8th International Command and Control Research and Technology Symposium,
June 17-19, 2003
National Defense University
Washington, DC

Topic: C2 Experimentation

Paper's Title: Information Extraction from Battlefield Reports

Author
Name: Dr. Matthias Hecking
Organization: FGAN/FKIE
Address: Neuenahrer Straße 20
53343 Wachtberg-Werthhoven
Germany
Phon: +49 228 9435 576
Fax: +49 228 9435 685
E-Mail: hecking@fgan.de

Information Extraction from Battlefield Reports

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 SOKRATES we use information extraction technology for written language to analyze German free-form battlefield reports. These reports are processed by transducers. The extraction result is formalized in feature structures, semantically enriched by the semantic analysis and the augmented result is then stored in the ATCCIS database. After storing in the database, triggers initiate a change in the position of a tactical symbol on the tactical map. In this paper, we first introduce to the promising field of information extraction. Then, we describe in detail our project SOKRATES.

Introduction

In the NATO technical report *Potentials of Speech and Language Technology Systems for Military Use: an Application and Technology Oriented Survey* (see [Steeneken, 1996]) the *processing of human language* was recognized as a critical capability in many future military applications, among other things in 'command and control'. Though, still a topic of research, there already exist natural language components in military systems, e.g. CommandTalk, a spoken-language interface to the ModSAF battlefield simulator (see [Moore *et al*., 1997], [Stent *et al*., 1999], [CommandTalk]) or the Phraselator (see [Phraselator, 2003]) used by the US Army in Afghanistan.

Today, the usability of human language technology 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 language technology.

A natural way to input information in a C2 system would be to use written or spoken natural language. In this context, we tried to show in a former approach that the available methods, techniques, and tools of computational linguistics are mature enough to test their applicability to C2 systems. In this former approach we used the ATCCIS database (cf. [NATO, 2000]) as the domain. We have already reported on the progress of our former project using a speech

recognizer (cf. [Hecking, 2001]). Furthermore, we showed (cf. [Hecking, 2002]) how to use *deep syntactic analysis* techniques to analyze simple written sentences. This approach however has various deficiencies. Especially, the high demand on processing time and the expectation, that the sentences are all well structured with respect to the grammar, hampers the application of these techniques. Therefore, we were looking for an alternative approach that avoids these deficiencies.

Information extraction (IE) is an engineering approach based on results of computational linguistics to build systems that process huge amount of texts of a specific sort. IE is an approach that avoids the deficiencies mentioned above. Each IE system is tailored to a specific domain and task. IE uses a *shallow syntactic approach*, i.e. that only parts of the sentences (so-called 'chunks') are processed with finite state automaton or transducers. These parts contain the relevant information about the Who, What, When, etc. To realize an IE system the desired output must be specified. This is done through *templates*. Templates represent feature-value structures. During the IE process a domain-specific lexicon and domain-specific rules are used to instantiate the templates.

In this paper, we will first give a short introduction into the promising field of IE. Then, we show how we use the SMES system in our project SOKRATES to realize an IE system for battlefield reports in German. We will describe the various steps during the IE process and we will explain in detail what transducers are used and how they are used.

Information Extraction

Information extraction (IE) is the task of identifying, collecting and normalizing information from natural language text (see [Appelt, 1999], [Pazienza, 1999]). Relevant information about the Who, What, When, etc. is looked for. The information of interest is described through patterns called *templates*. During the IE task these templates are filled with the collected information. IE therefore can be seen as the process of normalizing from free-form text into a defined structure. The templates are domain and task specific, i.e. for each new task and domain they must be newly created.

To realize an IE system, language resources (lexicon, grammar) and appropriated parsing software are necessary. This software must be language-specific. The IE tools for the English language are not appropriated for analyzing e.g. German text due to the free-order of the language.

In order to achieve robust and efficient IE systems, domain knowledge must be integrated and shallow algorithms must be used. The domain knowledge is tightly integrated with the language knowledge, e.g. the name 'Leopard' in the lexicon has the categorical information 'tank'. This association between words and semantic information is domain-specific and has to be change for other applications.

The current IE technology are used successfully in various application areas, e.g. intelligent information retrieval, linguistically based data mining, automatic term extraction, text classification.

The IE process itself is divided into sub steps. After tokenizing the text, the sentence boundaries must be identified. Then, the morphological component identifies the word stems, the abbreviation and detects the syntactic information (e.g. grammar case and gender). After this, the chunk parsing with transducers selects parts of the natural language text, which are relevant for the specific information extraction task. The chunks are then used to instantiate the templates, which represent the result of the IE process.

Various toolboxes are available to build IE systems. These toolboxes must be language specific. A powerful IE toolbox for German is the SMES system (cf. [Neumann, 2003]). This toolbox offers among other things a morphological analysis component with a huge lexicon, predefined grammars (transducers) for specific phrases (e.g. noun phrases) and the possibility to program arbitrary transducers.

The Project SOKRATES[1]

The overall objective of the SOKRATES project is to analyze written German battlefield reports. The result of the analysis is stored in the ATCCIS database (see [NATO, 2000]). These stored results can be used for different purposes. One purpose is that location changes of units initiate automatically changes of tactical symbols on the tactical map.

The Architecture

The architecture is shown in Fig. 1. The free-form reports are handed over to the *coordination module*, which is responsible for all the coordination in the system. In a first step, the *syntactic preprocessing* identifies the sentence boundaries. Next, the *information extraction* module uses the lexicon and the grammar transducers to identify and select the relevant parts in the natural language text. These parts are represented as typed feature structures that are coded as an XML document. The result of the information extraction is used by the *semantic analysis* component to deduce more information out of the extracted information with the help of an ontology and the context (see [Schade, 2003a], [Schade, 2003b]). After the semantic analysis the result is pushed into the ATCCIS database and then it is used to alter automatically the position of tactical symbols on the map.

The Information Extraction Module

During the information extraction the structure of a text 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. To be able to process large amounts of text an efficient approach must be used. Therefore, we use a *shallow syntactic approach*, i.e. that only parts of the sentences (so-called ‘chunks’) are processed. These transducers code the necessary grammar.

[1] The members of the SOKRATES project are: X. Casals Elvira, M. Frey, M. Hecking, U. Schade.

The result of the syntactic analysis is represented in templates. The necessary templates are formalized in SOKRATES by *typed feature structures* (see [Pollard and Sag, 1994]). These structures consist of name-value-pairs. In the simplest case, the feature values can be strings, numbers or other atomic values. But the values can also be whole feature structures. In the following example

```
(:TIME
     (:SECOND . "???")
     (:MINUTE . 35)
     (:HOUR . 10)
     (:DAY . 9)
     (:MONTH . "september")
     (:YEAR . "???")
     (:TIMEZONE . "???")
     (:TYPE . :TIME)
)
```

the feature with name `:TIME` of type 'time' (`(:TYPE . :TIME)`) has a feature structure (`(:SECOND . "???")` … `(:TIMEZONE . "???")`) as its value. In contrast, the feature with name `:HOUR` has an atomic value (`10`). Unknown values are represented by `"???"`. Feature values can be accessed through paths (e.g. `:TIME|:MINUTE` gives the '35' value).

The feature structures used form an inheritance hierarchy (see Fig. 2). This hierarchy describes completely all possible structures that the IE module can use and might instantiate. E.g. in Fig. 2 the type 'template' has two subtypes 'report' and 'order'. The report-type feature structure has various name-value-pairs, e.g. ':addressee *partner*'. The value *partner* is itself a feature structure.

The information extraction process by itself is realized with shallow algorithms. These are called *transducers*. Transducers are finite-state automaton that read from an input stream and write to an output stream. Theses automaton can be cascaded, i.e. that the output of one transducer is the input in another one. For example, the names of various locations are the input to a transducer, which constructs a feature structure that formalizes the recognized name of the location. This feature structure is than handed as an input to the transducer responsible for detecting e.g. goal expressions. During the syntactic analysis of the free-form reports with transducers more and more complex feature structures are constructed.

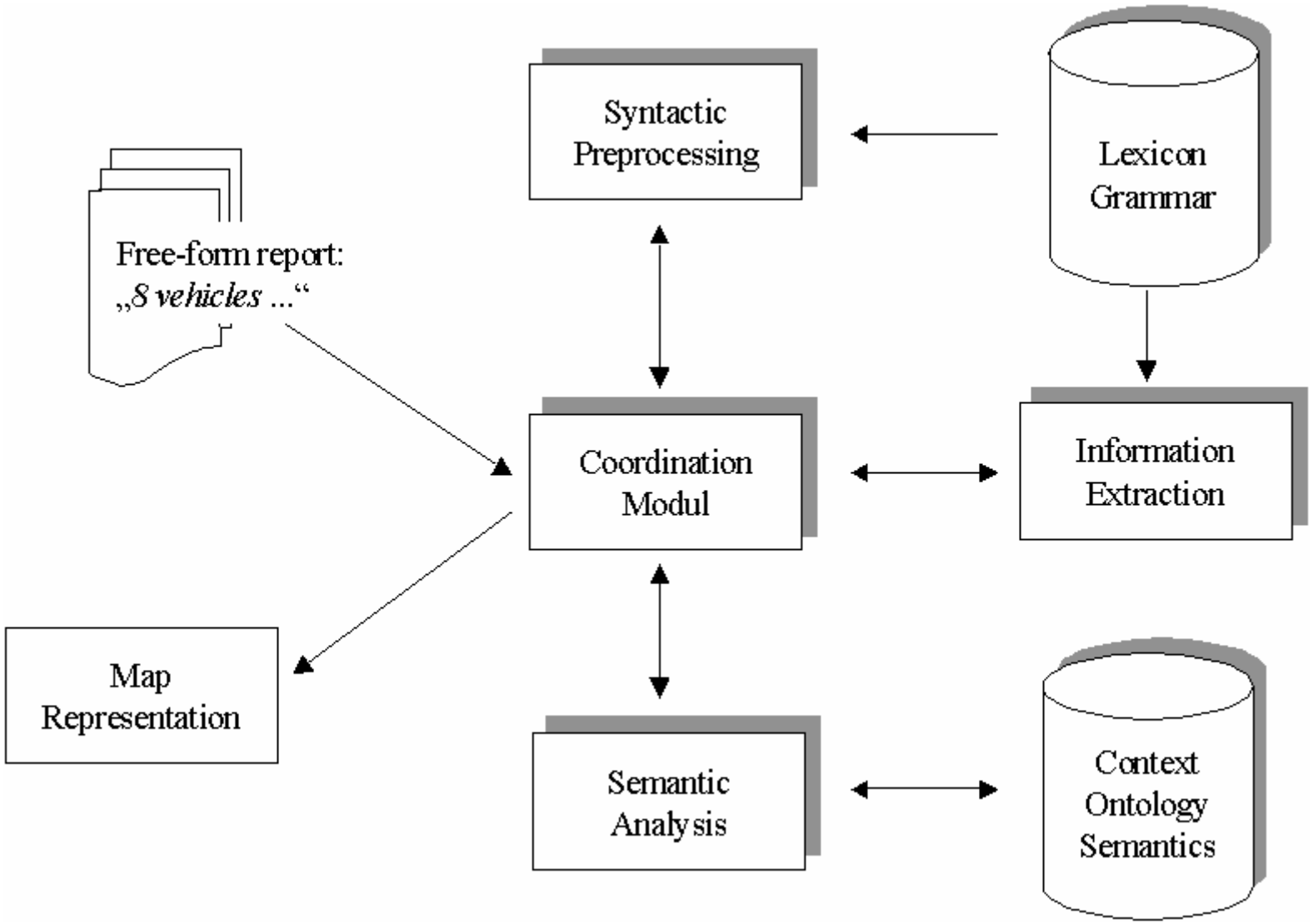

Fig. 1 The Architecture of SOKRATES

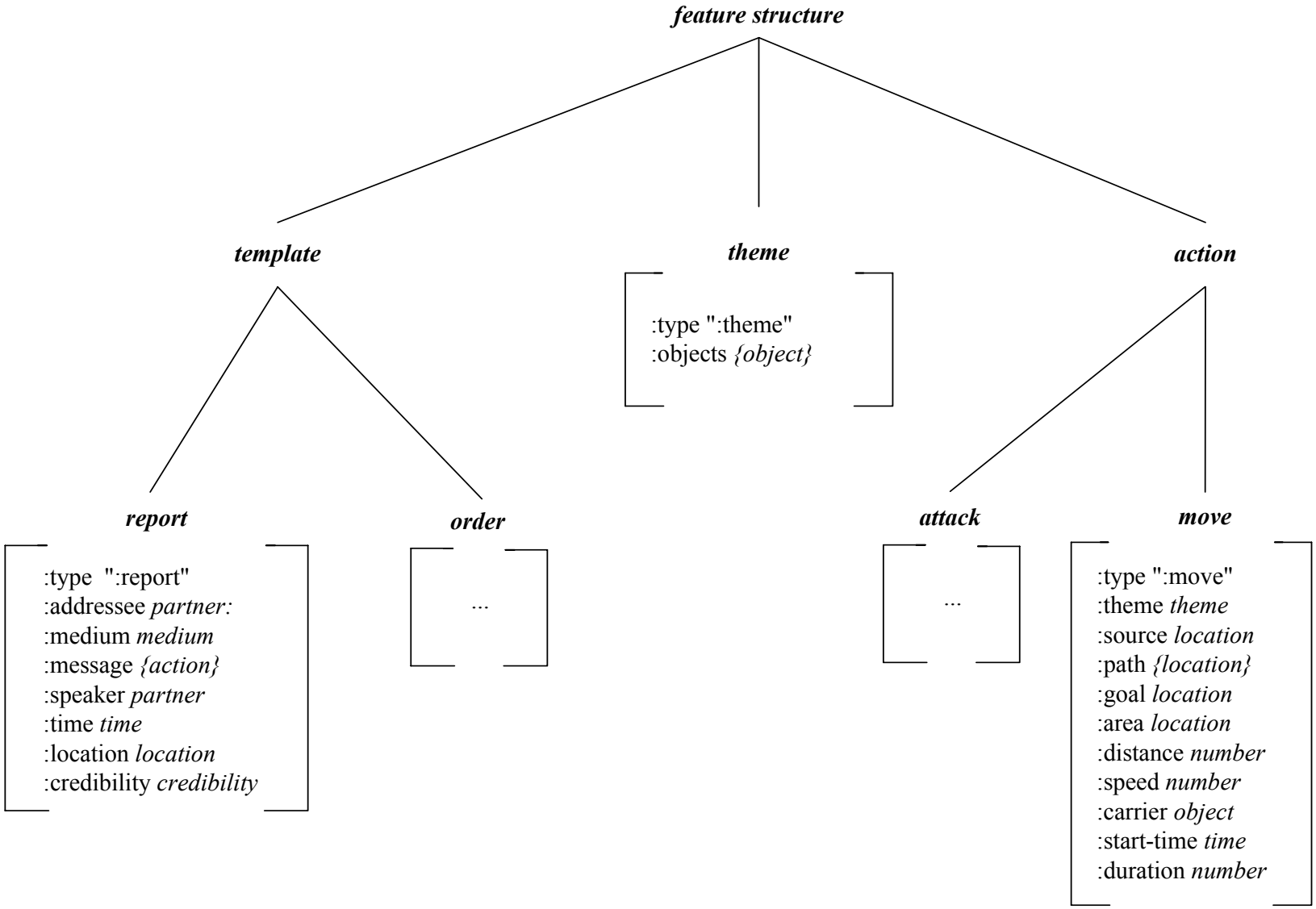

Fig. 2 A Part of the Feature Structure Hierarchy

In SOKRATES we use as an IE tool the SMES system (see [Neumann, 2003]). This system contains a huge German lexicon and offers the possibility to create transducers. SMES is implemented in Allegro Common Lisp (see [ACL, 2002]) and therefore offers also the whole functionality of Common Lisp as well. The following example shows a simple transducer for detecting who reports, when and where:

```
(compile-regexp
'(:conc
     (:current-pos start)
     (:seek so-date-time date)
     (:seek so-meldender meldender)
     (:seek so-standort-meldender standort)
     (:morphix-punctuation ":")
     (:current-pos end)
)
:debug *debug*
:register-types '((:register start date meldender standort
end))
:output-desc
'(:lisp (multi-acons :speaker meldender :time date :location
standort))
:prefix T
:suffix nil
:name 'so-meldung-prolog
:compile *compile*
:write-to-file *trace-file*
)
```

The name of this transducer is 'so-meldung-prolog' (`:name 'so-meldung-prolog`). It consists of a concatenation (`:conc`) of calls to other transducers (e.g. `(:seek so-date-time date)`) and a call to the morphological component (`(:morphix-punctuation ":")`). If the called transducers recognize successfully the appropriate parts in the report, they construct a feature structure describing their recognition result. This result is than handed over to the calling transducer, e.g. in variable `date` of the calling statement `(:seek so-date-time date)`. The shown transducer uses the recognition result to construct its own feature structure (`:output-desc '(:lisp (multi-acons :speaker meldender :time date :location standort))`) which is than passed up to the calling transducer of the shown transducer.

In the current implementation the SOKRATES system is able to process simple reports about moving objects. One example is the following report: "*09. September 10.35 Uhr von 6./PzMrs332-Zug B in Vinstedt: 18 Fahrzeuge marschieren bei Straßenkreuzung Kr3 nördlich Eppensen nach Ebstorf.*" (September 9th, 10.35 a.m. from 6./PzMrs332-Zug B in Vinstedt: 18 vehicles march at road crossing Kr3 north of Eppensen to Ebstorf.). After processing this report the result is represented as the following typed feature structure:

```
(
```

```
(:CREDIBILITY . "???")
(:LOCATION
     (:NAME . "vinstedt")
     (:TYPE . :LOCATION)
)
(:TIME
     (:SECOND . "???")
     (:MINUTE . 35)
     (:HOUR . 10)
     (:DAY . 9)
     (:MONTH . "september")
     (:YEAR . "???")
     (:TIMEZONE . "???")
     (:TYPE . :TIME)
)
(:SPEAKER
     (:NAME . "6/pzmrs/332/zug/b")
     (:TYPE . :UNIT)
)
(:MESSAGE
     (:SET
          (
               (:DURATION . "???")
               (:START-TIME . "???")
               (:CARRIER . "???")
               (:SPEED . "???")
               (:DISTANCE . "???")
               (:AREA . "???")
               (:GOAL
                    (:QUALIFIER . :TO)
                    (:NAME . "ebstorf")
                    (:TYPE . :LOCATION)
               )
               (:PATH . "???")
               (:SOURCE
                    (:QUALIFIERS
                         (:SET
                              (
                                   (:QUALIFIER . "nördlich")
                                   (:NAME . "eppensen")
                                   (:TYPE . :LOCATION)
                              )
                         )
                    )
                    (:COORDINATES . "kr3")
                    (:QUALIFIER . :EXACTLY-AT)
                    (:NAME . "Straßenkreuzung")
                    (:TYPE . :LOCATION)
               )
               (:THEME
                    (:OBJECTS
                         (:SET
                              (
                                   (:COUNT . 18)
```

```
                                        (:NAME . "Fahrzeuge")
                                        (:TYPE . :VEHICLE)
                                    )
                                )
                            )
                            (:TYPE . :THEME)
                        )
                        (:TYPE . :MOVE)
                    )
                )
            )
            (:MEDIUM . :LETTER)
            (:ADDRESSEE . "???")
            (:TYPE . :REPORT)
)
```

The above feature structure is of type 'report'. Each report might contain information about the addressee of the report (:ADDRESSEE), the medium in which it was formulated (in this case :LETTER), the message itself (:MESSAGE), the unit or person who sends the report (:SPEAKER), the time of reporting (:TIME), the location of the unit or person who reports (:LOCATION) and the credibility of the unit or person (:CREDIBILITY). If the IE module doesn't find the appropriate information the string "???" is delivered. The message contains a set (:SET) of action descriptions. In the example it is a move-action (:MOVE). Words like "marschieren", "fahren", "schwimmen", "fliegen" (to march, to drive, to swim, to fly) are mapped to the move-action. The description of the move-action contains various features: the duration (:DURATION), the start-time (:START-TIME), the carrier (:CARRIER), the speed (:SPEED), the distance (:DISTANCE), the area (:AREA), the goal (:GOAL), the path (:PATH), the source (:SOURCE) and the theme (:THEME) of the action. The goal of the move is described with the help of a feature structure of type :LOCATION. It formalizes in our example the city Ebstorf as the goal of the march. The starting point of the move-action is given after the feature name :SOURCE. This is also a feature structure of type :LOCATION. It gives the coordinates ((:COORDINATES . "kr3")) of the crossing ((:NAME . "Straßenkreuzung")) and it gives also a qualifying statement that the crossing is in the north ((:QUALIFIER . "nördlich")) of the city Eppensen ((:NAME . "eppensen")). The theme describes which objects (:OBJECTS) are moving. In our example 18 vehicles ((:COUNT . 18)(:NAME . "Fahrzeuge")(:TYPE . :VEHICLE)) are moving.

The SOKRATES system is implemented and is able to process simple examples as shown above. Up to now, the lexicon was only extended with a few military specific words (but it contains already more then 120,000 German word stems). The next steps will be the extension of each module to enhance the processing capabilities.

Conclusion

In this paper, we introduced in the promising field of information extraction and we gave a description of our project SOKRATES. After presenting the overall architecture of our system, we have shown how the syntactic analysis is done with transducers and how feature structures

are used to describe and to store potential analysis results. We described how a simple German battlefield report was analyzed and how the analysis result was represented in a feature structure.

References

[ACL, 2002] Allegro Common Lisp. Franz Inc., 2002, http://www.franz.com.

[Appelt, 1999] Appelt, D. & Israel, D. *Introduction to Information Extraction Technology*. Stockholm: IJCAI-99 Tutorial, 1999, http://www.ai.sri.com/~appelt/ie-tutorial/.

[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.

[Hecking, 2002] Hecking, M. *Analysis of Spoken Input to C2 Systems*. In: Proceedings of the 7th International C2 Research and Technology Symposium (ICCRTS), Québec City, Kanada, 2002.

[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.

[Neumann, 2003] Neumann, G. http://www.dfki.de/~neumann/pd-smes/pd-smes.html, 2003.

[Pazienza, 1999] Pazienza, M. T. (ed.) *Information Extraction*. Berlin, 1999.

[Phraselator, 2003] *The Phraselator*. http://www.phraselator.com/products.htm, 2003.

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

[Schade, 2003a] Schade, U. *Ontologieentwicklung für Heeresanwendungen*. Forschungsgesellschaft für Angewandte Naturwissenschaften e.V. (FGAN), FKIE-Bericht, 2003.

[Schade, 2003b] Schade, U. *Towards an Ontology for Army Battle C2 Systems*. In: Proceeding of the 8th ICCRTS, 2003.

[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.