problems can arise in misunderstanding of the user query by the system, so irrelevant information can be delivered without the user make sense of this. The more general way to resolve these problems is to construct a NLG module generating NL query from the internal query representation. For the user could control the correctness of the system understanding the system generates the rephrased query. It is shown together with the fragment of DB as to be a sort of explanation the structure of the data delivered. On the other hand the generation facilities would be a step to the interactive mood.

There exist some question answering systems available in Internet, for example, START (http://www.ai.mit.edu/projects/infolab/ailab.html), answering questions about USA geography. START and other systems available in Internet are monothematic. The InBASE system is multithematic. It is tuned on register of database queries and can be adjusted to any thematic area [1]. This makes it robust and well thought over, but this demands the same from the generation module. We are thus constrained in inventing the system's reactions.

The second but not the last goal of the investigation is theoretical. The theme of NLG is not discussed in relation to the practical Internet systems like, for example, the sited START system. It is investigated in the research projects like, for example, DRAFTER [5], GhostWriter [6], AGILE [4]. DRAFTER and GhostWriter were aimed to generate formal representations oriented to the NL communication. The data source were specifications oriented for consuming by computer. In difference to this the aim of AGILE project was the generation of NL text from a formal representation oriented to NL communication which were created in automatized mood by the user of the system. In the AGILE project software instructions were generated in three Slavic languages - Bulgarian, Czech and Russian. Also the English resources could be involved. Theoretically speaking we can think of the Russian resources from AGILE system as the means to generate user queries. But there arise two problems: the first one is that the Russian resources are manageable in the KPML environment [9] which is powerful research system but is too large for usage in commercially oriented system InBASE. The second is the fact that in difference to AGILE our generation module unifies two functions of the above types of systems: it has as the source specifications oriented for consuming by computer - Q representation of user query (an internal XML based SQL similar representation of the InBASE system) but the aim is NL generation. So NL communication oriented representation can be considered as the intermediate one.

In section 2 we consider the source representation its knowledge nature in difference to the systems like AGILE. Then we consider generation in InBASE system as the process broken down into the following phases: interpretation of Q-representation into the form convenient for the further generation process - planning, sec. 3; associating of Q-representation items to items of generator vocabulary, sec. 4; Grammatical realization of relations between lexified ODM concepts in Q-representation sec. 5;

2 Source Representation

Generation systems often follow the descriptive logic paradigm. Some of them including KPML direction and the AGILE system are based on KL-ONE [7] tradition. So the source knowledge base is presented in two parts: one is Terms box (T-box) which contains the hierarchy of concepts for Domain Model and the others are multiple expressions which could be constructed from the DM - Assertion box (A-box). So they are named individuals with the features defined in T-box. In multilingual generation systems usually language independence of the source representations is declared. But it is usually not the case.For example, in the AGILE system the DM is based on two ontologies - domain ontology of things and Upper Model [3] - the language specific ontology of things and processes concepts. The DM structure is organized in full on UM. For example, let it be the process OPEN - *to open* which is classified as in the UM DIRECTED-ACTION, there also exist roles Actor and Actee linking to the process its participants - USER - *user*, and SCREAN-OBJECT, in particular case - *a dialog window*. So we can construct a configuration and realize it in Russian: *Poljzovatelj* (Actor) *otkryvaet dialogovoje okno* (Actee) and in English: *The user* (Actor) *opens* (Actee) *the dialog box*. Language conditionality is clear from grammar nature of the type DIRECTED-ACTION.

Q representation is not semantic but a knowledge representation. It comprise not abstract semantic notions as DIRECTED-ACTION, but exact notions from the DB augmented and restructured to be a "DB Object Model" (ODM) [1]. This takes it closer to naive human mentality but not to language conditionality. ODM consists of objects and attributes related to the attributes and values of DB.

ODB can be also augmented by some notions of DB access register which are in the Dictionary of the InBASE system. They are operations and logical links used is SQL - not, count, asc., and so on. So we can think of the augmented ODM as T-box for our generation system.

We consider Q representations as A-boxes in our module. The illustrative part of the Q representation for the query is presented in figure 1. Below we'll speak of this part as of Q representation, omitting the technical details. The example shows that if the ODM is multiobjectival the elements of Q representation can form multilink chains. Figure 1 shows the Q representation for a query to a DB about selling cars. In the SELECT part the chain reflects the fact that the Cost is the attribute of a car as an Offer - the thing which you can buy, and that it is related to the Object Model. The chain in WHERE part reflects the fact that in the ODM the Model have the attribute Company by means of the object attribute Mark.

```
Q result=
   'SELECT avg(Model.Offer.Cost)
    FROM Model
    WHERE ((Model.Mark.Company.Company LIKE '\%LADA\%'))'
```

Fig. 1. The Q representation for the user query: *Average cost of LADA?*

3 Planning the NL-Oriented Representation

The aim of planning is to represent the content in NL oriented form. First of all this includes segmentation into fragments equal to clauses that can be presented as a NL simple sentences.

We have investigated the NL forms of user queries. Most of them are one sentence presented by a question, imperative or a noun group. The letter is the most frequent form in Russian. In figure 2 three variants of query realization are presented. The list can be continued by style variations - synonyms for negation, concepts, slang elements, ellipsis.

Employees without high school education.
Who has no secondary education?
List the employees having no high school education!

Fig. 2. Query variations

For the beginning, we consider the planning and realization of the noun group form. Q representation matches into the form very naturally since it is a description of an object - the object of user's interest.

The second task of planning process is to make the representation close to human perception of content. In our case during the analysis and interpretation some information which was not presented in the NL query is added automatically by the analyzer. This information is: a) the default attributes in the SELECT part, for example, when you ask *"who from the stuff"*, it is interpreted by the system to general information from object *"employee"*; this general information is meant as Name, Surname and Position attributes of Employee object; b) some information is added by the system in concrete situations, for example, the Q representation for the query in figure 2 is shown in figure 3.

You can see that in formalized form conditions are augmented because the fact that a person having no secondary education also has no higher education is presupposed and obvious for the user, so only persons with lower education degree must be delivered. You can also see the default attributes in the SELECT area. The fourth element Employee.Education is superfluous for the NL form, because it is added from WHERE area for the Education attribute would be delivered.

```
SELECT Employee.Surname, Employee.Name, Employee.Position,
Employee.Education FROM Employee WHERE
(Employee.Education<>'secondary') AND
    (Employee.Education<>'BA') AND
    (Employee.Education<>'MS')
```

Fig. 3. Q representation for the query shown in figure 2

4 Vocabulary

For wording of the Q-representation elements we need a vocabulary that associates them to lexical items. There is a vocabulary of the InBASE understanding module that describes many ODM and other lexical units as typed items having a number of attributes. Type of a lexical unit defines its attributes, as well as their semantic role in query text. For example, Object, Conjunction (comma, co-ordinate conjunction), Value, Unit (litre, dollar,), Relation (equal, less,), Functor (operation above the Q-representation elements, for example, count, sum,) etc. The lexical items types belong to socio-cultural concepts layer, that communicates directly with language semantics [8], but is not language-based. Vocabulary of the InBASE understanding module supplies ODM with lexical information and extends it by typical attributes values, functors and relations.

The generation process is not reverse to the understanding process. The attempt to use the understanding module vocabulary directly for generation would be confronted with a number of difficulties. Lets explain it by the following example. Here are lexical items extracted from vocabulary, associated with the Object *Employee* of employee ODM: *Employee* $\rightarrow$ *anybod, each, employee, everybod, individual, member, official, people, person, personnel, staff, who, worker, collaborator, colleague, specialist, guy*

One ODM concept can be associated with many lexical items. Some of these variants can be used in generation process for the style management. For example, *Employee* $\rightarrow$ *employee* (literature style), *Employee* $\rightarrow$ *guy* (colloquial style). There are words that have no association to ODM items, such as *anybod, each, everybod* etc. In some cases we could find only interrogative pronounces associated with an ODM item, and so have no lexical realization for generator.

Thus, we need to build a special generator vocabulary. We can develop it independently from the understanding module vocabulary or adapt it from the latter by importing items to the new generator vocabulary. For the first step the latter variant was chosen, but in the general case we should maintain a new multilingual generation module vocabulary.

Generator vocabulary items are of one of four types:

- Feature-Object - is the language equivalent of ODM item. Attributes:
  - ODM item ID
  - Language attributes and preferred flag
- Feature-Value - is the language equivalent of ODM item value. Attributes:
  - Search pattern in Where section of Q-representation
  - ODM attribute item ID
  - Language attributes and preferred flag
- Operation - is the language equivalent of operator ($LIKE, =, >, <, \geq, \leq, <>$) from Where section of Q-representation. Attributes:
  - Operator ID
  - ODM attribute item ID
  - Language attributes and preferred flag
- Action - is the language equivalent of functor (*count, sum, avg, max, min*) from Select section of Q-representation. Attributes:
  - Functor ID
  - ODM attribute item ID, for that functor is used
  - Language attributes and preferred flag

Every generation module vocabulary entry has the link to Q-representation or to ODM item referred from Q-representation. But because of preferred flag presented in every unit the reference from Q-representation to generator vocabulary is unique.

5 Grammatical Realization

Select:N,Def	
Ru:	En:
Case=Nom	
"цена"	*"the cost"*

From-object:N,Pl	
Ru:	En:
SEMANTIC LINK	
attribute-object	
Case=Gen	Pr="of"
"машин"	*"of cars"*

Where:Def
see the next table

Fig. 4. Preselects of three Q-query parts, "attribute-object" relation

Most NLG systems have a deal with semantic representations of text propositions, built by semantic relations formalism, as intermediate representation between domain representation and NL-text. For example, in the AGILE system [2] the semantic representation is presented in terms of SPL (Sentence Plan Language) of the KPML environment. In our Project we begin with the Q-representation and keep its concepts in the text plan. We consider the relations between the parts of Q representation in the context of NL -communication, so as a generalized semantic links. We think about realization "attribute-object" relation in the SELECT-FROM construction, and "object-attribute" relation in the FROM-WHERE construction. The generated text in difference to the user query has a regular form. The principles we follow are: all values must be presented explicitly with their Attributes, and all the relations between elements must be expressed by grammar means (in particular, by zero means). In figures 4, 5 we present our grammar in a table form. It can be considered as a fragment of a grammar network similar to the network in SFG tradition based on UM notions: it has semantic relations, choices, and we can think of choosers formulation. The conceptual difference between the KPML grammar network and our grammar fragment presented in figures 4, 5 is that our network is syntagmatic (it describes what grammar means are available to express concrete object configuration) in difference to KPML's one which is paradigmatic (it describes the organization of language means). So our network also includes the syntactic motivations for the realization.

In the figures semantic and syntactic relations are presented bold, Generalized semantic relation "attribute-object" (figure 4) is presented and regularly expressed by Genitive construction - "of" preposition in English. The "object-attribute" (figure 5) relation is realized variously according to the object's semantics . It is shown by semantic relations that we took from [10].

"Object-attribute" relation can be realized also in subordinate proposition with some syntactic motivation of the expression. Proposition is represented as a relative clause with "ghost participant" from the main part expressed by relational pronoun, or as attributive clause with the predicate expressed by a participle form. The clause is organized on the base of a predicate. In our three elements sequence: From-object - Attribute - Value, we have three possibilities: Attribute realized by a verb, HAVE verb between From-object and Attribute with the Value adjoined to the Attribute, BE verb between Attribute and Value. The possibilities are presented in figure 5.

	SEMANTIC LINK		SYNTACTIC LINK		
	Ru:	En:	Ru	En	
Prepositional group	**comitative**		**attributive clause**		Clause
	Where:N,Sg		Where:V,Active participle		
	Case=Instr Pr="c"	Pr="with"			
	с ценой [.] *со стоимостью [.]*	*with the price of* *with the cost of*	*весящие [.]*	*weighing [.]*	
	transitive		**have attributive clause**		
	Where:N,Sg		HAVE,Where:N,Sg		
	Case=Dative Pr="по"	Pr="at"	Case=Acc		
	по цене [.] **по стоимости [.]*	*at the price of [.]* *at the cost of [.]*	*имеющие цену [.]*	*having the price [.]*	
Nominal group	**mediative**		**relative clause**		
	Where:N,Sg	*	Where:V,Present tense, 3p Ghost participant:From-object,Nom		
	Case=Instr				
	**ценой [.]* *стоимостью [.]*		*которые весят [.]*	*that weighs [.]*	
	qualitative		**have relative clause**		
	Where:N,Sg	*	HAVE,Where:N,Sg Ghost participant:From-object,Nom		
	Case=Gen		Case=Acc		
	завода [.] **цены [.]*		*которые имеют цену [.]*	*that have the price [.]*	
			be relative clause		
			BE,Where:N,Sg	*	
			Case=Nom Pr="у"		
			у которых цена - [.]		
			inv be relative clause		
			Where:N,Sg Ghost participant:From-object,Nom		
			Case=Nom		
			цена которых - [.]	*the price of which is [.]*	

Fig. 5. "Object-attribute" relation

All the elements are pairs of Attributes SELECT-FROM and FROM-WHERE form chains which must be organized into coordinative chains. There is no problem in the SELECT part, but some problems arise in the WHERE part having different means for realization. Rules ought to be formulated, for example, SEMANTIC realization first, SYNTACTIC - after, for example, *a man with a stick that has a hat* rather than *a man that has a stick with a hat.*

6 Conclusion

We tried to think about large scale generation machine in a small fragment of generation task. We have reduced the form of the generated text, but we didn't reduced the problems arising. For the first step we tried to think about linear organization of generation process. Maybe it is more relevant for the real text generation activity, that constructing at first full structure of a sentence. The generator interface was developed under C++ platform as COM-Interface like other components of the InBASE system. The morphology stage was realized using external Morph Generator "Lemmatizer" COM-Interface.

References

1. Zhigalov, V.A., Sokolova, E.G.: InBASE Technology: Constructing Natural Language Interface with Databases. (in Russian) Computational Linguistics and Its Applications International Workshop Proceedings, Vol.2., Dialogue, Aksakovo, (2001) 123–135
2. Sokolova, E.G., Sharoff S.A.: Multilingual User-Guides Generation in Slavonic Languages. (in Russian) Seventh National Conference in Artificial Intelligence KII-2000. Pereslavlj-Zalesskij (October, 2000) 409–417
3. Bateman, J., Renate, H., Fabio, R.: The Generalized Upper Model 2.0. GMD-IPSI Technical Report. URL: http://www.darmstadt.gmd.de/publish/komet/gen-um/newUM.html (1996)
4. Kruijff, G-J., Bateman, J., Dochev, D., Hana, J., Kruijff-Korbayova, I., Skoumalova, H., Sharoff, S., Sokolova, E., Staykova, K., Teich, E.: Multilinguality in a Text Generation System for Three Slavic Languages. Proc. COLING-2000, Luxemburg, (August, 2000)
5. Paris, C., Vander, Linden K., Fischer, M., Hartley, A., Pemberton, L., Power, R., Scott, D.: A support tool for writing multilingual instructions. Proc. IJCAI'95. Montreal, August 20-25 (1995) 1398–1404.
6. Reiter, E., Mellish, C., Levine, J.: Automatic generation of technical documentation. Applied Artificial Intelligence, Vol. 9, no. 3 (1995).
7. Woods, W.A., Schmolze, J.G.: The KL-ONE Family. In: Computers and Mathematics with Applications: 23:2-5, (1992) 133–178.
8. Halliday, M.A.K., Edward, A.: An Introduction to Functional. London (1985)
9. Bateman, J.A.: Enabling technology for multilingal natural language generation: the KPML development environment. Natural Language Engineering, 3 (1), (1997) 15-55
10. Zolotova, G.A.: Syntactic dictionary. Repertory of the element units for Russian syntax. (in Russian) Nauka, Moscow, (1988)