

Knowledge Representation and Semantics in a Complex Domain: The UNIX Natural Language Help System GOETHE

Gerhard Heyer, Ralf Kese, Frank Oemig, Friedrich Dudda
OLIVETTI D.O.R. - TA RESEARCH LAB
TA TRIUMPH-ADLER AG
Fürtherstr. 212
8500 Nuremberg (West Germany)

1 Abstract

Natural language help systems for complex domains require, in our view, an integration of semantic representation and knowledge base in order to adequately and efficiently deal with cognitively misconceived user input. We present such an integration by way of the notion of a *frame-semantics* that has been implemented for the purposes of a natural language help system for UNIX.

2 Introduction

It is commonly agreed that natural language systems for semantically rich domains require a level of *semantic* representation in order to provide for a sufficiently deep processing of the natural language input. The level of semantic representation is sometimes called a representation of linguistic knowledge. In addition, a natural language help system also requires a *knowledge base* of the application domain in order to answer the requests for domain specific help. The level of knowledge representation is sometimes called a representation of world knowledge. Most present day natural language processing systems, including, amongst others, SRI's Core Language Engine (Alshawi et.al. 1986), ESPRIT I project ACORD (Bez et.al. 1990), and the UNIX natural language help systems UNIX-Consultant (Wilensky et.al. 1988), SINIX-Consultant (Wahlster et.al. 1988), and AQUA (Quilici et.al. 1986), keep the two levels of representation distinct. In addition, there usually is no feed-back of information between the semantic representation and the knowledge base. Thus, parsing is supposed to result in a *complete*

semantic representation of the user input which then is passed on to the knowledge base manager for further processing. This kind of architecture follows a strategy that can be called *stepwise* processing. We claim that for complex domains this kind of approach is inadequate because it ignores the user's cognitive misconceptions about the particular application. Instead, we wish to argue that at least with respect to semantics and knowledge representation in natural language help systems an *integrated* approach should be preferred. In the approach we advocate, semantics and knowledge representation interact to correct (or complete) a possibly incorrect (or incomplete) semantic representation. The mechanism by which this is achieved is based on the notion of a *frame-semantics* (cf. Heyer et.al. 1988, Hausser 1989). We demonstrate our integrated approach with examples from GOETHE, a natural language help system for UNIX as a complex domain.

GOETHE (cf. Kese/ Oemig 1989) has been developed together with OLIVETTI AI Center, Ivrea, and Tecciel AI Lab, Rome, for UNIX V on the OLIVETTI LSX 30xxComputer Series. The present prototype includes a protocol for monitoring user's actions and has the natural language mode of interaction fully integrated into a graphical DeskTop under InterViews (based on a cooperation with Fraunhofer Society, Stuttgart), thus allowing also for deictic natural language and graphical interactions. It covers all of UNIX' file handling, containing a static knowledge base of more than 70 UNIX programs. It is written in Quintus-PROLOG and C, and takes by average less than 10 seconds for generating an answer to a user's request.

3 Requirements on Knowledge Representation and Semantics for a UNIX natural language help system

It is the task of the *knowledge base* in a UNIX natural language help system to serve as a basis for correctly and adequately answering a user's questions in one of the following situations: (1) the user needs to know a UNIX command, or series of commands, in order to carry out a certain task, (2) he has sent off a UNIX command and the system has resulted in a different state than he expected, or (3) he wants to get information about a UNIX command. In general, this will require two knowledge sources: *static* knowledge about UNIX as a collection of possible man-machine interactions, and *dynamic* knowledge about the respective UNIX state (in particular, i-nodes and the associated files with their permissions), the user's actions, and the system's reactions (in particular, error messages). It is the task of the *semantic* representation to provide the knowledge base manager with a correct and adequate semantic representation of the user's input (in a specific situation). There basically are two strategies available at this point. On the one hand, adhering to the idea that co-occurrence restrictions (also for missing elements) need to be treated as real restrictions, possibly even as syntactic restrictions (Chomsky 1965), we can insist that if there is a semantic representation of an input sentence at all, it will be *correct* and *complete* (with respect to the domain of application). Hence, the system will fail to provide an answer to a user's request unless he phrases his question in a correct and complete way. On the other hand, co-occurrence restrictions may not be taken as real restrictions but rather as *semantic defaults* which may be overwritten by additional knowledge base information. This allows for a much more user-friendly and cooperative natural language processing, but requires that the semantic representation is closely tied to the knowledge base.

For the purposes of the GOETHE system, we have opted for the second alternative, because the cognitive misconceptions a user may have about UNIX not only cause him to invoke the help system, but also cause him in most cases to *phrase* his questions in the way he does: If the system is presented with a semantically incorrect question, this is to be taken as an indication that the user needs help, and a reminder that he better rephrase his question in a correct way will not be of much use to him. Of course, it would have also been possible to relax the syntactic co-occurrence restrictions. In effect, however, this would have resulted in a duplication of knowledge base information in the lexicon. The second alternative, therefore, not only appears to be the more adequate, but also the more efficient solution.

4 Frame Semantics

Output of the parser in GOETHE is a possibly incorrect, or incomplete, semantic representation where the meaning of the individual- and predicate-constants of the logical representation are represented as frame-theoretic *icons* (Heyer et.al. 1988). We call this kind of semantic representation frame-semantics, or database-semantics (Hausser 1989). Taking the frame representation of UNIX (including attached procedures and, additionally, the protocolled history) as the context-model relative to which a user's input is interpreted, this frame-semantics allows for a simple and efficient processing of a semantic representation for correction, completion, or the retrieval of the requested information via the knowledge base manager. As an illustration, consider the following examples:

1) "How can I edit a text named 'test'?"

```
[qword(how),  
 [action(edit),  
  [mo(file),attr(name,test)]]]
```

2) "Why didn't you list dir 'testdir' sorted by date!"

```
[qword(why-not),  
  [action(show),  
    [so(directory),attr(name,testdir)],  
    [mo(file),quant(all)],  
      attr(name,_),  
      app(descending, attr(date,_))]]]
```

(Note that "list directory" = "show all files"; "so" = source object, "mo" = main object, "attr" = attribute).

In these lists (which might equally be represented as trees), each argument points to a frame in the UNIX knowledge base. Semantic processing then basically consists of successively unifying each of these frames (where the predicates are slots in the frame referred to by the respective mother-node).

In case the unification of a set of frames fails, GOETHE tries a number of heuristics to achieve unification, including: *Identification* (identifying the referents of proper names as denoting a file, a directory, a UNIX command, an owner, or a group), *generalisation* (taking the frame-generalisation of the (first) action node as a candidate for unification), and *precondition-check* (checking whether existence of a file, ownership, and read-, write-, execution-rights are fulfilled as required).

Once a set of frames is consistent, retrieving the answer to a request is stirred by the frames for How, Why, and Why-not, always appearing on the top-level node of the semantic representation. These frames can be understood as representing *strategies* for searching the knowledge base as follows:

How: search for a frame with a goal component containing a special command entry

Why: search for a frame representing a program in the history and compare the used commands with the intended goal with respect to identities

Why-not: search for a frame representing a program in the history and compare the used commands with the intended goal with respect to differences.

Literature

Alshawi et.al. 1986, "Feasibility Study for a Research Programme in Natural-language Processing", SRI International, Cambridge, 1986.

Bez et.al. 1990, "Construction and Interrogation of Knowledge-Bases using Natural Language Text and Graphics: ESPRIT Project 393, ACORD. Final Report", Springer Verlag, 1990 (in print).

Chomsky 1965, "Aspects of the Theory of Syntax", M.I.T. Press, 1965.

Hausser 1989, "Computation of Language", Springer Verlag, 1989.

Heyer et.al. 1988, "Specification of the KB-Manager: A Frame-Extension of DRSSs for Supporting Conceptual Reasoning", ACORD Project, Deliverable T5.4, 1988.

Ralf Kese, Frank Oemig 1989, "GOETHE: Ein kontextsensitives Beratungssystem für UNIX", LDV-Forum, Vol.6, 1989.

A.E.Quilici, M.G. Dyer, M.Flowers, 1986, "AQUA: An intelligent UNIX Advisor", Proceedings of the 7th European Conference on Artificial Intelligence (ECAI), 1986, Vol.II.

W.Wahlster, M.Hecking, C.Kemke, 1988, "SC: Ein intelligentes Hilfesystem für SINIX", in: Gollan, Paul, Schmitt (eds.): "Innovative Informationsinfrastrukturen", Informatik-Fachberichte Nr.184, 1988.

R.Wilensky, D.Chin, M.Luria, J.Martin, J.Mayfield, D.Wu: "The Berkeley UNIX Consultant Project", Computational Linguistics, Vol.14, No.4, 1988.