ANNELISE BECH

The Design and Application of a Domain Specific Knowledgebase in the TACITUS Text Understanding System

Abstract

TACITUS is a text understanding system being developed at SRI International. One of the main components in the system is a knowledgebase which contains commonsense and domain specific world knowledge encoded as axioms in a first order predicate calculus language. The prime function of the knowledgebase is to provide extra-linguistic facts to be used in the resolution of a range of ambiguities such as compound nominal constructions, definite reference, and in drawing conclusions on the basis of the implicatures in the text. The paper discusses the methodology used in building a knowledgebase for analyzing news reports about terrorist attacks, and demonstrates how it is used in an application extracting information to be stored in a simulated database.

1 Preamble

During my term as International Fellow at SRI International, California, this past winter, I had the opportunity to familiarize myself with the TACITUS text understanding system. Under the supervision of Jerry Hobbs, who is head of the TACITUS project, I developed a domain specific knowledgebase for the TACITUS system. The present paper is a brief and fairly high-level and non-technical overview of the enterprise.

Section 2 of the paper presents the methodology used in the construction of the knowledgebase for news reports about terrorist attacks; a crude outline of the TACITUS system is given in section 3 as necessary background information before we go on to looking in detail at an example text in sections 4 and 5. We conclude with some final remarks in section 6.

Proceedings of NODALIDA 1989, pages 114-126

2 The Methodology behind the Construction of the Knowledgebase

Our goal was to build a fairly large knowledgebase for a specific domain, namely terrorist attacks, to be used as a basis for automated understanding of texts falling within this domain, and subsequent automatic extraction of specific information. We decided to work on the basis of a set of sample texts, and we compiled a corpus consisting of several news reports about terrorist events. This corpus then constituted the backbone in our work.

Rather than adopt what might be termed a strict sublanguage approach to the descriptive task (cf. Hirschman 1986, and Hobbs 1984 for more detailed discussions), we employed a methodology of stepwise refinement (cf. Hobbs 1984).

The three steps of our working methodology, which will be elaborated on below, consisted in:

- An (informal) analysis of the corpus texts in order to establish a basic vocabulary, determine and select relevant facts for the domain.
- Breaking up the domain into self-contained and coherent sub-domains.
- Axiomatizing the facts of the subdomains.

2.1 The Analysis of Corpus Texts

Firstly, the corpus texts served the purpose of establishing the basic vocabulary in our system. Secondly, they constituted a picture of the world we intended to model in our knowledgebase, i.e. what are the settings, what are the typical actions, who are the agents, what are the roles and relations between the entities in our 'terrorist' universe, etc. Thus they indicated what linguistic and extralinguistic information would be needed in our knowledgebase.

Using a full-sentence concordance of the sample texts, we looked at each single lexical item in context, and noted down, in an informal manner, what facts were linguistically presupposed and what general background knowledge would be needed in order to understand a given occurrence of a lexical item in its context. (We will not discuss the meaning of 'understand' here, but we use it in a sense similar to that of Eco's term 'actualisation' (Eco 1979)).

The analysis results in a first breaking down of each item into component parts and explicit statements about the implicatures (Grice 1975) carried by the text.

2.2 Structuring the Domain Information

The aim of the second step was to structure the domain information by sorting facts into sub-domains or 'clusters' (Hayes 1985). The prime reason for imposing a structure on the domain is to enhance conceptual clarity, attain modularity, and to be able to discover gaps and logical dependencies in the knowledgebase.

Sorting facts into sub-domains is generally a straightforward process. The first crude distinction which can be made, is that between facts pertaining to commonsense knowledge and domain specific or specialized knowledge. The former is facts about the world in general and not particularly tied to a specific domain (be it terrorist actions, information technology, or what have you), whereas the latter characterizes the facts which are quite often found to be restricted and highly specialized.

Facts pertaining to for example space, time, and belief are considered commonsense knowledge, whereas various facts about terrorist organizations are clearly domain specific, and essential for the understanding of reports about terrorist events. Geographical facts about the location of cities and countries seem to fall somewhere between the more abstract commonsense notion and the specialized domain knowledge.

On the basis of the results from our fact-finding, i.e. step one above, we defined 30 sub-domains. The overall conceptual structure for the knowledge base, the sub-domains and the relations between them, can be schematically rendered by the illustration in figure 1.

Apart from providing conceptual clarity, the advantage of this modular approach is obviously that it permits you to later enhance or modify the subdomains in the knowledgebase independently of each other.

2.3 Axiomatization of the Facts

The final step in the construction of the knowledgebase consisted in creating precise ontologies for the individual sub-domains, i.e. what entities exist and what are the relations between them, and axiomatizing the facts.

The main task here was to decide on which predicates to decompose, i.e. characterize by other or new predicates, and which were to be basic predicates, i.e. ground terms for which no further description is provided.

The idea behind the adopted approach is neither to fully define each lexical item in the sense of providing necessary and sufficient conditions, nor to decompose it into a predefined set primitives in the Schankian tradition. Rather, the purpose was to characterize the predicates used in the knowledge-base. Consider as an example the following axioms from the 'organization' sub-domain.

```
organization (o)   ->   E s (Vx. x∈s ->   person (x) &
                                          member (x,o)) &
                        E p,g  plan (p,g,o)

member (x,o)       ->   E e. role (e,x,o)

role (e,x,o)       <-   agent (x,e) & in_service_of (e,g,p) &
                        plan (p,g,o)
```

These axioms give the basic facts about organizations, i.e. that an organization has persons as members, and that they have a plan. Furthermore, a member

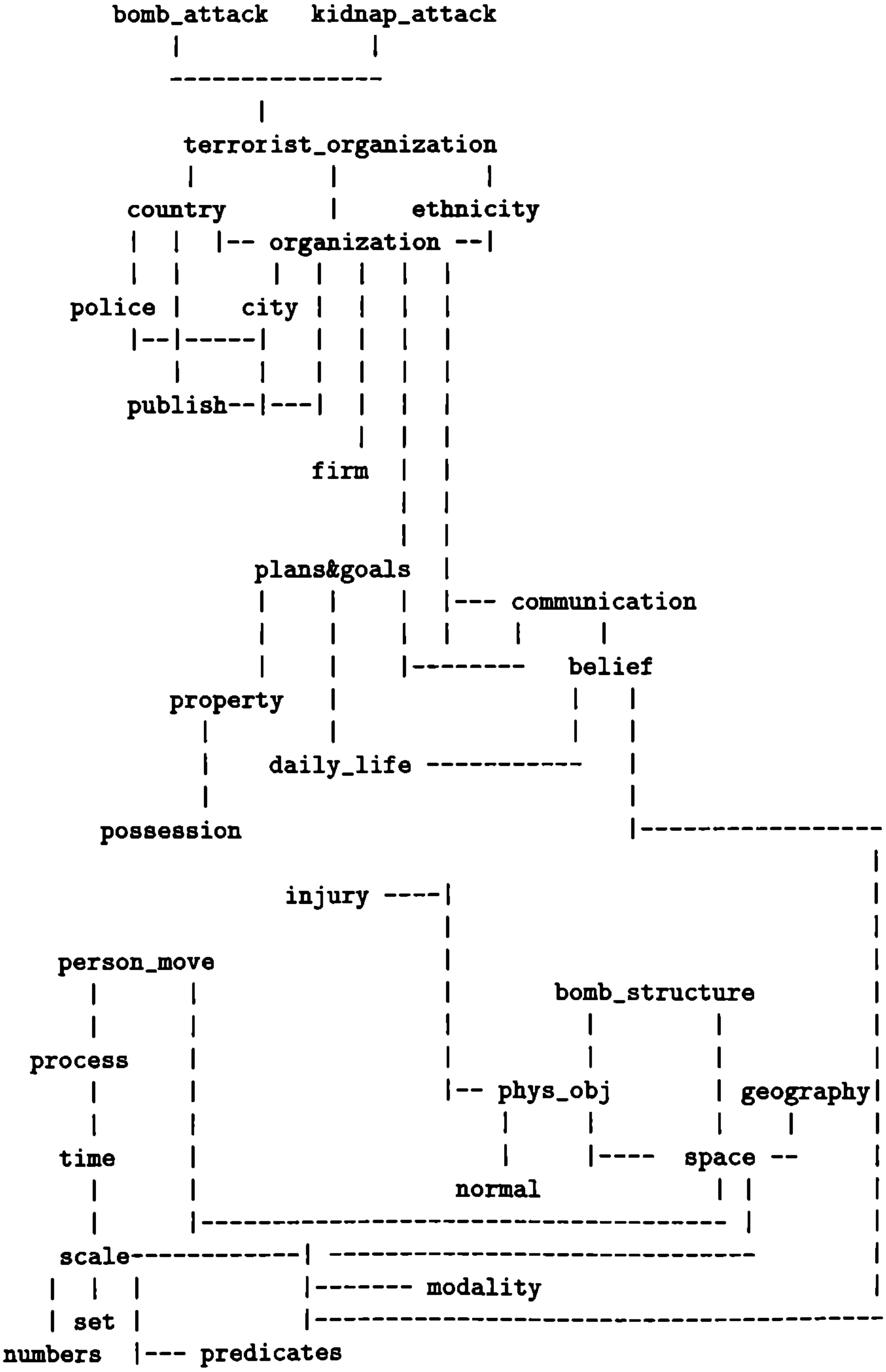
bomb_attack
kidnap_attack
terrorist_organization
country
ethnicity
organization
police
city
publish
firm
plans&goals
communication
belief
property
daily_life
possession
injury
person_move
bomb_structure
process
phys_obj
geography
time
space
normal
scale
modality
set
numbers
predicates

Figure 1:

has a role, which is being the agent of some action which is in service of the plan of the organization.

3 The Knowledgebase and the TACITUS System

To test our knowledgebase, we implemented a subset (app. 100) of the axioms we had defined on the system, and ran different types of sentences. The axioms are stated in the 'ontological promiscuous' notation developed by Hobbs (cf. Hobbs 1985b).

This notation is a first order predicate calculus language with the addition of a nominalization operator, written '!', and an extra argument, informally referred to as the 'self' argument.

To be more concrete and to convey the basic intuition of the notation to the reader, let us consider a simple example:

explode (b) which is to be read as: b explodes

explode!(e1*, b) which is to be read as: the explosion of b.

Where p(x) says that p is true of x, p!(e,x) says that e is the eventuality or possible situation of p being true of x. Consequently, Hobbs' notation can be related to standard first order predicate expressions by the following axiom:

(Vx) p(x) <=> (Ee) p!(e,x) & Rexists(e)

where Rexists(e) says that the eventuality 'e' does in fact really exist.

In sum, the basic idea of the notation is that of splitting a sentence into its propositional content and an assertional/existential claim. Furthermore, the self argument, i.e. the 'e', provides a 'handle' for referring to a predication, i.e. a predicate and its argument, in other predicates.

Before we go on to discussing a sample text, we will give a crude overview of the basic components and their functioning in the TACITUS system. We deliberately ignore some of the more advanced features of TACITUS in order not to get bogged down by too many technical details. Unfortunately, this means that we do not do TACITUS full justice (but for more detailed and comprehensive descriptions of the system, see for example Hobbs 1986c and later).

The system, which is implemented in LISP and runs on Symbolics, comprises an interpretation component and a task specific analysis component. In the interpretation component, there is a parsing and a pragmatics module. The parsing module handles the syntactic and what we will call the basic semantic analysis; this module is a further development of the DIALOGIC system (Grosz et al. 1982) used in TEAM (Grosz et al. 1987). As output, it produces a logical form of the parsed sentence in a first-order predicate calculus language. The logical form is elaborated on, or more precisely, further processed by the second module of the interpretation component, the pragmatics module. The task of the pragmatics module is to resolve referential expressions and some syntactic ambiguities, to expand metonymies, and to interpret the implicit relations in compound nominals. The pragmatics module works by constructing a logical expression for the basic semantic analysis result, and calling the KADS theorem prover (Stickel 1982) to prove or derive it using a scheme of abductive inferencing in which it is permitted to assume the existence of 'new' facts. The theorem prover draws on the knowledgebase of commonsense and domain knowledge to complete the task.

Abductive inference is, of course, a logically invalid mode of inference, i.e. given p(X) → q(X) and q(a) we conclude p(a). However, we may argue, as does Hobbs (cf. Hobbs et al. 1988), that it is a reasonable way of looking at text understanding because abduction is inference to the 'best explanation' in a given context. q(a) can be thought of as the observerable evidence, the implication as the general principle that could explain the occurence of q(a), and the antecedent of the implication as the underlying cause or explanation of q(a).

An interesting feature of the pragmatics module is that it uses a scheme for abductive inferencing in which weights and costs are assigned to the axioms (for further details, see e.g. Stickel 1988). Thus if we cannot prove an antecedent, we assume its existence at some cost. Some basic heuristic principles controlling the weights and assumability costs are hardwired into the system (e.g. it is more expensive to assume a fact than to prove it, and it is less expensive to assume an indefinite entity than a definite one), but the axioms in the knowledgebase may be assigned costs manually (cf. 4.2). The interpretation of a text in this abductive and assumption-based framework, amounts to producing the minimal explanation of why the text would be true (cf. Hobbs et al. 1988 for a detailed discussion).

The analysis component, i.e the component for extracting task specific information from an interpreted text, is basically a specialized call to the theorem prover (see further below). The enhanced logical form, i.e. the result output from the pragmatics module, is abductively proved by back-chaining over the axioms in the knowledgebase.

In the next sections, we will have a look at an example text and show how the knowledgebase is used for disambiguation and computation of implicit information.

4 An Example

Let us now consider the following two sentences as an example text to be treated within our framework:

(1) A bomb exploded at a Renault showroom in Bilbao. A person claiming to represent the ETA-M had warned of the blast in a call to the police.

Linguistically, the sentences present us with problems of resolving a compound nominal construction, 'Renault showroom', and locating a possible antecedent for 'the blast'.

The extra-linguistic knowledge needed in order to achieve some reasonable level of understanding of the text is among other things: Renault is a French firm manufacturing products, i.e. cars, a showroom is a building owned by a firm where the products of that firm are on display, Bilbao is a city in the country Spain, ETA-M is a terrorist organization, and terrorist organizations have members, certain plans and goals and violent methods for reaching their goals, and an explosion generally involves a blast.

The basic facts such as for instance Spain being a country and ETA-M being a terrorist organization, are encoded as existential axioms in the knowledgebase. E.g:

```
(1a) (Defaxiom COUNTRY-SPAIN-1 (terror)
        ''Spain is a country''
        ((SOME ((e1* . ev) (country! e1* spain)))

(1b) (Defaxiom TERORG-ETA-M-1 (terror)
        ''ETA-M is a terrorist organization''
        ((SOME ((e1* . ev) (terorg! e1* eta-m)))
```

The quantified variables in the axioms are marked for their type such that 'ev' denotes event and 'nev' non-event variables.

4.1 Axioms for Disambiguating Compound Nominal Constructions

From the linguistic point of view, the TACITUS framework offers interesting possibilities for disambiguating compound nominal expressions using linguistic as well as extra-linguistic knowledge.

The individual nouns in a compound nominal construction are analyzed as arguments of the generic 'nn'-predicate. That is, the expression 'Renault showroom', would appear as **nn(e1*,Renault,Showroom)** in the initial logical form of the sentence produced as output from the parsing module.

In formulating the axioms for resolving such nn-relations, we adopted a strategy combining the line of analysis for coumpound nominals proposed by Downing (1977), and that advocated by Levi (1978). In summary, Downing argues that the semantic relationship between the elements of a coumpound cannot be characterized in terms of a finite list of appropriate compounding relationships, whereas Levi tries to establish such a list for the most common cases on the basis of the transformational relationship between the elements.

Our combined approach can be seen in the following sample axioms, where the first two axioms encode the possible general relationship as expressed in terms of prepositions, and the subsequent two axioms state further specific constraints.

```
(2a) (Defaxiom NN-1 (terror)
     ''An nn-relation: for''
     (ALL ((e1* . ev) (p . nev) (s . nev))
          (IMPLY (for! e1* s p)
               (SOME ((e2* . ev))
                    (nn! e2* p s)))))

(2b) (Defaxiom NN-2 (terror)
     ''An nn-relation: of''
     (ALL ((e1* . ev) (f . nev) (s . nev))
          (IMPLY (of! e1* s f)
               (SOME ((e2* . ev))
                    (nn! e2* f s)))))

(3a) (Defaxiom FOR-1 (terror)
     ''A showroom is for products''
     (ALL ((e2* . ev) (s . nev) (e3* . ev) (p . nev) (e4* . ev) (f . nev))
          (IMPLY (AND (showroom! e2* s) (product! e3* p) (firm! e4* f))
               (SOME ((e1* . ev))
                    (for! e1* s p)))))

(3b) (Defaxiom OF-1 (terror)
     ''A showroom is owned by a firm''
     (ALL ((e2* . ev) (s . nev) (e3*. ev) (e4* . ev) (f . nev))
          (IMPLY (AND (showroom! e2* s) (own! e3* f s) (firm! e4* f))
             (SOME ((e1* . ev))
                      (of! e1* s f)))))
```

In trying to abductively prove a relevant logical form output from the parsing module and to make implicit information explicit, the pragmatics module has the theorem prover back-chain over the axioms in the knowledgebase. Thus an nn-relation as the above is resolved against 2a and 2b, then the new goals, of!(e1* s f) and for!(e1* s f), are resolved against 3a and 3b respectively, yielding new goals to be resolved.

4.2 Axioms for Resolving Referring Expressions

As mentioned above, one of the basic heuristic assumption hardwired into TACITUS' pragmatics module is that an indefinite noun phrase introduces new information and a definite noun phrase refers to a known entity, i.e. something which is either in the knowledgebase or has been introduced in the previously processed text. Hence the cost of assuming an indefinite noun phrase is cheaper than assuming a definite noun phrase.

In the example sentences given in (1), the noun phrase 'the blast', is related to the event of the explosion mentioned in the preceeding sentence. Simplifying somewhat (cf. further below), we could say that 'the blast' is in a sense a nominalization of 'a bomb exploded'.

In order to establish reference connections of this type, we define the following kind of axiom in our knowledgebase:

```
(4) (Defaxiom EXPLOSION-BLAST-1 (terror)
    ''An explosion generates a blast''
    (ALL ((e1* . ev) (x . nev) (y . nev) (z . nev))
         (IMPLY (AND (ASSUMABLE (etc-expl e1* x y z ) 0.3)
                     (explode! e1* x y z))
                (SOME ((e2* . ev) (b . nev))
                     (AND (blast! e2* b) (genn e1* e2*))))))
```

Essentially, this axiom says that a blast (e2*) implies the occurrence of some explosion event (el*), and that the latter generates the former, which is stated by way of the primitive predicate 'genn'. The predicate 'etc-expl', which can be seen as 'additional', but not spelled out properties relating to the explode predicate, is introduced because we do not want to state flatly that 'a blast' and 'an explosion' is the same thing.

Since an 'explosion' is known (it was introduced in the previous sentence), it is free of charge to resolve the second predicate in the antecedent of the axiom against this known fact. The first predicate in the antecedent has been assigned such a low assumability cost (0.3), that proving 'blast' by use of the axiom is cheaper than to assume its existence.

5 Extracting Specific Information from the Texts

The logical form encapsulating the interpretation found for a text, i.e. the output from the interpretation component, is the input to the task specific analysis component. The analysis is performed on the basis of the logical form and a 'task schema specification' given to the theorem prover.

5.1 The Schema

Let us here consider a simplified example of the kind of event related specific information we would like the system to compute. For a given text describing a terrorist event, we would like to find answers (if any) to 'questions' such as the following:

```
INCIDENT TYPE:
TARGET TYPE:
TARGET NATIONALITY:
INCIDENT CITY:
INCIDENT COUNTRY:
RESPONSIBLE ORGANIZATION:
.
.
etc.
```

The above actually simulates a database record to be automatically filled in. However, as the system was not yet hooked up to produce actual database entries, the answers found are printed out on the screen. The slots in the 'record' are filled by the values found for variables when presenting the theorem prover with goals to be abductively proven by using the information from the text interpreted and the facts in the knowledgebase.

The goals of the schema appear as the consequent in what might informally be called the 'linking axioms' in the application task specific part of the knowledgebase. Linking axioms can be thought of as guidelines for how to find answers to the 'questions' posed by way of the schema specification.

The schema itself is a metalogical LISP expression in a first-order predicate calculus form annotated by non-logical operators for search control and resource bounds. The two non-logical operators are 'proving' and 'enumerated-for-all'. Without going into technical details about these two operators (for more details, see Tyson and Hobbs 1988), let us simply present a small excerpt from the schema for the above example 'record', and make some explanatory comments in order to convey the basic intuitions of the process to the reader:

```
(proving
     (enumerated-for-all ((e1 . ev))
          (proving
               (some ((it . nev)) (incident-type e1 it))
               (terror-limits default-time)
               print-incident)
               (and
                    (enumerated-for-all ((it . nev))
                         (proving
                              (incident-type e1 it)
                              (terror-limits default-time)
                              print-incident-type)
                             :true)
                                 :
                                 :
                    (enumerated-for-all ((ro . nev))
                         (proving
                              (responsible-organization e1 ro)
                              (terror-limits default-time)
                              print-responsible-organization)
                             :true)
               (terror-limits default-time)
          print-sentence-finished)))
```

The linking axiom in the knowledgebase for 'responsible organization' could be the following statement:

```
(5) (Defaxiom RESP-ORG-1 (terror)
    ''The organization responsible for the attack''
    (ALL ((e1* . ev) (e . ev) (e2* . ev) (o . nev) (e3* . ev))
         (IMPLY (AND (terattack! e1* e) (responsible! e2* o e)
                                             (terorg! e3* o))
                (responsible-organization e o))))
```

Thus, we find the organization (o) responsible for an attack (e) by proving that e is a terrorist attack, that o is a terrorist organization, and that o is responsible for e.

Contrary to the pragmatics module, no assumptions are made in the task specific analysis phase when trying to prove the goals of the schema; this step is meant to extract information only. However, the process is still back-chaining controlled abductive inferencing. This means that everything has to be proved against the knowledge in the database in conjunction with the interpretation of the text.

Proving the antecedents of the linking axioms may of course involve resolving the new goals with knowledge asserted in the text or in this case, proving further axioms in the knowledgebase.

There may also be different axioms for the same goal, indicating that a goal can be explained, or more correctly proved, in different ways. Actually, this is only a reflection of the fact that a given phenomena can be brought about in different ways. For example, there are actually three different axioms for 'responsible' in our knowledgebase.

5.2 The Information Extracted from the Interpretation Result

Let us now return to our example text. For illustration, we first show an excerpt from the result of the interpretation of the sentences in external format (6) — note the resolved compounding relationship; and then the print-out of the information automatically extracted by the analyze component from the interpretation (7) of the two example sentences.

(6)

```
INTERPRETATION 1 OF SENTENCE:
          Cost: 34
New and Assumed Information:
x1:                 bomb!(e2, x1)
y1:                 explode!(e4, y1, x1)
x12:                bilbao!(e13, x12)
x8:                 renault!(e9, x8)
x6:                 showroom!(e7, x6)
                    in!(e11, x6, x12)
e4:                 at!(e5, e4, x6)
                    past!(e15, e4)
Given or Inferred Information:
x8:                 renault!(e9, x8)
                    nn!(e10, x8, x6)
                    own!(e25, x8, x6)
                    firm!(e26, x8)
x6:                 of!(e29, x6, x8)
```

(7)

```
INCIDENT TYPE: explosion
TARGET TYPE: commercial
TARGET NATIONALITY: french
INCIDENT CITY: bilbao
INCIDENT COUNTRY: spain
PROPERTY DAMAGE: <unknown>
WARNING: yes
METHOD: phone
RESPONSIBLE ORGANIZATION: eta-m
```

6 Final Remarks

TACITUS offers an interesting framework for experimenting with knowledge-based natural language processing, and in fact it is a quite sophisticated system. Previously, the TACITUS team at SRI has been experimenting with implementations of knowledgebases for domains such as the break-down or malfunctioning of mechanical parts in ships (Hobbs 1987). Constructing a knowledgebase for the terrorist attack domain was the first attempt to deal with a slightly less restricted subject field in the TACITUS system. The main conclusion to be drawn from the experiment with the terrorist texts is that very careful axiomatization of the facts is necessary in order to achieve good results, i.e. 'nuts and bolts' have to be carefully fitted together to create 'delusions of grandeur'.

Acknowledgements:
The Danish Carlsberg Foundation provided the financial support for my stay at SRI International. Constructing and testing the domain specific knowledgebase for terrorist texts in the TACITUS system described here, was suggested to me by Jerry Hobbs and carried out under his supervision. I am indebted to Jerry for his guidance and many useful hints. Needless to say, if the present paper contains errors or misconceptions in the presentation of TACITUS, the author alone can be blamed.

References

Eco, U. 1979. *Lector in Fabula.* Milan.

Grice, H.P. 1975. Logic and Conversation. R. Schank and B. Nash-Webber [Eds.], *Theoretical Issues in Natural Language Processing*169–174. Cambridge, Mass.

Grosz, B., N. Haas, G. Hendrix, J. Hobbs, P. Martin, R. Moore, J. Robinson, and S. Rosenschein. 1982. *DIALOGIC: A Core Natural-Language System.* SRI Tech. Note 270. SRI, Menlo Park, California.

Grosz, B., D.E. Appelt, P.A. Martin, and F.N.C. Pereira. 1987. TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces. *Artificial Intelligence*, 32:173–243.

Downing, P. 1977. On the Creation and Use of English Compound Nouns. *Language*, 53:810–842.

Hayes, P.J. 1985. The Second Naive Physics Manifesto. J.R. Hobbs and R.C. Moore [Eds.], *Formal Theories of the Commonsense World*:1–36. Ablex, New Jersey.

Hirschman, L. 1986. Discovering Sublanguage Structures. R. Grishman and R. Kittredge [Eds.], *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*211–234. Erlbaum, New Jersey.

Hobbs, J.R. 1978. Coherence and Coreference. SRI Tech. Note 168. SRI, Menlo Park, California.

Hobbs, J.R. 1984. Sublanguage and Knowledge. SRI Tech. Note 329. SRI, Menlo Park, California.

Hobbs, J.R. 1985a. Granularity. In: *Proceedings of IJCAI-85*:1–4.

Hobbs, J.R. 1985b. Ontological Promiscuity. In: *Proceedings of ACL-85*:61–69. University of Chicago, Illinois.

Hobbs, J.R. 1986a. Commonsense Metaphysics and Lexical Semantics. SRI Tech. Note 392. SRI, Menlo Park, California.

Hobbs, J.R. 1986b. Discourse and Inference. Ms. SRI, Menlo Park, California.

Hobbs, J.R. 1986c. Overview of the TACITUS Project. *Computational Linguistics*, 12:220–222.

Hobbs, J.R. 1987. Local Pragmatics. SRI Tech. Note 429. SRI, Menlo Park, California.

Hobbs, J.R., W. Croft, T. Davies, D. Edwards, and K. Laws. 1988. The TACITUS Commonsense Knowledge Base. Ms. SRI, Menlo Park, California.

Hobbs, J.R., M. Stickel, P. Martin, and D. Edwards. 1989. Interpretation as Abduction. Ms. SRI, Menlo Park, California.

Levi, J. 1978. *The Syntax and Semantics of Complex Nominals*. Academic Press, New York.

Stickel, M.E. 1982. A Nonclausal Connection-Graph Resolution Theorem-Proving Program. *Proceedings of the AAAI-82 National Conference on Artifical Intelligence*: 229–233. Pittsburgh, Pennsylvania.

Stickel, M.E. 1988. A Prolog-like Inference System for Computing Minimum Cost Abductive Explanations in Natural Language Interpretation. *Proceedings of ICCSC 88*:343–350. Hong Kong.

Tyson, M. and J.R. Hobbs. 1988. Domain-Independent Task Specification in the TACITUS Natural Language System. Ms. SRI, Menlo Park, California.

EUROTRA-DK
University of Copenhagen
Njalsgade 80
DK-2300 Copenhagen S.
annelise_bech@eurokom.ie