Analysis of Spoken Input to C2 Systems

Dr. Matthias Hecking
FGAN/FKIE
Neuenahrer Straße 20
53343 Wachtberg-Werthoven
Germany
Phone: +49 228 9435 576
Fax: +49 228 9435 685
hecking@fgan.de

Abstract

A natural way to communicate with C2 systems would be to use natural language. There are already natural language components used in military systems, e.g. CommandTalk is a spoken-language interface to the ModSAF battlefield simulator. In our project NATLAC we try to realize a query component for the ATCCIS database and we try to show that the available methods, techniques, and tools of computational linguistics are mature enough to look whether they are applicable to C2 systems. In this paper, we first describe several military systems with HLT (human language technology) components. In the main part, we will present the project NATLAC, how the syntactic analysis is done and how the logic-based meaning representation is transformed into SQL statements for querying the ATCCIS database.

1. **Introduction**

New military operations and growing coalitions demand for more complex C2 systems. A natural way to communicate with C2 systems would be to use *natural language*. If the C2 systems would have the ability to process spoken language, this would simplify the usage of the systems.

In the NATO technical report *Potentials of Speech and Language Technology Systems for Military Use: an Application and Technology Oriented Survey* (see [Steeneken, 1996]) efficient speech communication was recognized as a critical capability in many military applications, among other things in 'command and control'. There are already natural language components used in military systems, e.g. CommandTalk is a spoken-language interface to the ModSAF battlefield simulator (see [Moore et al., 1997]). Operational C2 systems with HLT components are not known.

CommandTalk is one of the rare systems with *language processing* capabilities. This term comprises those technologies to understand what was spoken (the meaning). Language processing includes syntactic, semantic, and dialogue analysis.

Today, the usability of human language technology (HLT) is restricted to narrow and well defined application areas (domains). Another requirement is that the language must be restricted as well. This means, that the vocabulary and the grammatical structures must be limited enough such that processing time becomes acceptable. The military domain and the stereotyped military command language seem to be appropriated for using HLT.

In our project NATLAC (NATural Language ACcess) we try to show that the available methods, techniques, and tools of computational linguistics are mature enough to look whether they are applicable to C2 systems. The NATLAC system uses the ATCCIS database (see [NATO, 2000]) as the domain. We have already reported on the progress of the project using a speech recognizer (see [Hecking, 2001]).

In the next section, we will first introduce several military systems with HLT (human language technology) components. In the main part of this paper, we will present the project NATLAC. After explaining the overall architecture the realization of the *syntactic analysis* will be described. During the syntactic analysis a formal description of the *meaning* of the natural language input is constructed. We will show how the logic-based meaning is transformed into SQL statements for querying the ATCCIS database.

2. **Military HLT-Systems**

There are military systems in use, which contain HLT components. In [Hunt, 2000] a cross-language automatic interpreting system is described which the NATO forces use. The system is able to recognize 4000 phrases in English (e.g. "I am a member of the NATO peacekeeping forces.") and to play the corresponding spoken phrase in another language through a loudspeaker. The system was developed with a COTS speech recognizer and is based on a standard portable PC. In this approach only a speech recognizer is used. The relation between the pairs of sentences in English and the other language is realized through simple mapping. A further development of this system is the *Phraselator* (see [Phraselator, 2002]). This commercial available system is used by the American troops in Afghanistan. In this system no natural language processing technique, i.e. no analysis of the meaning is used. This is different in the *CommandTalk* system.

CommandTalk is a spoken-language interface to the ModSAF battlefield simulator (see [CommandTalk], [Moore *et al*., 1997], [Stent *et al*., 1999]). SRI International developed the system. The simulation user can use ordinary spoken English to create forces (e.g. "Create an M1 platoon designated Charlie 4 5."), assign missions to forces, change missions during execution, and control all the functionality of the simulator program (e.g. "Center on M1 platoon"). The principal design goal of the CommandTalk system was to let the commanders interact by voice with simulated forces as if they would command actual forces. The development of the simulator is based on various components developed at SRI. As the speech recognizer component the Nuance system (see [Nuance]) was used. The natural language parsing and semantic interpretation is done by the Gemini subsystem. For the CommandTalk system an application-specific grammar and wordset was developed. Beside the information from the utterance the system uses linguistic context, situational context, and defaults to produce a complete interpretation of the utterance. The system is also able to handle dialogues and combinations of language and mouse input.

3. **The Project NATLAC**

The overall objective of our research project NATLAC (NATural Language ACcess) is to evaluate whether various techniques from computational linguistics can be used to realize a spoken access to C2 databases. For this evaluation we have to realize a prototype. The ATCCIS database (see [NATO, 2000]) delivers the domain model. As a scenario for our prototype system the planning of a multinational operation is used. In this scenario an action, action tasks, units, objectives, geographical points, reporting data, and contexts are used to model the situation. In the first step, we are realizing the natural language front-end that will be able to answer simple spoken questions concerning this situation, e.g., "Gehört das 9. Deutsche Batallion zu den verfügbaren Einheiten?" ("Does the 9th GE battalion belong to the available units?"). The scenario is elaborated enough, so that more complicated language and domain problems can be modeled too in the future, e.g., complex questions or dialogues. The long-term objective of the project NATLAC is the construction of a dialogue system for a subset of spoken German referring to the scenario in the ATCCIS database.

3.1 ***The Architecture***

The architecture is shown in Fig. 1. For converting the acoustic input into a stream of words we use a COTS product (the *speech recognizer*). We have experimented with the Nuance (see [Nuance], [Hecking, 2001]) system. The speech recognizer uses a *lexicon* and a *grammar* of the subset of the German language. In addition, the recognizer needs an acoustic model. In this model each written word is associated with its pronunciation given as a sequence of phonemes. This acoustic model can be extended to those words not included in the scope of supply.

The recognized words are then analyzed through linguistic means. First the input is *syntactically analyzed* with the help of a parser, a lexicon, and a grammar. The grammar and the lexicon determine the ability of the natural language component. The result of the syntax analysis is a *feature structure* and a first formal representation of the meaning. This feature structure contains all available syntax information that can be determined only from the input sentence. No other sources are used.

Then, the formal representation of the meaning is *semantically analyzed* to build a final semantic representation of the stream of words. To realize this the semantic analysis component uses the *context* of the dialogue to resolve ambiguities e.g. to decide which platoon is meant in the sentence “The platoon attacks hill 234”. The *ontology* component and the *semantics* component provide semantic information e.g. of how concepts of the domain are interrelated (e.g. a tank is an armored vehicle).

Finally, the semantic representation is used as the foundation for the access to the ATCCIS database. The *DB access component* constructs from the semantic representation an SQL statement that is dispatched to the underlying database system. The database delivers the result, and it is presented to the user as spoken output.

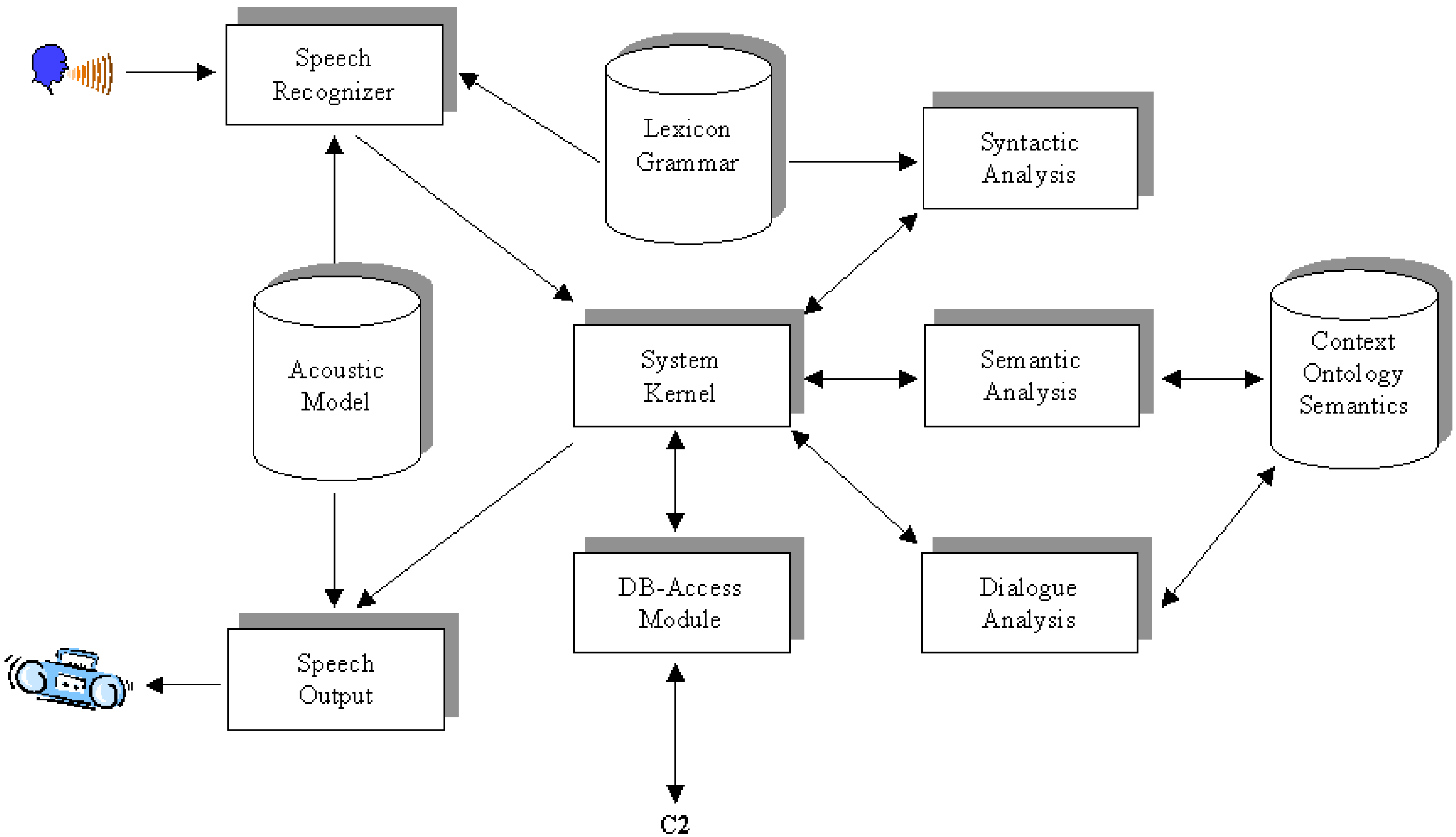

Fig. 1 The Architecture of NATLAC

To give NATLAC dialogue capabilities a *dialogue analysis* component is necessary. This component is capable of identifying individual dialogues and dialogue steps. This is the foundation for the capability of the system to become active by itself and to control the flow of the dialogue (e.g. to insert a clarification dialogue).

All components of the system are coordinated through the system kernel. Most of the components have been or will be realized through Prolog (see [Prolog, 2000]). Java is also used.

So far, we have

- tested the speech recognizer,
- implemented parts of the kernel system,
- chosen the grammar formalism *Head-Driven Phrase Structure Grammar*,
- realized a first version of the syntactic analysis with the tool *The Attribute Logic Engine*,
- chosen the logic-based representation languages *Quasi-Logical Form* and *Target Reasoning Language* (TRL), and
- realized the transformation process from TRL formulae to SQL statements

The implemented parts of NATLAC are not integrated yet.

3.2 ***Syntactic Analysis***

During the syntactic analysis the structure of a sentence must be determined. To do this, a grammar with a lexicon and a parser is necessary. There are a lot of different grammar formalisms for natural language processing. In NATLAC the formalism *Head-Driven Phrase Structure Grammar* (HPSG, see [Pollard and Sag, 1994]) is used. This formalism allows the integrated processing of syntactical and semantical information and is widely used in research. Grammar fragments for German (e.g. [Meurers, 1994]) and the free tool *The Attribute Logic Engine* (ALE) (see [Carpenter and Penn, 1999]) are available.

In contrast to context-free rules HPSG has the following properties:

- lexicalisation, i.e. structural information is not represented in rules but in the lexical entries,
- it uses feature structures as complex symbols,
- it works head-oriented, i.e. one component of the right-hand side of a rule is the important one (the head).

$$\text{HEAD}\ {}_{\textit{head}}\!\begin{bmatrix} \text{AGR}\ {}_{\textit{agr}}\!\begin{bmatrix} \text{PERSON person} \\ \text{NUMBER number} \\ \text{GENDER gender} \end{bmatrix} \end{bmatrix}$$

Fig. 2 Feature Structure

In HPSG each linguistic object (e.g. words, phrases, and sentences) is represented through *feature structures* (see Fig. 2). Each feature has a name (e.g. HEAD) and a value (e.g. the expression with the square brackets after HEAD). Each value has a type (e.g. *head*) as well. A feature value can be a complex feature structure (e.g. the value of AGR consists of the three features PERSON, NUMBER and GENDER). Feature values can be accessed through paths (e.g. HEAD|AGR| PERSON gives the 'person' value).

The main work of designing a HPSG grammar is to develop complex feature structures and represent them in the *type hierarchy*. An example of a type hierarchy is shown in Fig. 4. As already mentioned, the important part of the HPSG grammar is the *lexicon*. Each word has a complex feature structure representing the important syntax (and semantic) information. In Fig. 3 a part of an example entry for the word 'Panzer' (tank) is shown. Beside the lexicon, there are various *lexical rules* to produce further lexicon entries, e.g. the plural forms of nouns.

```
WORD: panzer
ENTRY:
word
LR e_list
QSTORE eset
RETR e_list
SYNSEM synsem
       LOC loc
           CAT cat
               HEAD noun
                    CASE dat
                    DECL decl
                    MOD nothing
                    PRD minus
               MARKING unmarked
               NPCOMP minus
               VAL val
                   COMPS e_list
                   SPR ne_sign_list
                       HD sign ...
```

Fig. 3 Lexicon Entry

Schemata and *principles* are used in HPSG to produce complex feature structures for the sentence or parts of the sentence. This analysis starts with the feature structures for every recognized word and complex feature structures are constructed during the parsing of the whole sentence.

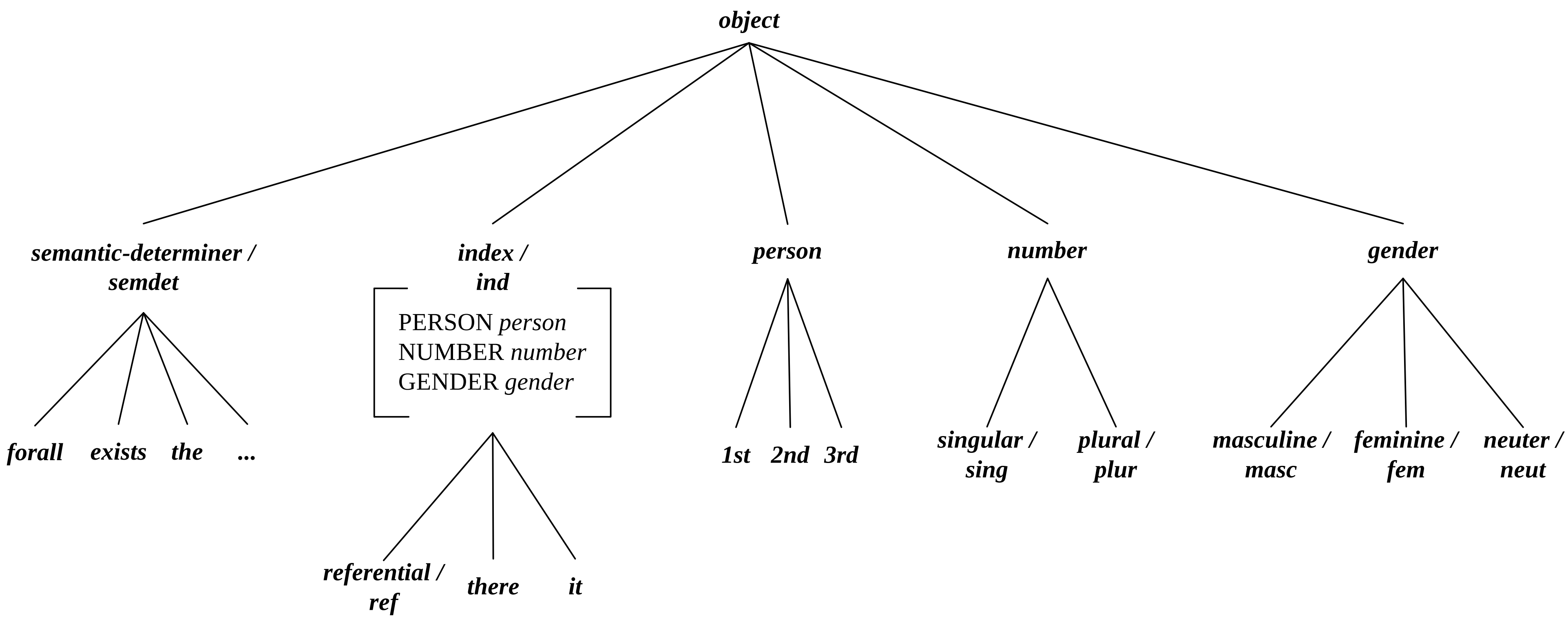

Fig. 4 Type Hierarchy

To realize the grammar for NATLAC, we started with the grammar fragment of W. D. Meurers (see [Meurers, 1994]). We made various modifications to suit the grammar to our scenario. The lexicon is still very small. The implementation uses the free *The Attribute Logic Engine* (ALE, see [Carpenter and Penn, 1999]) tool. ALE is a development system for unification-based parsing systems and was developed with an HPSG-orientation. ALE compiles into SICStus Prolog (see [Prolog, 2000]) and a chart parser is a part of it. Fig. 5 shows the result of an example run of the parser for the sentence "Der Panzer muss fahren." (The tank must drive.).

3.3 ***From Logical Formulae to SQL Statements***

```
| ?- rect([der, panzer, muss, fahren]).
0 der 1 panzer 2 muss 3 fahren 4
O.k.    ** CPU time used (sec): 0.0,  Solutions:  1
***************************************
```

Fig. 5 Result of ALE

As the result of the semantic analysis phase a representation of the sentence meaning in the predicate logic based *TRL* (Target Reasoning Language) is delivered. TRL and *QLF* (Quasi Logical Form) were developed in the *Core Language Engine* (see [Alshawi *et al.*, 1992a]) and the *CLARE* project (see [Alshawi *et al.*, 1992b]). The TRL formula is passed to the DB-Access Module. In this module the *Abductive Equivalential Translation* (AET) is realized (see [Rayner, 1993]). During this translation the meaning formula is logically sound transformed into an SQL statement. For this transformation the mapping between linguistic predicates (e.g. l_unit1(X)) and database relations must be specified. This is done through equivalences in the *Linguistic Domain Theory* (LDT). Fig. 6 shows an example. The linguistic predicate "l_unit1(X)" can be replaced by the conceptual predicate "c_unit1(X)" under the assumption that "unit_is_military_unit1(X)" is true.

```
(l_unit1(X) <=> c_unit1(X)) <= unit_is_military_unit1(X).
```

Fig. 6 TRL Formula

The translation process comprises different phases:

1. Transformation of linguistic to conceptual predicates.
2. Transformation of conceptual to database predicates.
3. Construction of the SQL string from the database predicates.

During the translation logical simplification can be applied. During this simplification an access to the database might also be necessary to prove assumptions.

Fig. 7 and 8 show an example of the input and output of the translation process. In Fig. 7 the TRL representation of the question "Hat die Einheit den Spitznamen 'The Big Red One'?" (Does the unit have the nickname ‘The Big Red One’?) is represented. After applying the AET process the SQL query in Fig. 8 was produced.

```
and(l_unit1(77001), l_nickname1(77001,The Big Red One))
```

Fig. 7 TRL Input Formula

```
SELECT
        oi_1.name
  FROM
        obj_item oi_1,
        org o_2
  WHERE
        oi_1.obj_item_id='77001' AND
        oi_1.cat_code='OR' AND
        o_2.org_id='77001' AND
        o_2.cat_code='UN' AND
        o_2.nickname_name='The Big Red One'
```

Fig. 8 Produced SQL Statement

The transformation component is described in detail in [Casals Elvira, 2002].

4. Conclusion

In this paper we have described the level of development of our project NATLAC. We have presented how the syntactic analysis is done and how the logic-based meaning representation is transformed into SQL statements for querying the ATCCIS database. To realize the first prototype we have to do next:

- The used German grammar must be adapted to missing syntactical structures.
- The lexicon must be extended.
- The representation of the words must be integrated into the lexicon and the grammar must be extended to build up the meaning representation of the whole sentence.
- The syntax analysis must be connected with the AET process.
- All the components must be connected with the kernel system.

In the first prototype the semantic analysis will be only a minimal one. No dialogue capabilities, and only written input and output will be available.

5. References

[Alshawi *et al.*, 1992a] Alshawi, H. et al. *The Core Language Engine*. The MIT Press, Cambridge, Massachusetts, 1992.

[Alshawi *et al.*, 1992b] Alshawi, H. et al. *CLARE: A Contextual Reasoning and Cooperative Response Framework for the Core Language Engine*. SRI International, Cambridge, 1992.

[Carpenter and Penn, 1999] Carpenter, B. & Penn, G. *ALE The Attribute Logic Engine – User's Guide*. Version 3.2. Beta, May 1999.

[Casals Elvira, 2002] Casals Elvira, X. *Mapping of Logical Formulae to SQL-Statements for a Natural Language Access Component*. Dept. Llenguatges i Sistemes Informàtics (LSI) - Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, 2002.

[CommandTalk] *CommandTalk*. SRI International, http://www.ai.sri.com/natural-language/projects/arpa-sls/ commandtalk.html.

[Hecking, 2001] Hecking, M. *Natural Language Access for C2 Systems*. Paper presented at the RTO IST Symposium on „Information Management Challenges in Achieving Coalition Interoperability", held in Quebec, Canada, 28-30 May 2001, and published in RTO MP-064.

[Hinrichs *et al*., 1994] Hinrichs, E. W. et al. *Partial-VP and Split-NP Topicalization in German – An HPSG Analysis and its Implementation*. Arbeitspapier des SFB 340 Nr. 58, Universität Tübingen, 1994.

[Hunt, 2000] M. Hunt et al. *A Military Operational Automatic Interpreting System*. In: [RTO-MP-28, 2000], pp.87-90.

[Meurers, 1994] Meurers, W. D. *On Implementing an HPSG Theory – Aspects of the Logical Architecture, the Formalization, and the Implementation of Head-Driven Phrase Structure Grammars*. In: [Hinrichs, 1994].

[Moore *et al.*, 1997] Moore, R. et al. *CommandTalk: A Spoken-Language Interface for Battlefield Simulations*. In: Proc. of the 5th Conf. on Applied Natural Language Processing, Washington, DC, pp. 1-7, ACL, 1997.

[NATO, 2000] NATO. *The Land C2 Information Exchange Data Model*. AdatP-32 Edition 2.0, 31 March 2000.

[Nuance] http://www.nuance.com

[Phraselator, 2002] http://www.phraselator.com, 10th June 2002.

[Pollard and Sag, 1994] Pollard, C. & Sag, I. A. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press, Chicago, London, 1994.

[Prolog, 2000] Intelligent Systems Laboratory. *SICStus Prolog User's Manual*. Swedish Institute of Computer Science, Kista, Sweden, 2000, http://www.sics.se/sicstus.

[Rayner, 1993] Rayner, M. *Abductive Equivalential Translation and its application to Natural Language Database Interfacing*. Ph.D. thesis, Royal Institute of Technology, Stockholm, 1993.

[RTO-MP-28, 2000] NATO IST Panel. *Multi-Lingual Interoperability in Speech Technology*. Cedex: RTO/NATO, RTO Meeting Proceedings 28, RTO-MP-28, AC/323(IST)TP/4, 2000.

[Steeneken, 1996] Steeneken, H. J. M. *Potentials of Speech and Language Technology Systems for Military Use: an Application and Technology Oriented Survey*. NATO, Technical Report, AC/243(Panel 3)TP/21, 1996.

[Stent *et al*., 1999] Stent, A. et al. *The CommandTalk Spoken Dialogue System*. In: Proc. of the 37th Annual Meeting of the ACL, pp. 183-190, University of Maryland, College Park, MD, ACL, 1999.