Using Natural Language for Database Design

Edith Buchholz * and **Antje Düsterhöft**

Department of Computer Science
University of Rostock, A.-Einstein-Str.21
18059 Rostock, Germany
Email: {buch,duest}@informatik.uni-rostock.de

Abstract.

This paper deals with a natural language dialogue tool for supporting the database design process. We want to illustrate how natural language (German) can be used for obtaining a skeleton design and for supporting the acquisition of semantics of the prospective database. The approach is based on the assumption that verbs form a central part in defining the meaning of sentences and imply semantic roles in the sentences which have to be filled by objects. We are using a moderated dialogue for drawing the designer's attention to these objects in order to extract comprehensive information about the domain.

1 Introduction

The quality of database design is a decisive factor for the efficiency of a database application. A database designer has to use a high level of abstraction for mapping his real-world application onto an entity relationship model. The designer has to learn the model and the constraints to use it.

Natural language can be exploited in order to overcome this bottleneck. From our point of view a user-friendly design system has to have two supporting tools: firstly, a tool which makes available an interface for obtaining a natural language description of an application and secondly, a tool for paraphrasing database schemes in a natural language way (see also [FloPR85]).

[ColGS83], [TseCY92], [TjoB93] are presenting various methods dealing with natural language as input for database design systems. These systems are based on natural language texts for the requirement specification in the data base design process. This paper illustrates how natural language in a dialogue tool can be used for gathering the knowledge of the designer and how it can be transfered into an extended entity-relationship model. The dialogue together with the knowledge base will be used for drawing to the designer's attention special facts resulting from the syntactic, the semantic and the pragmatic analyses. The tool makes suggestions for completing the design applying the knowledge base.

* This work is supported by DFG project TH 456/2-2.

In the database design project RAD ([ThaA94]) we have implemented a rule-based dialogue design tool for getting a skeleton design on the basis of the extended entity-relationship model HERM [Tha91]. The designer describes the structure of an application in German. The specification and formalisation of semantic constraints is one of the most complex problems for the designer. Within natural language sentences the designer uses semantic constraints intuitively. For that reason, within the natural language design process we focus on extracting comprehensive semantic information about the domain from natural language utterances. The results of the dialogue are available in the internal DataDictionary for the other tools (grahical interface, integrity checker, strategy adviser,...) of the system. Within the RAD system the designer can use these results for various forms of representation, e.g. a graphical representation. The skeleton design with the semantic constraints is also the basis for further semantic checks, e.g. of key candidates, and will restrict the search areas in the checking process.

For the theoretical and pragmatic analyses of the language used within the design dialogue it was necessary to do this with a practical example. So we decided to choose the field of library - its tasks and processes. As a method of obtaining the linguistic corpus we carried out a number of interviews with librariens and library users. The extracted corpus was analysed statistically to obtain the frequency of word forms and the occurence of synonyms and homonyms. Starting from this domain we developed relations to other domains (see [BucD94]).

The dialogue tool will be implemented in PROLOG.

2 The structure of the dialogue tool

For the acquisition of designer knowledge we decided to choose a moderated dialogue tool. A moderated dialogue can be seen as a question-answer-tool. The tool asks for input or additional questions considering the acquisition of database design information. These questions are frames which will be updated in the dialogue process. The designer can formulate the answer in natural language

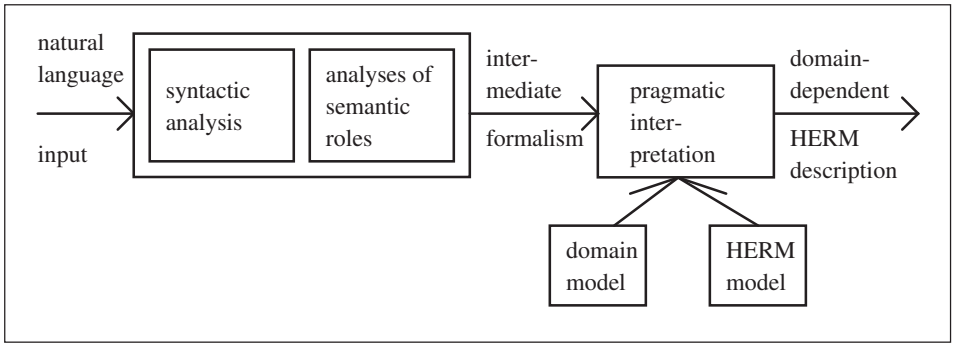

Fig. 1. Two-stage Dialogue interpretation tool

sentences. Each sentence will be analysed syntactically as well as semantically and then transformed into HERM stuctures.

Within the dialogue the results of the syntactic, semantic and pragmatic analyses will be used for controlling the dialogue. That means, if an incomplete designer input is received a question will be initiated. Inputs are incomplete if either semantic roles are not complete or the newly generated design model is incomplete. Semantic roles are filled within the semantic analysis. The pragmatics realizes the transformation of the natural language sentences into HERM structures.

2.1 Syntactic analysis

The syntactic analysis of the natural language input of the designer is based on a GPSG parser (Generalized Phrase Structure Grammar) [Gaz85]. GPSG belongs to the family of Unification Grammars. A basic feature is the introduction of ID/LP Rules (Immediate Dominance/ Linear Precedence). Immediate Dominance determines the immediate dominance of a root over its followers, Linear Precedence determines the order in which the follower, e.g. syntactic categories are to be processed.
The parser implemented in our tool uses the Earley algorithm [Ear70].

2.2 Semantic analysis

Interpreting the semantics of the designer input we are using the model of Bierwisch [Bie88] which inserts a semantic level between the syntax level and the conceptual level (HERM data model).

We assume that verbs form a central part in defining the meaning of sentences and the relationships between parts of sentences. Basically they describe actions, processes and states. We have tried to find a classification of verb semantics that can be applied to all verbs in the German language. Our aim was to keep the number of classes small and fairly general but large enough to identify their function in a sentence correctly. This classification (see also [BucD94]) is, at this stage, independent of the domain to be analysed (cf.Fig.2).

To identify the meaning of sentences we have used the model of semantic roles. Verbs of a special class imply the occurence of semantic roles. The units in a sentence or an utterance are seen to fulfil certain roles. Our role concept is mainly based on the hypothesis by Jackendoff [Jac83] and consists of the following roles which refer to the objects partaking in the action: Cause, Theme, Result/ Goal, Source, Locative, Temporal, Mode, Voice/Aspect. The following example illustrates the role concept.

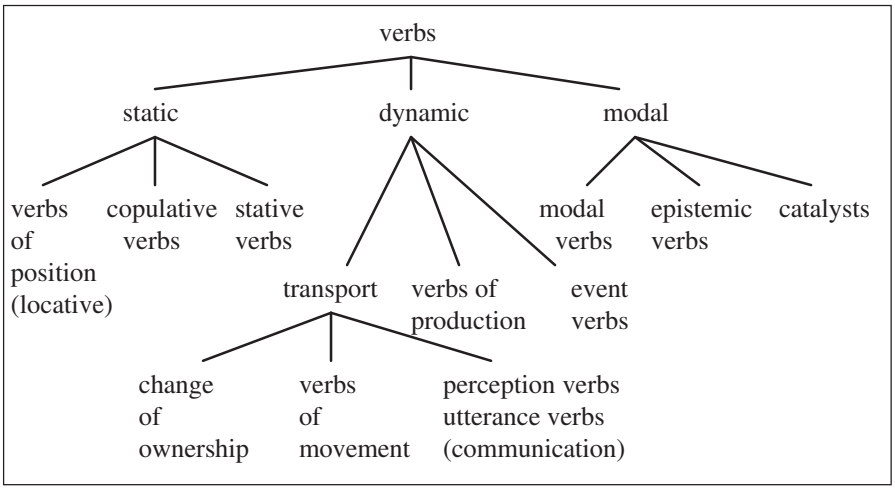

Fig.2. Verb classification

Example. 'The user borrows a book with a borrowing-slip'

```
results of the semantic analysis:
verb type:   change of ownership
subject:     the user
object:      a book
locative:    ?*
temporal:    ?*
mode:        with a borrowing-slip
```

(* an additional question will be initiated)

2.3 Pragmatic interpretation

2.3.1 Obtaining a skeleton design

The transformation of the structure of natural language sentences into EER model structures is a process which is based on heuristic assumptions, e.g., we assume that all nouns are entities. [TjoB93] illustrate a large number of such heuristics in an informal way. If we accept these heuristics then we can formalize them using contextfree and contextsensitive rules.

Example.
/* all nouns are transferred into entities */
`N(X) → entity(NAME,X).`

/* sentences with the main verb 'have' are transferred into an entity (the subject) and the according attribute (the object of the sentence) */

```
N(X),subject(X),V(haben),N(Y),object(Y)
   →entity(X),attre(X,Y).
```

Considering the results of the syntactic analysis of a natural language sentence we can describe these results using a tuple structure.

Example. The tuple structure of the sentence 'the user borrows a book with a borrowing-slip' is:

```
S(NP(DET(the),N(user)),
  VP(VP(V(borrows),NP(DET(a),N(book)),
                  PP(PRAEP(with),
                     NP(DET(a),
                        N(borrowing-
                          slip)))))
```

The tuple can be seen as a language which can be described by a grammar, e.g. terminals are N, DET or VP. The HERM model can also be seen as a language if predicates are used to describe the elements of the model. *Now we can handle the transformation as a compiler process using an attribute grammar.* The heuristics are integrated into grammar rules as well as into semantic rules. A compiler for this purpose has been developed. The following example illustrates how the transformation is realized.

Example. Transforming the utterance 'at the library' into an entity named 'library' using a contextfree grammar formalism. (The small letters identify nonterminals, and the capital letters are terminals. '$x' is a variable. 'assert(X)' asserts 'X' to the model description.)

tuple structure:
`S(PP(PRAEP(at),NP(DET(a),N(library))))`

grammar rules:

```
start → S(phrase)
phrase → PP(pp_phrase)
pp_phrase → PRAEP($x),NP(np_phrase)
np_phrase → NP(det_phrase,n_phrase)
det_phrase → DET($x)
n_phrase → N($x) {assert(entity($x))}
```

The advantage of this approach is that we can define actions at the word category level as well as at the sentence phrase level. So, it is possible to define database design actions, e.g. when considering the occurence of a genitive nominal phrase connected with another nominal phrase in the sentence. The heuristics underlying is that a genitive nominal phrase has an attribute function concerning the corresponding nominal phrase.

We are using a dialogue in which the designer can formulate a description of an application in several sentences. For that reason we have to deal with the problem of inserting a new part of a design into an existing design. We have implemented a two-step approach. Firstly, a seperate design will be generated from the sentence of the user. Secondly, the design description will be updated inserting the new design part. Common heuristics are the basis of the updating process (cf. [Düs94]).

2.3.2 Extracting information on behaviour

In most cases a database will be used for complex processes. In order to be able to maintain the database we have to define transactions. (For the reasons of using transactions see [Tha94:114].) The behaviour of the database can help to make the system more efficient and faster and thus to save time and money.

Behaviour can best be gained from a knowledge base. One form of presenting the domain is by classification of the processes involved as a conceptual graph. The knowledge base will be used for gathering relevant processes of the application and is based on the results of the semantic analysis. Each application can be classified. Lending processes are identified by verbs of the class

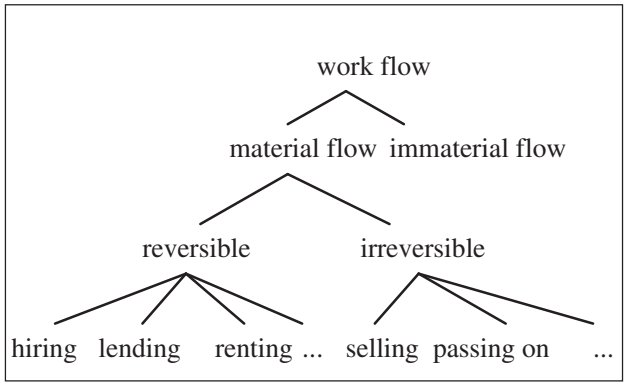

Fig. 3. Part of the process classification

'change of ownership'. The library processes or the 'rent a car' processes (cf. Fig. 3) belong to this group.

The lending process as a complex process can be further classified into a number of pre and post processes (cf. Fig. 4). These processes are included in the knowledge base. If a user input contains one of these processes a possible classification will be defined and an action within the dialogue will be initiated.The pre and post processes in Fig. 4 can be further subdivided into processes which are summarized in the above classification. Lending thus requires the processes of obtaining a user card, updating the user card if need be checking whether the book is held and available, filling in a borrowing-slip and signing it.

Example. The sentence 'the user borrows a book with borrowing-slip' implies the following general questions (borrowing has the synonym lending):

```
preprocesses:
1) Is the process 'obtaining' situated before
     'lending' ?
2) Is the process 'registration' situated
     before 'lending' ?
main processes:
3) Is the process 'document exists' situated
     before 'lending' ?
4) Is the process 'document valid' situated
     before 'lending' ?
     ...
postprocesses:
5) Is the process 'returning' situated after
     'lending' ?
```

The designer has to give correct answers.

3 Conclusions/ Future Topics

We have presented a dialogue tool consisting of a syntax analyser, a semantic role definer and a pragmatics interpreter. The dialogue tool gathers information on structure, semantics and behaviour of the prospective database. By means of transformation rules this information is mapped onto the HERM model.

The advantage of the dialogue tool is that the designer can describe the requirements of the database system in a natural language (German) and thus can specify the knowledge of a domain in a natural way. This knowledge is then employed for gathering database constructs such as entities, attributes, cardinalities, constraints, etc.

The efficiency of the database greatly depends on the exact interpretation and transformation of the natural language input analysis. The accuracy, on the other hand, depends on the size and complexity of the grammar used and the scope of the lexicon.

Work in future has to concentrate on extending the grammar to comprise all types of sentences and other hitherto excluded parts of grammar and on ways of steadily increasing the lexicon. For reasons of integrity we cannot leave updating of the lexicon to the chance designer who may have no linguistic training. Much work will have to go into completing and maintaining the linguistic background before it can finally be used for any type of systems design.

A second future topic is the application of the linguistic knowledge for acquiring further semantic information of the prospective database, e.g. acquiring key attributes or functional dependencies.

Acknowledgements

We are grateful to Bernhard Thalheim for his guidance and support of our work and for his helpful criticism and suggestions.

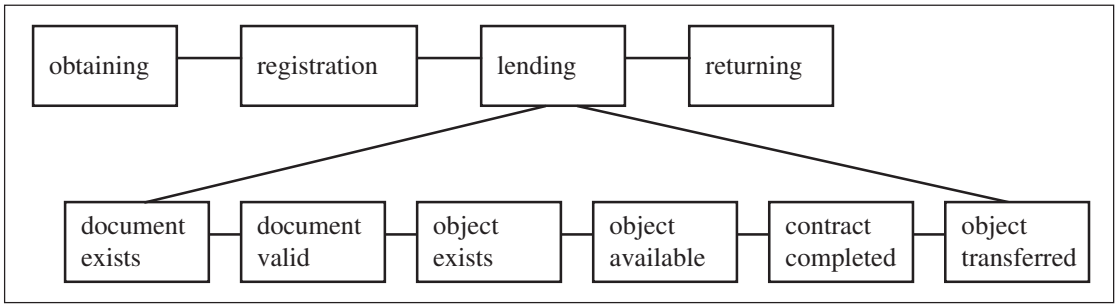

Fig. 4. Part of the knowledge base: pre, main and post processes of the act/borrowing/ lending

References

[Bie88] Bierwisch, M., Motsch, W., Zimmermann, I. : Syntax, Semantik und Lexikon. Berlin,Akademie Verlag, 1988

[BucD94] Buchholz, E., Düsterhöft, A.: The linguistic backbone of a natural language interface for database design. In: LLC ?/94, Oxford University Press

[ColGS83] Colombetti, M.; Guida, G.; Somalvico, M.: NLDA: A Natural Language Reasoning System for the Analysis of Data Base Requirements. In: Ceri, S. (ed.): Methodology and Tools for Data Base Design. North-Holland, 1983

[Düs94] Düsterhöft, A.: Zur Vorgehensweise bei der pragmatischen Inter pretation natürlichsprachiger Äußerungen Im Datenbankentwurf. Preprint 4/94, Fachbereich Informatik, Universität Rostock

[Ear70] Earley, J.: An efficient context-free parsing algorithm. Comm. ACM13:2, S.94-102

[Eic84] Eick, Ch.F.: From Natural Language Requirements to Good Data Base Definitions - A Data Base Design Methodology. In: Proc. of the International Conference on Data Engineering, pp.324-331, Los Angeles, USA, 24.-27.4.1984

[FloPR85] Flores, B.; Proix, C.; Rolland, C.: An Intelligent Tool for Information Design. Proc. of the Fourth Scandinavian Research Seminar of Information Modeling and Data Base Management. Ellivuori, Finnland, 1985

[Gaz85] Gazdar, G.; Klein, E.; Pullum, G.; Sag, I.: Generalized Phrase Structure Grammar. Havard University Press Cambridge, Mass. 1985

[GolS91] Goldstein, R.C.; Storey, V.C.: Commonsense Reasoning in Database Design. Proc. of the 10th International Conference on Entity-Relationship Approach, San Mateo, California, USA, 23.-25.October 1991, pp.77-92

[Jac83] Jackendoff, R.: Semantics and cognition. MIT Press, Cambridge Mass., 1983

[Tha91] Thalheim, B.: Intelligent Database Design Using an Extended Entity-Relationship Model. Berichte des Fachbereiches Informatik 02-1991, Universität Rostock.

[Tha94] Thalheim B.: Fundamentals of Entity-Relationship Modeling. Springer Verlag 1994, Forthcoming

[ThaA94] Thalheim, B., Albrecht, M., Altus, M., Buchholz, E., Düsterhöft, A., Schewe, K.-D.: Die Intelligente Tool Box zum Datenbank entwurf RAD. Workshop "Benutzerschnitstellen“,17.-19.März1994, Kassel

[TjoB93] Tjoa, A.M., Berger, L.: Transformation of Requirements Specifications Expressed in Natural Language into an EER Model. Proceeding of the 12thInternational Conference on ER-Approach, Airlington, Texas USA,Dec. 15-17th, 1993