

Generating DB Queries for Web NL Requests Using Schema Information and DB Content

Bernhard Thalheim¹, Thomas Kobiemia²

¹ Computer Science, BTU Cottbus, D - 03013 Cottbus
thalheim@informatik.tu-cottbus.de

² AVC Kommunikationssysteme GmbH, L. Braille-Str. 2, D - 03044 Cottbus
kobiemia@avc-online.de

Abstract. In the internet, ad-hoc natural language based querying is the normal way to access internet sites. To support this we built a tool which analyses NL utterances, compares them with the information known for the database which is under request and generates SQL queries on the basis of this analysis. The SQL queries can be used for querying the corresponding database. The queries are ordered by similarity of schema matching.

1 The Problem

NL Access to SQL Databases

Natural language access to databases has been already considered several times in the literature. In most cases it was based on the access to relational database systems, e.g. [Cuf84,Him88,KhN95]¹. The nonsuccess of these approaches is caused by one of the drawbacks of the relational database model implementation: its lack of representation of semantical associations among types beside the attribute names and foreign keys. Furthermore, the scope of the projects usually has been 'holistic', i.e. targeting the general solution with an approach pictured in Figure 1. Also, critical surveys [Dek94,May80] did not give NL access to databases any real chance. Some approaches [BoF92,Luk89,Sab90,Wik97] successfully used semantic database models.



Fig. 1. Two Approaches to SQL Query Generation

¹ For a detailed analysis, a detailed presentation of the approach with a discussion of examples and the theoretical background see [ThK01]

The Challenge of Internet Applications

In the internet age a sophisticated dynamic support for users becomes even more crucial. Users of web systems did not get a CS education. Neither they know query languages. They want to dynamically ask their requests directly to the engine without knowing the specific vocabulary used in the engine. Thus, NL request to a variety internet databases is a must.

The Problem Tackled And Solved In This Paper:

Is it possible to dynamically generate the support which enables an internet service to answer questions to the system which are ask in natural language?

2 The Cottbus Intelligent NL Request Transformer

Our solution which has been successfully used in internet service sites such as city information systems, edutainment sites and B2C sites uses the following restrictions and specific approaches:

- The database content is known and can be partially used.
We use sophisticated analysis support for NL utterances developed for the RADD system [BDT96].
- The database structure and the functionality of the DBMS is known.

We use an extended entity-relationship model (HERM) that allows to complete this task [Thu00]

The Intelligent NL Request Transformer allows to

- *analyze natural language utterances* and transform these into syntax trees,
- *unify variables in the syntax tree with database content*, and
- *generate a set of SQL queries* which can be applied to the databases used for support of the information service.

Instead of approaches in Figure 1 we use the three-step approach in Figure 2.

The architecture of the tool developed so far is displayed in Figure 3. The tool consists of the following main components:

Query liquefaction: This component allows to melt the request into parts, to integrate the parts with the database and the meta-data contained in the database schema manager.

DB schema manager: The request is compiled in the liquefaction component by analyzing the utterance and matching the utterance against the thesaurus and against the database schema. Thus, the schema manager keeps the schema information in a form which enables in matching. It is based on an *ER schema storage engine*, an *ER2R translation suite* enabling in flexibility of translation or compilation from ER schemata to relational schemata and integrates *various DB design workbenches*.

Generation of SQL query candidates based on full information

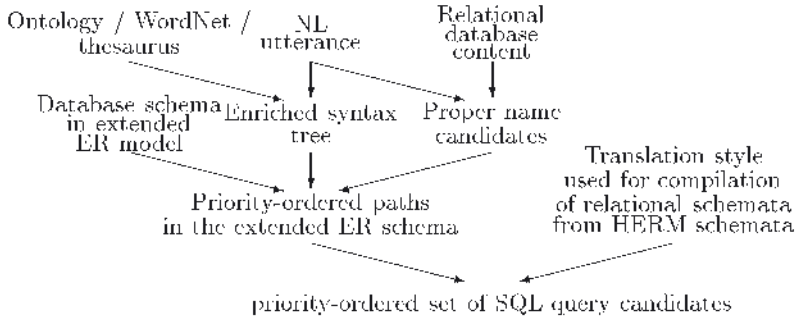


Fig. 2. Three-step Approach to SQL Query Generation

DB thesaurus manager: In order to support the user specific vocabulary is integrated based on *ontology* and specific *lexicons* and is to be extended by such as WordNet. Extraction of proper names in utterances is supported by *database content extraction*. Further, the thesaurus relates natural language expressions with short-cut and specific expressions used by the database developer.

The query liquefaction tool consists of three sub-components:

Syntactical analysis for analysis NL utterances and generation of a (small) number of syntax trees with priorities. The analysis uses the *parser* of the RADD project [BDT96]. It is based on *ID/LP grammars*. Further, a sophisticated *lexicon* is used. The architecture of the syntactical analysis component can be roughly represented as displayed in Figure 4. The number of trees generated might be large. We use heuristics and schema information for ordering this set. Only trees with high preference are considered in the next steps.

Intelligent path extraction: This sub-component uses the syntax tree and the information maintained in the database schema manager and the information provided by the thesaurus for generation of paths in the ER schema. The main functionality of this component is displayed in Figure 5.

Relational query melting-pot: After we have generated a number of paths in the ER schema we generate now SQL queries depending on the translation style. Then the ER path can be translated to a relational path with hinge attributes, and directed by applying known ER schema semantics.

The detailed description of the **database schema manager** and of the **database thesaurus manager** is out of the scope of the paper due to space limitations. It can be found in [ThK01] and cited there references.

3 Generalizations and Application

The tool has been integrated into various websites by industrial cooperation partners which used their ‘web presenters’ for presenting results of querying.

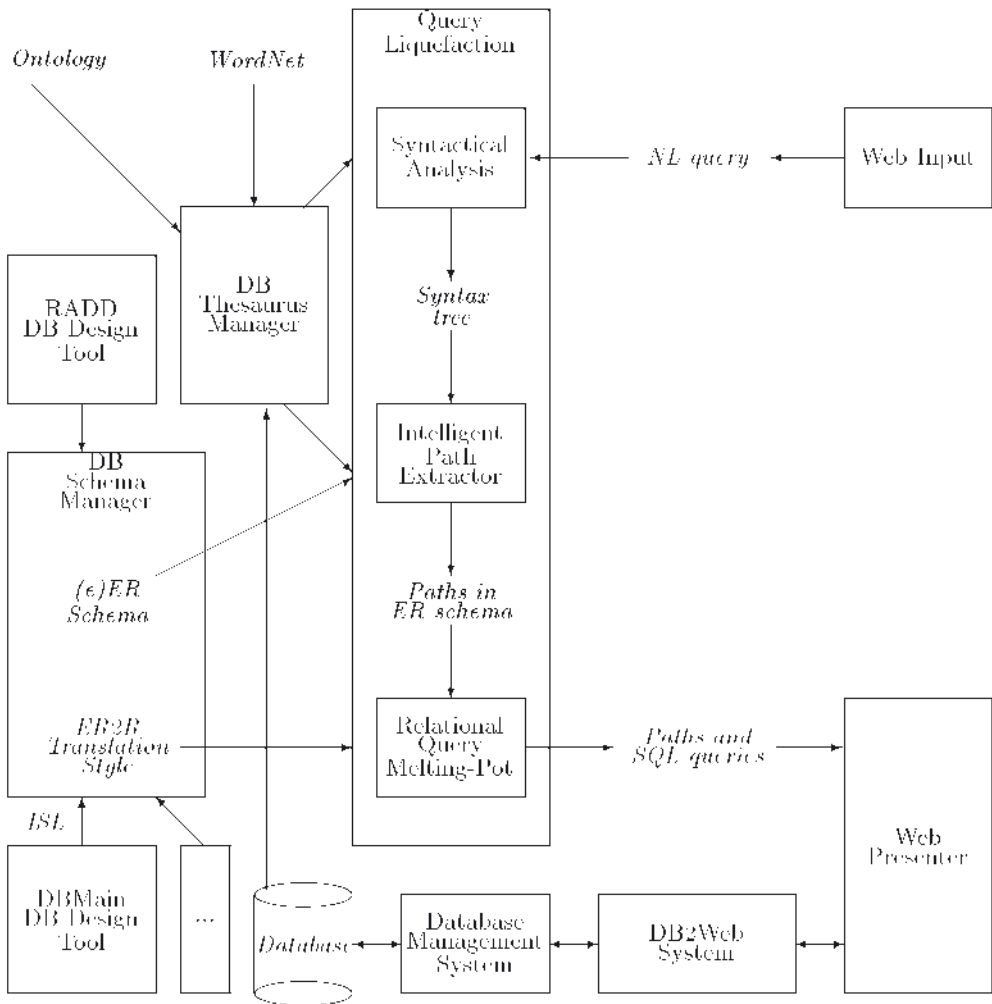


Fig. 3. The Cottbus Intelligent NL Request Transformer

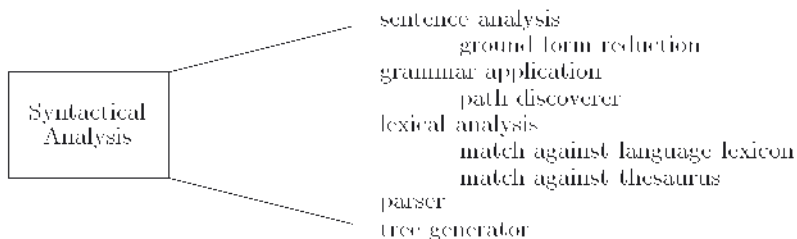


Fig. 4. The Syntactical Analysis Sub-Component

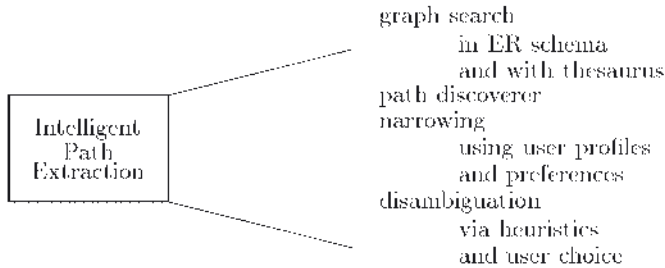


Fig. 5. The Intelligent Path Extraction Sub-Component

The *Cottbus Intelligent NL Request Transformer* allows to analyze NL utterances into SQL queries depending on the database schema, the database content and using additional lexicons and an ontology. The extension of the query and thesaurus vocabulary by systems such as WordNet is currently under development. Furthermore, a speech recognizer is currently integrated into the systems.

References

- [BDT96] E. Buchholz, A. Düsterhöft, and B. Thalheim, Capturing information on behavior with the RAD3-NLI: A linguistic and knowledge based approach. Proc. Second Workshop on Applications of Natural Language to Database Design (ed. J.F.M. Burg), Amsterdam, 1996.
- [BoF92] G. Bono, P. Fitorilli, Natural language restatement of queries expressed in a graphical language. ER 1992, 357-374.
- [Cuf84] R. N. Cuff, HERCULES: Database query using natural language fragments. BNCOD 1984, 133-149.
- [Dek94] S. M. Dekleva, Is natural language querying practical? DATA BASE 25(2), 24-36 (1994).
- [Him88] S. Himbaut, Tell-Me: A natural language query system. EDBT 1988, 583-587.
- [KhN95] H. Khelalfa, O. Nouali, Sigar : Generating responses in a natural language query system. NLD3 1995.
- [Luk89] W. S. Luk, Building natural language interface to an ER database. ER 1989, 345-360.
- [May80] E. Mays, Failures in natural language systems: Applications to data base query systems. AAAI 1980, 327-330.
- [Sab90] S. Sabbagh, SESAME: An application of entity-relationship models to a natural language user interface. ER 1990, 331-343.
- [Tha00] B. Thalheim, Entity-relationship modeling - Fundamentals of database technology. Springer, Berlin, 2000.
- [ThK01] B. Thalheim, T. Kobienia, From NL DB request to intelligent NL DB answer. Preprint I-6-2001, BTU Cottbus, Computer Science, 2001 (available through <http://www.informatik.tu-cottbus.de/~thalheim>).
- [Wik97] W. Winiwarter, Y. Kambayashi, DOA - The deductive object-oriented approach to the development of adaptive natural language interfaces (Abstract). BNCOD 1997, 137-138.