Understanding Spontaneous Negotiation Dialogue*

Michael Kipp and **Jan Alexandersson** and **Norbert Reithinger**
DFKI GmbH,
Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany
{kipp|janal|bert}@dfki.de

Abstract

In this paper we present the task-oriented context representation and the dialogue manager for the VERBMOBIL translation system. We show how to utilize statistical methods, shallow extraction and propositional representation to provide translation relevant information and most of all, to enable the system to automatically create a dialogue script and result summary.

1 Introduction

VERBMOBIL is a speech-to-speech translation system working in English, German and Japanese [Bub and Schwinn, 1996]. The first four years of the project resulted in a prototype being able to produce approximately 75% correctly translated contributions in the domain of appointment scheduling dialogues. In the second phase, different extensions in domain and functionality were to be implemented. The domain now includes travel planning and hotel reservations and one of the new features is the automatic generation of a dialogue summary or script [Alexandersson and Poller, 1998]. Since, in the second phase of VERBMOBIL, a possible scenario is that of a telephone server used by two participants with VERBMOBIL as a third party, a dialogue script does provide some kind of a status report where each participant can check what items have been agreed on already. A summary then lists all final decisions in a thematic order. Such documentation would reduce misunderstanding and the need for *clarification* (a dialogue partner forgetting the agreed on items). Lack of robustness can be compensated by such a transparent context component where the user can see and react to errors in speech recognition or analysis.

For translation in the VERBMOBIL system various fundamentally different methods are employed. We call them *deep* and *shallow* translation tracks [Bub *et al.*, 1997]. The deep track works with syntactical and semantical analyses and a logical, DRT-inspired representation to conduct a semantic transfer. Each language has its own generator for content-to-speech generation. The shallow track uses finite state technology and statistics as well as rule and plan based techniques to produce an approximate translation of task-relevant content [Reithinger, 1999].

One of the features of VERBMOBIL is its ability to take into account contextual and pragmatical information which is gained, accumulated and distributed by the dialogue and context component. It was the idea of making transparent this information that led to the new feature of an automatic summary/script. This paper describes work in progress, explaining how the context information is gathered and encoded, and what extensions to the dialogue manager and our interfaces were necessary to support the summary feature. We will also give examples of other consumers of context information within the VERBMOBIL system. We conclude the paper with open problems and future work.

2 Welcome to the Real World

Since work on the VERBMOBIL system always has to take into account the complete processing from speech input to speech output we have to deal with incorrect information due to shortcomings in other modules, especially speech recognition. The following two examples show how dramatic even small errors in recognition can be in terms of interpretation (recognized string in italics):

(a) Unfortunately I have only time in December.
unfortunately I have a meeting of December.

(b) When would be a good time for us to meet?
one would be a good time for us to meet.

Utterance (a) will be interpreted as a rejection of December as a possible date (since having a meeting usually signals an explained rejection of a date) whereas utterance (b) triggers the false suggestion of one o'clock for the meeting.

These utterances were taken from a corpus of end-to-end evaluation dialogues which were recorded in test runs under realistic conditions, i.e. with only the VERBMOBIL

*The research within VERBMOBIL presented here is funded by the German Ministry of Research and Technology under grant 01IV101K/1.

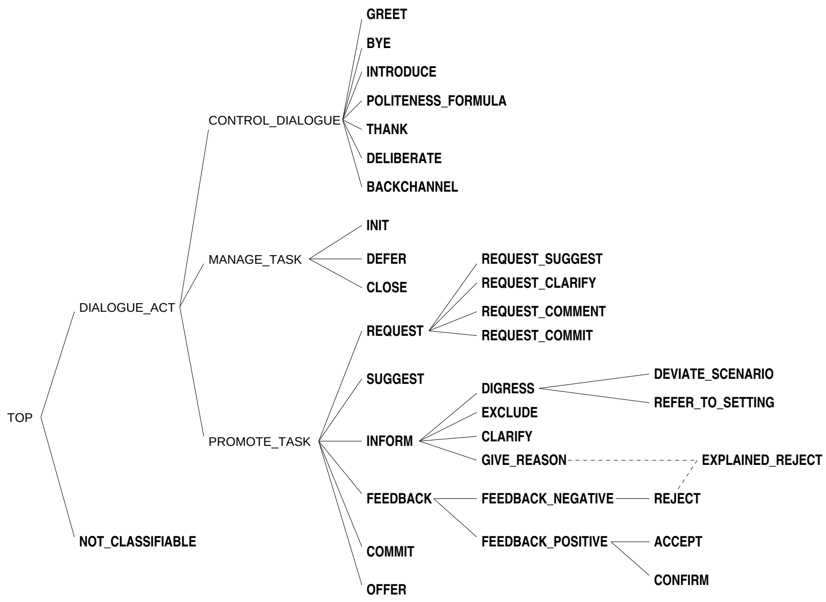

Figure 1: Dialogue acts hierarchy as employed in VERBMOBIL 2

system as a translator between an English and a German person. Although in utterances like the above it is almost impossible to recover the original meaning, other heavily damaged input like the next example is easily interpreted correctly by our shallow analysis approach:

(c) OK, could you repeat please. Do you have time on the fifth and sixth, or don't you?
repeat please do you have time. on the fifth and sixth why don't you.

Apart from imperfect input, two other factors influence our approach to dialogue understanding in VERBMBIL. First, users of VERBMOBIL in its current state have to deal with inaccurate translations and therefore, make ample use of confirmations and clarifications. In general, people behave differently in a real application of VERBMOBIL as opposed to the ideal conditions of monolingual sample dialogues from the VERBMOBIL corpus. Second, since VERBMOBIL acts as a mediator between two or more parties it is not supposed to pose questions itself (clarification dialogues). Its dialogue engine is not an interactive machine but solely tracking the dialogue. It has to obey the principle of *unobtrusiveness*. As a result, we pursue an approach using methods not unlike those of Information Extraction (IE, cf. [Hobbs *et al.*, 1996]): we know what to expect, we try to extract as much information as possible and put them into frames, checking consistency on the way.

To guarantee robustness under such working conditions, a number of domain-specific assumptions have to be made and implemented (see section 4.4).

3 Extraction of Utterance Contents

A major premise of our representation is the assumption that in a task-oriented dialogue it is sufficient to know the communicative function (dialogue act) and the propositional content of an utterance. The criterium for a successful shallow translation is the conservation of dialogue act and propositional content [Levinson, 1993]. Although our representational structure is custom-designed, we believe that it transfers well to other negotiation domains, including interactive scenarios, and information retrieval dialogues (in hotel, airport, train station or tourist information domains).

3.1 Representation

The communicative function of an utterance is modeled by the dialogue act which is one of the labels of a structured hierarchy as shown in fig. 1. Once a dialogue act is found (statistically or rule-based) it is attached to the internal representation of the utterance as a label (note that multiple, overlapping communicative functions of a single segment are possible in which case we attach a list of labels).

The propositional content of an utterance is modeled by a number of objects – abstract and physical – that are connected by relations. These objects are instances of an ISA class hierarchy and refer to real-world objects (a small part of the class tree is shown in fig. 2).

The representation language is called DRL (discourse representation language), its entities are called DIREXes (discourse representation expressions). Embedded in this language is a special time representation language whose expressions are called TEMPEXes [Endriss, 1998]. An example representation looks like this:

so how about a three o'clock flight out of Hannover
[has_move:[move,has_departure_time:{time_of_day:3},
has_source_location:[city,has_name='hannover']]]

The utterance itself is also represented by a DIREX. It contains slots for the dialogue act, the topic and pointers to above mentioned objects. Due to frequent occurrences of ellipses and anaphora the contents information delivered by shallow extraction is incomplete and has to be extended with the help of context knowledge (see section 4.3 below). The utterance object points to the extended object as well as to the original object delivered by shallow extraction.

3.2 Deep Extraction

VERBMOBIL makes use of different translation methods which are ordered along the line of one so-called translation track each. The *deep* track employs HPSG-grammars to obtain syntax and semantics which are in turn encoded in VIT's (VERBMOBIL interface terms) [Bos *et al.*, 1998]. From these VIT's, together with pragmatic information, direx expressions are extracted and transferred to the dialogue manager for completion, storage and evaluation.

3.3 Shallow Extraction

Triggered by the real world conditions of the system as explained in section 2, a number of shallow techniques were devised to cope with the incoming data. The shallow track employs a combination of statistical methods and finite state technology to extract dialogue act, TEMPEX and DIREX expressions [Reithinger, 1999].

The dialogue act is found by statistical methods using trigrams and deleted interpolation [Reithinger and Klesen, 1997]. This requires a data-base of manually annotated dialogues which serves as training material. We currently use a training corpus of about 660 dialogues for German, 300 for English and 400 for Japanese.

The string output of the speech recognizer plus the recognized dialogue act is then sent to a language-specific set of finite state automata which deliver valid TEMPEXes and a number of semantic chunks that are assembled to a DIREX. These automata were manually encoded with a graphically enhanced toolkit (see fig. 3). Each set can be seen as a number of ordered cascaded automata with special actions for state transition (allowing e.g. *global* search on the whole string for *one* transition). All automata are applied to the input in an ordered fashion and each one builds on the output of the preceding one. This output consists of bracketed *chunks* and the remaining string. The final output delivers a bracketed representation of a DIREX. Currently, we use 186 automata for German, 167 for English and 127 for Japanese extraction.

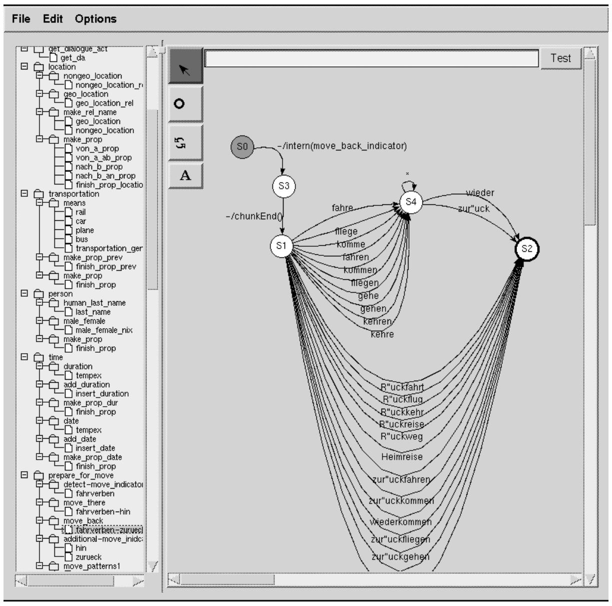

Figure 3: The graphical user interface for automaton development

Performance in dialogue act recognition achieves an accuracy of about 70% on unseen data and is even better on the actual evaluation dialogues (like the ones mentioned in section 2) since dialogues with the running system are often simpler than our sample data.

Content objects found by shallow extraction are also reliable since they are based on very local linguistic evidence and do not expect correct syntactic structures. The objects correspond to linguistic forms within the utterance. Inferences concerning co-reference and reference to real-world objects are drawn at a later stage.

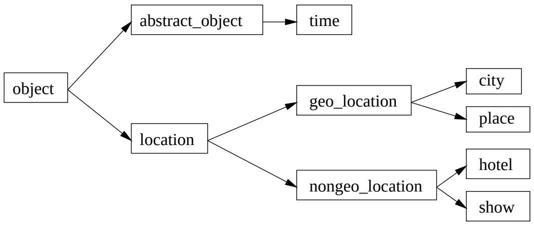

Figure 2: OBJECT subtree of direx class hierarchy

4 Tracking the Dialogue

In the first phase of VERBMOBIL we were most of all concerned with time expressions and inferences to find out the speakers attitude towards those expressions (accept, reject, uptake) [Maier, 1996; Alexandersson *et al.*, 1997]. In the new domain we have to deal with more complex suggestions, e.g. flights, hotel reservations, sparetime activities in the evening. Still, the limited domain and the final goal of devising a summary/script lets us make certain assumptions about how the incoming objects could look like.

4.1 Topics

First of all, we partition the task space into topics: SCHEDULING, TRAVELING, ACCOMMODATION and ENTERTAINMENT. For each topic we keep structural information about potential incoming data (so-called *blueprints*) which is used to complete/assemble the semantic chunks of each new utterance (see 4.3).

Each topic stores information in a frame. This information consists of a focus stack and a storage with accepted suggestions. The focus stack keeps track of the most recently mentioned suggestion. As soon as a suggestion is accepted it is also put to the 'accepted' slot. Topic shifts are recognized by using rules that work on current topic, key-words, dialogue act and extracted content. They are managed by the following algorithm:

if this is the first utterance
  then take SCHEDULING as new topic

elsif dialogue act is INIT
  then determine new topic with key-words and content objects

elsif key-words or content objects indicate other topic than current one
  then check evidence

else retain current topic

In case of a topic shift the respective focus stack is re-instantiated. This locality of focus has proven useful in the final phases of negotiation dialogues where confirmations for different topics are run through once more like in the following transcript:

`A01:` so that was Monday the twenty-first at the check-in counter [SCHEDULING]

`B02:` I'll do the flight reservations [TRAVELING]

`A03:` and I will let my secretary take care of the hotel [ACCOMMODATION]

The respective items (flights, hotel) can be found on the local focus stack of the respective topic frame. The topic shift in `B02` is recognized using key-word spotting.

4.2 Dialogue Manager

We do not keep an explicit dialogue model with a range of legal states and transitions. Instead we have a set of internal actions triggered mainly by the segment's dialogue act and using different sorts of information as arguments (topic, focus, stored suggestions etc.). It is certainly possible to formulate a dialogue model based on a formal definition of dialogue states but this may be of more importance in an interactive system where one wants strict control over the system's (visible) behaviour.

The dialogue manager needs to handle content: anaphoric references and ellipses in suggestions have to be resolved. The dialogue manager also keeps track of the speakers' attitudes towards the content objects (accept/reject etc.) and handles topic (see above) as well as focus. Each new segment/utterance is processed according to its *dialogue act*:

suggest: propositional content is completed (see 4.3), stored and kept in focus

feedback: the speaker's attitude (accept/reject) is annotated in focussed suggestion; a *strong* accept/reject is an utterance that mentions the accepted/rejected proposal explicitly, e.g.

`A01:` let's meet on Tuesday then.

`B02:` Tuesday is fine.

confirmations are annotated as strong accepts, e.g.

`A03:` so I see you Tuesday, 2 o'clock in your office

The annotation of the speakers' attitudes serves as evidence for the summarization (see section 4.4).

Question-answer pairs are dealt with by pushing the question item to a temporary short-term storage and waiting for a reply. The reply then triggers the treatment of the content data. We distinguish two types

yes/no-questions: in case of a positive reply, the propositional content is treated as if introduced as a fact at the point of the positive reply.

information requests: in this case the question usually provides one part of the object and the reply the other, e.g.

`A01:` when's that flight going?
  → [plane, has_date: ?DATE]

`B02:` two thirty.
  → {time_of_day:2:30}
  ↪ [plane, has_date: {time_of_day:2:30}]

Again the fact is added as if stated at the time of the answer.

4.3 Completing the Data

Ellipses and anaphora are commonplace in everyday conversation and thus, also in negotiation dialogues. Finding them is one problem, representing them another one. As for the representation there are two principal approaches, one using links (which replace missing data) and inferring the complete object at a later stage and one using instant completion of the data. For anaphora that is replacing the anaphor by the actual reference object and for ellipsis that is adding the missing object(s).

Time expressions

For time expressions it has proven useful to go the path of instant completion [Birkenhauer, 1998]. Wherever a time expression is encountered the system tries to find a sponsoring expression (either in the focus or in the *situative context*, i.e. the time of the dialogue) and completes the new expression (see examples[1]).

[1] in our examples we will use a simplified notation for the direx and TEMPEX formalism. TEMPEXes will appear in curly brackets, direxes in square brackets. Those parts of an expression that have been taken over from a sponsor are underlined.

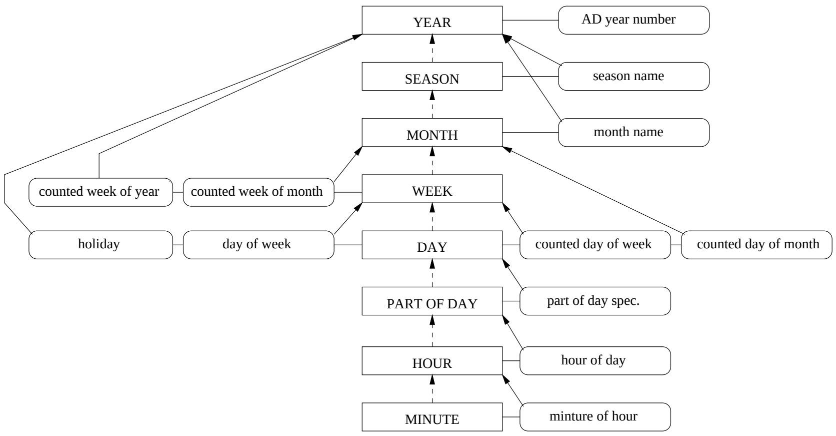

Figure 4: principal temporal units (capital letters) and their possible specifications

A01: Why not meet on the fourth of June?
⟶ {month:june, day_of_month:4}

B02: The sixth would be better, I'm afraid.
⟶ {month:june, day_of_month:6}

A03: So is it going to be the eigth or the ninth?
⟶ or({day_of_month:8}, {day_of_month:9})

B04: A Friday? Yes, Friday's good.
⟶ {day_of_month:9, day_of_week:friday}

A05: Six o'clock looks like a good time for me.
⟶ {time_of_day:6:0})

B06: Couldn't we do it before?
⟶ before({time_of_day:6:0})

Our approach makes use of the temporal specification tree in fig. 4. A complete temporal expression is defined as contiguous path from the most specific node (that would be *counted day of month* in the first example) to the root node (YEAR). The completion of a time expression uses this tree to find out the missing temporal data that has to be taken from the sponsoring expression.

Generalization

We employ a similar approach with the general representation in the direx formalism. Each topic contains a number of so-called *blueprints* which are frame structures for those objects that are expected to occur within a specific topic (see an example for topic TRAVELING in fig. 5). The blueprint corresponds to the temporal specification tree. Incomplete incoming data (due to ellipsis or anaphora) is extended according to this blueprint. In the following example a time expression that was found in utterance B02 is extended to form an object of type MOVE:

A01: Which plane do we take?
⟶ [move,has_transportation:[plane]]

B02: There's one at six fortyfive.
⟶ [move,has_transportation:[plane], has_date:{time_of_day:6:45}]

Completion is conducted using the blueprint and a focused object (compare fig. 5 and the example). The blueprint specifies what elements have to be added and the focused object acts as a sponsor.

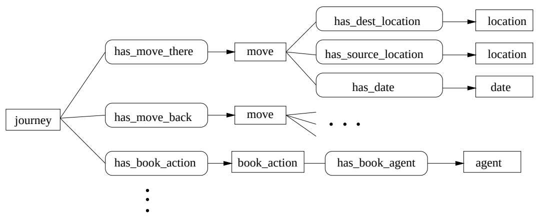

Figure 5: Completion blueprint for objects of topic TRAVELING

4.4 Wrapping it up

The summary information contained in the topic frames still need some checks and extensions before it can be transferred to generation. Since the result summary will have a thematic ordering in topics we want each topic to contain as much information as possible. Hence, we implement a set of rules inferring relatively secure data from what is known in the context memory:

- location of the accommodation equals the destination location of the trip
- source location of moves equals location of the dialogue if not otherwise mentioned
- the date of evening entertainments corresponds to that of the hotel stay (if it's only one night)

Consistency checks are necessary because of the known shortcomings of speech recognition and data extraction. Since human dialogue contains a lot of redundant information it is more than obvious to exploit this fact. Therefore, all suggestions have been annotated with speaker attitudes (accept, strong accept, reject etc.) which are used to compute the final agreement. If there are two conflicting dates for one meeting a score for each date is computed to select the one supported by more evidence.

5 Consumers of the Information

The accumulated context structures described so far are requested by other modules of the VERMOBIL system. We give a brief overview of how these modules use this data.

Dialogue script generator

The main consumer of the thematic structure is the dialogue script generator [Alexandersson and Poller, 1998]. It provides the user(s) with two different functionalities. The first, which is called *dialogue script*, is a condensed description of the most *salient* parts of the dialogue. The second, *result summary*, summarizes the *result* of the negotiation. In this paper we describe the result summary (the functioning of the dialogue minutes is described in [Alexandersson and Poller, 1998].). For both types, a set of VITs is construed which, when required, can be transfered to any VERBMOBIL language.

VERBMOBIL ERGEBNISPROTOKOLL Nr. 1

Teilnehmer: Herr Rosenstock, Frau Johann
Datum: 1.3.1999
Uhrzeit: 15:26 Uhr bis 15:27 Uhr

GESPRÄCHSERGEBNISSE:

Reiseplanung:

Am 14. Februar 1999 findet die Hinreise statt . Mit dem Flugzeug findet sie statt . Eine Stunde dauert die Hinreise nach Hannover . Am Montag findet die Rückreise statt . Herr Rosenstock bucht die Reise . Herr Rosenstock und Frau Johann reisen gemeinsam .

Unterkunft:

Frau Johann bucht die Unterkunft .

Protokollgenerierung automatisch am 8.3.1999 14:58:55 h

Figure 6: A sample result protocol

The result summary is based on a thematic view on context information based on a partitioning of the task space into the pre-defined topics mentioned in section 4. It basically mirrors the contents of the thematic memory. For the generation, it suffices to traverse the structure, select (what to say) and partition (how to say) the information collected during the course of the dialogue. A sample summary, generated from our corpus, is shown in figure 6[2].

Semantic transfer

For translation, the most important consumer is the transfer component. The semantic transfer has to cope with many ambiguities that remain after syntactic/semantic analysis concerning mainly discourse particles and verbs. For disambiguation they use dialogue act and focused object, e.g. in the case of the English verb *go* which can be translated to *gehen* (walk), *fahren* (drive) or *fliegen* (fly). Here the class of the focused object (PLANE, RAIL, BUS) provides sufficient information to determine a unique reading.

Deep analysis

Another consumer is the English HPSG analysis. We support the speaker and addressee for the correct reading in the following case (*"We have to meet Mr. Hallerman." vs. "We have to meet, Mr. Hallerman!"*). If it is known that the hearer is called Mr. Hallerman, one can discard the first reading.

6 Related Work

The most recent related work can be found in the *C-Star* project [Levin *et al.*, 1998] where a shallow representation is used for translation in a travel planning domain. The representation is basically a number of labels representing speech act, concepts and arguments like in

> *we have a single and a double available*
> `give-information+availability+room`
> `(room-type=(single & double))`

The shortcomings of this very simple representation is obvious: more complex relations between two or more objects are not possible to represent in natural way. Consider:

> *a single room in hotel A would cost you 110 dollars, in hotel B 106*

Since concepts (the hotels) and arguments (room-type and prices) are not linked by relations this utterance cannot be represented properly.

7 Discussion and Future Work

We have presented the work in progress on the VERBMOBIL dialogue component which had to adapt to an extended domain and the new functionality of summary/script generation in the second phase of the project. An overview of the shallow processing track was given, demonstrating the advantages of statistical and shallow finite state technology in a speech-to-speech system which requires a high level of robustness. We explained the structure of our representation of dialogue

[2] An English translation might look like this: VERBMOBIL SUMMARY: **Participants:** Mr Rosenstock, Mrs Johann. Date: 1. March 1999. Time 3:26 pm – 3:27 pm. **Dialogue Results:** *Travel planning:* The trip there will take place on the 14th of February. Transportation will be by plane. The trip to Hannover takes one hour. The trip back is on Monday. Mr Rosenstock will book the tickets. Mr Rosenstock and Mrs Johann travel together. *Accommodation:* Mrs Johann will arrange the accommodation.

context and how to manage topic, focus and agreement in a negotiation dialogue so that a summary can easily be generated, again taking into account the requirements for robustness.

There are many open questions to think about and work on. Here we list some of them:

- *how far do we get with this approach?* Will we be able to extend the coverage of the summary/script with this representation? How does it transfer to other domains?
- *how do we evaluate the dialogue manager?* We still have to check how decisions about negotiation agreements can be made more reliable.
- *is there an overall framework?* We have to compare our approach with related systems/techniques.
- *do the results apply for interactive systems?* We believe so but what extensions would we need?

Our future work in the further course of the VERBMOBIL project will focus on the following items:

- *multi-speaker*: do we need other dialogue acts? How do we know who's talking to whom?
- *combining deep and shallow information*: how do we handle different segmentation?
- *extending coverage*: so the summary can include timetables, excuses, reasons, social formulae

References

[Alexandersson and Poller, 1998] Jan Alexandersson and Peter Poller. Towards multilingual protocol generation for spontaneous speech dialogues. In *Proceedings of INLG-98*, Niagara-On-The-Lake, 1998.

[Alexandersson *et al.*, 1997] Jan Alexandersson, Norbert Reithinger, and Elisabeth Maier. Insights into the Dialogue Processing of VERBMOBIL. In *Proceedings of the Fifth Conference on Applied Natural Language Processing, ANLP '97*, pages 33–40, Washington, DC, 1997.

[Birkenhauer, 1998] Christoph Birkenhauer. Das Dialoggedächtnis des Übersetzungssystems Verbmobil. Universität des Saarlandes, 1998. Diplomarbeit.

[Bos *et al.*, 1998] Johan Bos, C. J. Rupp, Bianka Buschbeck-Wolf, and Michael Dorna. Managing information at linguistic interfaces. In *Proceedings of COLING–ACL '98*, Montreal, Canada, 1998.

[Bub and Schwinn, 1996] Thomas Bub and Johannes Schwinn. Verbmobil: The evolution of a complex large speech-to-speech translation system. In *Proceedings of ICSLP-96*, pages 2371–2374, Philadelphia, PA., 1996.

[Bub *et al.*, 1997] Thomas Bub, Wolfgang Wahlster, and Alex Waibel. Verbmobil: The combination of deep and shallow processing for spontaneous speech translation. In *Proceedings of ICASSP-97*, pages 71–74, Munich, 1997.

[Endriss, 1998] Ulrich Endriss. Semantik zeitlicher Ausdrücke in Terminvereinbarungsdialogen. Verbmobil-Report 227, Technische Universität Berlin, 1998. The report is available from the Verbmobil document server at `http://www.dfki.de/cgi-bin/verbmobil/htbin/doc-access.cgi`.

[Hobbs *et al.*, 1996] Hobbs, Appelt, Bear, Israel, Kameyama, Stickel, and Tyson. FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. In Roche and Schabes, editors, *Finite State Devices for Natural Language Processing*. MIT Press, 1996.

[Levin *et al.*, 1998] Lori Levin, Donna Gates, Alon Lavie, and Alex Waibel. An Interlingua Based on Domain Actions for Machine Translation of Task-Oriented Dialogues. In *Proceedings of International Conference on Spoken Language Processing (ICSLP'98)*, 1998.

[Levinson, 1993] Stephen C. Levinson. *Pragmatics*. Cambridge University Press, 1993.

[Maier, 1996] Elisabeth Maier. Context Construction as Subtask of Dialogue Processing - the VERBMOBIL Case. In Anton Nijholt, Harry Bunt, Susann LuperFoy, Gert Veldhuijzen van Zanten, and Jan Schaake, editors, *Proceedings of the Eleventh Twente Workshop* *on Language Technology, TWLT, Dialogue Management in Natural Language Systems*, pages 113–122, Enschede, Netherlands, June 19-21 1996.

[Reithinger and Klesen, 1997] Norbert Reithinger and Martin Klesen. Dialogue act classification using language models. In *Proceedings of EuroSpeech-97*, pages 2235–2238, Rhodes, 1997.

[Reithinger, 1999] Norbert Reithinger. Robust information extraction in a speech translation system. In *Proceedings of EuroSpeech-99*, pages 2427–2430, 1999.